Available online at www.sciencedirect.com

SCIENCE DIRECT®

ELSEVIER

Information Processing and Management 42 (2006) 1151–1162

INFORMATION PROCESSING & MANAGEMENT

www.elsevier.com/locate/infoproman

Does pseudo-relevance feedback improve distributed information retrieval systems?

Fernando Martínez-Santiago, Miguel A. García-Cumbreras *, L. Alfonso Ureña-Lòpez

Department of Computer Science, University of Jaén, Jaén, Spain

Received 3 October 2005; received in revised form 4 January 2006; accepted 4 January 2006
Available online 23 February 2006

Abstract

This paper presents a thorough analysis of the capabilities of the pseudo-relevance feedback (PRF) technique applied to distributed information retrieval (DIR). Previous studies have researched the application of PRF to improve the selection process of the best set of collections from a ranked list. This work emphasizes the effectiveness of PRF applied to the collection fusion problem. Usually, DIR systems apply PRF in the same way as traditional Information Retrieval systems. For each collection, local results are improved through PRF. A first question which arises is whether this local improvement is preserved in the final result. In addition, DIR systems merge the documents of rankings that are returned from a set of collections. Since a new global list of documents is available, we could use that list to apply PRF again, but on global level rather than on a local level. In order to apply global PRF, we have developed a merging approach called two-step RSV. Finally, we describe a number of experiments involving the two levels, local and global, of application of the PRF techniques.

Keywords: DIR; Collection fusion; TREC; CORI; Pseudo-relevance feedback

1. Introduction

Typically, a distributed information retrieval (DIR) system must rank document collections for query relevance. It selects the best set of collections from a ranked list, and merges the document rankings, returned from a set of collections. This last issue is the problem called *collection fusion problem* (Voorhees, Gupta, & Johnson-Laird, 1995). The aim of this paper is a thorough analysis of the pseudo-relevance feedback (PRF), also named blind feedback, applied to the collection fusion problem.

* Corresponding author.
E-mail addresses: dofer@ujaen.es (F. Martínez-Santiago), magc@ujaen.es (M.A. García-Cumbreras), laurena@ujaen.es (L.A. Ureña-Lòpez).

0306-4573/$ - see front matter © 2006 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ipm.2006.01.003

Relevance feedback (Rocchio, 1971) is an appreciated process that improves the performance of the information retrieval. The goal of relevance feedback is to retrieve and rank those documents highly, which are similar to the documents that are found relevant by the user. On the other hand, (PRF) (Salton & Buckley, 1990) technique does not need user interaction but it makes the assumption that the top N retrieved documents are relevant. Relevance feedback has been applied to non-distributed scenarios as well as DIR systems, but it needs user collaboration to decide what documents are relevant.

Usually DIR systems apply PRF in the same way that traditional information retrieval system: PRF or query expansion is applied in a local environment by each individual IR System. In this paper, we use *local feedback* to refer to this way of application of PRF. In addition our proposal is to apply PRF globally over the final top ranking merged document list. In this paper it is said that this kind of PRF is *global PRF.* Previous works are focused on local PRF as a way of improving the collection selection process, obtaining poor results (Ogilve & Callan, 2001). We explore global PRF as a way of improving the document merging process, not the collections selection process.

The documents returned for each IR engine are merged by using an algorithm called two-step RSV (Martínez-Santiago, Martín, & Ureña, 2003a, 2005). This algorithm works well in CLIR systems based on query translation, but the application of two-step RSV at DIR environments requires an additional effort: learning of collection issues such as document frequency, collection size and so on. On the other hand, since two-step RSV makes up a new global index based on query terms and the whole of retrieved documents, it is possible the application blind feedback at global level, by means of the DIR monitor, better than at local one, by means of each individual IR engine. Note that two-step RSV algorithm does not implement the whole of the DIR problems. Two-step RSV deals only with the document merging problem. Thus, *ranking and* selection of collections are realized by using the known CORI algorithm, described in the next section (see Table 1).

1.1. The Collection Retrieval Inference Network (CORI)

The Collection Retrieval Inference Network (CORI) model (Callan, Lu, & Croft, 1995) is a well-known algorithm used in DIR, and it addresses three problems. Briefly, CORI is inspired by the TF * IDF document ranking method (TF is the term frequency and IDF is the inverse document frequency) as an analogy for collection ranking. Each collection is depicted as a virtual document. The whole of those virtual documents build up a virtual collection. Given an user query, collections are selected in the same way that traditional IR systems select documents. Thus, CORI uses the TF * IDF formula by replacing TF with DF (document frequency) and IDF with ICF (inverse collection frequency). The required resource description is built up by document frequency, size of the collection and so on. (Callan, Connell, & Du, 1999) and (Callan, French, Powell, & Connell, 2000) propose a sampling technique (query-based sampling) in order to learn the description of the collections by interacting with every database by sending queries and analyzing the outcomes.

In a recent paper (Si & Callan, 2003c) studied the limitations of CORI when collection size varies. They found that CORI rarely ranks large collections highly, even though the collections are often the best source of relevant documents. They propose to modify CORI based on estimated database size to compensate for this effect but they do not address the difficult issue of parameter choice.

Table 1
DIR systems and algorithms implemented for the experiments

	CORI	Two-step RSV
Ranking collections	CORI	CORI
Selecting collections	CORI	CORI
Merging documents	CORI	Two-step RSV
Local PRF available	Yes	Yes
Global feedback available	No	Yes

In our experiments we have used the last version of CORI. Note that the main interest of this paper is not to evaluate an alternative of CORI or another distributed model, but to evaluate a well-known method in IR, as PRF is, in distributed environments.

2. Calculation of two-step retrieval status value

Given a query term distributed over several selected collections, their document frequencies are grouping together (Martínez-Santiago, Martín, & Ureña, 2003b). Then, the method needs to recalculate the document score by changing the document frequency of each query term. The new document frequency will be calculated adding up each local document frequency of the term for each selected collection. Given an user query, our method has two steps:

1. The document pre-selection phase. It consists in searching relevant documents locally for each selected collection. The result is a single collection of pre-selected documents (I' collection) that is the result of the union of the top retrieved documents for each collection.
2. The re-indexing phase. It consists in re-indexing the global collection I', but considering only the query vocabulary. Only the query terms are re-indexed: given a term, its document frequency is the result of grouping together the document frequency of each term from each selected collection. Finally, a new index is created by using the global document frequency, and the query is executed against the new index. Thus, for example, if two collections are selected, I_1 and I_2, and the term "government" is part of the query, then the new global frequency is $df_{I_1}(\text{government}) + df_{I_2}(\text{government})$.

The hypothesis of this method is as follows. Given two documents, the score of both documents will be comparable if the document frequency is the same for each meaningful term query. By grouping together the document frequency for each term, we ensure the hypothesis compliance. Thus, DIR system must send the query to each selected collection, download the most relevant documents, create a new index with such documents by means of some weighting formula, and finally the query is executed against the new index. The I' documents are ranked by using this new index. Note that to create this index at query time is possible because the vocabulary of the new index is made up only by query terms.

The two-step RSV merging algorithm has been applied successfully in cross lingual information retrieval (CLIR). CLIR is a subfield of information retrieval dealing with retrieving information written in languages different from the language of the user's query. For example, a user may pose their query in English but retrieve relevant documents written in French, Spanish and Russian. Thus, given a user query, a CLIR system obtains a multilingual list of documents. Usually, a CLIR system obtains as many lists of documents as languages a re in the multilingual collection of documents. Thus, the last step of such a CLIR system is merging every monolingual list of retrieved documents in order to obtain a only multilingual list. This merging process is very similar to the merging of documents in DIR environments, so we have applied largely two-step RSV as merging algorithm in several multilingual systems. The obtained results have ever been about 20–40% of improvement respect of other traditional merging algorithms such as Round-Robin or Raw scoring algorithm (Martínez-Santiago et al., 2003a, 2004). Even two-step RSV obtains better results than merging approaches based on machine learning such as logistic regression or neural networks, obtaining an improvement about 10–20% (Martínez-Santiago, García-Cumbreras, Díaz-Galiano, & Ureña-López, 2005).

2.1. Required elements for the global index calculus

In this work, we have used OKAPI BM-25 (Robertson, Walker, & Beaulieu, 2000) in the creation of the global index for the second step of the two-step RSV approach. Anyway, other weighting formulae require similar information. OKAPI BM-25 in which the weight w_{ij} assigned to a given term t_j in a document D_i was computed according to the following formula:

$$w_{ij} = \sum_{T\in Q} W^{(1)} \frac{(k_1+1)tf}{K+tf} qtf \qquad (1)$$

where

- Q is the query made up by T terms.
- K represents the ratio between the length of the document measured by the sum of tf_{ij} and the collection mean denoted by *avgdl*. K is calculated as $K = k_1((1 - b) + b * dl/avgdl)$.
- k_1 and b are constant values fixed at $k1 = 1, 25$ and $b = 0, 75$. This values are obtained empirically.
- *tf* is the term frequency into the document.
- *qtf* is the term frequency into the query Q.
- *dl* and *avgdl* are the length of the document and the average length document in the collection.
- $W^{(1)}$ is the weight of the term T in Q proposed by Robertson and Jones (1976). R and r factors are taken into account when PRF is applied.

$$W^{(1)} = \log \frac{(r + 0.5)/(R - r + 0.5)}{(n - r + 0.5)/(N - n - R - r + 0.5)} \quad (2)$$

N is the number of documents in the collection, n is the document frequency, R is the number of the top-ranked documents in the collection. In the initial search these top documents are relevant, and the rest of the documents in the collection are irrelevant, r is the number of top-ranked documents containing the term.

The question is how to apply such formula globally (over I' collection) better than locally (for each I_i collection). Several elements are known. Other must be learned or estimated.

- The term frequency is known because the documents must be downloaded by the DIR system before the second step of two-step RSV is committed.
- Note that the document frequency n is the sum of the local documents frequency n_i. Every local document frequency n_i is obtained by means of several methods: even non-collaborative IR Systems return the number of documents containing every term. If this number is not returned, we could obtain document frequency of the term by sending an only-term query for each term presented into the query. The number of returned documents is the document frequency *df* of the term. In addition, there are several approaches to learn this factor, such as the well-known *Query-based sample* (Callan et al., 1999) algorithm.
- The number of documents in the collection, N, is obtained in the same way as the global document frequency: by grouping together each local collection size. If the size of each collection is not available, this is estimated by using *Capture-recapture* (Liu, Yu, Meng, Santos, & Zhang, 2001) or *sample-resample* (Si & Callan, 2003b) algorithms.
- Finally, the average document length is learned by means of *Query-based* algorithm.

2.2. A DIR system proposal based on two-step RSV calculus

Of all the factors involved in the formula 1, the most expensive is the document frequency since this requires every document to be downloaded before the global ranked list is obtained. This restriction is feasible at query time because the ranked list is calculated according to the user requirements of more documents. For example, DIR system obtains the 10 first documents, these documents are shown to the user. If the user requests more documents, the DIR system downloads, re-weights and shows next 10 documents. An algorithm to minimize the download charge is proposed as follows:

1. Let I, a distributed collection made up by L subcollections, $I = I_1, \ldots, I_L$. Given an user query, let L' ($L' \leqslant L$) the number of subcollections selected by the DIR System (by using CORI algorithm, by example).
2. The DIR system downloads m documents for each selected subcollection $I_i, 1 \leqslant i \leqslant L'$. In this way, the DIR system has downloaded the set of documents $I' = \{I'_1, \ldots, I'_{L'}\}$, $|I'| = L' * m \geqslant 10$, and $I'_i = \{d^i_1, \ldots, d^i_n\}$, $|I'_i| = m$ is the downloaded set of documents from I_i.
3. I' documents are re-weighted and ranked according to two-step RSV.

4. The 10 first documents into I' ranked list are eliminated from every I'_i documents set. If there is any empty I'_i subset, download m documents from I_i, and re-weight the new I'_i documents set. Repeat the previous step until every I'_i subset contains at least one document.
5. Show top-10 documents to the user.
6. If the user requests more documents, return to the step 2.

If the system shows H documents for each user request and the top-H documents are into the same collection I_i (this is the worst case scenario), the DIR monitor needs to download.
$L' * m + H - m$ documents, where

- L' is the number of subcollections selected by the DIR system.
- m is the number of documents downloaded for each selected subcollection.
- H is the number of documents showed for each user request.

For example, given $H = 10$, $L' = 10$, $m = 2$, the DIR monitor needs to download 28 documents in the worst possible case before the system shows the first 10 documents. Note that 28 documents is the worst case for the first 10 documents, and the next user request needs to download at most 18 documents. On the other hand, several systems allow the more than one document to be downloaded per access. Even if this possibility is not available, the DIR system could download these documents by using a parallel process.

At this point, the application of blind feedback is simple. Given the R top-ranked documents at global level, the formula 2 is applied. In this work, Top-10 documents are analyzed. Then, the expanded-query is applied to re-weight every downloaded document. Note that the expanded query is only applied at global level, not at local one.

3. Experiments and results

3.1. Experimental method

The steps to run each experiment are summarized as follows:
The scenario is set up *off-line* as follows:

1. A local index is compiled for every collection (OKAPI).
2. *Query-based sampling* and *sample-resample* algorithms are applied in order to obtain a language model: vocabulary, document average length and collections size. Document frequency of the terms of the learned vocabulary is returned from every local collection by sending one-term queries. The sum of returned documents for each collection is just the document frequency for this query.

After the indices are built and the language model is learned, the system is evaluated. To evaluate the proposed method, we have used TREC1 and TREC2 query set, 100 queries in total. For each query:

1. CORI algorithm is applied to rank document collections.
2. The most relevant collections are selected by means of the *clustering* algorithm shown in (Callan et al., 1995).
3. Every selected collection obtains a local list of documents by running each query against each local index.
4. Some experiments make use of PRF for each collection, at local level. The PRF approach is OKAPI-BM25 (Eqs. 1 and 2). Top-10 terms of the top-10 documents are taken into account.
5. Merging process is carried out by using CORI and two-step RSV merging algorithms.
6. Some experiments apply PRF at global level by using the global index created by two-step RSV. The configuration of PRF at global level is the same that the local one.
7. On the other hand, the whole of documents are indexed in a only index. The query is executed against this huge index, too. The obtained results are labeled as *centralized* and it simulates a non-distributed IR system. Thus, a optimal distributed IR system should obtain at least the same performance that the *centralized* one.

Table 2
Description of the collections sets

	# of docs.			Size (in MB)		
	Min.	Avg.	Max.	Min.	Avg.	Max.
TREC-1	741,991	741,991	741,991	2168	2168	2168
TREC-13	10,163	57,066	226,087	33	159.23	260
TREC-80	2473	9273	32,401	23	25.81	30

Finally, the evaluation has been accomplished by using two measurements: *R*-precision at 5, 10, 20 and 100 documents, and 11-pt average precision.

3.2. Test collections description

The experiments are made using three partitions of TREC1 and TREC2. The TREC1 and TREC2 collections belong to the text published between 1987 and 1990 in various newspapers, news agencies and editorials. There is a total of more than two gigabytes of data, divided into about 740,000 documents.

Over these 13 collections we have made three test, described in Table 2:

- *TREC-1*. The 13 collections indexed with OKAPI. It shows the best case.
- *TREC-13*. Each collection of the 13 has been indexed separately.
- *TREC-80*. The original 13 collections have been divided into 80 collections, and indexed separately. These eighty collections have been created by source and with a random distribution.

Since two-step RSV builds a new dynamic index, it is not important how the terms in the different collections are distributed, because the algorithm two-step RSV does not take into account the source of each document. The document is re-indexed with the others docs selected, whatever its source (Martín, Martínez-Santiago, & Ureña, 2005).

The queries used appear in the first two editions of the TREC conferences; one hundred queries, in total. We have always and only used the title and description fields.

3.3. Experiments without query expansion

In this section we compare CORI and two-step RSV. We have applied neither local nor global query expansion.

Some parameters of this experiment are:

- We have used the last version of CORI.
- We have retrieved the first 1000 documents from each database.
- We have used the test collections TREC-13 and TREC-80. Over TREC-13, the most relevant collections are selected by means of the *clustering* algorithm (see section Experimental Method) and over TREC-80 we have selected the first 5, 10, 15 and 20 collections.

Tables 3 and 4 show the results obtained without blind-feedback. two-step RSV outperforms CORI both TREC-13 (38.2%) and TREC-80 (12.8%) collection set. These results show that the increase of two-step RSV performance over CORI is collection-depended, since the difference of improvement between both collection sets is about 25%. Several studies show that CORI performance is collection-depended (Si & Callan, 2003a, 2003b) while two-step RSV obtains a performance very stable.

Two-step RSV is more stable when the number of collections is increased, and it obtains a better performance. Thus, the difference of improvement could be due to the different performances of both algorithms.

Table 3
DIR experiments without feedback (set TREC-13, queries 101–150)

Fusion	5-prec	10-prec	20-prec	100-prec	Avg.-prec
CORI	0.468	0.416	0.363	0.265	0.130
Two-step RSV	0.480	0.458	0.426	0.329	0.181
Centralized	*0.492*	*0.492*	*0.444*	*0.346*	*0.194*

Table 4
DIR experiments without feedback (set TREC-80, queries 51–100)

Fusion	5-prec	10-prec	20-prec	100-prec	Avg.-prec
CORI	0.368	0.366	0.362	0.254	0.086
Two-step RSV	0.416	0.424	0.396	0.263	0.097
Centralized	*0.556*	*0.514*	*0.492*	*0.371*	*0.210*

3.4. Experiments with query expansion

In this section we study the impact of query expansion method based on local and global PRF approaches:

- Local pseudo-relevance feedback. Each local collection applies PRF locally with the purpose of increasing the local performance.
- Global pseudo-relevance feedback. This case can only be applied to two-step RSV. Since the DIR system generates a new global index, it is possible to apply PRF to that global index.
- Local and global pseudo-relevance feedback. Finally, it is possible to apply PRF firstly to each local collection, and also later to the DIR monitor.

3.4.1. Local PRF experiments

As Fig. 1 shows, the local feedback does not provide an increase in the two-step RSV case. If we use CORI the situation is a little worse over 13 collections (top of Fig. 1). The use of PRF over 80 collections sometimes,

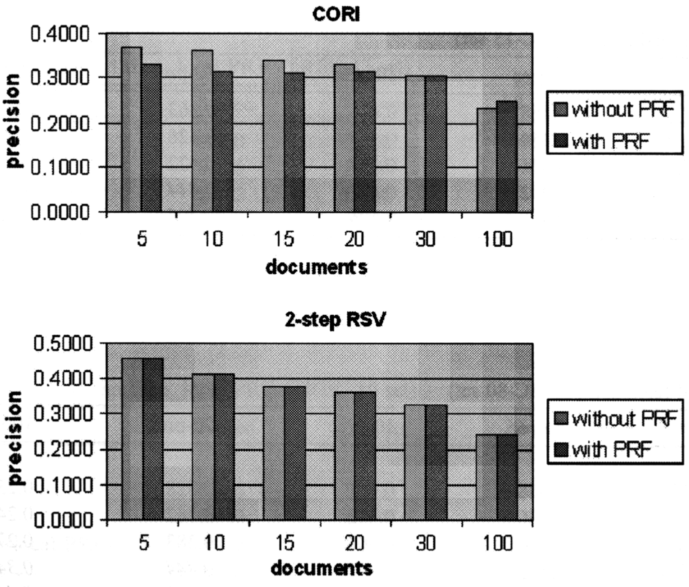


Fig. 1. Local feedback impact (TREC-13 set).

but not always, produces even worse results. In any case, the differences are so small (not more than two points) that the only conclusion is that PRF does not affect the final result, either with CORI or with two-step RSV. This result was already known by CORI for non-collaborative systems. Thus, (Ogilve & Callan, 2001) shows that the use of query expansion methods (*Local Context Analysis* in this case) increases neither the collections selection nor the documents selection. A possible cause is the expanded query length, because this new length makes difficult the score normalization. This reason cannot be applied to two-step RSV. Some previous experiments showed that the performance of the two-step RSV does not depend on the length of the query (Martínez-Santiago et al., 2003a). In the case of two-step RSV the causes can be two:

- The DIR system only works with the original query vocabulary. Thus, a new relevant document is not selected due to the query expansion.
- Two-step RSV does not use the local score obtained for each document. The sole relevant condition of a document is that this document belongs to the list given by the local collection, and the query vocabulary that this document contains, and never to the local score obtained.

3.4.2. Experiments with global PRF

Whether local PRF is applied or not, the application of query expansion methods is feasible, not using each collection as a single unit but using the index created in the second phase of the two-step RSV method. The computational cost in this case is not very high, if compared with the computational cost of a centralized system, because it only needs to analyze a few number of documents, in our experiments the 10 first documents, so in general it is only necessary to download two or three documents per selected collection, using any procedure such as those shown in Section 2.

The obtained results are depicted in Tables 5 and 6. The increase of the average precision introduced using global PRF in relation to the original two-step RSV is quite a lot significant, around 20%.

The increase of the centralized model with PRF is 41%, much more than the 20% obtained with two-step RSV. The reason is clear: the centralized model has access to *all* documents of *all* collections, so it is possible to include relevant documents which have not been selected previously. This is impossible with the index created by means of two-step RSV, because this method only includes *some* documents and *some* collections.

Table 5
DIR experiments with global feedback (TREC-13 set)

Fusion	5-prec	10-prec	20-prec	100-prec	Avg.-prec
CORI	0.468	0.416	0.363	0.265	0.130
Two-step RSV	0.480	0.458	0.426	0.329	0.181
Two-step RSV + global PRF	0.436	0.438	0.422	0.357	0.216
Centralized	*0.492*	*0.492*	*0.444*	*0.346*	*0.194*
Centralized + PRF	*0.540*	*0.526*	*0.497*	*0.418*	*0.273*

Table 6
DIR Experiments with global feedback (TREC-80 set)

Fusion	5-prec	10-prec	20-prec	100-prec	Avg.-prec
Clustering					
CORI	0.368	0.362	0.328	0.234	0.079
Two-step RSV	0.456	0.412	0.362	0.241	0.089
Two-step RSV + global PRF	0.440	0.408	0.383	0.274	0.105
Centralized	*0.492*	*0.492*	*0.444*	*0.346*	*0.194*
Centralized + PRF	*0.540*	*0.526*	*0.497*	*0.418*	*0.273*

Thus, global PRF only resorts documents selected previously, but it never adds new documents. In order to add new documents to the pre-selected lists of documents, we have tested a variation of the searching process by adding some additional steps:

1. The user formulates the initial query.
2. The initial user query is sent throughout every selected collection.
3. The selected documents are merged by applying two-step RSV.
4. Global PRF is applied and the original user query is expanded.
5. Steps 2 and 3 are repeated but by using the expanded query instead of the initial query.

The results are depicted in Tables 7 and 8. The improvement in relation to the original two-step RSV is about 30% (for instance, in TREC-13, 0.181 with two-step RSV vs. 0.230 with two-step RSV + global PRF + local PRF), near to the 40% obtained by means of the application of PRF in a centralized IR System (for instance, in TREC-80, 0.194 with Centralized vs. 0.273 with Centralized + PRF).

Note that this procedure is quite different to apply local PRF and then, global PRF. Here, we apply global PRF and then, local PRF by using the expanded query given by global PRF. The procedure depicted above apply *the same* expanded query at local and at global level. In addition local PRF improves the local ranked list of document taking into account only local issues of the subcollection. Because the query expanded locally is quite different of the query expanded globally, the new documents added locally to the ranked list are largely ignored at global level.

The difference of the improvement obtained with the application of PRF over a distributed or a centralized model can be because of the fact that in a centralized model the IR system has access to all documents, so it is normal that the terms of the expanded query will be more representatives of the relevant documents.

Is the application of global PRF ever a good idea? It is clear that the global PRF increases the average precision, but it does not always increase the selection of the first documents, and worse levels of precision are frequently obtained with global PRF if compared with the results without PRF, when only the 5 or 10 first documents are considered. This conclusion is showed in Fig. 2. According to the increment in the number of documents the precision obtained with global PRF also increases, and finally, the average precision and the *R*-precision are a little higher (see Fig. 3). The use of global PRF increases the recall in general, because it introduces more relevant documents between the first thousands, but it does not increase the

Table 7
DIR experiments with global feedback and then, local PRF (TREC-13 set)

Fusion	5-prec	10-prec	20-prec	100-prec	Avg.-prec
Two-step RSV	0.480	0.458	0.426	0.329	0.181
Two-step RSV + global PRF	0.436	0.438	0.422	0.357	0.216
Two-step RSV + global PRF + local PRF	0.442	0.448	0.441	0.377	0.230
Centralized + PRF	*0.540*	*0.526*	*0.497*	*0.418*	*0.273*

Table 8
DIR Experiments with global feedback and then, local PRF (TREC-80 set)

Fusion	5-prec	10-prec	20-prec	100-prec	Avg.-prec
Clustering					
Two-step RSV	0.456	0.412	0.362	0.241	0.089
Two-step RSV + global PRF	0.440	0.408	0.383	0.274	0.105
Two-step RSV + global PRF + local PRF	0.482	0.428	0.395	0.274	0.121
Centralized	*0.492*	*0.492*	*0.444*	*0.346*	*0.194*
Centralized + PRF	*0.540*	*0.526*	*0.497*	*0.418*	*0.273*

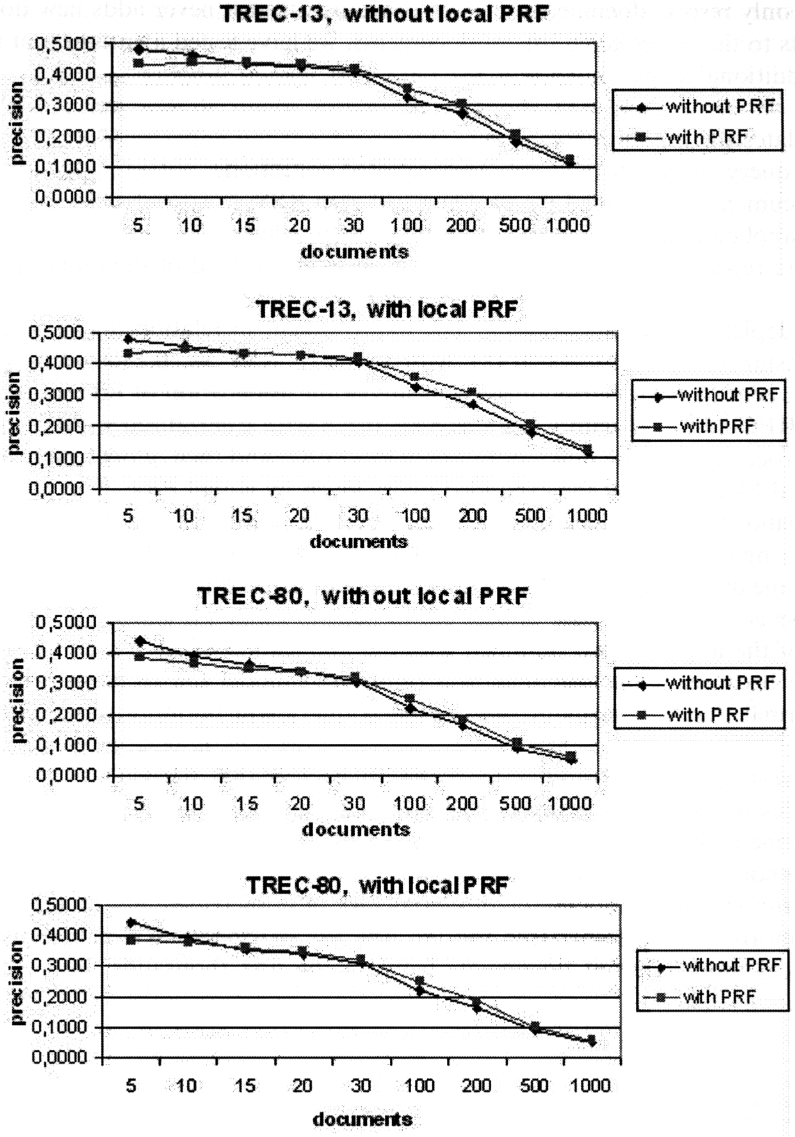
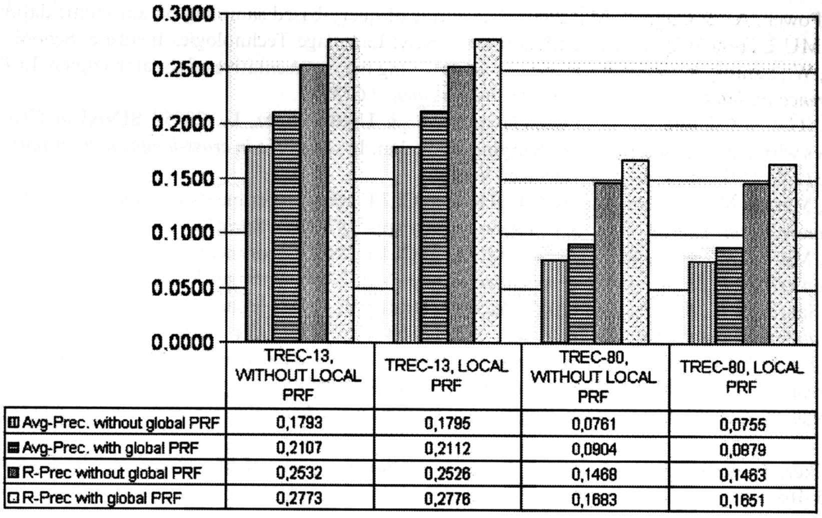


Fig. 2. Global feedback impact (I).

precision from the first documents. Is it always advisable to use global PRF? As it is usual in these cases, the answer depends on the user‘s needs. In general PRF worsen a little the ranking of the first 10 or 15 documents. Beyond these documents the result gets better. On the other hand the computational cost of applying PRF is moderate in an intranet environment but not null, because it needs to analyze the first documents at query time. In an Internet environment the cost is higher because of the communications cost, although this cost can be dramatically reduced using memory caches. The performance of the technique proposed depends on the number of selected collections by the distributed system. It does not matter the number of collections available, only how many collections are selected. Obviously, more collections selected involve more time consumption.

In addition, if we want improve even more the results, the expanded query should be sent again over each selected subcollection. It is possible that this computational cost will be the reason to apply or not this method: in general, the results are improved, but the user has to wait a few some more seconds.



	TREC-13, WITHOUT LOCAL PRF	TREC-13, LOCAL PRF	TREC-80, WITHOUT LOCAL PRF	TREC-80, LOCAL PRF
▥ Avg-Prec. without global PRF	0,1793	0,1795	0,0761	0,0755
▤ Avg-Prec. with global PRF	0,2107	0,2112	0,0904	0,0879
▦ R-Prec without global PRF	0,2532	0,2526	0,1468	0,1463
□ R-Prec with global PRF	0,2773	0,2776	0,1683	0,1651

Fig. 3. Global feedback impact (and II).

4. Conclusions and future work

DIR systems require merging list of documents from several collections. Usually, merging approaches take into account only original query terms in order to re-evaluate every document. On the other hand, blind feedback is a well-known IR technique to expand original query without user interaction. Experiments reported in this work pointed, indeed, in the same direction when blind feedback is applied at local level. The results are very different when PRF is applied at global level, reaching a noticeable improvement (between 20% and 30%). To aim the application of PRF at global level we have applied two-step RSV algorithm in two DIR scenarios. two-step RSV requires recalculating the document score by changing the document frequency of each query term. Given a query term, the new document frequency will be calculated by means of adding up each local document frequency of the term for each selected collection which creates an index at query time.

On the other hand, the experiments show that the performance of the two-step RSV merging algorithm is very stable. Don't mind the number of collections or the expansion of the user query, two-step RSV ever overcomes largely to the well-known merging formula of CORI, and the difference of performance between the centralized model and the distributed model is moderate. In order to corroborate the stability of the algorithm, we are interested in the capabilities of two-step RSV algorithm over other DIR scenarios such as more collections with very different sizes and weighting models.

In this paper, we prove that two-step RSV is a valid merging algorithm in distributed IR systems. In other papers such as (Martínez-Santiago et al., 2003a, 2005) we proved that the proposed merging approach obtains outstanding results in multilingual systems, too. Thus, the next step for our investigation is the development of a system capable of operating in a multilingual and distributed scenario. Since the Web is huge distributed a multilingual collection of documents, we are interested in the application of such a multilingual and distributed system over the Web.

Acknowledgement

This work has been supported by Spanish Government (MCYT) with Grant TIC2003-07158-C04-04.

References

Callan, J. P., Connell, M., & Du, A. (1999). Automatic discovery of language models for text databases. In *ACM-SIGMOD International conference on management of data*.

Callan, J. P., Lu, Z., & Croft, W. B. (1995). Searching distributed collections with inference networks. In *Proceedings of the 18th international conference of the ACM SIGIR'95*.

Callan, J., French, J., Powell, A., & Connell, M. (2000). The effects of query-based sampling on automatic database selection algorithms (Tech. Rep. No. CMU-LTI-00-162). Carnegie Mellon University: Language Technologies Institute, School of Computer Science.

Liu, K., Yu, C., Meng, W., Santos, A., & Zhang, C. (2001). Discovering the representative of a search engine. In *Proceedings of 10th ACM international conference on information and knowledge management (CIKM)*.

Martínez-Santiago, F., García-Cumbreras, M., Díaz-Galiano, M., & Ureña-López, L. (2005). SINAI at CLEF 2004: Using machine translation resources with a mixed two-step RSV merging algorithm. In *Advances in cross-language information retrieval. Lecture notes in computer science* (vol. 3491, pp. 156–164). Springer-Verlag.

Martínez-Santiago, F., Martín, M., & Ureña, L. (2003a). SINAI at CLEF 2002: Experiments with merging strategies. In *Advances in cross-language information retrieval. Lecture notes in computer science* (pp. 187–197). Springer-Verlag.

Martínez-Santiago, F., Martín, M., & Ureña, L. (2003b). SINAI at CLEF 2002: Experiments with merging strategies. In *Advances in cross-language information retrieval. Lecture notes in computer science* (Vol. 2785). Springer-Verlag.

Martín, M., Martínez-Santiago, F., & Ureña, L. (2005). Merging strategy for cross-lingual information retrieval based on learning vector quantization. *Neural Processing Letters, 22*(2).

Ogilve, P., & Callan, J. (2001). The effectiveness of query expansion for distributed information retrieval. In *Proceedings of the 10th international conference on information knowledge management (CIKM 2001)*.

Robertson, S. E., & Jones, K. S. (1976). Relevance weighting of search terms. *Journal of the American Society for Information Science, 27*, 129–146.

Robertson, S. E., Walker, S., & Beaulieu, M. (2000). Experimentation as a way of life: Okapi at TREC. *Information Processing and Management, 1*, 95–108.

Rocchio, J. J. (1971). *The smart retrieval system: Experiments in automatic document processing*. Prentice Hall.

Rogati, M., & Yang, Y. (2004). Multilingual information retrieval using open, transparent resources in CLEF 2003. In *Advances in cross-language information retrieval. Lecture notes in computer science* (Vol. 3237, pp. 133–139). Springer-Verlag.

Salton, G., & Buckley, G. (1990). Improving retrieval performance by relevance feedback. *Journal of American Society for Information Sciences, 21*, 288–297.

Si, L., & Callan, J. (2003a). Distributed information retrieval with skewed database size distributions. In *Proceedings of the national conference on digital government research.*

Si, L., & Callan, J. (2003b). Relevant document distribution estimation method for resource selection. In *Proceedings of the 26th annual international ACM SIGIR conference on research and development in information retrieval.*

Si, L., & Callan, J. (2003c). Relevant document distribution estimation method for resource selection. In *ACM SIGIR international conference on research and development in information retrieval.*

Voorhees, E., Gupta, N. K., & Johnson-Laird, B. (1995). The collection fusion problem. In *Proceedings of the 3rd text retrieval conference TREC-3.*