Learning Similarities for Collaborative Information Retrieval

Armin Hust

armin.hust@onlinehome.de

Abstract. The accuracy of ad-hoc information retrieval (IR) systems has plateaued in the last few years. At DFKI, we are working on so-called collaborative information retrieval (CIR) systems which have the potential to overcome the current limits. We focus on a restricted setting in CIR in which only old queries and correct answer documents to these queries are available for improving a new query. For this restricted setting we propose new approaches for query expansion procedures. We show how collaboration of individual users can improve overall information retrieval performance.
In our first steps towards techniques, we proposed new algorithms for query expansion in CIR systems. Now in this paper we focus on learning similarity measures. We do not try to invent new similarity measures, but learn weighting schemes to be applied to the standard cosine similarity measure. After learning the new weightings we re-evaluate our previously proposed CIR algorithms on standard IR test collections. It turns out that retrieval performance of previously developed algorithms is improved after learning the weightings for the involved similarity measure.

1 Introduction

In this section we introduce the research area of Collaborative Information Retrieval (CIR). We motivate and characterize the primary goals of this paper, query expansion procedures for CIR and outline the structure and contents.

The ultimate goal in IR is finding the documents that are useful to the information need expressed as a query. Much work has been done on improving IR systems, in particular in the Text Retrieval Conference series (TREC). 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 [13]. We are working on new approaches which learn to improve retrieval effectiveness from the interaction of different users with the retrieval engine. Such systems may have the potential to overcome the current plateau in ad-hoc retrieval.

CIR is a methodology where an IR system makes full use of all the additional information available in the system, especially

- the information from previous queries

- the relevance information gathered during previous search processes, independent of the method used to obtain this relevance information, i.e., explicitly by user relevance feedback or implicitly by unobtrusively detected relevance information.

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) as long as the relevance information gathered from these previous users has some significant meaning.

Subject to these assumptions we expect that collaborative searches will improve overall retrieval quality for all users.

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.

This paper is organized as follows: Section 2 describes related work in the field of query expansion, section 3 introduces the vector space model and query expansion procedures that have been developed for use in the vector space model. Section 4 describes the method for learning similarity functions and describes one of the functions in detail. Then section 5 describes the document collections we have used for evaluating our new algorithms and describes the evaluation methodology, section 6 describes the results of the evaluation. Finally section 7 summarizes this paper, draws some conclusions, and shows the essential factors for improving retrieval performance in CIR.

2 Related Work

Usage of short queries in IR produces a shortcoming in the number of documents ranked according to their similarity to the query. Thus IR systems try to reformulate the queries in a semi-automatic or automatic way. Several methods, called query expansion methods (QE), have been proposed to cope with this problem [3], [10]. These methods fall into three categories: usage of feedback information from the user (e.g. interactive QE), usage of information derived locally from the set of initially retrieved documents, and usage of information derived globally from the document collection. The goal of all QE methods is to finally find the optimal query which selects all the relevant documents. A comprehensive overview of newer procedures is available from Efthimiadis in [6]. Another newer technique, called local context analysis (LCA), was introduced by Xu and Croft in [15].

Newest procedures in the field of query expansion are dealing with query bases, a set of persistent past optimal queries, for investigating similarity measures between queries (refer to Raghavan, Sever and Alsaffar et al. in [11], [12] [2]). Wen et al. [14] are using query clustering techniques for discovering frequently asked questions or most popular topics on a search engine. This query clustering method makes use of user logs which allows to identify the documents the users have selected for a query. The similarity between two queries may be deduced from the common documents the users selected for them ([4]). Cui et al. [5] take into account the specific characteristics of web searching, where a large amount of user interaction information is recorded in the web query logs, which may be used for query expansion. Agichtein et al. [1] are learning search engine specific query transformations for question answering in the web.

3 Basics and Terminology

In this section we introduce the vector space model (VSM) which is employed in our work. We introduce the pseudo relevance feedback method for query expansion and two of our newly developed methods for CIR.

Vector Space Model Documents as well as queries are represented in a common way using a set of terms. Terms are determined from words of the documents, usually during preprocessing phases (e.g. stemming and stopword elimination). In the following a term is represented by $t_i, \ 1 \leq i \leq M$, where M is the number of terms in the document collection.

The vector space model assigns weights to terms in queries and in documents and represents them as M dimensional vectors

$$d_j = (w_{1j}, w_{2j}, ..., w_{Mj})^T, \quad 1 \leq j \leq N, \tag{1}$$

$$q_k = (w_{1k}, w_{2k}, ..., w_{Mk})^T, \quad 1 \leq k \leq L, \tag{2}$$

where T indicates the transpose of the vector, w_{ij} or w_{ik} is the weight of term t_i in document d_j or query q_k, N is the number of documents and L is the number of queries contained in the document collection.

The result of the execution of a query is a list of documents ranked according to their similarity to the given query. The similarity $sim(d_j, q_k)$ between a document d_j and a query q_k is measured by the cosine of the angle between these two M dimensional vectors:

$$\mathrm{sim}(d_j, q_k) = \frac{d_j^T \cdot q_k}{\|d_j\| \cdot \|q_k\|}, \tag{3}$$

where $\| \cdot \|$ is the Euclidean norm of a vector. In the case that the vectors are already normalized (and hence have a unit length) the similarity is simply the dot product between the two vectors d_j and q_k.

Query Expansion by Pseudo Relevance Feedback (PRF) After retrieval of the list of documents (in a first stage) highly ranked documents are all assumed to be relevant [15] and their terms (all of them or some highly weighted terms) are used for expanding the original query. Then documents are ranked again according to their similarity to the expanded query.

In this work we employ a variant of pseudo relevance feedback described by Kise et al. [9]. In our comparison with the newly developed methods, we will use the PRF method.

Let E be the set of document vectors given by

$$E = \left\{ d_j \middle| \frac{\mathrm{sim}(d_j, q_k)}{\max_{1 \le i \le N}\{\mathrm{sim}(d_i, q_k)\}} \ge \theta \right\} \tag{4}$$

where q_k is the original query and θ is a threshold parameter of the similarity. Then the sum D_k of the document vectors in E, $D_k = \sum_{d_j \in E} d_j$ is used as expansion terms for the original query. The expanded query vector q'_k is obtained by

$$q'_k = q_k + \alpha \frac{D_k}{\|D_k\|} \tag{5}$$

where α is a parameter for weighting the expansion terms. Then the documents are ranked again according to their similarity $\mathrm{sim}(d_j, q'_k)$.

Parameters θ in Equation 4 and α in Equation 5 are tuning parameters. During evaluation best parameter value settings have been obtained by experiment and those which give the highest average precision were selected for comparison against other methods.

Query Expansion by Methods developed for CIR In our approaches we use global relevance feedback which has been learned from previous queries; this is in contrast to local relevance feedback which is produced during execution of an individual query. All our new query expansion procedures work as follows:

- for each new query to be issued compute the similarities between the new query and each of the existing old queries
- select the old queries having a similarity to the new query which is greater than or equal to a given threshold
- from these selected old queries get the sets of relevant documents from the ground truth data
- from this set of relevant documents compute some terms for expansion of the new query
- use this terms to expand the new query and issue the new expanded query

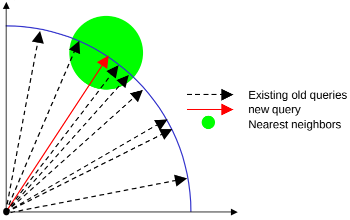

Fig. 1. Motivation for CIR methods: usage of the nearest neighbors

The algorithmic description is given here:

```
for each new query $q$ do
    compute the set $S = \{q_k | \mathrm{sim}(q_k, q) \ge \sigma, 1 \le k \le L\}$
    compute the sets $RD_k = \{d_j | q_k \in S \wedge d_j \textit{ is relevant to } q_k\}$
    compute the expanded query $q' = cirf(q, S, RD_k)$
end
```

where S is the set of existing old queries q_k with a similarity of σ or higher to the new query q, RD_k are the sets of the documents being relevant to the queries q_k and $cirf$ is a function for query expansion.

The goal now is to find suitable functions $cirf$ which can be efficiently computed and which maximize the effectiveness of the new query q' in terms of recall and precision. As is shown in figure 1 our approach is searching for neighbors of the new query. If suitable neighbors of a query q within a given distance are found, we try to derive information about the documents which are relevant to q from its nearest neighbors.

These functions introduce a new level of quality in the IR research area: while the term weighting functions such as tf-idf only work on documents and document collections, and relevance feedback works on a single query and uses information from their assumed relevant and non-relevant documents only, CIR now works on a single query, and uses the information of all other queries and their known relevant documents.

Methods Description. Due to lack of space we describe the methods informally very short. For detailed description and evaluation we point the reader to the referenced papers and articles.

***Q**uery **S**imilarity and Relevant **D**ocuments.* Method QSD ([7]) uses the relevant documents of the most similar queries for query expansion of a new query. The new query is rewritten as a sum of selected relevant documents of existing old queries, which have a high similarity to the new query, i.e.,

$$q' = q + \sum_{k=1}^{|S|} \sigma_k \frac{RD_k}{\parallel RD_k \parallel}, \tag{6}$$

where $|S|$ is the number of selected queries, σ_k are the similarities $\mathrm{sim}(q_k, q) \geq \sigma$ (σ is the threshold value) and RD_k are the sets of relevant documents.

***Q**uery **L**inear Combination and Relevant **D**ocuments.* Method QLD ([8]) uses the relevant documents of the most similar queries, which are used in re-writing the new query as a linear combination of the most similar queries. This query expansion method reconstructs the new query as a linear combination of existing old queries. Then the terms of the relevant documents of these existing old queries are used for query expansion, i.e.,

$$q' = q + \sum_{k=1}^{|S|} \tilde{\lambda_k} \frac{RD_k}{\parallel RD_k \parallel}, \tag{7}$$

where the $\tilde{\lambda_k}$ are parameter for weighting the expansion terms. The $\tilde{\lambda_k}$ are computed as follows: in most cases we cannot represent the new query q exactly as a linear combination of the old queries q_k, i.e., $q = \sum_{k=1}^{|S|} \lambda_k q_k$ will not have a solution for the coefficients λ_k. This equation is equivalent to a system of linear equations $Q\lambda = q$, where $Q = (q_1, q_2, \ldots, q_{|S|})$ is a matrix of M rows and $|S|$ columns and $\lambda = (\lambda_1, \lambda_2, \ldots, \lambda_{|S|})^T$ is a column vector consisting of $|S|$ elements. Because Q is normally singular ($M \gg |S|$) and there is no solution to the system, we find a vector $\tilde{\lambda}$ so that it provides a closest fit to the equation in some sense. Our approach is to minimize the Euclidean norm of the vector $Q\lambda - q$, i.e we solve

$$\tilde{\lambda} = argmin_\lambda \|Q\lambda - q\| \tag{8}$$

where $\tilde{\lambda} = (\tilde{\lambda_1}, \tilde{\lambda_2}, \ldots, \tilde{\lambda_{|S|}})^T$ is called the least squares solution for the system $Q\lambda = q$.

Limiting Factors in CIR Performance *Similarities of Queries to Documents.* One of the limiting factors for CIR retrieval performance is the similarity between the query and its non-relevant documents (as it is for non-CIR retrieval performance).

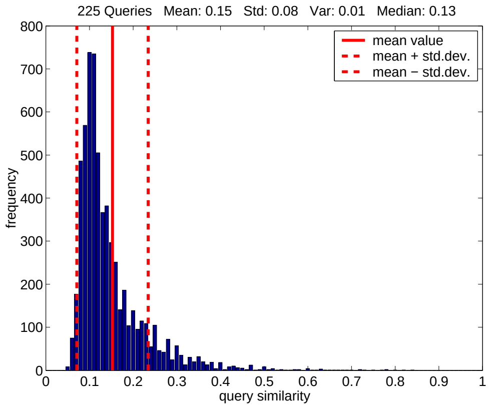

Fig. 2. CRAN: distribution of query similarities

Inter-Query Similarities. In our considerations for usage of similarities between different queries for retrieval performance improvements, we decided to analyze the inter-query similarities. We did not expect to have queries having highly correlated similarities as we would expect in real world applications. Indeed, the histograms show very low inter-query similarity for most of the text collections. Figure 2 displays the distribution of the inter-query similarity, excluding those similarities which are 0. Also the mean and the median value as well as the variance and the standard deviation are indicated in the graph. The vertical lines are: the mean similarity (solid line), and the values of the mean similarity ± the standard deviation (dotted lines).

Overlap of Relevant Documents. Another limiting factor for our CIR methods is some "overlapping" in relevant documents for different queries. We define the overlap of relevant documents as follows: Let $q_k, q_l \in Q$, $k \neq l$ be two different queries. Let RD_k and RD_l be the sets of documents which are relevant to queries q_k and q_l respectively. Then the overlap of relevant documents for these two queries is the number of documents in the set $O_{kl} = RD_k \cap RD_l = \{d_j | \; d_j \in RD_k \wedge d_j \in RD_l\}$. We expect retrieval performance improvements if the overlap of relevant documents is high.

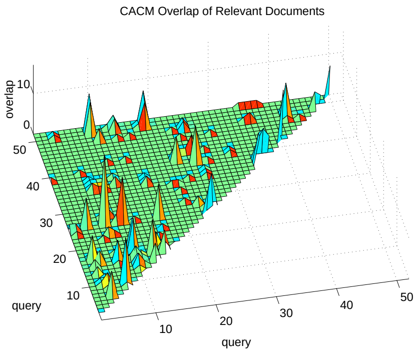

Fig. 3. CACM: overlap of relevant documents

4 Learning Similarity Functions

The motivation for learning similarity functions arises from the achieved performance improvements of our query expansion methods QSD and QLD.

Similarity between queries as it is used up to now is solely based on syntactical elements. Although we have used some normalization and cleaning operations (stemming and stop-word-elimination) there is no further processing beyond the syntactical level. Similarity between two queries is high if they use the same words. Similarity is low if they use different words.

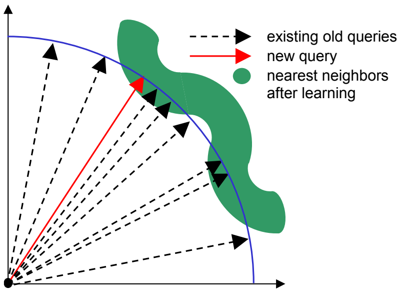

Fig. 4. Motivation for Learning Similarities: area of nearest neighbors changes dramatically

The same information need can be expressed in different queries, having a low similarity, although they are querying for the same facts and thus may have the same relevant documents. However, the methods developed up to now only use the inter-query similarity on the syntactical level, they do not consider the information need of the user. Figure 4 illustrates the proposed effect of learning, where the area of nearest neighborhood may change dramatically if the newly learned similarity functions are applied. In this way we can identify queries as nearest neighbors of a new query, even if they are far away (according to the standard cosine-similarity) from the new query.

The Learning Problem We now formulate the learning of similarity functions as a minimization problem.

We measure the similarity of sets of relevant documents by

$$\mathrm{dsim}_{kl} := \mathrm{sim}(rd_k, rd_l), \tag{9}$$

where $\mathrm{sim}(\cdot,\cdot)$ is defined in Equation 3, and rd_i are the summarized and centered document vectors consisting of the relevant documents of query q_i, i.e.,

$$rd_i = \frac{1}{|RD_i|} \sum_{d_j \in RD_i} d_j \tag{10}$$

and the similarity between queries by

$$\mathrm{qsim}_{kl}(x) := \mathrm{sim}(g(x, q_k), h(x, q_l)) \tag{11}$$

where x is a vector of weights to be applied against the queries q_k and/or q_l with some functions g and h, each returning an M-dimensional vector which can be fed into the standard cosine similarity measure described in Equation 3.

The motivation for learning the weights for similarity functions is as follows: If the similarity between two different vectors rd_k, rd_l is high, then the similarity between the two queries q_k, q_l having these document vectors assigned as relevant documents should be high. If the similarity between vectors rd_k, rd_l is low, then the similarity between the corresponding two queries should be low. This directly leads to the functions $f_{kl},\ 1 \leq k, l \leq L$ to be minimized as

$$f_{kl}(x) = \mathrm{qsim}_{kl}(x) - \mathrm{dsim}_{kl} \tag{12}$$

and considers all pairs of queries. Let F be a vector-valued function consisting of L^2 functions, where each of these functions uses an M dimensional input vector x, i.e.,

$$\begin{aligned} F : \mathbb{R}^M &\rightarrow \mathbb{R}^{L^2} \\ (x_1, x_2, \cdots, x_M)^T &\mapsto (f_{11}(x), f_{12}(x), \cdots, f_{kl}(x), \cdots, f_{LL}(x))^T \end{aligned} \tag{13}$$

Then we can state our learning problem as

$$\hat{x} = (\hat{x}_1, \hat{x}_2, \cdots, \hat{x}_M)^T = argmin_x \|F(x)\|^2 = argmin_x \sum_{k,l=1}^{L} f_{kl}(x)^2 \tag{14}$$

i.e., we are searching for a vector $\hat{x}$ that minimizes the Euclidean norm of the function F.

The Similarity Functions The goal is to find reasonable functions $\mathrm{qsim}_{kl}(x)$ which give us significant performance improvements for IR whilst having a moderate computational complexity both in the learning process as well as during the application of the similarity measure in the query expansion methods QSD and QLD.

We have developed 9 reasonable functions. Due to lack of space we describe only one of them here.

Similarity Function F2 We first define the component-wise multiplication of the individual components of two vectors, denote it by $\dot{*}$ and use it in infix-notation:

$$\dot{*} : \mathbb{R}^n \times \mathbb{R}^n \rightarrow \mathbb{R}^n$$

$$x \dot{*} y = (x_1, x_2, \cdots, x_n)^T \dot{*} (y_1, y_2, \cdots, y_n)^T = (x_1 y_1, x_2 y_2, \cdots, x_n y_n)^T$$

Then we define the new similarity function using the weights to be learned and denote it by a superscript

$$\mathrm{qsim}_{kl}^2(x) = \mathrm{sim}(q_k, x \dot{*} q_l) \tag{15}$$

leading to our minimization problem

$$\hat{x} = argmin_x \sum_{k,l=1}^{L} (\mathrm{qsim}_{kl}^2(x) - \mathrm{dsim}_{kl})^2 \tag{16}$$

5 Experimental Design

We use standard document collections and standard queries and questions provided by the SMART project and the TREC conferences. In addition we use some special collections that we have generated from the TREC collections to show special effects of our algorithms. In our experiments we used the following 10 collections:

- the SMART collections ADI, CACM, CISI, CRAN, MED and NPL.
- the TREC QA (question answering) collection prepared for the Question Answering track held at the TREC-9 conference, the QA-AP90 collection containing only those questions having a relevant answer document in the AP90 (Associated Press articles) document collection, the QA-AP90S collection (extracted from the QA-AP90 collection) having questions with similarity of 0.65 or above to any other question, and the QA-2001 collection prepared for the Question Answering track held at the TREC-10 conference.

On the one hand by utilizing these collections we take advantage of the ground truth data for performance evaluation. On the other hand we do not expect to have queries having highly correlated similarities as we would expect in a real world application. So it is a challenging task to show performance improvements for our methods.

Preparation of the Text Collections Terms used for document and query representation were obtained by stemming and eliminating stopwords. Then document and query vectors were created according to the so called tf-idf weighting scheme, where the document weights d_{ij} are computed as

$$d_{ij} = \sqrt{f_{ij}} \cdot idf_i \tag{17}$$

where f_{ij} is the raw frequency of term t_i, idf_i is the inverse document frequency $\log \frac{N}{n_i}$ of term t_i, and the query weights q_{ik} are computed as

$$q_{ik} = \sqrt{f_{ik}} \tag{18}$$

where f_{ik} is the raw frequency of term t_i in a query q_k.

Properties of the Text Collections Table 1 lists statistics about the collections after stemming and stopword elimination has been carried out; statistics about some of these collections before stemming and stopword elimination can be found in Baeza-Yates [3] and Kise et al. [9].

	ADI	CACM	CISI	CRAN	MED	NPL	QA	QA-AP90	QA-AP90S	QA-2001
size(MB)	0.1	1.2	1.4	1.4	1.1	3.8	28.2	3.7	3.7	20.1
number of documents	82	3204	1460	1400	1033	11429	6025	723	723	4274
number of terms	340	3029	5755	2882	4315	4415	48381	17502	17502	40626
mean number of terms	17.9	18.4	38.2	49.8	46.6	17.9	230.7	201.8	201.8	220.5
per document	(short)	(short)	(med)	(med)	(med)	(short)	(long)	(long)	(long)	(long)
number of queries	35	52	112	225	30	93	693	353	161	500
mean number of terms	5.7	9.3	23.3	8.5	9.5	6.5	3.1	3.2	3.5	2.7
per query	(med)	(med)	(long)	(med)	(med)	(med)	(short)	(short)	(short)	(short)
mean number of relev.	4.9	15.3	27.8	8.2	23.2	22.4	16.4	2.8	3.2	8.9
documents per query	(low)	(med)	(high)	(med)	(high)	(high)	(med)	(low)	(low)	(med)

Table 1. Statistics about the test collections

Methodology of Evaluation The numerical methods used for function minimization do not guarantee that they will find a global minimum of the function. However they will find a local minimum in an area surrounding the initial start value. Thus we did the same experiment several times with different initial values.

The result of each experiment was the vector $\hat{x}$. We then fed these values into the QSD and QLD methods using the similarity measure $\mathrm{qsim}^2_{kl}(x)$ as defined in Equation 15 for the query expansion methods.

The evaluation follows the "leave one out" technique used in several areas such as document classification, machine learning etc. From the set of L queries contained in each text collection we selected each query one after the other and treated it as a new query $q_l, 1 \leq l \leq L$. Then for each fixed query q_l we used the algorithm as described in section 3. Of course the now fixed query q_l itself does not take part in the computation of the query expansion. We varied parameters of the algorithms according to suitable values, and selected those parameters where highest performance improvements (in terms of average precision over all queries) were achieved.

6 Results

The methods are denoted by adding the name of the learned similarity function to the basic name, i.e., QSDF2 denotes the QSD method after learning similarity function F2 using the similarity measure $\mathrm{qsim}^2_{kl}(x)$ as defined in Equation 15.

Interpolated Average Precision Table 2 shows the interpolated average precision obtained by using the best parameter values for different methods. For each collection the best value of average precision is indicated by bold font, the second best value is indicated by italic font. In those cases, where our new methods outperform the PRF method, the value is underlined.

	ADI	CACM	CISI	CRAN	MED	NPL	QA	QA-	QA-	QA-
								AP90	AP90S	2001
VSM	0.375	0.130	0.120	0.384	0.525	0.185	0.645	0.745	0.643	0.603
PRF	0.390	0.199	0.129	0.435	**0.639**	**0.224**	0.685	0.757	0.661	**0.614**
QSD	0.374	0.237	0.142	0.428	0.503	0.184	0.727	*0.810*	0.786	0.603
QSDF2	*0.433*	**0.293**	**0.184**	*0.463*	*0.525*	*0.202*	*0.753*	**0.818**	*0.796*	*0.604*
QLD	0.369	0.227	0.171	0.436	0.507	0.185	0.734	0.812	0.789	0.603
QLDF2	**0.436**	*0.286*	*0.182*	**0.465**	*0.525*	0.196	**0.754**	**0.818**	**0.798**	*0.604*

Table 2. Interpolated average precision in CIR methods

Significance Testing Significance tests were applied to the results. Table 3 shows the results. Each row contains the results of two tests, i.e., test method X against method Y and vice versa.

- The indicator ++ (+) shows that method X is performing better than method Y at significance level $\alpha = 0.01$ ($\alpha = 0.05$).
- The indicator o shows that there is low probability that one of the methods is performing better than the other method.
- The indicator −− (−) shows that method Y is performing better than method X at significance level $\alpha = 0.01$ ($\alpha = 0.05$).

methods		ADI	CACM	CISI	CRAN	MED	NPL	QA	QA-	QA-	QA-
X	Y								AP90	AP90S	2001
PRF	VSM	+	++	++	++	++	++	++	+	o	++
QSD	PRF	o	o	o	o	−−	−−	++	++	++	−−
QSDF2	PRF	++	++	++	+	−−	−	++	++	++	−−
QSDF2	QSD	+	++	++	++	o	+	++	+	o	o
QLD	PRF	o	o	++	o	−−	−−	++	++	++	−−
QLDF2	PRF	++	+	++	+	−−	−−	++	++	++	−−
QLDF2	QLD	++	++	o	++	o	o	++	o	o	o

Table 3. Paired t-test results for significance levels $\alpha = 0.05$ and $\alpha = 0.01$

Relative Performance Improvements Table 4 shows the relative performance improvements for different methods. The ratio of improvement is computed as follows: let X be the average precision obtained by one of the methods and let Y be the average precision obtained by another method. Then the ratio is calculated by $ratio = \frac{X-Y}{Y}$. A positive value for the ratio indicates an improvement of method X over method Y, a negative value indicates a degradation in average precision from method X to Y.

methods		ADI	CACM	CISI	CRAN	MED	NPL	QA	QA-	QA-	QA-
X	Y								AP90	AP90S	2001
QSD	PRF	-4.0%	+18.8%	+9.8%	-1.7%	-21.3%	-17.8%	+6.1%	+6.9%	+18.8%	-1.8%
QSDF2	PRF	+10.9%	+47.0%	+42.2%	+6.3%	-17.9%	-9.7%	+10.0%	+8.0%	+20.4%	-1.7%
QSDF2	QSD	+15.5%	+23.7%	+29.6%	+8.2%	+4.3%	+9.9%	+3.6%	+1.0%	+1.3%	+0.1%
QLD	PRF	-5.5%	+14.1%	+32.3%	+0.1%	-20.7%	-17.2%	+7.2%	+7.2%	+19.3%	-1.8%
QLDF2	PRF	+11.9%	+43.5%	+40.8%	+6.8%	-17.8%	-12.5%	+10.1%	+7.9%	+20.6%	-1.6%
QLDF2	QLD	+18.3%	+25.7%	+6.5%	+6.7%	+3.6%	+5.7%	+2.7%	+0.7%	+1.1%	+0.2%

Table 4. Average precision improvement in different methods

Analysis of the Results From the average precision analysis we see that the QSDF2 method and the QLDF2 method perform best and second best in most cases. In all cases they also perform better than the basic method before learning.

For the MED and NPL text collections, the basic methods QSD and QLD do not perform better than the PRF method, nor do any of the methods after learning. We think that, in the case of the MED collection, this effect comes from the missing overlap of relevant documents, and in the case of the NPL collection from the high similarity of non-relevant documents to the queries.

In all but in one case we observe performance improvements after learning compared to the basic methods without learning. The highest performance improvement achieved is +40.3% (for the CACM collection). Only for the QA-2001 collection we observe a performance degradation for one method of -0.1% after learning; it should also be noted that for this collection the performance improvements achieved after learning are the lowest of all collections.

7 Conclusions

We have studied learning methods for improving retrieval performance in a restricted CIR environment where information about relevant documents from previous search processes carried out by several users is available for the current query.

Specifically, we developed, evaluated and analyzed new algorithms for query expansion, since query expansion methods are known to be successful in improving retrieval performance.

Results of the newly developed methods are encouraging. Retrieval performance improvements were achieved in most cases. For some text collections no significant retrieval performance improvements could be achieved, neither in the basic methods nor in applying the methods after learning similarity functions. We identified three essential factors for retrieval performance improvements:

- similarity between queries, also called inter-query similarity: we can not achieve performance improvements, if there are no pairs of queries with high similarities
- similarity of queries to their relevant documents and non-relevant documents: precision decreases, if non-relevant documents are ranked higher than relevant documents
- the overlap of relevant documents for pairs of queries: if there is no or low overlap in relevant documents, there are no document terms which are used for query expansion

We think that the first factor is the most important for our CIR methods. Best performance improvements have been achieved in text collections where the inter-query similarity is high, although the overlap in relevant documents is not high. Low or no retrieval performance improvements were achieved in those cases were the inter-query similarity is on average low.

For text collections, where similarity of queries to their non-relevant documents is high on average, we achieved low performance improvements.

For text collections, where the overlap of relevant documents is low or where no overlap in relevant documents exists, we did not achieve performance improvements, neither in the basic methods nor in the methods that have been applied after learning.

References

1. Eugene Agichtein, Steve Lawrence, and Luis Gravano. Learning search engine specific query transformations for question answering. *10th International WWW Conference*, pages 169–178, 2001.
2. Ali H. Alsaffar, Jitender S. Deogun, and Hayri Sever. Optimal queries in information filtering. *Foundations of Intelligent Systems, 12th International Symposium, Proceedings*, volume 1932 of *LNCS*, pages 435–443, 2000.
3. Ricardo Baeza-Yates and Berthier Ribeiro-Neto. *Modern Information Retrieval.* Addison-Wesley Publishing Company, 1999.
4. Brian T. Bartell, Garrison W. Cottrell, and Richard K. Belew. Optimizing similarity using using multi-query relevance feedback. *JASIS*, 49(8):742–761, 1998.
5. Hang Cui, Ji-Rong Wen, Jian-Yun Nieand, and Wei-Ying Ma. Probabilistic query expansion using query logs. *11th International WWW Conference*, 2002.
6. Efthimis N. Efthimiadis. Query expansion. *Annual Review of Information Science and Technology*, 31:121–187, 1996.
7. Armin Hust, Stefan Klink, Markus Junker, and Andreas Dengel. Query Expansion for Web Information Retrieval. *WIR Workshop*, volume P-19 of *LNI*, pages 176–180, 2002.
8. Armin Hust, Stefan Klink, Markus Junker, and Andreas Dengel. Query Reformulation in Collaborative Information Retrieval. *IKS 2002*, pages 95–100, 2002.
9. Koichi Kise, Markus Junker, Andreas Dengel, and Keinosuke Matsumoto. Experimental evaluation of passage-based document retrieval. *ICDAR-01*, pages 592–596, 2001.
10. Christopher D. Manning and Hinrich Schütze. *Foundations of Natural Language Processing.* MIT Press, 1999.
11. Vijay V. Raghavan and Hayri Sever. On the reuse of past optimal queries. *18th ACM SIGIR*, pages 344–350, 1995.
12. Hayri Sever. *Knowledge Structuring for Database Mining and Text Retrieval Using Past Optimal Queries.* PhD thesis, University of Louisiana, Lafayette, 1995.
13. 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.
14. Ji-Rong Wen, Jian-Yun Nie, and Hong-Jiang Zhang. Clustering user queries of a search engine. *10th International WWW Conference*, pages 162–168, 2001.
15. Jinxi Xu and W. Bruce Croft. Improving the effectiveness of information retrieval with local context analysis. *ACM TOIS*, 18(1):79–112, 2000.