Improving Retrieval Feedback with Multiple Term-Ranking Function Combination

CLAUDIO CARPINETO, GIOVANNI ROMANO
Fondazione Ugo Bordoni
and
VITTORIO GIANNINI
WIND Telecomunicazioni S.p.A

In this paper we consider methods for automatic query expansion from top retrieved documents (i.e., retrieval feedback) which make use of various functions for scoring expansion terms within Rocchio's classical reweighting scheme. An analytical comparison shows that the retrieval performance of methods based on distinct term-scoring functions is comparable on the whole query set but considerably differs on single queries, consistent with the fact that the ordered sets of expansion terms suggested for each query by the different functions are largely uncorrelated. Motivated by these findings, we argue that the results of multiple functions can be merged, by analogy with ensembling classifiers, and present a simple combination technique based on the rank values of the suggested terms. The combined retrieval feedback method is effective not only with respect to unexpanded queries but also to any individual method, with notable improvements on the system's precision. Furthermore, the combined method is robust with respect to variation of experimental parameters and it is beneficial even when the same information needs are expressed with shorter queries.

Categories and Subject Descriptors: H.3.3. [**Information Storage and Retrieval**]: Information Search and Retrieval - retrieval models; relevance feedback; query formulation; H.3.1. [**Information Storage and Retrieval**]: Content Analysis and Indexing – indexing methods
General Terms: Algorithms, Theory, Design, Experimentation
Additional Key Words and Phrases: Information retrieval, Automatic query expansion, Retrieval feedback, Method combination, Short queries

1. INTRODUCTION

With the advent of the World Wide Web, information retrieval applications have been increasingly characterized by very short user queries and by large heterogeneous document collections (Kobayashi and Takeda 2000, Spink *et al*. 2001). These operational requirements may exacerbate well known limitations of current information retrieval systems related to the difficulty of dealing with synonymy (different words for describing the same things) and polysemy (same word to describe different things).

The paucity of query terms increases the difficulty of handling synonymy, because there is less chance of some important word cooccurring in the query and in the relevant documents. The large number of heterogeneous documents being searched makes the

Authors' addresses: C. Carpineto and G. Romano, Via B. Castiglione, 59, I-00142 Roma, Italy; email:{carpinet, romano}@fub.it. V. Giannini, Via del Giorgione, 21, I-00147 Roma, Italy; email: Vittorio.Giannini@inwind.it.

effects of polysemy more severe, as there is more chance of some ambiguous word cooccurring in the query and in the nonrelevant documents. As a result, the system may fail to retrieve the relevant documents while retrieving many slightly relevant or totally irrelevant documents.

To address the word mismatch problem, research in information retrieval has focused on methods for the automatic creation of a query "context". A number of approaches have been used, including term clustering (Sparck Jones 1971), statistical factor analysis (Deerwester et al. 1990), similarity thesauri (Qiu and Frei 1993, Jing and Croft 1994), collection independent thesauri (Aronson 1994), natural language processing (Strzalkowski 1995), and formal concept analysis (Carpineto and Romano 2000a). Most of these techniques, although employing different mathematical tools, can be seen as exploiting the term coocurrences in the documents to extract particular content relationships between all the terms contained in the collection. These various forms of query expansion are thus referred to as global (Xu and Croft 1996, 2000) or based on terminological knowledge structures (Efthimiadis 1996).

A simple, alternative approach to automatic query expansion is based on the identification of useful terms from the top retrieved documents, which is termed retrieval feedback, or pseudo-relevance feedback, or even local feedback. Retrieval feedback has been shown to produce good retrieval performance (Buckley *et al*., 1995; Xu and Croft, 1996; Fitzpatrick and Dent, 1997, Hawking *et al*. 1998**,** Mitra *et al*. 1998, Carpineto *et al*. 2001). Furthermore, it is much more efficient than query expansion based on knowledge structures, because the source of expansion terms is typically restricted to the set of top-ranked documents while the global term statistics possibly being used are usually obtained cheaply from the first pass retrieval.

However, the benefits of different retrieval feedback methods have been mostly evaluated with respect to using unexpanded queries and not by cross-system comparisons. Furthermore, there is a lack of microevaluations of the test performed, with little understanding of the behavior of the methods under study. The interest in retrieval feedback calls for a more analytic evaluation of competing approaches and motivates a research effort aimed at improving foundations and the performance of this technique.

We focus on retrieval feedback methods that combine Rocchio's classical reweighting scheme with various term-scoring functions. Such functions are used to select and weight the expansion terms that will be added to the unexpanded query by Rocchio's formula. In this paper we study how to take advantage of multiple term scoring functions to increase the effectiveness and the robustness of retrieval feedback.

The benefits of method combination for information retrieval have been investigated by a number of researchers, usually with a focus on merging multiple document rankings (Bartell *et al.* 1994, Lee 1997, Vogt and Cottrel 1998) or multiple document classifications (Larkey and Croft 1996, Hull et al. 1996). The exploitation of multiple evidence has been advocated also for query expansion, by emphasizing the advantages of using multiple query representations (Turtle and Croft 1991, Belkin *et al.* 1993) or multiple sources of terminological knowledge (Fidel and Efthimiadis 1995, Spink and Saracevic 1997). In this case, the representations are either provided by the user or generated by the systems, while the evidence combination is typically performed manually. To our knowledge, the use of method combination for fully automatic query expansion is new, with the exception of the work by Carpineto and Romano (1999), where this idea was put forward and some preliminary investigations were carried out.

The first contribution of this paper is an experimental comparison of several retrieval feedback methods based on distinct term-scoring functions. Such methods achieve comparable mean retrieval performance on the whole set of queries. Nonetheless, we learn through a query-by-query analysis that the same methods present large variations on individual queries, both with respect to the retrieval effectiveness and with respect to the characteristics of the ordered sets of terms suggested for query expansion. To analyze the latter, we adapt a formal method used in statistics, the Spearman rank-order correlation coefficient, to the experimental conditions that are of interest to retrieval feedback.

These observations suggest using combination strategies, by analogy with ensembling classifiers in the machine learning field. This is the second main contribution of the paper. We present a simple approach that combines the terms suggested by multiple term-scoring functions by using their rank values. The combined method obtains better retrieval performance results than each individual method on almost all evaluation measures, with tangible advantages for the first retrieved documents. Perhaps more interestingly from an application point of view, the benefits of the combined method are great even when the queries become shorter, provided that the quality of baseline retrieval is not too low. Our results hold across test collections of different nature and size, including the 10 gigabytes web dataset known as WT10g, and for different values of the other main method parameters.

The rest of the paper has the following structure. Section 2 introduces retrieval feedback and describes various term-scoring functions that can be used within Rocchio's formula to select and weight expansion terms. Section 3 is devoted to an experimental comparison of three retrieval feedback methods based on distinct term-scoring functions. After describing the experimental setting, we compare the retrieval performance of the three

methods both on the whole set of queries and on individual queries, then analyze term overlap and rank-order correlation between the lists of terms suggested by each methods. Section 4 describes an approach to combining the results produced by multiple term-scoring functions for use within Rocchio's formula. The performance of the combined method is evaluated with respect to the three methods introduced in Section 3 and the robustness of retrieval results with respect to parameter variation is discussed. Section 5 studies how the performance of retrieval feedback methods, including the combined one, varies as the queries become shorter. Finally, Section 6 provides some conclusions and directions for future work.

2. RETRIEVAL FEEDBACK TECHNIQUES

The main idea behind retrieval feedback dates back to the work by Attar and Fraenkel (1977) and Croft and Harper (1979). They assumed that the top few documents retrieved by an initial run were relevant, in the absence of any real relevance judgements, and considered all the terms contained in such documents as candidates for query reweighting, with (Attar and Fraenkel 1977) or without (Croft and Harper 1979) query expansion.

Recent variants of this general scheme for locating expansion terms make use of different knowledge sources such as past similar queries (Fitzpatrick and Dent 1997) and enriched collection (Singhal *et al*. 1999), or they try to improve the quality of the initial run by reranking the retrieved documents (Buckley et al. 1998, Mitra et al. 1998), or they use more sophisticated index units such as passages (Xu and Croft 1996, Hawking *et al.* 1998, Xu and Croft 2000) or document summaries (Lam-Adesina and Jones 2001).

By considering the lower ranked documents as nonrelevant documents, it is also possible to expand the original query with negatively weighted terms (Walker *et al*. 1998, Singhal *et al*. 1999), although it is not clear whether this approach improves retrieval performance (Hawking *et al*. 1998).

Having identified the candidate terms, the process which leads to a query with modified weights and terms consists of two main phases: selection of expansion terms to be included in the query, and reweighting of the expanded query.

2.1 Selection of expansion terms

The selection of expansion terms is usually performed by first ranking candidate terms, and then choosing the highest ranked terms. For ranking expansion terms, a number of different methods have been proposed, following two main conceptually distinct approaches.

One common solution is to rank the candidate expansion terms by using the term weights w_t computed for document ranking (Srinivasan 1996, Mitra *et al.* 1998, Singhal *et al.* 1999). Usually, the score assigned to each candidate is given by $\sum_{k=1}^{r} w_{t,k}$, where the summation index *k* ranges over the first *r* retrieved documents and $w_{t,k}$ is the weight of term t in document *k*. This approach is simple and computationally efficient, but it has the disadvantage that each term weight may more strongly reflect the usefulness of that term with respect to the entire collection rather than its importance with respect to the user query.

A different approach to expansion term selection is based on the difference between the distribution of terms in a set of relevant documents and the distribution of the same terms in the overall document collection. Several term-ranking functions based on distribution analysis have been used for selecting terms for query expansion, including Doszkoc's variant of CHI-square (Doszkoc 1978), Porter's simple difference in term distribution (Porter 1982), and RSV (Robertson 1990), which is, perhaps, the best known selection metric of this kind.

A more recent approach relies on the relative entropy, or Kullback-Leibler distance, between the two distributions, from which a computationally simple and theoretically justified method to assign relevance scores to candidate expansion terms can be derived (Carpineto *et al.* 2001).

2.2 Reweighting of expanded query

Most systems that perform retrieval feedback make use of Rocchio's formula (Rocchio 1971), as improved by Salton and Buckley (1990). Assuming that the relevant documents are the *r* top documents retrieved by the system and that the information about the non-relevant documents is absent, Rocchio's formula adapted to the retrieval feedback setting becomes:

$$w_{t,Q\exp} = \alpha \cdot w_{t,Qun\exp} + \frac{\beta}{\mathrm{r}} \cdot \sum_{k=1}^{r} w_{t,k} \qquad (1)$$

where $w_{t,Qexp}$ is the weight assigned to term *t* after query expansion, and $w_{t,Qunexp}$ and $w_{t,k}$ are the weights of term *t* in the unexpanded query and in the pseudo-relevant

documents, respectively, according to a weighting scheme applied to the whole collection.

Rocchio's formula is usually employed not only to reweight the expanded query but also to select the terms used in the expansion (e.g., Srinivasan 1996, Mitra *et al.* 1998, Singhal *et al.* 1999). In this case, as mentioned in the previous section, candidate expansion terms are ranked using their document-based, or Rocchio's, weights.

By contrast, one can use a different ranking scheme such as RSV to perform expansion term selection, and then apply expression (1) to the selected expansion terms. Various recent systems use this approach (e.g., Buckley *et al.*, 1995; Robertson et *al.*, 1995, Hawking *et al.*, 1998), although doubts have been expressed about the benefits for retrieval effectiveness in this case (e.g., Salton and Buckley 1990, Harman 1992, Carpineto *et al.* 2001).

A third possibility is to use term-scoring functions based on distribution analysis not only to select expansion terms, as in the previous case, but also to weight them in expression (1), instead of Rocchio's weights. The overall query reweighting function becomes

$$w_{t,Q\exp} = \alpha \cdot w_{t,Qun\exp} + \beta \cdot score_t \qquad (2)$$

where $score_t$ is the value assigned to term t by the term-scoring function being used. This method may better reflect the different importance of the same term with respect to the documents in the collection and with respect to the user query, thus resulting in better retrieval performance (Carpineto and Romano 1999, Carpineto *et al.* 2001).

The query expansion scheme given by expression (2) is the adopted choice in this paper. In the next section we evaluate its performance for several choices of the term-scoring functions, including Rocchio's weights.

3. COMPARING ALTERNATIVE TERM-SCORING FUNCTIONS FOR RETRIEVAL FEEDBACK

3.1 Experimental setting

In this section we describe the two test collections on which the experiments were performed, the baseline ranking system developed for each collection, and the term-scoring functions used for query expansion.

3.1.1 Test collections

The experiments were performed using two test collections. The first collection was TREC topics 401-450 on TREC disks 4 and 5, containing approximately 2 gigabytes of

data (i.e., the TREC-8 ad hoc task, hereafter referred to as TREC-8), and the second was TREC topics 451-500 on the WT10g 10 gigabytes web dataset (i.e., the TREC-9 web track, hereafter referred to as TREC-9). The TREC-8 collection consists mostly of newspaper or newswire articles, although there are also some government documents and computer science abstracts, while the TREC-9 document set is a snapshot of the web from 1997 from the Internet Archive (see (Voorhees and Harman 2000) and (Hawking 2001) for more detailed descriptions).

We chose these test collections because they are quite different from each other. Both are general domain databases, but the document set of TREC-9 is larger and much more heterogeneous than TREC-8. Furthermore, the TREC-8 documents are more controlled and reliable, whereas TREC-9 contains a huge number of typographic errors and spurious documents. The nature of their topics is also different, for TREC-9 topics were taken from real user queries.

For both collections, the full topic statement was considered, including "title", "description", and "narrative". The spelling errors contained in the title field of some TREC-9 topics were manually corrected. An example topic used in the experiments is the following:

<number> 451
<title> What is a bengal cat?
<description> Provide information on the bengal cat breed.
<narrative> Item should include any information on the bengal cat breed, including description, origin, characteristics, breeding program, names of breeders and catteries carrying bengals. References which discuss bengal clubs only are not relevant. Discussions of bengal tigers are not relevant.

3.1.2 Baseline document ranking systems

On a conceptual level, the one-pass ranking systems used for TREC-8 and TREC-9 were similar. The whole body of each document was indexed except for HTML tags, which were removed from TREC-9 pages. The systems performed word stopping and word stemming, using a very large *trie*-structured morphological lexicon for English (Karp et al. 1992). Single keyword indexing was performed for both test collections, and link

information in web data was not used.[1] In order to reduce the inherent web data noise, the TREC-9 system also performed word pruning by removing very rare, ill-formed or exceedingly long words. After indexing, the TREC-8 topics were described with an average of 12.25 distinct terms, with a minimum of 4 and a maximum of 33 terms; the TREC-9 topics contained, on average, 17.76 terms (minimum 10, maximum 25).

In the first pass ranking, the systems used the following Okapi formula (Robertson *et al*. 1999) for computing a similarity measure between a query *q* and a document *d*:

$$sim(q,d) = \sum_{t \in q \wedge d} w_{t,d} \cdot w_{t,q} \qquad (3)$$

$$\text{with } w_{t,d} = \frac{(k_1 + 1) \cdot f_{t,d}}{k_1 \cdot \left[(1-b) + b \cdot \frac{W_d}{avr\_W_d}\right] + f_{t,d}} \qquad (4)$$

$$\text{and } w_{t,q} = \frac{(k_3 + 1) \cdot f_{t,q}}{k_3 + f_{t,q}} \cdot log \frac{N - f_t + 0.5}{f_t + 0.5} \qquad (5)$$

where k_1, k_3 , and b are constants which were set to 1.2, 1000, and 0.75 respectively. W_d is the length of document *d* expressed in words and avr_W_d is the average document length in the entire collection. The value *N* is the total number of documents in the collection, f_t is the number of documents in which term *t* occurs, and $f_{t,x}$ is the frequency of term *t* in either document *d* or query *q*.

Although conceptually similar, the ranking systems used for the TREC-8 and TREC-9 test collections have distinct implementations. The former, coded in Common Lisp, was developed in the context of our participation in TREC-8 (Carpineto and Romano 2000b). Its first-pass ranking takes a few seconds per query on a SUN-Ultra workstation equipped with 512 megabytes of RAM. To deal with the new larger collection, the system used for

[1] The use of link information for topic relevance retrieval from the WT10g collection did

TREC-8 was completely re-engineered and implemented in ESL, a Lisp-like language that is automatically translated into ANSI C and then compiled by gcc compiler. Currently, the new system developed for experimenting with the TREC-9 test collection allows sub-seconds one-pass searches on the WT10g document set, using a 550 MHz Pentium III with 256 megabytes of RAM running Linux.

3.1.3 Term-scoring functions used for query expansion

In the second pass ranking, expression (3) and (4) were left unchanged while expression (5) was replaced by expression (2). More specifically, the first term of expression (2), concerning the term weight in the unexpanded query, was given by expression (5), and the second term of expression (2) was computed by using different term-scoring functions.

The following three functions were tested in our experiments ($p_{t,R}$ and $p_{t,C}$ indicate the probability of occurrence of a term *t* in the set of pseudo-relevant documents R and in the whole collection C, respectively, and $w_{t,k}$ is the weight of term t in document *k* :

- Rocchio's weights: $\text{score}_t = \sum_{k=1}^{r} w_{t,k}$ (6)

- Doszkocs' variant of CHI-squared (CHI-1):[2] $\text{score}_t = [p_{t,R} - p_{t,C}] / p_{t,C}$ (7)

- Kullback-Leibler distance (KLD): $\text{score}_t = p_{t,R} \cdot \log [p_{t,R} / p_{t,C}]$ (8)

To estimate $p_{t,R}$, we used the ratio between the frequency of *t* in *R*, treated as a long string, and the number of term tokens in *R*; analogously, to estimate $p_{t,C}$, we used the ratio between the frequency of *t* in *C* and the number of term tokens in *C*. The estimation of probabilities is an important issue because it might affect performance results. In addition to the maximum likelihood estimate, we also tried different estimation functions such as the number of pseudo-relevant documents that contain the term (Buckley *et al.*,

---

not produce any significant benefit in earlier experiments (Hawking et al. 2001).

[2] This variant, unlike the standard CHI-squared function, has the advantage of rejecting those terms that are good indicators for nonrelevance; i.e., such that $p_C(t) >> p_R(t)$.

1995; Robertson *et al*., 1995), but the latter method was found to produce worse retrieval effectiveness for any term-scoring function tested in the experiments.[3]

As the document-based weights used for the unexpanded query and the distributional scores used for the expansion terms had different scales, the computation of expression (2) for the distributional methods CHI-1 and KLD required normalization. Each weight in the original query and each weight in the set of expansion terms was normalized by the maximum corresponding weight, then the normalized values were summed up.[4]

It should be noted that the choice of the term-scoring functions to test was influenced by our interest in method ensembling. As the combination works well when the methods being combined have comparable performance (Dietterich 1997, Schwenk and Gauvain 2000), we focused on those term-scoring functions exhibiting similar retrieval effectiveness. Other possible functions, such as RSV and CHI-squared, were not chosen for analytical comparison because they yielded worse mean retrieval performance (see (Carpineto et al. 2001) for the retrieval effectiveness of RSV and CHI-squared on the TREC-8 test collection).

All term-scoring functions tested in the experiment required four user-supplied values: the number of pseudo-relevant documents, the number of terms considered for inclusion in the expanded query, and the values of alpha and beta in expression (2). Consistently with other experimental studies on the TREC-8 collection, the values of the first two parameters were set at 10 and 40, and the values of alpha and beta at 1 and 2. For the TREC-9 test collection, characterized by a lower quality of initial retrieval, we used the same number of expansion terms but fewer pseudo-relevant documents and a higher value of the ratio between alpha and beta ($\alpha = 1$, $\beta = 0.2$). The effects of different parameter choices will be discussed later.

For practical implementation, the execution of the query expansion step required the computation of the collection frequencies, which were directly stored in the inverted file built from the set of documents, and of the retrieval feedback frequencies, which were determined through one iteration over the first retrieved documents. Thus, the additional time necessary to perform just query expansion was usually short for both test collections. The overall time necessary to compute the second-pass ranking was

---

[3] It should be noted that for other term-scoring functions such as RSV the latter estimation method may produce better results than the maximum likelihood estimate (Carpineto *et al.* 2001).

[4] We have not attempted to optimize the normalization scheme. Alternative methods that not only scale data into a same range but also increase its uniformity could be more effective (e.g., Montague and Aslam 2001).

comparable to that of first-pass ranking, as long as the expanded query did not contain many more terms than the original query.

3.2 Mean retrieval performance

For each database and for each query, we ran the system with query expansion three times, one for each possible selection of the term-scoring function used in the expansion scheme. Table 1 and Table 2 show the retrieval performance of each method on TREC-8 and TREC-9, respectively, along with the performance improvement over unexpanded query, used as a baseline. Performance was measured using the following evaluation measures:

AV-PREC: average precision,

PREC-AT-5: precision at five retrieved documents,

PREC-AT-10: precision at ten retrieved documents.

The measures are averaged over the query set. Asterisks are used to denote that the difference is statistically significant, using a two-tailed paired *t* test with a confidence level in excess of 95%.

Table 1. Comparison of mean retrieval performance on TREC-8.

	Unexpanded	Rocchio	CHI-1	KLD
AV-PREC	0.2718	0.2972	0.2918	0.3053
		+9.33%*	+7.35%*	+12.32%*
PREC-AT-5	0.5960	0.6040	0.5480	0.6000
		+1.34%	-8.05%	+0.67%
PREC-AT-10	0.4920	0.5160	0.4840	0.5120
		+4.88%*	-1.63%	+4.07%

Table 2. Comparison of mean retrieval performance on TREC-9.

	Unexpanded	Rocchio	CHI-1	KLD
AV-PREC	0.2110	0.2201	0.2234	0.2303
		+4.28%*	+5.88%*	+9.11%*
PREC-AT-5	0.4000	0.4160	0.4280	0.4280
		+4.00%	+7.00%	+7.00%
PREC-AT-10	0.3320	0.3600	0.3540	0.3700
		+8.43%*	+6.63%*	+11.45%*

The results of Table 1 and 2 show that each expanded method performed better than the unexpanded method for all evaluation measures and on both test collections, except for CHI-1's PREC-AT-5 and PREC-AT-10 on TREC-8. In particular, the expanded methods clearly outperformed the unexpanded method with respect to AV-PREC, with the differences being always statistically significant. For the first retrieved documents, where it is harder to improve over unexpanded query, the performance of query expansion was much better than the baseline on TREC-9, notably for PREC-AT-10, while it was only marginally better, or even worse, than baseline retrieval on TREC-8.

The performance of query expansion on TREC-9 was thus really good, even better than TREC-8. Compared to some recent findings about the low utility of query expansion for the former collection (Hawking 2001), our results were much more favourable. This may be attributed at least in part to the fact that we had a greater opportunity for parameter tuning (we will discuss this issue in depth in Section 4.3). Furthermore, the removal of spurious words performed at indexing time (see Section 3.1.2) considerably reduced the number of typographical errors in documents, which was pointed out as one of the causes for poor query expansion (Kraaij and Westerveld 2001).

Table 1 and 2 also suggest that although KLD achieved better results on most data points the three expansion methods obtained comparable average performance on both collections. Perhaps more importantly, the pairwise differences for each performance measure were never statistically significant. Thus, the above analysis confirms that no single method was clearly superior to the others.[5]

[5] This results hold for TREC-8 and TREC-9; a similar experiment conducted on the TREC-7 test collection gave more favourable results for the KLD method (Carpineto *et al.* 2001)

As the three methods rely on distinct theoretical foundations and use different mathematical functions, we hypothesized that despite their similar mean effectiveness they would present considerable performance variation on individual queries. Therefore we decided to test this hypothesis through a query by query analysis.

3.3 Retrieval performance on individual queries

For each query and for each expansion method, we computed the difference between the retrieval performance of expanded query and that of unexpanded query, considering average precision as the performance measure. In Figure 1 and Figure 2 we report, for each query, the minimum and maximum of such differences on the test collections TREC-8 and TREC-9. Thus, the length of each bar depicts the range of performance variations over the unexpanded case attainable by the three methods on each query.

A first result of this experiment is that for the majority of queries (62 out of 100), the variations with respect to unexpanded query (x axis) were either all positive (48 times) or all negative (14 times), as might be expected, whereas in each of the remaining 38 queries at least one method hurt and one improved performance. The extent to which the retrieval performance was affected by query expansion was substantial, as the results differed by more than 5% from the baseline for 118 cases on TREC-8 and 135 cases on TREC-9.

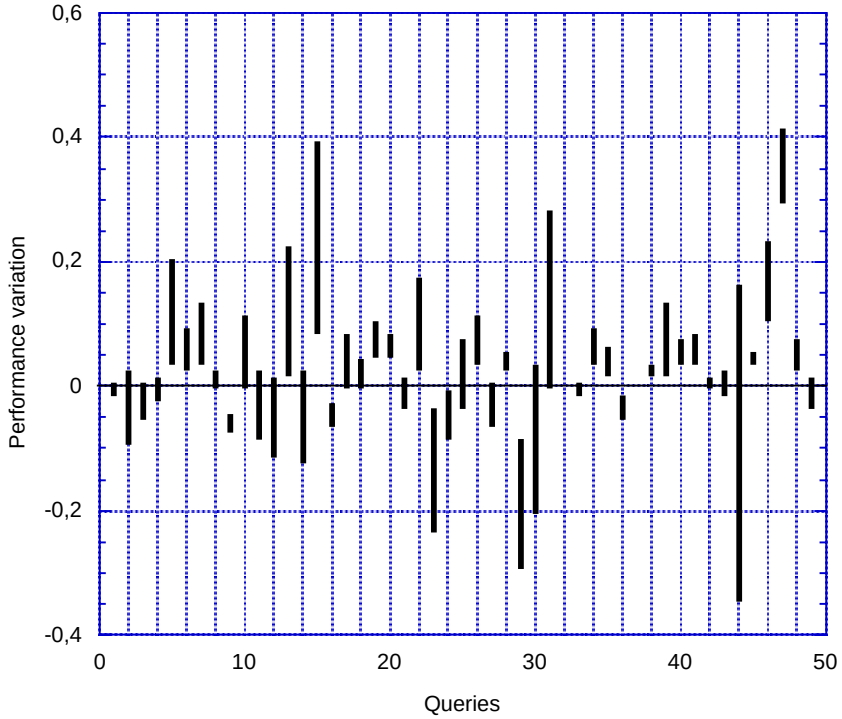

Figure 1. Performance variation of retrieval feedback methods on individual queries for TREC-8 (unexpanded query is the baseline).

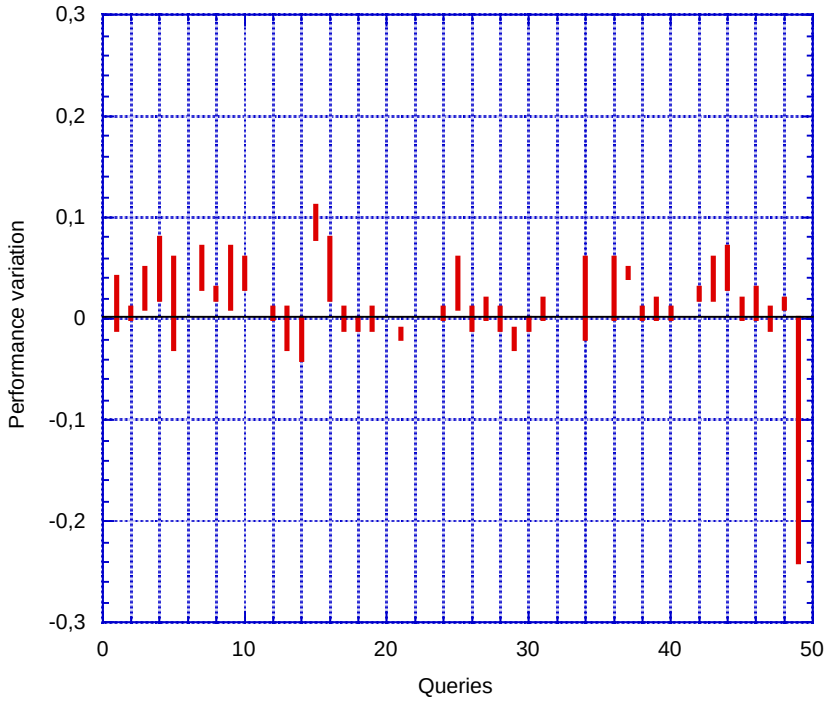

Figure 2. Performance variation of retrieval feedback methods on individual queries for TREC-9 (unexpanded query is the baseline).

Furthermore, and perhaps more importantly, the inter-method variations on single queries were ample, with a mean absolute value of 0.087 for TREC-8 and 0.029 for TREC-9. The variations observed for TREC-9 were, on an absolute scale, more limited than TREC-8, but we should also consider that the baseline retrieval performance for the former collection was considerably lower than the latter. As the three methods achieved similar mean performance over the query set, it appears that methods which generated better terms on some queries produced poorer terms on others.

To make this observation more precise, we counted the number of queries for which each method, achieved the best, median, or worst performance. The results are shown in Table 3. The main finding is that each method behaved better (or worse) than the others a comparable number of times, thus ruling out the possibility that the performance of any method was determined by an exceptionally good (or bad) result on a few specific queries.

Table 3. Ranked performance.

	TREC-8			TREC-9		
	Rocchio	CHI-1	KLD	Rocchio	CHI-1	KLD
1st	15	12	23	16	13	21
2nd	19	18	13	19	11	20
3rd	16	20	14	15	26	9

Figure 1 and 2 and Table 3 demonstrate the potential of multiple term-ranking functions to improve retrieval performance. It turns out that if we were able to select the best method for each query, we would get a mean AV-PREC value as remarkably high as 0.3422 for TREC-8, with a performance gain of 25.90% over unexpanded query and of 12.08% over the best expansion method, and a value of 0.2380 for TREC-9, with a gain of 12.79% over baseline and of 3.33% over the best method. Thus, there really seems to be a lot of scope for performance improvement with method combination, although at this point it is still not clear how this can be achieved.

Before addressing the issue of combining the results produced by different methods, we need a better understanding of why those results were different. This, in turn, might provide useful clues as to how to proceed to combine such results. In the next section, we analyze and compare, still at the query level, the ranked term lists produced by each method, prior to their utilization in the query expansion scheme.

3.4 Term overlap

The differences in performance on single queries could be easily explained if the individual methods were suggesting distinct terms. A simple method to evaluate this aspect is to use the overlap coefficient for pairwise combinations of the three sets of suggested terms (Katzer *et al.* 1982, Xu and Croft, 1996). The results, restricting each term ranking to the first 40 terms, are shown in Table 4.

The overlap varied depending on which pair of methods was considered, while remaining substantially stable across the collections. In fact, the results for TREC-9 were slightly higher than TREC-8, probably due to the smaller number of pseudo-relevant documents used for term suggestion (3 versus 10).

Some statistics about the terms that were exclusively suggested by the single methods were also collected. The mean number of unique terms was 15.96 (σ =3.26) for

Rocchio, 18.60 (σ =5.33) for CHI-1, and 5.16 (σ =3.25) for KD, on the TREC-8 test collection. The corresponding figures for TREC-9 were: 15.16 (σ =4.62) for Rocchio, 15.30 (σ = 9.37) for CHI-1, and 4.82 (σ = 4.00) for KLD.

It is interesting to note that Rocchio and CHI produced a similar substantial number of unique terms, while KLD yielded very few unique terms. This behavior held across both collections. Thus, it seems that in this respect Rocchio and CHI-1 are quite different from each other, with KLD lying in between.

One possible explanation is that the sensitivity of each method with respect to the main term-ranking variables is different. Specifically, Rocchio's weights, computed through expressions (6) and (4), are more strongly biased towards promoting terms which occur frequently in the pseudo-relevant documents, even though they are relatively frequent in the collection, whereas CHI-1, using expression (7), tends to give more importance to terms which are very infrequent in the collection. The behavior of KLD, according to expression (8), is intermediate, and thus it shares most of its terms with either Rocchio or CHI-1. We experimentally computed the mean probability of occurrence of the unique terms suggested by each method in the whole collection; in fact, the highest values were found for Rocchio (0.000706 on TREC-8, 0.001300 on TREC-9), the lowest for CHI-1 (0.000016 on TREC-8, 0.000021 on TREC-9), and the intermediate ones for KLD (0.000247 on TREC-8, 0.000227 on TREC-9).

The results reported here may represent an indication that CHI-1 and Rocchio behaved differently, because the two methods produced many unique terms and presented a low overlap, but for the other two pairs of methods the indications are elusive because the degree of overlap was, on average, fairly high. Furthermore, Table 3 shows that the variability across queries was high, especially for the CHI-1/Rocchio pair. A more powerful method to evaluate the differences between term rankings should consider not only which terms are suggested by each method but also how those terms are ranked.

Table 4. Overlap between pairwise term rankings (restricted to the first 40 terms).

	TREC-8			**TREC-9**		
	Rocchio	CHI-1	KLD	Rocchio	CHI-1	KLD
Rocchio	40.00 (σ = 0.00)			40.00 (σ = 0.00)		
CHI-1	10.52 (σ = 4.07)	40.00 (σ = 0.00)		14.18 (σ = 8.65)	40.00 (σ = 0.00)	
KLD	23.96 (σ = 3.22)	21.32 (σ = 5.24)	40.00 (σ = 0.00)	24.66 (σ = 4.55)	24.52 (σ = 9.23)	40.00 (σ = 0.00)

3.5 Term-ranking correlation

The Spearman rank-order correlation coefficient is perhaps the best known statistics based on ranks (Siegel and Castellan 1988). It is a measure of association between two variables which requires that both variables be measured in at least an ordinal scale so that the individual under study may be ranked in two ordered sets. When there are no ties in the data,[6] the Spearman rank-order correlation coefficient is given by:

$$r_s = 1 - \frac{6 \sum_{i=1}^{N} d_i^2}{N^3 - N} \qquad (9)$$

where d_i is the difference in ranks on the two variables for each individual and N is the number of individuals . The value of r_s ranges from -1 to +1, regardless of the number of individuals chosen.

In our case, the application of expression (9) is not straightforward. Since only restricted rankings are used for query expansion (i.e., the first 40 terms), we need a way of calculating the contribution of those terms in the restricted ranking produced by one method that are not contained in the restricted ranking of the other method. To overcome this problem, we assume that, for each method in a given pair, the missing terms are randomly added to the corresponding restricted ranking, right after its last item. Thus, each augmented ranking used to calculate expression (9) will contain the union of the two original restricted rankings. With this assumption, the contribution of the missing terms to the summation in expression (9) will, in general, be lower than that obtained by considering the rank of those terms in the full ranking, and thus the result can be seen as a lower bound of the actual difference. Clearly, if our hypothesis that the rankings are different were confirmed under this assumption, it would *a fortiori* hold for the more general case.

At this point we are left with the problem of calculating the Spearman rank-order correlation coefficient given by expression (9) using the two augmented rankings including a random component. Let R_1 and R_2 be the two restricted rankings; the order of

[6]Since we deal with numerical variables, ties can hardly occur; if any, they are resolved randomly.

choice is irrelevant because the measure is symmetrical. Let l_1 and l_2 be the length of restricted rankings R_1 and R_2, respectively; in our case, $l_1 = l_2 = 40$. Let m_1 be the number of terms contained only in R_2 that have been added to R_1 to form the augmented ranking R_1*, and m_2 be the number of terms contained only in R_1 that have been added to R_2 to form the augmented ranking R_2* (in our case, $m_1 = m_2$).

For each term contained in R_1*, we need to compute the difference d in ranks on R_1* and R_2*. There are three possible cases. (a) The term was contained in both R_1 and R_2: the calculation is straightforward because we merely use the ranks of that term in each restricted ranking. (b) The term was contained in R_1 but not in R_2: we need to use the terms added to R_2. (c) The term was contained in R_2 but not in R_1: we need to use the terms added to R_1. In the latter two cases, we need to compute a difference between an exact rank on one restricted ranking and a random rank on the other augmented ranking. To compute such a difference, one can consider all possible ranks that can be assigned to the missing term when it is added to the original restricted ranking in which it was not contained, then average over the corresponding differences in ranks.

Thus, the difference in ranks for each term contained only in R_1 is given by:

$$d^2 = \frac{\sum_{x=1}^{m_2} (l_2 + x - pos_1)^2}{m_2} \quad (10)$$

where pos_1 is the rank of the term on R_1.

By simple mathematical passages, expression (10) can be rewritten in a closed form:

$$d^2 = (l_2 - pos_1)^2 + (l_2 - pos_1)(m_2 + 1) + \frac{2m_2^2 + 3m_2 + 1}{6} \quad (11)$$

For the terms contained only in R_2, the dual expressions of expression (10) and (11) can be derived by simply replacing l_2 with l_1, pos_1 with pos_2, and m_2 with m_1.

We are now able to compute the value of r_s between the augmented rankings, using the just described procedures to compute the values of d_i within expression (9).

The next step is to test the significance of r_S. As the number of individual terms in the augmented rankings being tested ranged from 58 to 80,[7] the significance of an obtained r_S under the null hypothesis that the two methods under study are independent may be conveniently tested by the statistics

$$z = r_s \sqrt{N-1} \quad (12)$$

where a z as large as 1.960 is significant at the .05 level for a two-tailed test (Siegel and Castellano 1988).

Using the procedure described above, we computed, for each query, for each pair of methods, and for each collection, the value of r_S, from which we derived the value of z by expression (12).

In Figures 3 and 4, we show for the pair CHI-1/Rocchio on TREC-8 and TREC-9, respectively, the value of z – 1.960 for each query. Analogous figures are shown for the pair KLD/CHI-1 (i.e., Figures 5 and 6) and KLD/Rocchio (i.e., Figures 7 and 8). Using this representation, the hypothesis that there is no association between the two methods should be accepted or rejected depending on whether the bar is below or above zero.

Figures 3 and 4 clearly show that the term rankings produced by CHI-1 and Rocchio were quite different, because there was no association at all on the TREC-8 collection and no association for 48 of the 50 queries on TREC-9, with a higher confidence level than usually necessary to support such a claim.

Figures 5 and 6 suggest that although KLD and CHI-1 presented a substantial amount of term overlap (see Table 4) they were mostly uncorrelated. No association was found for 48 queries on TREC-8 and for 29 queries on TREC-9.

For the KLD/Rocchio pair (Figures 7 and 8), a higher correlation was found, although the majority of queries remained uncorrelated. The association results were split evenly on TREC-8, while the term rankings were uncorrelated for 34 queries on TREC-9. This was also the pair with the highest term overlap.

Thus, the results of the rank-order correlation were related to but not coincident with the term overlap statistics. For instance, it may be the case that a higher correlation will be found for a pair with a lower overlap (e.g., KLD-Rocchio on TREC-8 versus KLD-Rocchio on TREC-9). More importantly, it is possible that low rank-order correlation coefficients correspond to high overlap coefficients.

[7] The latter value, implying that the sets of terms suggested by the two methods have no term in common, was obtained by CHI-1 and Rocchio on query 401 (TREC-8).

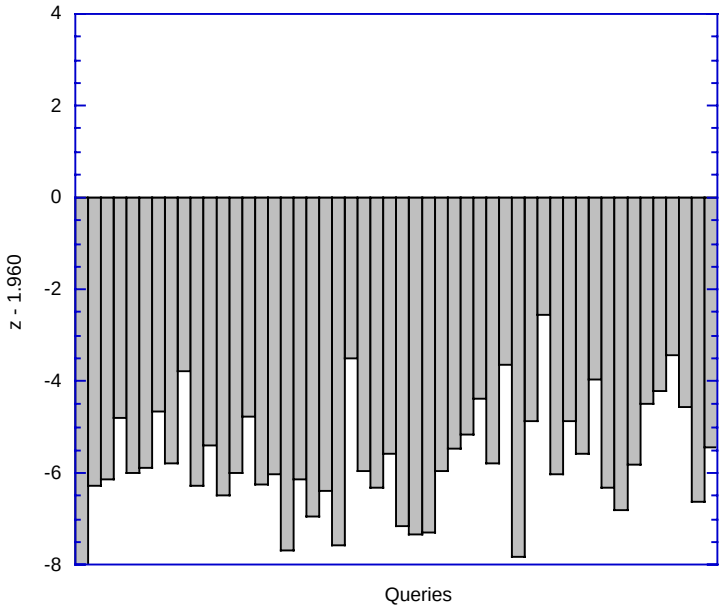

Figure 3. Significance of Spearman rank-order correlation coefficient between CHI-1 and Rocchio on TREC-8.

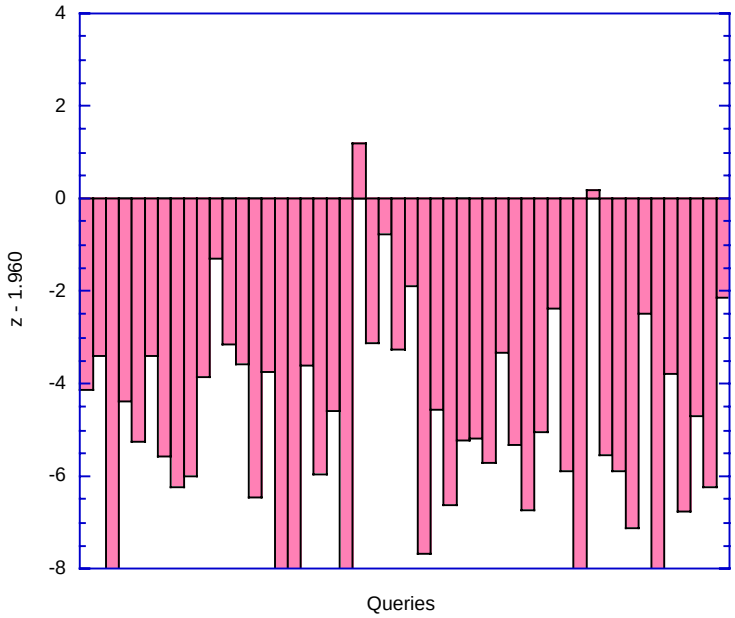

Figure 4. Significance of Spearman rank-order correlation coefficient between CHI-1 and Rocchio on TREC-9.

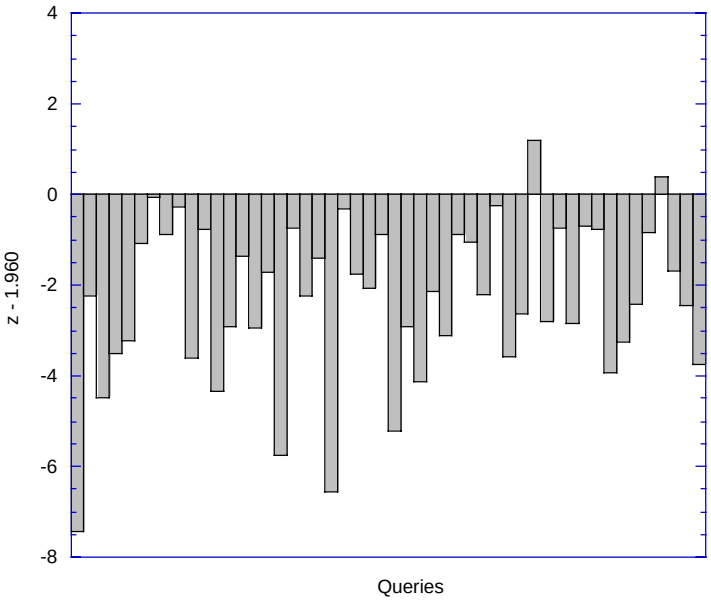

Figure 5. Significance of Spearman rank-order correlation coefficient between KLD and CHI-1 on TREC-8.

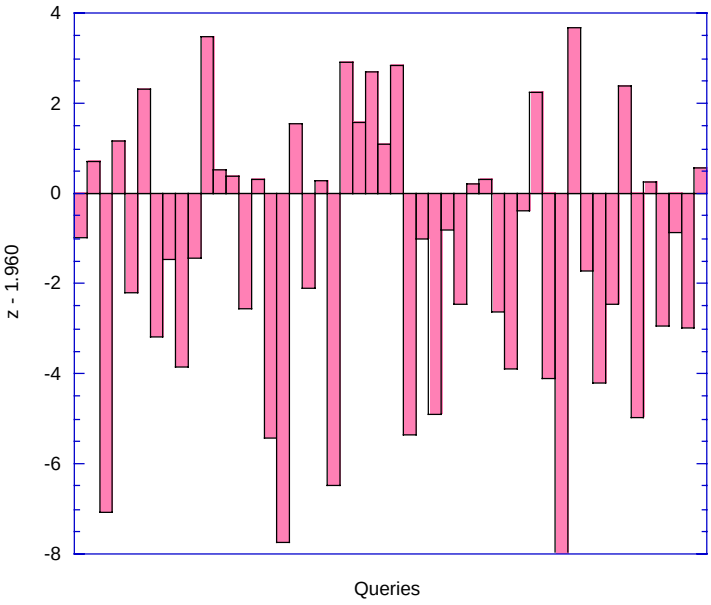

Figure 6. Significance of Spearman rank-order correlation coefficient between KLD and CHI-1 on TREC-9.

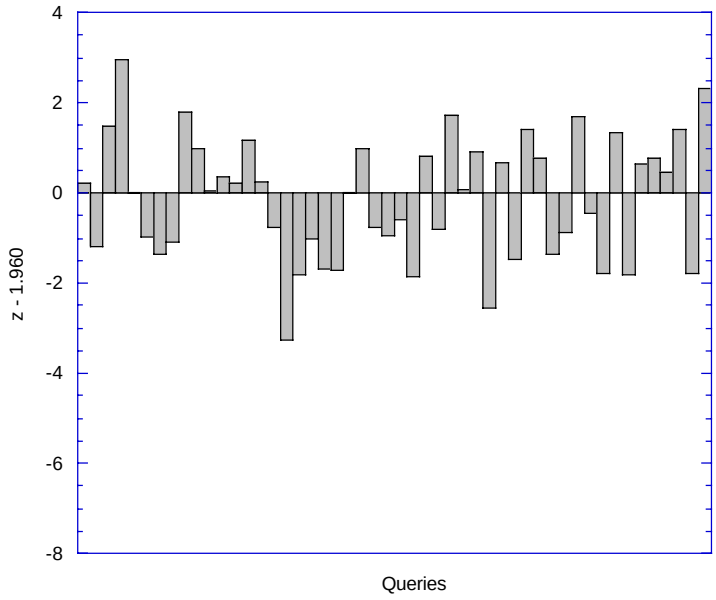

Figure 7. Significance of Spearman rank-order correlation coefficient between KLD and Rocchio on TREC-8.

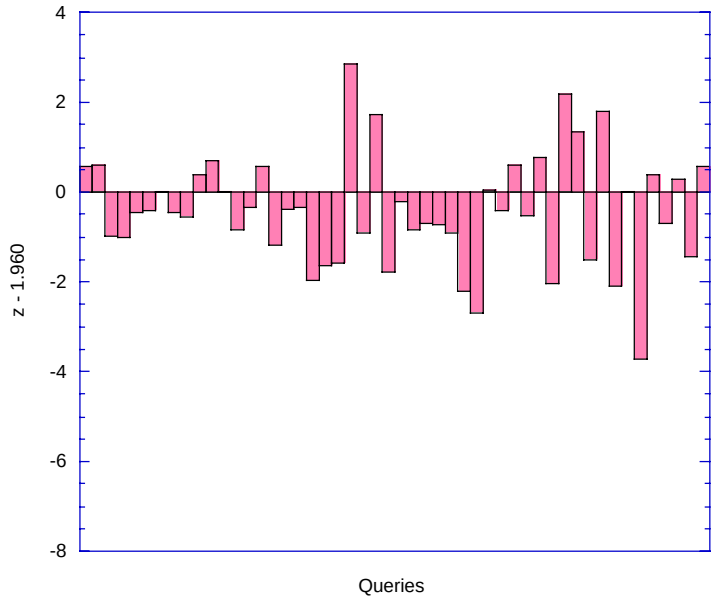

Figure 8. Significance of Spearman rank-order correlation coefficient between KLD and Rocchio on TREC-9.

The general lesson that one can learn from the above analysis is that the functions tested in our experiments behaved differently, either by suggesting distinct terms, or, when this was not the case, by ranking the same terms in a different order. The retrieval performance of expanded queries produced with different functions was thus found to differ considerably, consistently with other results that different representations of a query retrieve different sets of documents (Katzer *et al*. 1982, Saracevic and Kantor 1988, Belkin *et al.* 1993).

It is useful to compare our results with those reported in (Efthimiadis 1993). This is one of the few studies that compare the performance of multiple term-ranking functions for query expansion by analyzing the composition of the lists of suggested terms and not the system's overall retrieval effectiveness. Efthimiadis found that several term-ranking functions produced much more similar performance results than those reported in this paper. Nonetheless, his results may not be in contradiction with ours because the former were determined under very different and somewhat specific experimental conditions mimicing an interactive query refinement task. More specifically, the differences between alternative methods were evaluated either by using the percentage of relevant terms contained in each ranking without considering their position, or by averaging the ranks

assigned by each method to five particular terms considered as the best terms for a given query. While this experimental protocol may be suitable for interactive query refinement, because the ranked order is of lower importance and the user may be interested in only a few terms, it does not seem able to disclose more regular and subtle differences between term rankings which may affect the outcome of automatic query expansion techniques.

The fact that in our experiments the individual methods produced different results on individual queries while predicting, on average, equally good terms suggests trying combination strategies with the aim of retaining, for each query, the most informative terms. This issue is discussed below.

4. COMBINING MULTIPLE TERM-SCORING FUNCTIONS WITHIN ROCCHIO'S FORMULA

4.1 Combination method

One simple way to combine multiple term rankings is to take a linear combination of their relevance scores. This technique has been widely used to merge multiple document rankings (e.g., Selberg and Etzioni 1995, Lee 1997, Vogt and Cottrel, 1998), mainly based on Saracevic and Kantor (1988)'s early observation that the more runs a document is retrieved by the more likely it is that the document is relevant.

However, taking a linear combination did not work well in our case because the relevance scores were not comparable across the tested methods. We also tried normalization of relevance scores prior to combination, by normalizing each score by the maximum score in the corresponding method, but the obtained retrieval performance was poor again, probably due to the different distribution of the scores produced by each method.

Besides numerical problems, it is possible that a linear combination was altogether an unsuitable model for the situation at hand. The analysis presented above suggests that each function tested in our experiment was producing unusually inaccurate (or accurate) estimates of relevance for at least some terms, relative to the other functions. This phenomenon, which can be seen as an instance of the dark horse effect (Vogt and Cottrel 1998), cannot be exploited by a linear combination mode because it would require taking into account the relative behaviors of the methods being combined. These observations motivated a different approach, inspired by combination of classifiers.

Recent research in machine learning and information retrieval has shown that ensembling multiple classifiers, whether produced by single or different learning algorithms, may be a viable technique for improving classification accuracy (Larkey and Croft, 1996; Breiman, 1996; Dietterich, 1997). Two keys to success are that the individual classifiers

must disagree with one another and that their average accuracies must be comparable. In this case one can try to guess the right prediction by taking a majority vote, in the hope that the single classifiers make uncorrelated errors. A similar approach has been also used with good results to combine single words predicted for a given position by different continuous speech recognizers (Fiscus 1997, Schwenk and Gauvain 2000).

Going back to the retrieval feedback setting, we have seen that the methods tested in our experiments obey the first two assumptions mentioned above, i.e., they present comparable mean performance and produce different results on individual queries. At this point one can hypothesize that the individual term-ranking functions make uncorrelated errors in suggesting new terms, i.e., when a term erroneously gets a high (low) rank in one method the same term gets a low (high) rank in the other methods, so that a majority procedure can correctly rank the term.

The next step is the specification of such a majority procedure to ensemble the results of the individual methods. The output of each term-scoring function is represented by a set of variables with numerical scores rather than with discrete values as in concept classification, so a simple majority vote cannot be used. As the individual functions presented quite large variations on the order in which terms were ranked, we decided to focus on the rank values, ignoring the relevance scores. Although relevance is more often exploited to combine multiple results than rank, the latter has been used in several combination studies (e.g., Kantor 1995) and it has produced better results in certain cases (Lee 1997, Cohen *et al*. 1999).

Several rules for fusion of ranks can be used, such as taking their minimum, maximum, or sum (Fox and Shaw 1994, Kantor 1995, Lee 1997, Carpineto and Romano 1999). Here we set the combined rank equal to the median of the three separate ranks, similar to Kantor (1995). The main justification for our choice is as follows. If each method makes a limited number of errors, and if the three methods make uncorrelated errors, it is likely that there will be at most one wrong evaluation on each single decision. The median rule ensures that a given term is ranked correctly whenever that term is ranked correctly by at least two methods. As a result, the negative effects of wrong preferences assigned to single terms on retrieval performance should be reduced. We will test this hypothesis experimentally.

Once the ranks have been merged, each term should be assigned a relevance score for use in Rocchio's formula through some mathematical function of its combined rank. An important requirement of such a function is that its value should decrease monotonically as the rank increases, because in this way we are sure that if one expansion term, for a given query, was correctly ranked ahead of another term, then the former will receive a

greater weight in the expanded query. This is one of the main motivations for using a weighting function other than the document-based one within Rocchio's formula (Carpineto *et al.* 2001). In our experiments, the relevance score of a term was tranformed from the combined rank of the term using the following simple function:

$$score_t = 1 / rank_{t,COMB} \quad (13)$$

although more elaborate functions are conceivable (Lee 1997). The relevance scores obtained this way were then used in expression (2) to reweight the expanded query.

4.2 Retrieval performance of combined method

The combined expansion method was tested for performance using the same parameter setting as previous experiments with individual methods. The results are reported in Table 5 and 6. We show also the percentage improvement of the combined method over the retrieval performance of the unexpanded query and individual methods (see Table 1 and 2). It should be noted that such relative improvements are consistent with those obtained in a similar experiment by Carpineto and Romano (1999), for a different data set and combination method. The absolute retrieval results reported in by Carpineto and Romano (1999) were however much lower than those reported here due to the use of a simplistic indexing function.

Table 5 and 6 show that the combined method obtained better results than any individual method for all evaluation measures and across both collections, with most differences being statistically significant. The performance improvement was almost always more marked over CHI-1 and Rocchio than over KLD, consistent with the comparatively better results of KLD, although some significant gains were also achieved with respect to the retrieval performance of KLD.

Furthermore, the performance improvement due to method combination observed for TREC-9 was, in general, higher than that obtained for TREC-8. The gain was clearly larger for the unexpanded case and Rocchio, it was partially better for KLD, and partially worse for CHI-1, with an overall increase of the number of statistically significant differences.

Table 5. Performance improvement of combined method on TREC-8.

	Combined method	Improvement over unexp.	Improvement over ROCCHIO	Improvement over CHI-1	Improvement over KLD
AV-PREC	0.3088	+13.61%*	+3.93%	+5.85%*	+1.17%
PREC-AT-5	0.6200	+4.02%	+2.65%	+13.14%*	+3.33%
PREC-AT-10	0.5460	+10.97%*	+5.81%*	+12.81%*	+6.64%*

Table 6. Performance improvement of combined method on TREC-9.

	Combined method	Improvement over unexp.	Improvement over ROCCHIO	Improvement over CHI-1	Improvement over KLD
AV-PREC	0.2413	+14.36%*	+9.63%*	+8.01%*	+4.77%
PREC-AT-5	0.4480	+12.00%*	+7.69%*	+4.67%	+4.67%
PREC-AT-10	0.3900	+17.47%*	+8.33%*	+10.16%*	+5.40%*

Interestingly, for TREC-9, the performance of combined method was even better than the performance that we would obtain by selecting for each query the best individual method (see Section 3.3). Although somewhat surprising, this is perfectly consistent with the approach proposed here, because the ensembling strategy does not combine the performance results of individual methods but truly integrates their performance components. On the other hand, it is clear that this phenomenon was observed for TREC-9 and not for TREC-8 because the former database was characterized more by lower inter-method performance variations on single queries than the latter.

It is also worth noting that the good performance results obtained by the combined method were achieved by using term rankings which, although largely uncorrelated, presented a relatively high overlap of terms. This can be seen as analogous to recent findings in document ranking combination that show that improvements are possible even if the individual methods retrieve similar sets of documents (Turtle and Croft 1991, Lee 1997).

A second main finding of our experiments is that the combined method significantly improved over unexpanded query even when only the first retrieved documents were considered for evaluation, namely for PREC-AT-5 and PREC-AT-10. This particular case is especially important for Web-based retrieval, because the number of documents indexed by most search engines has been increasing by many orders of magnitude while

the user's ability and willingness to look at documents has remained very limited, usually confined to the top ten (or tens) retrieval results. The reportedly limited benefits of retrieval feedback for the system's precision may have been one of the causes that have so far hindered the utilization of this technique by Web retrieval systems. Our results indicate that retrieval feedback may be effective even at this task.

We should emphasize that in this experiment the function used to reweight the expanded query by the combined method was different from that used by each single method; i.e., rank values versus relevance scores. In order to rule out the possibility that the differences in performance were due to this different experimental condition, we reran the experiments concerning individual methods using the inverse of term rank as term-scoring function, similar to the combined method. The performance of individual methods was found to decrease slightly, compared to the results shown in Table 1, and thus the improvement due to method combination in this case was similar, and occasionally better, than that obtained using the relevance scores given by expressions (6) - (8).

Besides evaluating its retrieval performance, it is interesting to understand when it is convenient to use the combined method. We analyzed the ranked performance of each tested method, including unexpanded query, before and after the introduction of the combined method. In Table 7 we consider the unexpanded method and the three expansion methods (i.e., Rocchio, CHI-1, and KLD), showing the number of times that each method was ranked first, second, third, or fourth. Table 8 is the analogous table for the case when all methods, including the combined one, are considered. The chosen performance measure was AV-PREC, as usual throughout the paper.

Table 7. Ranked performance before method combination.

	TREC-8				TREC-9			
	Unexp	Rocchio	CHI-1	KLD	Unexp	Rocchio	CHI-1	KLD
1st	9	8	10	23	8	13	9	20
2nd	9	21	14	6	5	18	12	15
3rd	6	19	15	10	12	12	18	8
4th	26	2	11	11	25	7	11	7

Table 8. Ranked performance after method combination.

	TREC-8					TREC-9				
	Unexp	Rocchio	CHI-1	KLD	Comb.	Unexp	Rocchio	CHI-1	KLD	Comb.
1st	8	7	6	12	17	8	5	7	7	23
2nd	9	12	10	14	5	4	13	10	20	3
3rd	1	17	13	10	9	4	19	9	13	5
4th	7	12	11	6	14	12	11	14	6	7
5th	25	2	10	8	5	22	2	10	4	12

By comparing Table 7 and Table 8, the following observations can be made.

When all three expansion methods were ranked ahead of the unexpanded method (26 times for TREC-8, 25 times for TREC-9), the combined method almost always fared better than the unexpanded method (25 and 22 times, respectively). Furthermore, we found that of these 25 (22) queries, the combined method achieved the best overall performance 13 (16) times. Thus, the combined method worked very well when query expansion improved performance.

When all three expansion methods were ranked behind the unexpanded method (9 times for TREC-8, 8 times for TREC-9), the combined method almost always remained behind the unexpanded method (8 times for both collections). Thus, the combined method was not useful when query expansion caused harm.

The most interesting case is when there is at least one expansion method that improved and one that hurt performance (15 times for TREC-8, 17 times for TREC-9). These queries can be split into two groups, one containing the queries for which there was one method that improved and two methods that hurt (9 for TREC-8, 5 for TREC-9), and the other containing the queries for which there were two methods that improved and one that hurt (6 and 12, respectively). In the first case, the combined method was almost always (8 times for TREC-8, 4 times for TREC-9) unable to improve over unexpanded query.[8] By contrast, for the second group of queries, the combined method did always improve over the unexpanded method on TREC-8 and improved for 9 of the 12 cases on TREC-9 . Thus, these results seem to confirm our hypothesis that the combination is especially suitable when only one of the three methods hurts performance.

[8] Note that, for TREC-8, when passing from Table 7 to Table 8 the number of times that the unexpanded method is ranked second does not change, but in the 9 queries of Table 8

4.3 Effect of method parameters on retrieval performance

The parameter combination chosen for performing query expansion on the TREC-8 test collection was consistent with that used with good results by several groups experimenting with the same document set at recent TRECs. By contrast, the WT10g document set used for the web track at TREC-9 was introduced more recently and the use of automatic query expansion in that environment is still exploratory.

Using the same parameters as TREC-8 on TREC-9, query expansion caused harm. A decrease in retrieval performance was observed for any expansion method, including the combined one, on each evaluation measure.

The different behavior of the two test collections was probably due to the very different quality of the set of pseudo-relevant documents upon which query expansion was based. When passing from TREC-8 to TREC-9, the precision at ten retrieved documents (i.e., those used for query expansion on TREC-8) dropped sharply, from 0.4920 to 0.3320. If we want, for TREC-9, a quality comparable to that used for TREC-8, we should consider only the very first retrieved documents (e.g., PREC-AT-2 = 0.5400, PREC-AT-3 = 0.4667). Choosing a smaller number of pseudo-relevant documents may also be useful to reduce the chance of selecting terms from mostly nonrelevant documents due to the higher presence of topics with very few relevant documents; e.g., there are 10 TREC-9 topics with fewer than 10 relevant documents, as opposed to only 2 in TREC-8 (Ogawa et al. 2001). Our choice, for TREC-9, was to use three pseudo-relevant documents.

To account for the lower quality of the terms used for expansion, the values of α and β should be also adjusted. Based on the observation that the original query should become more important as the quality of the expansion terms and their weights diminishes (Buckley and Salton 1995), we increased the ratio between α and β in expression (2) to 5 (i.e., $\alpha = 1$, $\beta = 0.2$). In this way, it should be easier for the expanded query to keep the focus on the original topic, even in the presence of bad term suggestions.

To compensate for their lower quality, one might also reduce the number of terms used for expanding TREC-9 queries; however, we decided not to do so, mainly because this would have compressed the diversity of the term rankings used for method combination, thus narrowing down the scope for experimenting with the approach presented in the paper. The number of expansion terms was the same as with TREC-8 (40). In fact, it turned out that this choice did not hurt the retrieval effectiveness of query expansion (see discussion below).

---

there is one query for which the unexpanded method was formerly (i.e., in Table 7)

In order to gain some insights into the robustness of the proposed combination method, we evaluated how changes in the parameter combination used for each collection will affect retrieval performance. We considered all three main parameters, namely the number of pseudo-relevant documents, the ratio between α and β, and the number of expansion terms.

For each parameter and each collection, we let the parameter vary around the value used for performing the main experiments on that collection, while keeping the other two parameters constant. For each resulting parameter combination, we measured the retrieval performance of the combined method as well as of any individual expansion method. The results are shown in Table 9, Table 10, and Table 11, with the highest performance values obtained for each parameter and each collection shown in bold.

In all, we tested 26 distinct parameter combinations. The first result is that the expansion methods always performed better than the unexpanded method (we recall that the baseline average precision was 0.2718 for TREC-8 and 0.2110 for TREC-9). The second, perhaps more important, result is that for 24 times the combined method achieved better performance than any individual method; the only two exceptions were observed on TREC-9, for [# docs = 4, # terms = 40, β = 0.2] and [# docs = 5, # terms = 40, β = 0.2], where CHI-1 was occasionally better. Thus, the combination method showed itself to be quite robust with respect to parameter variation in the chosen ranges.

It is also interesting to examine how the retrieval effectiveness of query expansion varied as a function of individual parameters. We consider each of them, in turn.

As shown in Table 9, the retrieval effectiveness was, in general, moderately affected by the size of the set of pseudo-relevant documents. The performance varied at most by 7.42% for individual methods (i.e., for KLD, when passing from 5 to 2 documents on TREC-9) and by 9.88% for combined method (i.e., when passing from 5 to 3 documents on TREC-9).

---

ranked first.

Table 9. Performance versus number of pseudo-relevant documents for query expansion methods on TREC-8 and TREC-9.

	TREC-8 (# terms = 40, α =1, β = 2)					TREC-9 (# terms = 40, α =1, β = 0.2)				
	# docs 6	*# docs 8*	*# docs 10*	*# docs 12*	*# docs 14*	*# docs 1*	*# docs 2*	*# docs 3*	*# docs 4*	*#docs 5*
Rocchio	0.2952	0.3028	0.2972	0.2954	0.2923	0.2144	0.2140	0.2201	0.2082	0.2124
CHI-1	0.2836	0.2940	0.2918	0.2959	0.2931	0.2170	0.2195	0.2234	0.2247	0.2252
KLD	0.3028	0.3048	0.3053	0.3001	0.2964	0.2284	0.2315	0.2303	0.2183	0.2155
Comb.	0.3050	**0.3097**	0.3088	0.3057	0.3038	0.2316	0.2336	**0.2413**	0.2219	0.2196

Table 10. Performance versus β (α = 1) for query expansion methods on TREC-8 and TREC-9.

	TREC-8 (# docs = 10, # terms = 40)					TREC-9 (# docs = 3, # terms = 40)				
	β = 0.5	*β = 1*	*β = 2*	*β = 3*	*β = 4*	*β = 0.1*	*β = 0.2*	*β = 0.3*	*β = 0.4*	*β = 0.5*
Rocchio	0.2845	0.2905	0.2972	0.2988	0.2987	0.2191	0.2201	0.2206	0.2226	0.2226
CHI-1	0.2925	0.2944	0.2918	0.2877	0.2837	0.2179	0.2234	0.2239	0.2262	0.2282
KLD	0.3071	0.3068	0.3053	0.3005	0.2962	0.2218	0.2303	0.2312	0.2384	0.2412
Comb.	0.3104	**0.3133**	0.3088	0.3021	0.2993	0.2292	0.2413	**0.2438**	0.2422	0.2416

Table 11. Performance versus number of expansion terms for query expansion methods on TREC-8 and TREC-9.

	TREC-8 (# docs = 10, α =1, β = 2)					TREC-9 (# docs = 3, α =1, β = 0.2)				
	#terms 10	*#terms 20*	*#terms 30*	*#terms 40*	*#terms 50*	*#terms 10*	*#terms 20*	*#terms 30*	*#terms 40*	*#terms 50*
Rocchio	0.2871	0.2943	0.2984	0.2972	0.2973	0.2182	0.2171	0.2174	0.2201	0.2180
CHI-1	0.2840	0.2898	0.2921	0.2918	0.2924	0.2181	0.2182	0.2190	0.2234	0.2201
KLD	0.2949	0.3032	0.3037	0.3053	0.3040	0.2288	0.2311	0.2317	0.2303	0.2337
Combi.	0.3008	0.3070	**0.3100**	0.3088	0.3067	0.2348	0.2369	0.2343	**0.2413**	0.2378

For TREC-8, following suggestions made by Carpineto et al. (2001), the range of values chosen for test probably represented a good compromise between the maximization of the percentage of relevant documents and the presence of enough relevant documents. If we used fewer or, symmetrically, more documents, a decrease in performance would be observed for some or all basic expansion methods, and thus also for the combined method. For TREC-9, the variations were higher, especially as the number of documents grew. A further increase in the number of documents would cause degradation in retrieval performance.

Table 10 shows that the retrieval performance was moderately affected also by the ratio between α and β, with a maximum difference of 8.74% for individual methods (i.e., for KLD, when passing from $\beta = 0.1$ to $\beta = 0.5$ on TREC-9) and 6.36% for the combined method (i.e., when passing from $\beta = 0.1$ to $\beta = 0.3$ on TREC-9). Thus, the performance remained relatively stable as the parameter values varied in the chosen ranges; if we chose greater or smaller values, the retrieval effectiveness would decrease.

Finally, the results reported in Table 11 suggest that that the retrieval performance was weakly affected by variations in the number of terms used for query expansion. The performance varied at most by 3.93% for individual methods and by 3.05% for combined method, with the best result on TREC-9 obtained just for 40 expansion terms. The performance would remain substantially stable even for higher values of the number of expansion terms, while it would decrease for smaller values. This result confirms and extends to the WT10g collection earlier findings about the limited importance of this parameter for retrieval effectiveness (Salton and Buckley 1990, Carpineto *et al.* 2001).

As already stated, the overall utility of the combined method depends on the availability of a set of individual methods exhibiting comparably good retrieval performance. On the whole, our experiments show that although parameter setting remains a critical step for successful query expansion, there was at least a large range of values for each parameter where this requirement was fulfilled and the combined method consistently improved over the individual methods.

Table 9, 10, and 11 also reveal clues as to how to optimize performance. The parameter combination used for each collection in our main experiments was not the best possible choice. There were five parameter combinations, namely [# docs = 8, # terms = 40, β = 2], [# docs = 10, # terms = 30, β = 2], [# docs = 10, # terms = 40, β = 1], and [# docs = 10, # terms = 40, β = 0.5] for TREC-8, and [# docs = 3, # terms = 10, β = 0.3] for TREC-9, which produced better retrieval effectiveness.

5. EFFECT OF QUERY LENGTH ON RETRIEVAL FEEDBACK PERFORMANCE

The queries used in our previous experiments were rather long. As short queries may better simulate a real searching scenario, it is useful to evaluate the performance of the proposed method when the queries contain fewer terms.

In general, there are good reasons to believe that short queries could improve as likely as hurt retrieval feedback. On the one hand, word mismatch may be a more serious problem when the same information need is expressed using a smaller number of terms, and thus query expansion should be potentially more useful for a shorter query. On the other hand, it is conceivable that a shorter query yields poorer text from which to mine retrieval feedback terms, thus reducing the benefits of query expansion. Experimental studies have not shown conclusive results, although they have usually reported greater advantages for short queries (Voorhees 1994, Hawking *et al.* 1998, Xu and Croft 2000).

To mimic the short query scenario, we used only the title field of TREC-8 and TREC-9 topics. After indexing, TREC-8 topics were described with an average of 2.32 distinct terms (versus 12.25 for the long query set), with a minimum of 1 and a maximum of 3 terms; TREC-9 topics contained, on average, 2.46 terms (versus 17.76 for long queries), with a minimum of 1 and a maximum of 6 terms.

We reran the experiments using the short query set and keeping the other experimental conditions constant. The performance of the unexpanded method and of each expanded method, including the combined one, is shown in Table 12 and Table 13 for TREC-8 and TREC-9, respectively.

Table 12. Performance comparison for short queries on TREC-8

	Unexpanded	Rocchio	CHI-1	KLD	Combined
AV-PREC	0.2246	0.2641	0.2425	0.2883	0.2887
		+17.59%*	+8.01%*	+28.40%*	+28.58%*
PREC-AT-5	0.4720	0.5040	0.4320	0.5280	0.5360
		+6.78%	-8.47%	+11.86%*	+13.56%*
PREC-AT-10	0.4440	0.4800	0.4000	0.4660	0.4900
		+8.11%*	-9.91%*	+4.95%	+10.36%*

Table 13. Performance comparison for short queries on TREC-9.

	Unexpanded	Rocchio	CHI-1	KLD	Combined
AV-PREC	0.1630	0.1902	0.1720	0.1877	0.1928
		+16.67%*	+5.54%	+15.15%*	+18.27%*
PREC-AT-5	0.3200	0.3320	0.3280	0.3400	0.3400
		+3.75%	+2.50%	+6.25%	+6.25%
PREC-AT-10	0.2380	0.2500	0.2420	0.2560	0.2600
		+5.04%	+1.68%	+7.56%	+9.24%*

The results of Table 12 and 13 show that each expansion method, including the combined one, considerably improved performance over the baseline, with the exception of CHI-1 for the first retrieved documents on TREC-8. Under this respect, the results were similar to those obtained for the long query sets (see Table 1 and 2 for the basic expansion methods and Table 5 and 6 for the combined method).

A comparison between the figures reported in those tables shows that, as expected, the absolute results for long queries were better than short queries for all methods and evaluation measures. In contrast, query expansion improved retrieval performance by a greater or lesser extent for short queries than it did for long queries depending on the test collection being tested.

For TREC-8, the performance improvement was larger for 9 times and smaller for 3 times, the number of differences being statistically significant increased (9 versus 6), and the percentage improvements grew markedly for several methods and performance measures. In six cases, the improvement was higher than 10%, with a peak of 28.58% for KLD on AV-PREC, whereas the maximum improvement for long queries was 12.32%. The situation for TREC-9 was very different. The performance improvement was larger

for 3 times and smaller for 9 times, while the number of differences being statistically significant decreased from 9 to 4.

Considering both the long and short query experiments, the best results were obtained for TREC-8 short queries and TREC-9 long queries, with 9 statistically significant improvements over the baseline performance measures, followed by TREC-8 long queries, with 6 statistically significant differences, and TREC-9 short queries, with 4 statistically significant differences. It is possible that these results were influenced by the different quality of baseline retrieval, following Hawking *et al.* (1998)'s conjecture that queries producing very good baseline results may be closer to optimal, thus restricting the scope of potential gains from retrieval feedback. In fact, our results suggest that retrieval feedback is most useful when the information need is expressed by a query producing baseline retrieval of intermediate quality (i.e., TREC-8 short and TREC-9 long queries), while its benefits become smaller not only for queries with low quality baseline retrieval (i.e., TREC-9 short queries), as expected, but also for queries with very high quality baseline retrieval (i.e., TREC-8 long queries). Analogous results in support of this hypothesis were found by Carpineto *et al.* (2001) for the case when the queries refer to different information needs.

Table 12 and 13 also show that the performance of the combined method on the short query set was very good, even if the individual methods being combined presented less comparable mean retrieval performance than the long query experiments. The combined method obtained better results than any individual method for all evaluation measures, with more marked performance improvements over CHI-1 and Rocchio. A more detailed analysis revealed that, on the short query set, the combined method improved in a statistically significant manner over CHI-1 five times (i.e., for all evaluation measures on TREC-8, for TREC-9 AV-PREC, and for TREC-9 PREC-AT-10), it improved over Rocchio two times (i.e., for TREC-8 AV-PREC and TREC-9 AV-PREC), and over KLD just once (i.e., for TREC-8 PREC-AT-10).

The performance improvement due to method combination obtained for short queries was also comparable with that obtained for long queries. For TREC-8, the combined method worked, in general, comparatively better on the short query set, while its benefits were greater for long queries on TREC-9.

In addition to evaluating the mean retrieval performance, we reran for the short query set the experiments concerning the performance on individual queries. The main difference between the results obtained for short queries and those reported for long queries concerned the inter-method variations, which were more ample (0.095 versus 0.087 for TREC-8 and 0.0451 versus 0.029 for TREC-9). Interestingly, if we were able to select for

each query the best method, we would get for the TREC-8 short query set a mean absolute AV-PREC value of 0.3099, with an amazing performance gain of 37.97% over unexpanded query, and for the TREC-9 short query set a value of 0.2055 (+ 26.07% over baseline).

The observed increase in inter-method performance variation did not, however, correspond to a similar increase in the difference between term rankings, because the overlap and the correlation between the ordered lists of terms suggested by each method to expand the short queries were similar to those found for long queries. This behavior may be due to the fact that while the documents used to expand short queries and those used to expand long queries were often the same, thus producing the same term rankings, the contribution of query expansion to the final query was greater for short queries, thus increasing the differences in retrieval performance.

6. CONCLUSIONS AND FUTURE WORK

In this paper we have presented an evaluation of the performance of three different functions for scoring expansion terms within Rocchio's reweighting scheme, followed by a proposal for combining their results. From our experimental evaluation, the following main conclusions can be drawn.

Retrieval feedback methods employing distinct term-scoring functions produce different expanded representations of each query, with large variations on retrieval performance, even when the same methods present a comparable retrieval performance over the whole set of queries.

The combined method is more effective than baseline retrieval and any individual retrieval feedback method, especially when only the first documents retrieved by the system are considered for evaluation, with a marked performance improvement over CHI-1 and Rocchio and a smaller one over KLD.

This research can be extended in several ways. As the results relating to the effectiveness of the combined method were obtained for particular experimental conditions, one direction for future work is to perform a more robust evaluation of retrieval performance. Although there is some reason to believe that such results should hold, or might even improve, across different experimental settings, for we used very simple and untuned functions and two large document collections, this cannot be taken for granted. It would be useful to conduct further experiments to evaluate how the performance results change when controlling a wider range of factors that might affect the system's overall performance, including the primary weighting scheme, the expansion methods being combined, and the technique used to combine them.

The use of more powerful ensembling strategies is a second topic for future research. As combination methods work best when the results being combined are generated independently (Saracevic and Kantor 1988, Turtle and Croft 1991, Hull *et al.* 1996, Dietterich 1997), it is conceivable that the improvements over individual methods could be higher if we weakened some experimental factors that are likely to increase the correlation between the term-relevance estimates of the different functions, for instance by varying the number of documents used for query expansion or their representations. Furthermore, the merging of multiple term rankings could be performed by using more sophisticated techniques, such as minimizing the number of disagreements in the preference graphs associated with the rankings (Cohen *et al.* 1999), or learning optimal combination strategies from user feedback (Bartell *et al.* 1994). Finally, it might be interesting to evaluate the effect of combining the results of more than three basic expansion methods, which may require the use of a more effective function than median to merge the rank values produced by the single methods.

ACKNOWLEDGMENTS

We are grateful to Jamie Callan and four anonymous referees for their very useful comments and suggestions.

REFERENCES

ARONSON, ALAN R.; RINDFLESCH, THOMAS C.; BROWNE, ALLEN C. 1994. Exploiting a Large Thesaurus for Information Retrieval. In Proceedings of RIAO 94: Intelligent Multimedia Information Retrieval Systems and Management, New York, NY, 197-216.

ATTAR, R., AND FRAENKEL, A. S. 1977. Local feedback in full-text retrieval systems. *Journal of the Association for Computing Machinery*, *24*, 3, 397-417.

BARTELL, B., COTTRELL, G., AND BELEW, R. 1994. Automatic combination of multiple ranked retrieval systems. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'94)*, Dublin, 173-181.

BELKIN, N. J., COOL, C., CROFT, W. B., AND CALLAN, J. P. 1993. The effect of multiple query representations on information retrieval system performance. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'93)*, Pittsburgh, PA, 339-346.

BREIMAN, L. 1996. Bagging predictors. *Machine Learning*, *24* , 2, 123-140.

BUCKLEY, C., AND SALTON, G. 1995. Optimization of relevance feedback weights. In *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'95)*, Seattle, WA, 351-357.

BUCKLEY, C., SALTON, G., ALLAN, J., AND SINGHAL, A. 1995. Automatic query expansion using SMART: TREC3. In *Proceedings of the 3rd Text REtrieval Conference (TREC-3)*, Gaithersburg, MD, NIST Special Publication 500-226, 69-80.

BUCKLEY, C., MITRA, M., WALTZ, J., CARDIE, C., 1998. Using clustering and superconcepts within SMART. In *Proceedings of 6th Text Retrieval Conference (TREC-6)*, Gaithersburg, MD, NIST Special Publication 500-240, 107-124.

CARPINETO, C., DE MORI, R., ROMANO, G., AND BIGI, B. 2001. An information theoretic approach to automatic query expansion. *ACM Transactions on Information Systems*, *19*, 1, 1-27.

CARPINETO, C., AND ROMANO, G. 1999. Towards better techniques for automatic query expansion. In *Proceedings of the 3rd European Conference on Digital Libraries (ECDL'99)*, Paris, Springer Verlag, 126-141.

CARPINETO, C., AND ROMANO, G. 2000a. Order-theoretical ranking. *Journal of the American Society for Information Sciences*, *51*, 7, 587-613.

CARPINETO, C., AND ROMANO, G. 2000b. TREC-8 Automatic Ad-Hoc Experiments at FUB. In *Proceedings of the 8th Text REtrieval Conference (TREC-8)*, Gaithersburg, MD, NIST Special Publication 500-246, 377-380.

COHEN, W. W., SCHAPIRE, R. E., AND SINGER, Y. 1999. Learning to order things. *Journal of Artificial Intelligence Research, 10*, 243-270.

COOPER, J., AND BYRD, R. 1997. Lexical navigation: visually prompted query expansion and refinement. In *Proceedings of the 2nd ACM Digital Library Conference*, Philadelphia, 237-246.

CROFT, B., AND HARPER, D. J. 1979. Using probabilistic models of document retrieval without relevance information. *Journal of Documentation, 35*, 285-295.

DEERWESTER, S., DUMAIS, S. T., FURNAS, W., LANDAUER, T. K., AND HARSHMAN, R. 1990. Indexing by latent semantic analysis. *Journal of the American Society for Information Sciences, 41*, 6, 391-407.

DIETTERICH, T. (1997). Machine-learning research: four current directions. *AI Magazine, Winter 1997*, 97-135.

DOSZCOCKS, T.E. (1978). AID: an associative interactive dictionary for online searching. *Online Review*, *2*, 2, 163-174.

EFTHIMIADIS, E. 1993. A user-centered evaluation of ranking algorithms for interactive query expansion. In *Proceedings of the 16th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'93)*, Pittsburgh, PA, 146-159.

EFTHIMIADIS, E. 1996. Query Expansion. In Williams, Martha E., ed., Annual Review of Information Systems and Technology (ARIST), *31*, 121-187.

FIDEL, R., AND EFTHIMIADIS, E. 1995. Terminological knowledge structure for intermediary expert systems. *Information Processing & Management*, 31, 1, 15-27.

FISCUS, J. 1997. A post-processing system to yield reduced error word rates: recognizer output voting error reduction (ROVER). Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding, Santa Barbara, CA, 347-354.

FITZPATRICK, L., AND DENT, M. 1997. Automatic feedback using past queries: social searching? Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'97), Philadelpia, PA, 306-313.

FOX, E. A., SHAW, J. A. 1994. Combination of multiple searches. In *Proceedings of 2nd Text Retrieval Conference (TREC-2)*, Gaithersburg, MD, NIST Special Publication 500-215, 243-252.

HARMAN, D. K. 1992. Relevance feedback and other query modification techniques. In *Information Retrieval - Data Structures and Algorithms*, W.B. Frakes and R. Baeza-Yates, Eds., Englewood Cliffs: Prentice Hall. 241-263.

HAWKING, D., THISTLEWAITE, P., AND CRASWELL, N. 1998. ANU/ACSys TREC-6 Experiments. In *Proceedings of 6th Text Retrieval Conference (TREC-6)*, Gaithersburg, MD, NIST Special Publication 500-240. 275-290.

HAWKING, D. 2001. Overview of the TREC-9 Web Track. In *Proceedings of the 9th Text Retrieval Conference (TREC-9)*, Gaithersburg, MD, in press.

HULL, D., PEDERSEN, J., AND SCHUTZE, H. (1996). Method combination for document filtering. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'96)*, Zurich, 279-287.

JING, Y., CROFT, W.BRUCE. 1994. The Association Thesaurus for Information Retrieval. In Proceedings of RIAO 94: Intelligent Multimedia Information Retrieval Systems and Management, New York, NY, 146-160.

KANTOR, P. 1995. Decision level data fusion for routing of documents in the TREC-3 context: a best case analysis of worst case results. In *Proceedings of the 3rd Text REtrieval Conference (TREC-3)*, Gaithersburg, MD, NIST Special Publication 500-226, 319-332.

KARP, D., SCHABES, Y., ZAIDEL, AND M., EGEDI, D. 1992. A freely available wide coverage morphological analyzer for English. In *Proceedings of the 14th International Conference on Computational Linguistics (COLING '92)*, Nantes, France. 950-954.

KATZER, J., MCGILL, M. J., TESSIER, J. A., FRAKES, W., AND DASGUPTA, P. 1982. A study of the overlap among document representations. *Information Technology : Research and Development, 1*, 261-274.

KOBAYASHI, M., AND TAKEDA, K. 2000. Information retrieval on the web. *ACM Computing Surveys, 32*, 2, 144-173.

KRAAIJ, W., AND WESTERVELD, T. 2001. TNO/UT at TREC-9: How different are web documents? In *Proceedings of the 9th Text Retrieval Conference (TREC-9)*, Gaithersburg, MD, in press.

LAM-ADESINA, A. M., AND JONES, G.J.F. 2001. Applying Summarization Techniques for Term Selection in Relevance Feedback. In *Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2001)*, New Orleans, 1-9.

LARKEY, L., AND CROFT, B. 1996. Combining classifiers in text categorization. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'96)*, Zurich, 289-297.

LEE, J. H. 1997. Analyses of multiple evidence combination. In *Proceedings of the 20th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'97)*, Philadelpia, PA, 267-276.

MITRA, M., SINGHAL, A., AND BUCKLEY, C. 1998. Improving automatic query expansion. In *Proceedings of the 21th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98)*, Melbourne, 206-214.

MONTAGUE, M., AND ASLAM, J. 2001. Relevance score normalization for metasearch. To appear in *Proceedings of the 10th International ACM Conference on Information and Knowledge Management (CIKM 2001*), Atlanta, Georgia.

OGAWA, Y., MANO, H., NARITA, M., AND HONMA, S. 2001. Structuring and expanding queries in the probabilistic model. In *Proceedings of the 9th Text Retrieval Conference (TREC-9)*, Gaithersburg, MD, in press.

PORTER, M. F. 1982. Implementing a Probabilistic Information Retrieval System. *Information Technology: Research and Development, 1*, 2, 131-156.

PORTER, M.F., AND GALPIN, V. 1988. Relevance feedback in a public access catalogue for a research library: Muscat at the Scott Polar Research Institute. *Program, 22*, 1, 1-20.

QIU, Y.; FREI, H.P. 1993. Concept-Based Query Expansion. In Proceedings of the 16th Annual International ACM SIGIR *Conference on Research and Development in Information Retrieval (SIGIR'93)*, Pittsburgh, PA, 160-169.

ROBERTSON, S.E. 1990. On term selection for query expansion. *Journal of Documentation, 46*, 4, 359-364.

ROBERTSON, S.E. WALKER, S., JONES, G.J.F., HANCOCK-BEAULIEU, AND GATFORD, M. 1995. Okapi at TREC-3. *Proceedings of the 3rd Text REtrieval Conference (TREC-3)*, Gaithersburg, MD, NIST Special Publication 500-226. 109-126.

ROBERTSON, S. E., WALKER, S., AND BEAULIEU, M. 1999 Okapi at TREC-7: Automatic ad hoc, filtering, VLC, and interactive track. In *Proceedings of the 7th Text REtrieval Conference (TREC-7*), Gaithersburg, MD, NIST Special Publication 500-242. 253-264.

ROCCHIO, J. (1971). Relevance feedback in information retrieval. In *The SMART retrieval system - experiments in automatic document processing*, Salton, G., Ed., Prentice Hall, Englewood Cliffs.

SALTON, G. AND BUCKLEY, C. 1990. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Sciences, 41,* 288-297.

SARACEVIC, T., AND KANTOR, P. 1988. A Study of Information Seeking and Retrieving. III. Searchers, Searches, and Overlap. *Journal of the American Society for Information Science, 39,* 3, 197-216.

SCHWENK, H., AND GAUVAIN J.L. 2000. Combining multiple speech recognizers using voting and language model information. Proceedings of IEEE International Conference on Speech and Language Processing (ICSLP), Beijing, China, 915-918.

SELBERG, E., AND ETZIONI, O. 1995. Multi-Service Search and Comparison using the MetaCrawler. In Proceedings of the 4th International World Wide Web Conference, Boston, MA, 195-208.

SIEGEL, S., AND CASTELLAN, N.J., Jr. 1988. Nonparametric statistics for the behavioral sciences, second edition. McGraw-Hill Book Company.

SINGHAL, A., CHOI, J., HINDLE, D., LEWIS, D., AND PEREIRA, F. 1999. AT&T at TREC-7. *Proceedings of the Seventh Text Retrieval Conference (TREC-7),* Gaithersburg, MD, NIST Special Publication 500-242, 239-252.

SPARCK JONES, KAREN. 1971. Automatic Keyword Classification for Information Retrieval. London: Butterworths..

SPINK, A., AND SARACEVIC, T. 1997. Interaction in information retrieval: selection and effectiveness of search terms. *Journal of the American Society for Information Sciences, 48,* 8, 741-761.

SPINK, A., WOLFRAM, D., JANSEN, B.J., AND SARACEVIC, T. 2001. Searching the Web: The Public and Their Queries. *Journal of the American Society for Information Science, 53,* 2, 226-234.

SRINAVASAN, P. 1996. Query expansion and MEDLINE. *Information Processing & Management, 32,* 4, 431-443.

STRZALKOWSKI, T. 1995. Natural language information retrieval. *Information Processing & Management, 31,* 3, 395-416.

TURTLE, H., AND CROFT, W. B. 1991. Evaluation of an inference network-based retrieval model. *ACM Transactions on Information Systems, 9,* 3, 187-222.

VOGT, C. C., AND COTTREL, J. W. 1998. Predicting the performance of linearly combined IR systems. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98)*, Melbourne, 190-196.

VOORHEES, E. M. 1994. Query Expansion Using Lexical-Semantic Relations. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'94)*, Dublin, Ireland, 61-69.

VOORHEES, E. M., AND HARMAN, D. K. 2000. Overview of the Eighth Text Retrieval Conference (TREC-8). In *Proceedings of the 7th Text Retrieval Conference (TREC-7)*, Gaithersburg, MD, NIST Special Publication 500-246, 1-24.

WALKER, S., ROBERTSON, S.E., BOUGHANEM, M., JONES, G.J.F., AND SPARCK JONES, K. 1998. Okapi at TREC-6 Automatic ad hoc, VLC, routing, filtering and QSDR. In *Proceedings of the 6th Text Retrieval Conference (TREC-6)*, Gaithersburg, MD, NIST Special Publication 500-240, 125-136.

XU, J., AND CROFT, B. 1996. Query expansion using local and global document analysis. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'96)*, Zurich, 4-11.

XU, J., AND CROFT, B. 2000. Improving the effectiveness of information retrieval with local context analysis. *ACM Transactions on Information Systems, 18*, 1, 79-112.