Multiple Search Engines in Database Merging

Ellen M. Voorhees
National Institute of Standards and Technology
Gaithersburg, MD
ellen.voorhees@nist.gov

Richard M. Tong
Sageware, Inc.
Mountain View, CA
rtong@sageware.com

Abstract

A database merging technique is a strategy for combining the results of multiple independent searches into a single cohesive response. While a variety of techniques have been developed to address a range of problem characteristics, our work focuses on environments in which search engines work in isolation. This paper shows that the behavior of two previously developed isolated techniques is indeed independent of the particular search engines that participate in the search. Two very different search engines, SMART and TOPIC®, were each used to retrieve documents from five subcollections. The relative effectiveness of the merged result compared to the effectiveness of a corresponding single collection run is comparable for both engines.

The effectiveness of the merged result is improved when both search engines search the same five subcollections but participate in a single merging. The improvement is such that this 10-collection merge is sometimes more effective than the single collection run. This last finding suggests that these methods may be able to improve the effectiveness of World Wide Web searches by merging the output from several engines.

1 Introduction

Networked information systems such as digital libraries allow collections of text that are physically or administratively separate to be jointly accessible. Given the widespread accessibility, information system users expect to be able to search multiple databases with a single query without contending with multiple search protocols or fragmented, piecemeal search results. Producing an integrated search result can be difficult, however. The content of collections is likely to differ in range of topics, depth of coverage, and vocabulary. The search engines used by the collections to find documents of interest will be tuned for each individual collection and may thus produce search results that are incomparable to one another. This problem of combining the retrieval results from multiple independent textual databases into a single result that has the best possible effectiveness has become known as *database merging* [16] or *metasearching* [6].

Appropriate solutions for a particular database merging problem depend on the characteristics of the participating search engines. In situations where the component search engines are known to compute document-query scores in compatible ways, the simple strategy of retrieving the N documents with the best scores across all collections produces the same results as obtained when all the documents are in a single collection [8]. One way to guarantee compatible scores is to use instances of a single search engine that does not use collection-dependent statistics when computing the scores. Different search engines can be accommodated by this method provided each engine computes the same function (such as the probability of relevance) as the score. However, this "best-N" approach to database merging requires each collection to participate in each search and severely limits the retrieval strategies that can be used on the individual collections.

Other approaches are needed when the component search engines compute incompatible scores. The approach taken by Callan and his colleagues [5], called CORI for *collection retrieval inference network*, is appropriate when the merging strategy has access to document frequency (the number of documents a term appears in) statistics for each collection. Document frequencies are a subset of what the Stanford Protocol Proposal for Internet Retrieval and Search (STARTS) advocates to be exported as collection metadata [6]. CORI generalizes the inference network structure used for document ranking within a collection to collection ranking based on the term occurrence statistics.

The retrieval experiments conducted by Callan et. al showed that CORI is able to maintain accurate retrieval results for moderate numbers of retrieved documents while searching only about half of the collections per search. These results demonstrate the effectiveness of the CORI method when the metadata is available. Obtaining such metadata may be problematic, however. In the CORI experiments, the document frequencies could be obtained from the same indexes that were used to do the actual document retrieval, since the same search engine was used for each collection and the CORI algorithm knew of and had access to those indexes. In the general case (assumed in STARTS), each collection must publish the frequencies in a generic format, adding to the storage overhead of the collection and adding another facet to collection maintenance. The volume of metadata that must be processed for efficient query processing precludes all but a few large sites from performing searches. In addition, some database owners may be willing to allow users to search the database with the owner's search software, but would be unwilling to divulge as much about their proprietary documents as the metadata would expose.

We have developed two database merging strategies (the Modeling Relevant Document Distributions (MRDD) and Query Clustering (QC) strategies) that are designed for environments where search engines work in isolation: we assume only that a search engine will accept a query in natural language and return a ranked list of documents in response to that query [13, 16, 17, 18]. Instead of relying on collections publishing metadata, MRDD and QC construct their own (peculiar form of) metadata from the results of past queries. For both strategies, the required metadata is small enough that it would not provide a burden to the average site. In keeping with the isolation philosophy, only the site that performs a merged search incurs the extra storage burden.

To process a new query, the MRDD and QC strategies use the metadata to decide the number of documents to retrieve from each collection, obtain those documents from the search engines, and then produce a single ranked list that is returned to the user. Since usually the methods will decide to retrieve zero documents from some databases, only those engines actually contributing documents to the final result need be contacted at search time. Additionally, experiments show that the effectiveness of the merged results is only slightly compromised as compared to the effectiveness of the results obtained by treating the entire set of documents as a single collection for retrieved set sizes of interest to most users (50 documents retrieved per query or fewer). For example, in the experiments reported later in this paper, the merged search results are generally within 15% of the single collection results, which amounts to approximately one fewer relevant document retrieved per query.

As with any learning strategy, the effectiveness of the final ranked list built by MRDD or QC depends upon how well the training queries reflect the actual query pool. Thus these methods are unlikely to be effective in applications where database content fluctuates wildly or users' queries are completely uncorrelated with one another. On the other hand, the adaptation afforded by such learning may be able to be exploited to provide more personalized information services [14, 15].

This paper furthers our investigation of the MRDD and QC isolated database merging strategies. While the strategies are designed to be completely independent of the search engines participating in the retrieval, we have not had multiple search engines available to test the claim of search-engine independence until now. The results in this paper demonstrate that the two techniques are indeed able to accommodate multiple search engines. Two search engines, SMART and TOPIC®, are used to search portions of the TREC collection. The relative effectiveness of the merged result as compared to the single collection result is comparable when either TOPIC or SMART is used as the search engine. More importantly, the effectiveness of the merged result can improve when both search engines are involved in the same search.

The next section of the paper describes the various systems and algorithms used in this work. The following section describes the experiments in detail, reports on the results obtained in the experiments, and discusses the implications of the results. The final section comments on areas that remain to be explored.

2 Experimental Design

The purpose of this work is to investigate the behavior of the MRDD and QC database merging strategies when different retrieval systems are used to search the component databases. To accomplish this goal, two very different retrieval systems, SMART and TOPIC, are each used to search subsets of the TREC test collection as well as the entire set of TREC documents as a single collection. For each retrieval system, each merging strategy is applied to the output of the subset searches and the result is compared to the single collection run. Finally, each merging strategy is applied to the subset searches from both retrieval systems and that result is compared to the single collection run. Since the merging strategies require the set of queries to be split into training and test sets, the entire process is repeated with different combinations of training and test queries. To provide the necessary background for discussing the results of these experiments, this section provides brief descriptions of the TREC test collection, the two retrieval systems, and the two database merging strategies.

2.1 The TREC test collection

The TREC test collection is a test collection being developed in a series of workshops sponsored by the U.S. National Institute of Standards (NIST) and the Advanced Research Projects Agency (ARPA) [7]. The collection is composed of distinct subcollections that are different sizes, cover diverse topics, and have different retrieval characteristics. The retrieval experiments in this paper use the approximately 742,000 documents on TREC disks one and two, TREC query statements 51–200, and relevance assessments produced after TREC-3. TREC disks one and two contain texts taken from five distinct sources: the AP newswire, (U.S.) Department of Energy publications, the *Federal Register*, the *Wall Street Journal*, and Ziff-Davis Publishing's *Computer Selects* disks. Five databases are used in the merging experiments, with each database consisting of the set of documents from a single source. In the tables below, these databases are designated AP, DOE, FR, WSJ, and ZIFF respectively.

2.2 Retrieval systems

2.2.1 SMART

The SMART system is a text retrieval system developed at Cornell University [4]. In its basic configuration, SMART implements a vector-space model of retrieval [11]. Documents and queries are represented as vectors in a T-dimensional space where T is the number of distinct content terms in the set of documents. A measure of similarity between vectors in the space is used to compute the similarity between texts.

For the experiments performed here, each of the five TREC subcollections and the entire collection was indexed using the standard SMART routines to produce term frequency weighted document vectors and term frequency times inverse document frequency weighted query vectors. Similarities were computed using the cosine measure. In SMART parlance, all of the retrieval runs are "lnc-ltc" runs.

2.2.2 TOPIC

For the TOPIC experiments we used a standard "off-the-shelf" version of TOPIC, a full-text retrieval system that is available for a wide-range of hardware and software environments. The TOPIC system provides the user with a wide-range of query operators including: fuzzy equivalents of the Boolean operators, proximity operators, pattern matching operators, and structured field operators. These can be combined into complex, named search expressions that generate a relevance score when executed against the document database. The TOPIC document database is organized into a series of inverted indexes that are optimized for speed and size.

The document collection for these experiments was indexed using TOPIC's standard database building tools and we used the TOPIC queries developed for the TREC-2 and TREC-3 experiments without modification. See Tong [12] for an example of a TOPIC query and to get further details on the system itself.

To generate the data for these experiments, we simply ran the TOPIC queries against the appropriate subsets of the TREC test collection, reporting the first 100 documents retrieved. The single collection results were formed by selecting the top 100 documents for each query across all the subcollections: since TOPIC does not make use of corpus-dependent statistics, such as document frequency, this is precisely the ranking that would be produced if TOPIC were run on all the documents at once.

In the case where a query retrieves fewer than 100 documents for a specific subcollection, we use just those retrieved. This is a significant difference between TOPIC and SMART, which almost always retrieves at least 100 documents as shown in Table 1. The table gives the average over sets of 50 queries of the number of documents retrieved per subcollection for both TOPIC and SMART. This difference in the average number of retrieved documents affects performance scores as described in Section 3.1 and the behavior of the merging techniques as described in Section 3.2.

2.3 Database merging strategies

Both the MRDD and QC database merging strategies use relevance assessments from past queries to select the number of documents to request from each collection for the current query so as to retrieve N documents total for the current query. If a non-zero number of documents, λ, is to be retrieved from a given collection, the (natural language) query is submitted to that collection and the λ most highly ranked documents are returned. The methods differ in how they use the training data to determine the λ's.

2.3.1 Query clustering

The basic steps in the Query clustering (QC) merging method are presented in Figure 1. This method summarizes the behavior of queries by the number of relevant documents they retrieve in the top L (a parameter of the method) documents. For all of these experiments, we use $L = 100$ since 100 has worked well in previous experiments [18]. The training phase is depicted in Step 1 of the figure. For each collection, the set of training queries is clustered using Ward's algorithm and the inverse of the number of documents retrieved in common between two queries in the top L documents as the distance metric. Query vectors are created in a vector space formed from the set of training queries, and the vectors of the queries contained within each cluster are averaged to create cluster centroids. Each cluster is also assigned a weight that reflects how effective queries in the cluster are on that collection. The weight is computed as the average number of relevant documents retrieved by queries in the cluster, where a document is considered to be retrieved if it is in the top L documents.

Step 2 of Figure 1 depicts how the merged result is created for new queries. The cluster whose centroid vector is most similar to the query vector is selected for the query and the associated weight is returned. The set of weights returned by all the collections is used to apportion the retrieved set: when N documents are to be returned to the user and w_i is the weight returned by collection i, $\frac{w_i}{\sum_{i=1}^{C} w_i} * N$ documents are retrieved from collection i. For example, assume the total number of documents to be retrieved is 100, and there are five collections. If the weights returned by the collections are 4, 3, 3, 0, 2, then 33 documents would be retrieved from collection 1, 25 each from collections 2 and 3, none from collection 4, and 17 from collection 5. However, if the weights returned were 4, 8, 4, 0, 0 then 25 documents would be retrieved from each of collections 1 and 3, and 50 documents would be retrieved from collection 2. The weight of a cluster for a single collection in isolation is not meaningful; it is the relative difference in weights returned by the set of collections over which the fusion is to be performed that is important.

The last step in the process is computing a final ranking of the documents to be returned. Since the document scores are not necessarily compatible, they cannot be used to rank documents from separate collections, and since the merging strategy has no access to collection statistics, there is no basis for normalizing the scores. A simple round-robin approach whereby the first documents retrieved from each collection are placed in the ranking before the second documents from each collection, which in turn are placed before the third documents, etc. is known to produce poor rankings [17, 5]. We assign the final document ranking by a random process. To select the document for rank r, a collection is chosen by rolling a C-faced die that is biased by the number of documents still to be picked from each of the C collections. For example, a collection with 10 documents remaining to be placed will be twice as likely to be selected as a collection with only 5 documents remaining to be placed. The next document from that collection is placed at rank r and removed from further consideration. Rankings produced in this way are guaranteed to respect the rankings produced by the individual search engines while giving preference to collections that are most likely to contain relevant documents.

2.3.2 Modeling relevant document distributions

Figure 2 summarizes the steps of the second database merging technique, Modeling Relevant Document Distributions (MRDD). This method summarizes the behavior of queries by the ranks of the retrieved relevant documents. Once again, the first step is a training phase. One set of query vectors is created in a vector space constructed from all of the training queries[1]. The system also stores an explicit representation of the relevant document distribution for each training query in each collection. This distribution is equivalent to the ranks of the relevant documents in the top 100 retrieved.

The first step in processing a new query, q, is to determine the k training queries that are most similar to it.

[1]For the MRDD method, we created this vector space using an augmented SMART stop word list. The stop word list was augmented by words that occur frequently in the stylized TREC query statements but do not convey query content, such as *relevant*, *document*, *including*, *system*, and *identify*.

	TOPIC					SMART				
	AP	DOE	FR	WSJ	ZIFF	AP	DOE	FR	WSJ	ZIFF
Queries 51–100	83.8	59.2	71.5	93.3	75.4	100	98	100	100	98
Queries 101–150	90.4	67.7	73.6	99.0	82.6	100	98	100	100	100
Queries 151–200	51.2	26.4	37.1	79.6	30.2	100	98	100	100	100

Table 1: The average number of documents retrieved when 100 documents are requested for the TOPIC and SMART retrieval systems.

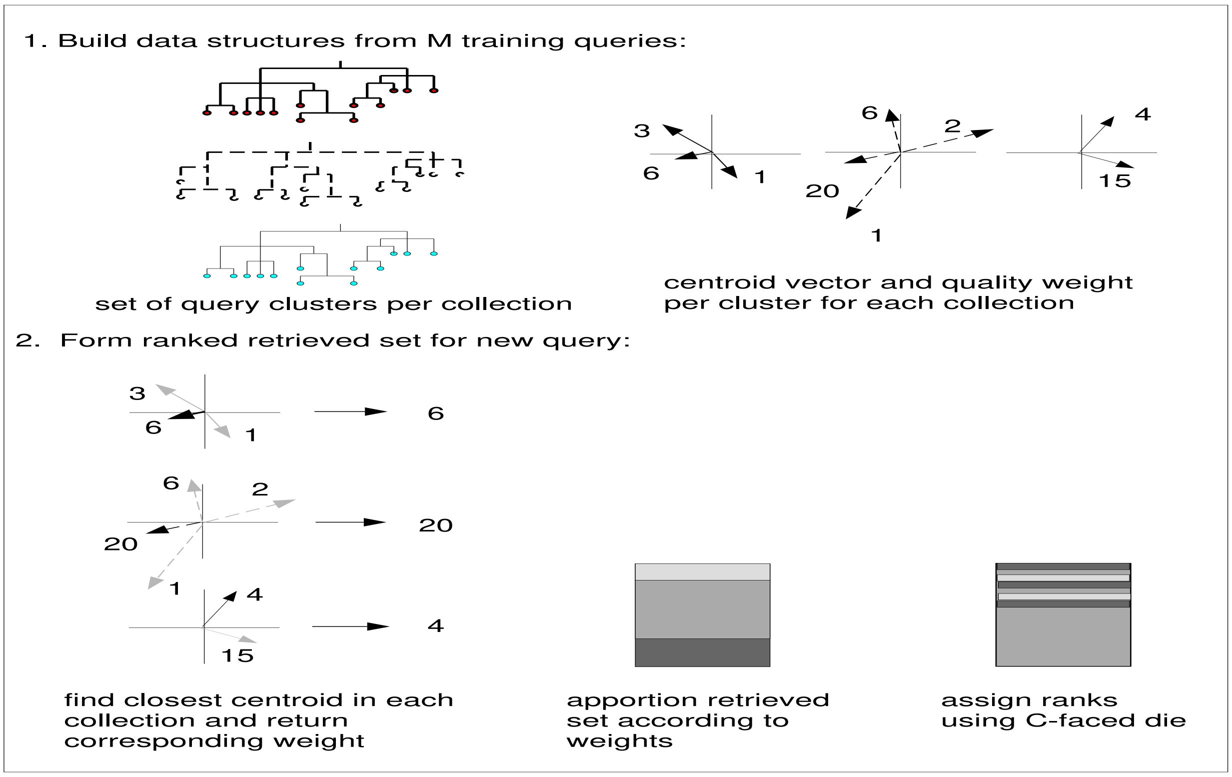

Figure 1: The QC database merging strategy.

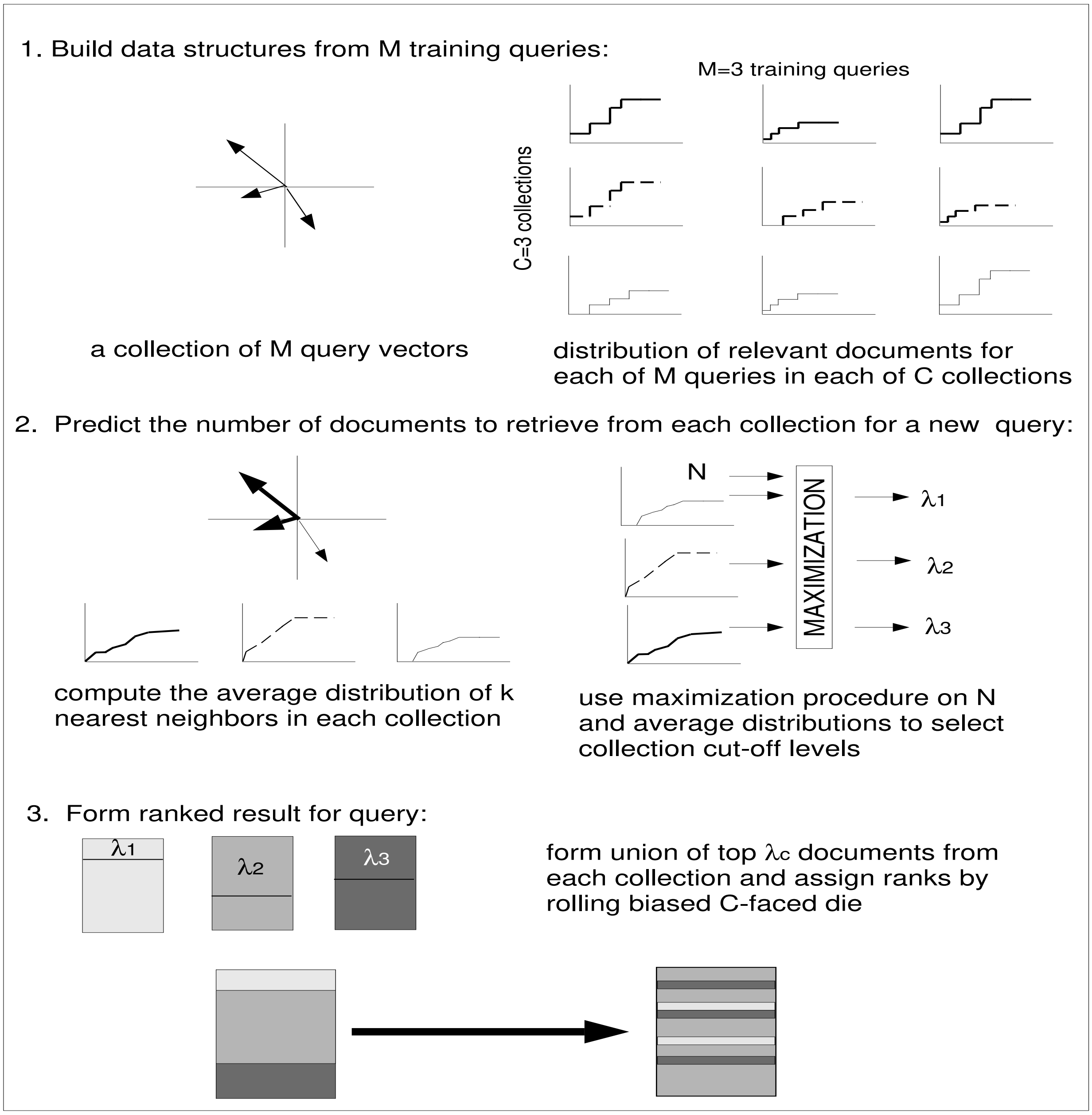

Figure 2: The MRDD database merging strategy.

On the basis of earlier experiments, we use $k = 6$ in this work [18]. A model of q's retrieval behavior in each collection is constructed by averaging the relevant document distributions of these k nearest neighbors. The value of the average relevant document distribution over k queries at rank r is defined as the mean number of relevant documents retrieved by the set of queries in ranks 1 to r.

Using the model distributions to predict the number of relevant documents that would be retrieved for q from each of the collections at different cut-off levels, the system computes the number of documents to retrieve from each collection such that the total number of relevant documents that would be retrieved is maximized. For a particular set of distributions, there may be different combinations that retrieve an equal number of relevant documents, and any given combination may have a "spill": a number of documents remaining to reach N retrieved but for which no additional relevant documents can be retrieved regardless of which collection the documents are selected from. For example, given the two distributions A: 1 1 0 0 1 and B: 1 0 0 1 0 where 1 represents a relevant document and 0 a nonrelevant document and $N = 5$, there are several different combinations that produce the maximum of 3 relevant retrieved: 2 from A, 3 from B; 5 from A; 4 from A, 1 from B; etc. A combination that takes 2 from A and 1 from B has a spill of 2 since no more relevant documents can be retrieved. When multiple different combinations exist, we use the combination that is computed first. The spill is distributed among the collections in proportion to the number of documents that would be otherwise retrieved from it. The final ranking of the retrieved documents is produced by the same procedure as is used in the QC method.

3 Experimental Results

To measure the effectiveness of a merged result, we compare the precision of the merged ranking with that of a ranking produced by searching the corresponding single collection. The precision of a retrieved set is the proportion of retrieved documents that are relevant. To compute the precision of a ranked list of documents, the precision is computed at a specified cut-off level. For example, Prec(15) is the precision of the set of documents in ranks 1–15.

This section reports the precision obtained in two sets of retrieval experiments. In the first experiment, the QC and MRDD strategies are used to merge the results of multiple databases when each database is searched using a common retrieval system. The purpose of this experiment is to determine if the behavior of a merging strategy changes depending on the retrieval system used. The second experiment combines the search results from different retrieval systems within a single merged run.

3.1 Merging the results from one retrieval system

In the first experiment, each of the TOPIC and SMART retrieval systems is used to retrieve 100 documents from each of the five TREC subcollections. The 150 TREC queries used in the study are divided into a set of 100 training queries and 50 test queries. The 100 training queries are used to train the QC and MRDD methods. These methods are then used to create merged rankings for the remaining 50 queries. The 50 test queries are also run against the single TREC collection to provide a benchmark for comparison.

The precision of these runs averaged over the 50 test queries is given in Tables 2 and 3. The precision is measured after 10, 20, 50, and 100 documents have been retrieved. The reported precision figures are somewhat depressed since if a given search did not retrieve at least 100 documents the empty ranks are assumed to be non-relevant. This procedure affects the TOPIC searches much more than the SMART searches since, as was discussed earlier, SMART almost always retrieves at least 100 documents[2]. Also given in the table is the percentage difference between the precision obtained by a run and its corresponding single collection run.

At small (10 and 20) numbers of retrieved documents, the QC merging strategy is consistently between 10–13% less effective than the single collection run for both retrieval engines. Such consistency provides strong evidence that the merging strategy is independent of the search engine used to produce the underlying document rankings.

The MRDD strategy appears to be less consistent: for the TOPIC searches the MRDD strategy is always more effective than the QC strategy, while it is usually less effective for the SMART searches. The explanation for this difference lies in method's dependence on the training data, which is much greater for the MRDD method than it is for the QC method [16]. The MRDD method stores and exploits the entire rankings of the training data instead of summarizing their performance in a set of weights. While theoretically this extra information should improve the MRDD performance, it also means that the method occasionally makes fine distinctions on the basis of dubious data. In general, the distribution of relevant documents in the TOPIC searches varies less across queries than does the distribution in the SMART searches. Since the TOPIC searches have less variance in which collections contain relevant documents, the training data is a more accurate representation of the test data for the TOPIC system, and the MRDD method performs relatively better.

3.2 Merging results from multiple retrieval systems

The second experiment examines the performance of the merging strategies when multiple search engines are used within a single merged run. The subcollection searches used in the previous experiment — the five subcollections searched by SMART and the same five subcollections searched by TOPIC — are again used in this experiment, and the merging algorithms are told that there are ten collections from which the merged results can be created. The precision obtained by these 10-collection searches, averaged over the 50 test queries, is given in Table 4. The table also gives the percentage change over the single collection TOPIC run ("%T") and the single collection SMART run ("%S"). For convenience, the table repeats the precision obtained by the TOPIC and SMART single collection runs.

The part of the merging algorithm that actually produces the final ranked list of documents was modified for this experiment to eliminate duplicate documents. If the current document to be inserted into the ranked list is already in the list, the next document in that particular collection's ranking is attempted to be inserted. Note that we do not solve the general problem of duplicate document detection [10, 19]; in our experimental set-up, duplicate documents have identical document identifiers.

[2]This procedure is used since we are not also reporting recall, the percentage of relevant documents that are retrieved. Given identical retrieved set sizes, a run with higher precision than another run will also have higher recall. However, when retrieved set sizes differ, precision may be increased simply by retrieving fewer documents.

	Prec(10)		Prec(20)		Prec(50)		Prec(100)	
Single Collection	.5220	—	.4620	—	.3836	—	.3062	—
QC Merging	.4680	–10%	.4160	–10%	.3328	–13%	.2372	–23%
MRDD Merging	.5140	–2%	.4520	–2%	.3672	–4%	.2844	–7%

a) Training queries 51–150; Test queries 151–200

	Prec(10)		Prec(20)		Prec(50)		Prec(100)	
Single Collection	.5640	—	.5140	—	.4464	—	.3506	—
QC Merging	.4820	–14%	.4460	–13%	.3803	–15%	.3054	–13%
MRDD Merging	.5520	–2%	.5030	–2%	.4220	–5%	.3314	–5%

b) Training queries 51–100,151–200; Test queries 101–150

	Prec(10)		Prec(20)		Prec(50)		Prec(100)	
Single Collection	.5612	—	.5255	—	.4522	—	.3769	—
QC Merging	.5040	–10%	.4700	–11%	.3852	–15%	.3064	–19%
MRDD Merging	.5180	–8%	.4710	–10%	.3972	–12%	.3198	–15%

c) Training queries 101–200; Test queries 51–100

Table 2: The effectiveness of the QC and MRDD merged results as compared to the single collection results for the TOPIC retrieval system.

	Prec(10)		Prec(20)		Prec(50)		Prec(100)	
Single Collection	.5320	—	.4970	—	.4320	—	.3648	—
QC Merging	.4800	–10%	.4390	–12%	.3880	–10%	.3298	–10%
MRDD Merging	.4660	–12%	.4330	–13%	.3712	–14%	.3198	–12%

a) Training queries 51–150; Test queries 151–200

	Prec(10)		Prec(20)		Prec(50)		Prec(100)	
Single Collection	.5920	—	.5570	—	.5020	—	.4646	—
QC Merging	.5360	–9%	.5060	–9%	.4580	–9%	.4184	–10%
MRDD Merging	.4980	–16%	.4760	–15%	.4420	–12%	.3950	–15%

b) Training queries 51–100,151–200; Test queries 101–150

	Prec(10)		Prec(20)		Prec(50)		Prec(100)	
Single Collection	.6000	—	.5790	—	.5460	—	.4786	—
QC Merging	.5380	–10%	.5050	–13%	.4496	–18%	.3948	–18%
MRDD Merging	.5300	–12%	.5180	–11%	.4672	–14%	.4148	–13%

c) Training queries 101–200; Test queries 51–100

Table 3: The effectiveness of the QC and MRDD merged results as compared to the single collection results for the SMART retrieval system.

	P(10)	%T %S	P(20)	%T %S	P(50)	%T %S	P(100)	%T %S
TOPIC	.5220	— —	.4620	— —	.3836	— —	.3062	— —
SMART	.5320	— —	.4970	— —	.4320	— —	.3648	— —
QC	.4720	–10% –11%	.4520	–2% –9%	.3980	+4% –8%	.3230	+5% –11%
MRDD	.5380	+3% +1%	.4850	+5% –2%	.4204	+10% –3%	.3420	+12% –6%

a) Training queries 51–150; Test queries 151–200

	P(10)	%T %S	P(20)	%T %S	P(50)	%T %S	P(100)	%T %S
TOPIC	.5640	— —	.5140	— —	.4464	— —	.3506	— —
SMART	.5920	— —	.5570	— —	.5020	— —	.4646	— —
QC	.5260	–7% –11%	.4890	–5% –12%	.4420	–1% –12%	.3978	+13% –14%
MRDD	.5500	–2% –7%	.5040	–2% –10%	.4608	+3% –8%	.4036	+15% –13%

b) Training queries 51–100,151–200; Test queries 101–150

	P(10)	%T %S	P(20)	%T %S	P(50)	%T %S	P(100)	%T %S
TOPIC	.5612	— —	.5255	— —	.4522	— —	.3769	— —
SMART	.6000	— —	.5790	— —	.5460	— —	.4786	— —
QC	.5300	–5% –12%	.5060	–4% –13%	.4548	+1% –17%	.3974	+5% –17%
MRDD	.5560	–1% –7%	.5310	+1% –8%	.4736	+4% –13%	.4152	+10% –13%

c) Training queries 101–200; Test queries 51–100

Table 4: The effectiveness of the QC and MRDD merged when merging collections searched by different retrieval systems.

	QC						MRDD					
	TOPIC		SMART		both		TOPIC		SMART		both	
AP	297	6.5%	1140	24.8%	68	1.5%	85	1.8%	971	20.0%	13	0.3%
DOE	19	0.4%	152	3.3%	3	0.1%	0	0.0%	429	8.8%	3	0.1%
FR	158	3.5%	103	2.2%	5	0.1%	2	0.0%	41	0.9%	1	0.0%
WSJ	813	17.7%	1074	23.4%	347	7.6%	1006	20.7%	1546	31.9%	428	8.8%
ZIFF	74	1.6%	332	7.2%	5	0.1%	19	0.4%	302	6.2%	4	0.1%
total	1361	29.7%	2801	60.9%	428	9.4%	1112	22.9%	3289	67.8%	449	9.3%

a) Training queries 51–150; Test queries 151–200

	QC						MRDD					
	TOPIC		SMART		both		TOPIC		SMART		both	
AP	467	9.4%	1136	22.9%	66	1.3%	121	2.4%	929	18.6%	31	0.6%
DOE	14	0.3%	104	2.1%	7	0.1%	3	0.1%	101	2.0%	1	0.0%
FR	51	1.0%	89	1.8%	2	0.1%	0	0.0%	107	2.2%	0	0.0%
WSJ	1132	22.9%	1101	22.2%	491	9.9%	1287	25.8%	1557	31.2%	519	10.4%
ZIFF	90	1.8%	195	4.0%	10	0.2%	22	0.5%	302	6.1%	5	0.1%
total	1754	35.4%	2625	53.0%	576	11.6%	1433	28.8%	2996	60.1%	556	11.1%

b) Training queries 51–100,151–200; Test queries 101–150

	QC						MRDD					
	TOPIC		SMART		both		TOPIC		SMART		both	
AP	386	7.9%	1172	24.0%	113	2.3%	98	2.0%	1394	28.4%	59	1.2%
DOE	50	1.0%	142	2.9%	4	0.1%	8	0.1%	116	2.4%	4	0.1%
FR	37	0.8%	147	3.0%	1	0.0%	3	0.1%	85	1.7%	2	0.0%
WSJ	1097	22.4%	1264	25.8%	304	6.2%	1180	24.0%	1463	29.8%	318	6.5%
ZIFF	56	1.1%	117	2.4%	5	0.1%	28	0.6%	150	3.0%	4	0.1%
total	1626	33.2%	2842	58.1%	427	8.7%	1317	26.8%	3208	65.3%	387	7.9%

c) Training queries 101–200; Test queries 51–100

Table 5: The source of retrieved documents in the 10-collection merged runs.

To verify that the merging algorithms are indeed combining the rankings from both search engines, and not consistently ignoring collections searched by one system, we examined the source of the documents in the final rankings. These distributions are shown in Table 5 which gives the number and percentage of documents retrieved in a merged run for each collection summed over all fifty test queries. The collection entries are divided into whether the document was only available in the TOPIC ranking, only available in the SMART ranking, or was available in both, where "available" means retrieved by the system at a rank lower than or equal to the λ value computed for that collection. The overall total number of documents retrieved is less then 5000 for each run because collections did not always retrieve as many documents as was requested from them. In general, approximately 30% of the documents retrieved are unique to the TOPIC searches, another 10% are common to both systems, and the remaining 60% are unique to SMART. More documents are retrieved from SMART searches since the SMART collection searches are somewhat more effective than the TOPIC searches and because the SMART searches retrieve more documents in general (see Table 1).

The merged result formed when both search engines are used is consistently more effective than the corresponding TOPIC-only merged run. The 10-collection MRDD ranking is also consistently better than the corresponding SMART ranking, although the 10-collection QC ranking is sometimes less effective than the 5-collection SMART ranking, particularly for test queries 101–150. TOPIC queries 151–200 (half of the training set for this set of test queries) retrieved comparatively few relevant documents from the AP collection. Thus the λ values computed for the 10-collection merged run contain many more WSJ documents and fewer AP documents than does the 5-collection SMART run. Unfortunately, for a noticeable number of queries this results in relevant AP documents that are retrieved in the SMART-only run being replaced by irrelevant WSJ documents in the 10-collection run.

One tantalizing aspect of the 10-collection results is that the merged results are often more effective than the TOPIC single collection run. While part of the improvement at 100 documents retrieved can be attributed to the fact that the 10-collection merged run retrieves more documents than the TOPIC run, the frequent improvement at lower ranks (and matching the effectiveness of the SMART single collection run for test queries 151–200) suggests that the MRDD and QC database merging techniques may be able to improve the results of individual queries by combining different formulations of the query as is in other query data fusion work [1, 2, 3, 9]. If MRDD and QR are indeed viable query data fusion methods, they can be used to improve searches of the World Wibe Web: most of the web search engines including Alta Vista, Excite, Lycos, and Infoseek return a ranked list of pages in response to a natural language query, and each searches the same collection — the Web — though in markedly different ways.

4 Conclusion

The Modeling Relevant Document Distributions (MRDD) and Query Clustering (QC) database merging strategies are isolated techniques, meaning that they are independent of the search engine that produces the component rankings and require no data from the component databases at runtime to decide how many documents to retrieve from each database. Isolated strategies are suitable for use in environments where there is no central authority. They are efficient to use since they require only moderate amounts of metadata and since only databases that contribute documents to the final retrieved set need be communicated with. Tests using two quite different search engines, TOPIC and SMART, confirm that the behavior of the MRDD and QC techniques is the same regardless of which engine, or combination of engines, is used.

The MRDD and QC techniques are dependent on the quality of the training data. The MRDD technique in particular can exploit good training data to produce excellent results, but can equally easily be led astray by poor training data. One way to improve the merging strategies' effectiveness may be to detect mismatches between the current query and previous queries, and to use a different strategy for computing the number of documents to retrieve from each collection if a mismatch is detected. For example, in the MRDD technique, if none of the nearest neighbors has a similarity greater than a threshold value, the number of documents to retrieve would be based on the relevant document distribution averaged over all the training queries.

Whether the MRDD and QC techniques are viable query data fusion methods (i.e., methods that improve query performance by combining the output of different formulations of that query) requires further investigation. In the experiments included in the paper, the fused result was often better than the TOPIC single collection run, but did not improve the SMART single collection run. It may be that fusing base runs whose individual levels of effectiveness were more similar would permit the combined result to be more effective than all the component runs, or that merging more than two results would cause more substantial improvement. We plan to address these issues in future work.

Acknowledgements

This work was undertaken when Voorhees was a member of the technical staff of Siemens Corporate Research, Princeton, NJ and Tong was director of the Advanced Technology Group at Verity, Inc, Mountain View, CA. Our thanks to Chris Buckley and Donna Harman who read a draft of this paper and suggested many improvements.

References

[1] Brian T. Bartell, Garrison W. Cottrell, and Richard K. Belew. Automatic combination of multiple ranked retrieval systems. In *Proceedings of the 17th International Conference on Research and Development in Information Retrieval (SIGIR-94)*, July 1994.

[2] N.J. Belkin, C. Cool, W.B. Croft, and J.P. Callan. The effect of multiple query representations on information system performance. In Robert Korfhage, Edie Rasmussen, and Peter Willett, editors, *Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 339–346, June 1993.

[3] N.J. Belkin, P. Kantor, E.A. Fox, and J.A. Shaw. Combining the evidence of multiple query representations for information retrieval. *Information Processing and Management*, 31(3):431–448, 1995.

[4] Chris Buckley. Implementation of the SMART information retrieval system. Technical Report 85-686, Com-

puter Science Department, Cornell University, Ithaca, New York, May 1985.

[5] James P. Callan, Zhihong Lu, and W. Bruce Croft. Searching distributed collections with inference networks. In Edward A. Fox, Peter Ingwersen, and Raya Fidel, editors, *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 21–28, July 1995.

[6] Luis Gravano, Kevin Chang, Hector Garcia-Molina, Carl Lagoze, and Andreas Paepcke. STARTS: Stanford protocol proposal for internet retrieval and search. Technical report, Digital Library Project, Stanford University, November 1993. http://www-db.stanford.edu/~gravano/starts.ps.

[7] Donna K. Harman. The first Text REtrieval Conference (TREC-1), Rockville, MD, U.S.A, 4–6 November, 1992. *Information Processing and Management*, 29(4):411–414, 1993.

[8] David Hawking and Paul Thistlewaite. Proximity operators — so near and yet so far. In Donna K. Harman, editor, *The Fourth Text REtrieval Conference (TREC-4)*, pages 131–143, October 1996. NIST Special Publication 500-236.

[9] David A. Hull, Jan O. Pedersen, and Hinrich Schutze. Method combination for document filtering. In *Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 279–287, August 1996.

[10] John W. Kirriemuir and Peter Willett. Identification of duplicate and near-duplicate full-text records in database search-outputs using hierarchic cluster analysis. In *Proceedings of the Electronic Library and Visual Information Research Conference*, May 1995.

[11] G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Communications of the ACM*, 18(11):613–620, November 1975.

[12] Richard M. Tong. Interactive document retrieval using TOPIC. In Donna K. Harman, editor, *Overview of the Third Text REtrieval Conference (TREC-3) [Proceedings of TREC-3.]*, pages 201–209, April 1995. NIST Special Publication 500-225.

[13] Geoffrey Towell, Ellen M. Voorhees, Narendra K. Gupta, and Ben Johnson-Laird. Learning collection fusion strategies for information retrieval. In *Proceedings of the 12th Annual Machine Learning Conference*, July 1995.

[14] Ellen M. Voorhees. Agent collaboration as a resource discovery technique. In *Working Notes of the CIKM Workshop on Intelligent Information Agents*. Third International Conference on Information and Knowledge Management (CIKM'94), December 1994.

[15] Ellen M. Voorhees. Software agents for information retrieval. In *Working Notes of the 1994 Spring Symposium on Software Agents*, pages 126–129. American Association for Artificial Intelligence, March 1994.

[16] Ellen M. Voorhees. Siemens TREC-4 report: Further experiments with database merging. In Donna K. Harman, editor, *The Fourth Text REtrieval Conference (TREC-4)*, pages 121–130, October 1996. NIST Special Publication 500-236.

[17] Ellen M. Voorhees, Narendra K. Gupta, and Ben Johnson-Laird. The collection fusion problem. In Donna K. Harman, editor, *Overview of the Third Text REtrieval Conference (TREC-3) [Proceedings of TREC-3.]*, pages 95–104, April 1995. NIST Special Publication 500-225.

[18] Ellen M. Voorhees, Narendra K. Gupta, and Ben Johnson-Laird. Learning collection fusion strategies. In Edward A. Fox, Peter Ingwersen, and Raya Fidel, editors, *Proceedings of the 18th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 172–179, July 1995.

[19] Tak W. Yan and Hector Garcia-Molina. Duplicate removal in information dissemination. In *Proceedings of VLDB-95*, September 1995.