The Collection Fusion Problem

Ellen M. Voorhees Narendra K. Gupta Ben Johnson-Laird
Siemens Corporate Research, Inc.
Princeton, NJ
{ellen,gupta,ben}@scr.siemens.com

Abstract

This paper examines the feasibility of merging the results of retrieval runs on separate, autonomous document collections into an effective combined result. In particular, we examine two collection fusion techniques that use the results of past queries to compute the number of documents to retrieve from each of a set of subcollections such that the total number of retrieved documents is equal to N, the number of documents to be returned to the user. The fusion techniques are independent of the particular weighting schemes, similarity measures, and retrieval models used by the component collections.

Our official TREC-3 runs are fusion runs in which $N = 1000$; other runs investigate the effects of varying N. These results show that the precision averaged over the 50 queries is within 10% of the precision of an effective single collection run for a wide range of values of N.

1 Introduction

Data fusion techniques have been used in information retrieval to improve the effectiveness of a given query by merging the results produced by different formulations of the query in the context of a single database [1, 2, 3, 7]. In this paper, we look at a different fusion problem — the *collection fusion* problem. The goal of a collection fusion technique is to combine the retrieval results from multiple, independent collections into a single result such that the effectiveness of the combination approximates the effectiveness of searching the entire set of documents as a single collection. Since the collections are assumed to be independent, the fusion techniques must cope with arbitrary similarity measures and weighting schemes.

The TREC environment is an attractive vehicle for investigating the collection fusion problem for two reasons:

- The TREC collection is composed of distinct subcollections that are different sizes, cover diverse topics, and have different retrieval characteristics.
- Other groups participating in the TREC workshops treat the set of documents as a single collection. We can therefore measure how closely the results of our fusion strategies approximate effective single-collection results.

In another report [9], we used the portions of the TREC collection available after TREC-2 to develop and evaluate the collection fusion strategies that are described here. We show there that for modest numbers of documents to be retrieved ($N < 100$), the effectiveness of the fused result is usually within 10% of the effectiveness of a run in which the entire set of documents is treated as one collection (hereafter called a *single collection* run).

In TREC-3, we look at larger values of N, and, as a consequence, investigate methods for producing a total ranking in a fused result. Our official TREC-3 runs, `siems1` and `siems2`, represent the output from the two fusion techniques when $N = 1000$. The runs are completely automatic, ad hoc runs; Siemens did not do any routing or interactive runs in TREC-3.

The next section of the paper gives a formal definition of the collection fusion problem and describes our fusion strategies. The following section compares the effectiveness of the fused results to a particular single collection run. The final section summarizes our findings.

2 The Collection Fusion Problem

Consider a set of collections each of which is accessed through its information server. For a given query Q, each collection I has a certain number of relevant documents, denoted by n_Q^I. We assume when the server for I is presented with Q, it returns a list of documents sorted by decreasing similarity to the query. Clearly, the number of retrieved relevant documents increases as the number of retrieved documents increases until all n_Q^I relevant documents are retrieved. We denote the distribution of the relevant documents in the retrieved set (i.e., the ranks at which the relevant documents occur) by $F_Q^I(S)$, a function of the number of retrieved documents S. The collection fusion problem can now be formally stated as follows: Given a query Q, information servers $I_1, I_2, ...I_C$ and N the total number of documents to be retrieved, find the values of $\lambda_1, \lambda_2, ...\lambda_C$ such that $\sum_{i=1}^{C} \lambda_i = N$ and $\sum_{i=1}^{C} F_Q^{I_i}(\lambda_i)$ is maximum, i.e. the total number of relevant documents retrieved is maximized. Of course, in practice F_Q^I is not known and must be approximated.

There are two simple approaches to approximating the relevant document distributions. One approach is to assume each collection has as many relevant documents as the next and the relevant documents are identically distributed across the servers' rankings. In this case, retrieving an equal number of documents from each collection maximizes the number of relevant documents retrieved on average. In practice, this is a poor approximation because the different collections have different specialties and thus do not have equal numbers of relevant documents. Tests comparing the effectiveness of this uniform strategy versus a single collection run using topics 1-150 demonstrate that the uniform strategy degrades the single collection performance by over 40% [9].

The second approach is to assume the similarity values across collections are comparable and to select the cut-off levels such that the documents with the N greatest similarities across all collections are retrieved. This approach is viable if the similarity measures are indeed comparable. However, the incorporation of collection-dependent frequency counts in the document or query weights (such as idf weights) invalidates this assumption. This effect, noted by Dumais in her TREC work [6], is illustrated by the example in Figure 1. The example shows TREC topic 144 and the rankings produced by each of the five subcollections. For this query, the top similarities strategy retrieves only seven relevant documents when $N = 100$ while the uniform strategy retrieves fourteen. The problem is that both "United Nations" and "organization" are rare terms in the DOE subcollection, and thus irrelevant documents that contain these words have a higher similarity to the query than do the relevant documents in the AP subcollection.

Our approach to approximating the relevant document distributions is to learn them from the results of past queries. That is, training queries are used to build

TREC query 144 (abbreviated):

management effectiveness of the United Nations (UN) and related organizations

Individual collection retrieval results:

	AP	DOE	FR	WSJ	ZIFF
# rel in top 100	17	0	0	6	0
# rel in top 20	9	0	0	5	0
greatest similarity	.1549	.2032	.1235	.2368	.1637
# docs with sim > .1549	0	7	0	6	4

Text of DOE document with greatest similarity:

The relations of the IAEA with the United Nations, with subsidiary UN organs, with other UN specialized agencies, with intergovernmental organizations and with non-governmental organizations are briefly described.

Figure 1: Example in which retrieving the N documents with highest similarities across subcollections is very ineffective.

models of both the content and the search behavior of each collection. Once training is complete, previously unseen queries are answered by matching the new query's content to that of the training queries and using the associated models to compute the number of documents to retrieve from each collection.

2.1 Modeling Relevant Document Distributions

In the relevant document distribution fusion technique, we use the training queries to explicitly build a model of the distributions of the relevant documents in the retrieved set of each information server I, and use the models in a maximization procedure to obtain the number of documents to retrieve from each collection. We model the relevant document distribution of a query, q, by averaging the relevant document distributions of the k most similar training queries (i.e., the k nearest neighbors of q). We use the vector space model to compute the similarities among the queries. The vector space is built from the set of training queries, and the nearest neighbors of q are those training queries that have the highest cosine similarity with q.

The average relevant document distribution over k queries is computed by taking the average of the number of relevant documents retrieved by the set of queries after each document retrieved. Once the average relevant document distribution is computed for the current query for each collection, the distributions and the total number of documents to be retrieved are passed to a maximization procedure. This procedure finds the cut-off level, λ_i, for each collection that maximizes the number of relevant documents retrieved (the current maximization procedure simply does an exhaustive search). The computed cut-off levels are the number of documents selected from each collection.

The union of the top λ_i documents of each collection produces a set of retrieved documents, but does not impose a total order on those documents. (It does produce a set of partial orders since, by assumption, all of the documents retrieved from the same server are ordered relative to one another.) However, a total ordering is necessary to produce ranked output. We impose a total ordering probabilistically. 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. 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 partial orders and give preference to collections that are the most likely to contain relevant documents. Figure 2 summarizes the steps of the relevant document distribution fusion process.

2.2 Query Clustering

The second fusion strategy does not form an explicit model of a collection's relevant document distribution. Instead, the system learns a measure of the quality of a search for a particular topic area on the collection. The number of documents retrieved from a collection for a new query is proportional to the value of the quality measure computed for that query.

As in the relevant document distribution approach, the query clustering fusion strategy uses query vectors in the vector space formed from the set of training queries. Topic areas are represented as centroids of query clusters. For each collection, the set of training queries is clustered using the number of documents retrieved in common between two queries as a similarity measure. The assumption is that if two queries retrieve many documents in common they are about the same topic. The centroid of a query cluster is created by averaging the vectors of the queries contained within the cluster. This centroid is the system's representation of the topic covered by that query cluster.

The training phase also assigns to a cluster 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 "retrieved" if it was among the first L (a parameter of the method) documents.

After training, queries are processed as follows. 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 are used to apportion the retrieved set such that when N documents are to be returned and w_i is the weight returned by collection i, $\frac{w_i}{\sum_{i=1}^{C} w_i} * N$ (rounded appropriately) 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.

As in the previous method, once the number of documents to be selected from each collection is determined, a final ranking of the documents must be created. To investigate the effects of a different rank-

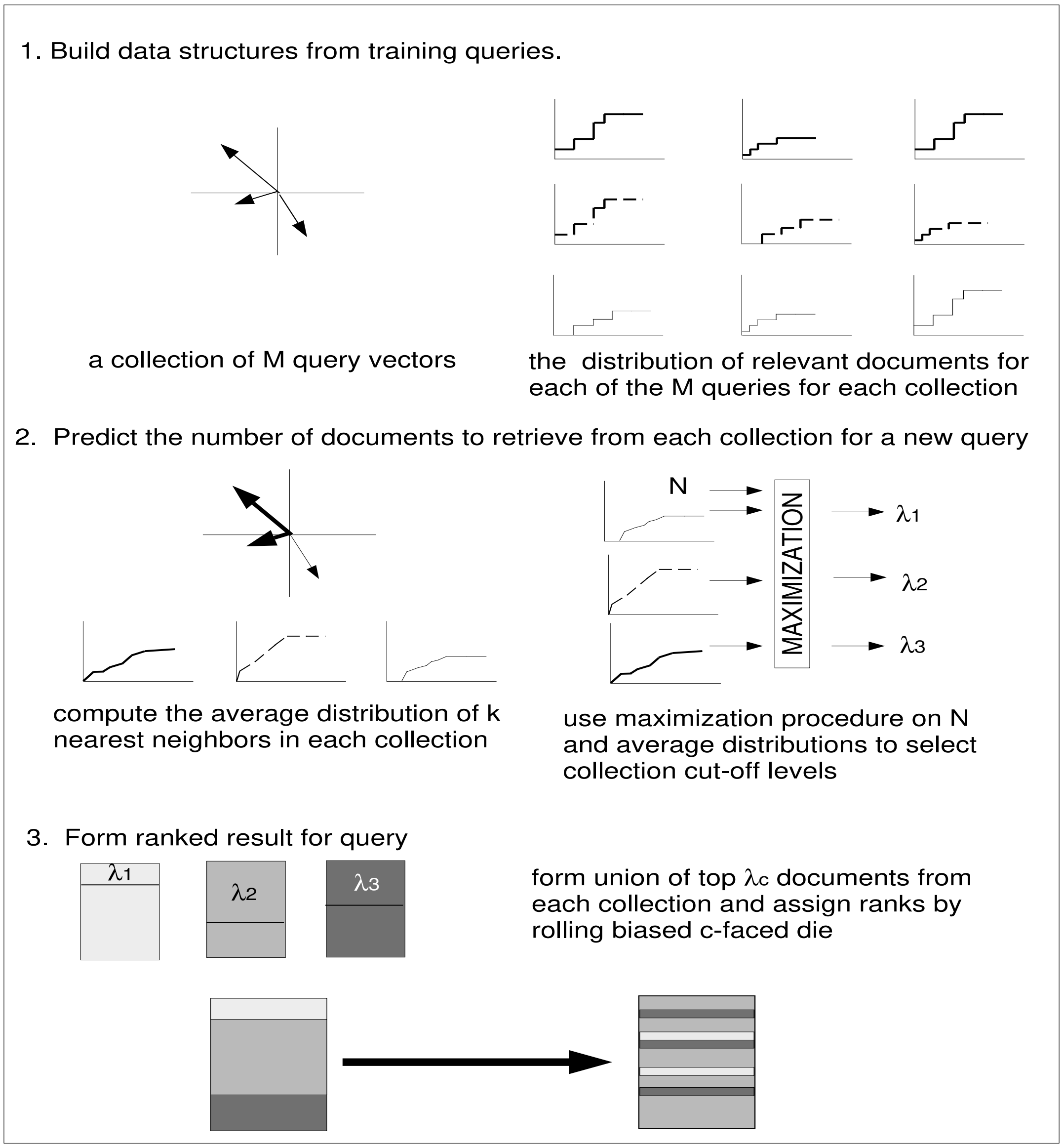

Figure 2: The relevant document distribution fusion strategy.

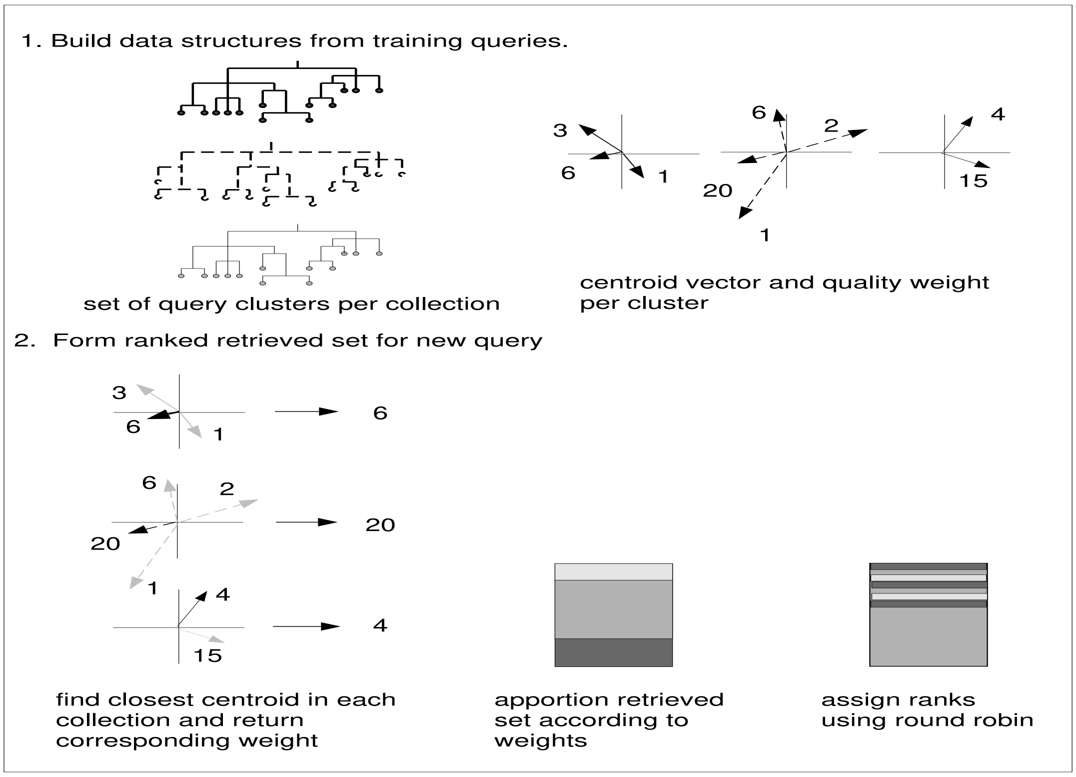

Figure 3: The query clustering fusion strategy.

ing method, for this fusion strategy we simply chose the documents round-robin style from each collection that had documents not yet inserted into the final ranking. While this total ordering also preserves each of the partial orders, it does not give preference to collections that are believed to have more relevant documents. Figure 3 summarizes the steps of the query clustering fusion strategy.

3 Retrieval results

The fusion results produced for TREC-3 were created using topics 1–150 as training topics and testing on topics 151–200. Queries were produced from the topics using the standard SMART [4] indexing routines on all of the fields except the Definition and Summary fields. The document set consisted of the documents on disk volumes one and two. Each of the five collections on those disks, AP, DOE, FR, WSJ, and ZIFF, was indexed separately, again using the standard SMART indexing routines. The retrieval runs on the component collections (that is, both the retrieval runs that were used for training and the runs that produced the rankings to be fused) used SMART's "`lnc`" weights for the document collections, "`ltc`" weights for the queries, and the inner product as the similarity measure (this is equal to the cosine of the vectors given the weights used) [5].

TREC run `siems1` was produced by the relevant document distribution fusion method. Four nearest neighbors were used to produce the average relevant document distributions. The nearest neighbors were selected using the cosine similarity metric on term-frequency weighted vectors.

Run `siems2` was produced by the query clustering fusion method. The training queries were clustered using the Ward clustering method [8] with the reciprocal of the the number of documents retrieved in common in the top 100 documents as the distance metric. Final clusters were formed by cutting the resulting dendrogram at a distance of 1. Cluster centroids were created using term-frequency-weighted vectors produced from the "Concepts" field only. (Since the "Narrative" section of each topic begins with *A relevant document will contain...*, using the Narrative field to build the centroids caused clusters about document processing to absorb all the queries that were dissimilar to the other queries.) The parameter L, the number of documents used to create the weights associated with each cluster, was set to 100.

Table 1 shows how the two official TREC runs compared to the median of the other TREC submissions. The fusion techniques were radically different from the median values for precision after 100 and 1000 documents once: the relevant document distribution technique on Topic 151 obtained the worst results (5 relevant documents retrieved after 1000 documents). This topic concerns how inmates cope with prison overcrowding and is not very similar to any of topics 1–150. We did not control for the common phrasing used in the Narrative section in the relevant document distribution method, and as a result the four nearest neighbors of Topic 151 were Topics 65, 63, 35, and 33 which have to do with document processing, machine translation, information retrieval and the like. Using these queries to compute the number of documents to retrieve from each collection, the vast majority of the documents returned were from the ZIFF collection, which, unfortunately, has no known relevant documents about prison overcrowding. We hope to eliminate this effect by incorporating inverse query frequency weights (weighting terms inversely proportional to the number of training queries they appear in) when calculating nearest neighbors.

For comparison purposes, we ran queries 151–200 against a single collection consisting of all of the documents on disks one and two using `lnc-ltc` weights and inner product similarity. The non-interpolated average precision, the average precision after 1000 document retrieved, and the total number of retrieved relevant documents for the single collection run, `siems1`, and `siems2` is given in Table 2. The percentage difference of these measures for the fused runs over the single collection run is also included in the table. The number of retrieved relevant documents (and thus the average of the precision at 1000 documents) for the fused runs is comparable to the single collection run, especially for the query clustering fusion run. The non-interpolated average precision suffers in comparison to the single collection run, however, demonstrating that the single collection run is retrieving the relevant documents at smaller ranks. Clearly the round-robin method of imposing a total ordering on the final retrieved set is not very effective. Run `siems2` retrieved 600 more relevant documents over the 50 queries than run `siems1`, but has a smaller average precision.

Users are seldom going to want 1000 documents returned in a single reply, so to be useful the fusion techniques must approximate the behavior of a single collection run at small ranks. Table 3 gives the average precision of the single collection and fused runs when they are evaluated after 10, 30, 100, and 200 documents. The relevant document distribution fusion run (`siems1`) is within 11% of the single collection performance at these ranks, but the query clustering fusion run (`siems2`) is still harmed by the poor ranking algorithm.

Finally, earlier tests with the fusion

	Rel ret @ 100			Rel ret @ 1000		
	#<med	#=med	#>med	#<med	#=med	#>med
Rel doc dist (`siems1`)	25	3	22	25	4	21
Query clust (`siems2`)	41	2	7	24	5	21

Table 1: Number of queries whose precision after 100 and 1000 documents retrieved is less than, equal to, or greater than the median precision averaged over all TREC submissions.

	Non-interpolated average precision		Average precision at 1000		Total number relevant retrieved	
Single collection	.2643	—	.1218	—	6088	—
Rel doc dist (`siems1`)	.2088	−21%	.1076	−12%	5381	−12%
Query clust (`siems2`)	.1873	−29%	.1196	−2%	5981	−2%

Table 2: Effectiveness of fused runs compared to single collection run for 1000 documents retrieved.

	Eval at 10		Eval at 30		Eval at 100		Eval at 200	
Single collection	.5320	—	.4707	—	.3648	—	.2885	—
Rel doc dist (`siems1`)	.4720	−11%	.4187	−11%	.3268	−10%	.2607	−10%
Query clust (`siems2`)	.3940	−26%	.3467	−26%	.2820	−23%	.2434	−16%

Table 3: Average precision of official TREC runs evaluated at several small ranks.

	Eval at 10		Eval at 30		Eval at 100		Eval at 200	
Single collection	.5320	—	.4707	—	.3648	—	.2885	—
Rel doc dist, $N = 10$	.4400	−17%						
Query clust, $N = 10$	.4760	−11%						
Rel doc dist, $N = 30$	.4320	−19%	.4087	−13%				
Query clust, $N = 30$	.4560	−14%	.4247	−10%				
Rel doc dist, $N = 100$	.4800	−10%	.4320	−8%	.3340	−8%		
Query clust, $N = 100$	.4060	−24%	.3940	−16%	.3392	−7%		
Rel doc dist, $N = 200$	.5080	−5%	.4287	−9%	.3368	−8%	.2650	−8%
Query clust, $N = 200$	.4040	−24%	.3640	−23%	.3226	−12%	.2714	−6%

Table 4: Average precision for varying numbers of requested documents.

techniques suggest that they are more effective when smaller numbers of documents are requested [9]. We repeated the evaluations in Table 3 while varying N, the number of documents to be retrieved by the fusion algorithms (the official TREC runs used $N = 1000$). These results are given in Table 4. The diagonal entries in the table eliminate any ranking effect, and therefore the relative performance of the query clustering fusion technique is better there. The relevant document distribution fusion technique needs to request a moderate number of documents (100–200) to eliminate spurious effects caused by the volatility of the relevant document distributions of the training queries at very small numbers of documents retrieved. In general, the results in Table 4 provide evidence that these fusion techniques can approximate the effectiveness of a single collection run at the ranks that will be of interest to most users.

4 Conclusions

Our goal in TREC-3 was to explore the feasibility of combining the retrieval results of searches on independent collections into a single ranked list such that the effectiveness of the combined result is comparable to that of a single collection run. We have described two collection fusion techniques that are able to produce effective retrieved sets for the TREC-3 ad hoc queries. These techniques require only the ranked lists and relevance assessments of previous queries to produce the final retrieved set, and are applicable for a wide range of number of documents to be retrieved.

A user's perception of the quality of a result is strongly influenced by how well the documents are ranked, especially when large number of documents are returned. From the results presented here, it appears that the fusion techniques select enough documents from marginally relevant collections such that round-robin ranking (taking all documents ranked first in their respective collection, then all second, etc.) is not effective. Ranking documents by perceived quality of the collection from which they were drawn is much more effective.

While these fusion results are encouraging, it remains to be seen to what extent these results are applicable in retrieval environments other than TREC. For example, as Figure 4 demonstrates, a fusion strategy that is better than the single collection retrieval method for the TREC-3 ad hoc topics is simply to retrieve $N/2$ documents from the AP collection and the remaining $N/2$ documents from the *Wall Street Journal* collection. This method works well because there is only one topic, Topic 190, for which the majority of the relevant documents are not in the combination of those two collections [1]. Further experimentation is required to determine if our fusion methods will generalize to other environments.

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*

[1] The majority of the relevant documents for Topic 190 is in Ziff. Both the relevant document distribution and the query clustering fusion methods take the majority of the documents they retrieve from the Ziff collection for Topic 190.

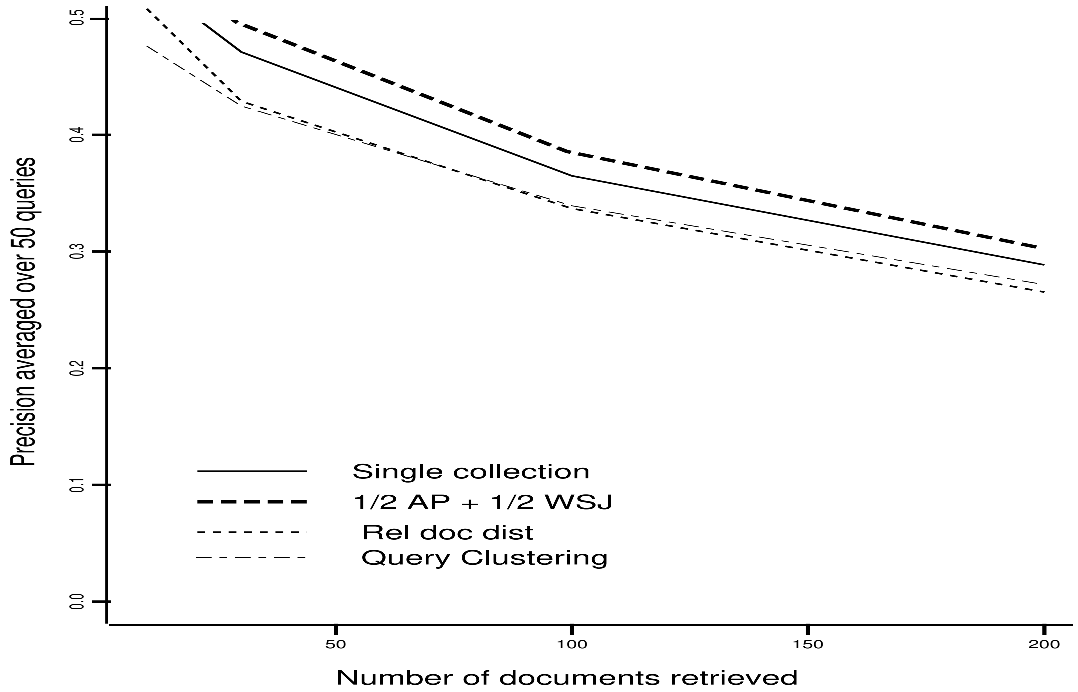

Figure 4: Effectiveness of four retrieval strategies.

Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pages 339–346, June 1993.

[3] N.J. Belkin, P. Kantor, C. Cool, and R. Quatrain. Query combination and data fusion for information retrieval. In Donna K. Harman, editor, *Proceedings of the Second Text REtrieval Conference (TREC-2)*, pages 35–44, March 1994. NIST Special Publication 500-215.

[4] Chris Buckley. Implementation of the SMART information retrieval system. Technical Report 85-686, Computer Science Department, Cornell University, Ithaca, New York, May 1985.

[5] Chris Buckley, Gerard Salton, and James Allan. Automatic retrieval with locality information using SMART. In D. K. Harman, editor, *Proceedings of the First Text REtrieval Conference (TREC-1)*, pages 59–72. NIST Special Publication 500-207, March 1993.

[6] Susan T. Dumais. LSI meets TREC: A status report. In Donna K. Harman, editor, *Proceedings of the First Text REtrieval Conference (TREC-1)*, pages 137–152, March 1993. NIST Special Publication 500-207.

[7] Edward A. Fox and Joseph A. Shaw. Combination of multiple searches. In Donna K. Harman, editor, *Proceedings of the Second Text REtrieval Conference (TREC-2)*, pages 243–252, March 1994. NIST Special Publication 500-215.

[8] L. Kaufman and P. Rousseeuw. *Finding Groups in Data — An Introduction to Cluster Analysis*. Wiley, 1990.

[9] Ellen M. Voorhees, Narendra Gupta, and Ben Johnson-Laird. Learning collection fusion strategies. In preparation.