Result Merging Strategies for a Current News MetaSearcher

Yves Rasolofo[†], David Hawking[‡], Jacques Savoy[†]

[†] Institut interfacultaire d'informatique
Université de Neuchâtel, Switzerland
Yves.Rasolofo@unine.ch, Jacques.Savoy@unine.ch

[‡] CSIRO Mathematical and Information Sciences
Canberra, Australia
David.Hawking@cmis.csiro.au

Abstract

Metasearching of online current news services is a potentially useful Web application of distributed information retrieval techniques. We constructed a realistic current news test collection using the results obtained from 15 current news websites (including ABC News, BBC and AllAfrica) in response to 107 topical queries. Results were judged for relevance by independent assessors. Online news services varied considerably both in the usefulness of the results sets they returned and also in the amount of information they provided which could be exploited by a metasearcher. Using the current news test collection we compared a range of different merging methods. We found that a low-cost merging scheme based on a combination of available evidence (title, summary, rank and server usefulness) worked almost as well as merging based on downloading and re-scoring the actual news articles.

Keywords: Metasearch, distributed information retrieval, evaluation, Web.

1. Introduction

Major Web search engines such as Google (www.google.com) and AllTheWeb (www.alltheweb.com) periodically traverse the Web link graph and fetch Webpages for inclusion in their centralised index. This process is known as "crawling" or "spidering" and it is subject to two significant limitations.

First, many large Web sites (such as the huge PubMed collection of medical abstracts www.ncbi.nlm.nih.gov) deny access to crawlers. Because of this, significant quantities of useful information are excluded from centralized search engines. Second, there are limits on how frequently centralized indexes can be updated -- a complete crawl is expensive and takes a very long time. Because of this, centralized search engines are not well suited to indexing volatile or ephemeral content. The version of "today's top story" indexed by a general search engine may be obsolete before the index update is published and may remain in the index for weeks.

Distributed information retrieval (or metasearching) techniques potentially solve the problems of incomplete coverage and data volatility by selecting and combining local search engines. Search facilities provided by sites like PubMed and at online newspapers can rapidly reflect changes in local content. The question is whether local search engines can be harnessed together to produce efficient and effective search over a broad scope.

Here we report our experiences in building a metasearcher designed to provide up-to-date search over a significant number of rapidly changing current news sites. We focus on how to merge the results from the search engines at each site into a high-quality unified list.

A metasearcher is a broker which forwards search queries to a set of primary search engines (each assumed to provide incomplete coverage[1] of the set of documents to be searched) and presents to the searcher a single merged list of results. This situation is depicted in Figure 1. In general, a metasearcher may incorporate solutions to the problems of:

1. identification and characterization of primary search services whose results are to be merged;
2. selection, for reasons of efficiency or effectiveness, of a subset of available search services (so called server or collection selection problem);
3. translation of the searcher's requests into the relevant query language of each primary search service and getting/parsing results;
4. merging of results from the primary search services into a single high-quality list (collection fusion or results merging problem).

[1] Coverage is the proportion of available documents which are actually indexed.

A number of Web metasearchers have been available to the public for some time. MetaCrawler (Selberg and Etzioni, 1997) is perhaps the best known example. Like others, this search engine merges the results of a number of general-purpose primary search engines, each of which attempts to index "the whole of the Web". Metasearchers like this can increase effective coverage when there is relatively poor overlap between the sets of documents indexed by the primary engines (Lawrence and Lee Giles, 1999). It also attempts to improve result quality by boosting the rank of documents ranked highly by several of the engines (voting). As another example, we may cite www.all4one.com which sends the user's request to only four search engines, (Alta Vista, Yahoo!, HotBot and Excite) and presents the retrieved items in four different windows, one for each search engine. When using a non-merging metasearcher like this, the user must inspect different windows, a situation that is subject of various criticisms from a usability point of view (Nielsen, 2000).

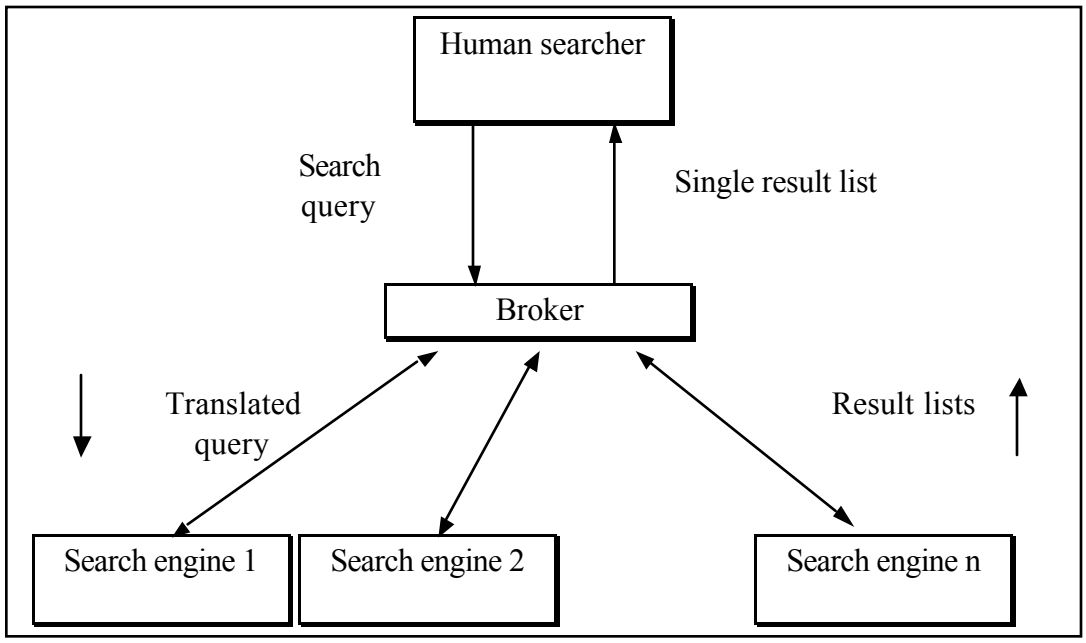

Figure 1: Human searcher, broker and primary search services

Metasearching, a search approach based on the paradigm of distributed information retrieval, have been promoted as a solution to some of the difficulties experienced by centralized search engines such as Alta Vista and Google, due to the huge size and rapid growth of the World Wide Web. Despite some doubts, the proportional coverage of the largest search engines is

believed to have increased despite several years of sustained Web growth. More importantly, several recent studies of search engine effectiveness (Craswell *et al.*, 2001), (Hawking *et al.*, 2001a), (Hawking *et al.*, 2001b) have failed to show an effectiveness benefit on precision-oriented tasks from this type of metasearching.

It is possible that the Web will grow to be so large that metasearching becomes essential. It is equally possible that other more clearly beneficial applications for metasearching techniques already exist.

This paper is organized as follows: The rest of this section reviews past work and describes the underlying characteristics of current news servers. Section 2 describes our experimental framework and the methodology used in this paper. Section 3 evaluates individually the various online news sites used in our study. In Section 4, we present and evaluate various merging strategies used to present to the user a single list of the retrieved items. Finally, we present some conclusions that can be drawn from our study.

1.1. Previous work in selection and results merging strategies

A large number of published studies have addressed the server selection and results merging problems in the context of test collections such as TREC ad hoc, TREC VLC and WT2g, as for example (Callan *et al.*, 1995), (Callan, 2000), (French *et al.*, 1998), (Gravano *et al.*, 1994), (Hawking and Thistlewaite, 1999), (Le Calvé and Savoy, 2000), (Rasolofo *et al.*, 2001). In these experiments, the test collection is divided into a set of disjoint partitions, each representing the documents indexed by a search service.

When analyzing selection procedures, suggested methods for identifying the partitions containing the greatest number of relevant or "good" documents are compared, and the results of retrieval over selected subsets of partitions are compared with retrieval results for the full collection. The methods proposed by Gravano *et al.* (1997), Hawking and Thistlewaite (1999) for server selection and by Kirsch (1997) are unrealistic in the sense that they rely on the widespread adoption of protocols by which search engines can communicate statistics or metadata about their holdings. At the time of writing, this seems very unlikely.

On the other hand, when studying merging experiments, the operation of a search service is typically simulated over each test collection partition and the quality of merged results lists corresponding to various merging strategies are compared. Merging of partitioned collections represents real-world search in the sense that several public Web search engines (such as Inktomi, Google and Fast) in fact operate by dividing the collection (the set of pages discovered and retrieved for indexing by a spider[2]) and indexing each partition separately. However, the suggested merging schemes are simple in this particular case because each partition is managed by the same search algorithm and because communication of global statistics is straightforward. When dealing with metasearching, we are faced with different retrieval engines for which the indexing and retrieval strategies are usually unknown. This fact invalidates most of the previous studies in results merging approach, with the exception of the work of Le Calvé & Savoy (2000) which however requires a learning phase and Craswell *et al*.'s work (1999) which used mixed engines.

1.2. The current study

As mentioned, it is probable that real-world search applications exist in which metasearching offers significant benefits over centralized indexes. We consider that search of current news services such as CNN, the BBC and ABCNEWS might constitute such an example, because:

- there are many online news services which provide a search interface to their own recent news stories;
- some news services may be only available to subscribers. However, article titles and summaries are often available for searching and they are enough for metasearching. It is up to the user to decide whether he or she will subscribe;
- high quality current news search (as opposed to search of news archives) requires that searchers be able to retrieve news stories the instant they are placed on the site. This is clearly not achievable by a centralized index which crawls news sites periodically. Of

[2] A spider (also known as a crawler or robot) discovers pages for indexing by recursively following hyperlinks from a seedlist of Web pages.

course, we recognize that there are ways in which centralized search engine companies can provide rapid indexing of news articles. For example, business relationships may be formed with news organizations in which search engine companies are notified of each new news story as it is posted, or in which subscriber-only data may be indexed for a fee. Alternatively, current news services may be identified and "spidered" much more frequently than other Web sites by centralized search engines.

Current news metasearch represents a more realistic and a more difficult application area in comparing metasearching methods than does disjoint partitioning of an existing test collection. The partitioning and distribution of a collection across multiple non-cooperating servers has no obvious application on the Web. By contrast, current news metasearch is realistic for reasons listed above. Characteristics of current news metasearch are: The search algorithms used by the various news sites are unknown and heterogeneous. These search servers do not cooperate with each other or with a broker, and the information available to guide selection and merging is variable in type and is generally small in quantity. Finally, response from current news servers is often slow and the issue of timeouts must be addressed.

In order to address these questions, we have created a current news metasearch test collection by broadcasting more than one hundred topical news queries to fifteen current news Web sites, and judging responses for relevance. In this paper, we will document this new test collection and report its use in comparing various results merging strategies based on information we were able to extract from news articles and servers.

Our objective is to study the issues involved in operating a metasearch engine in a real (Web) environment. It is immediately obvious that current news services do not provide the collection statistics whose availability is assumed in many previous studies, for example: Gravano *et al.* (1994, 1997), Callan *et al*. (1995), French *et al.* (1998), etc. Lawrence and Giles, (1998) and Craswell *et al.*, (1999) avoid this problem by downloading complete documents and rescoring them locally, but this increases latency and network traffic.

Callan *et al.*, (1999), and Craswell *et al.*, (1999) have investigated ways of estimating collection statistics by using streams of queries to sample the collections held by each server. It is uncertain that these techniques would be useful with current news search, given the rapidly

changing nature of current news collections. Here, we attempt to make use of information as is provided by some or all current news services in the form of document titles, summaries and dates. We also observe the relative effectiveness of the search engines operated by the current news services and attempt to take that into account in merging, following Craswell *et al.* (2000).

2. Experimental framework

In order to conduct metasearching experiments, we have written a metasearch engine which is the fruit of an international collaboration between Switzerland and Australia. The code of this metasearch engine was written in Perl and represents around 5,000 lines of code. While the user interface is available both in Switzerland (http://www.unine.ch/info/news/) and Canberra (http://peace.anu.edu.au/Yves/MetaSearch/), the core engine is running in Canberra. This section introduces the basic ideas underlying our metasearcher. Section 2.1 states issues we had to face during the implementation of the metasearcher. Section 2.2 describes the fifteen selected online news servers together with examples of queries generated by users. Section 2.3 presents the problem of unavailable documents or broken links included in the servers answers. Section 2.4 explains how we have established the relevance assessments of our selected queries. Finally, the last section describes the evaluation methodology used in this paper.

2.1. Issues related to current news servers

This study focused upon the evaluation of news metasearcher addressing fifteen servers. It is important to note that a metasearcher for current news servers differs in different aspects from those manipulating conventional search engines. These differences are the following:

- in general, the titles of documents returned by current news services are more accurate and reliable than those typically available with other Web documents. Thus, titles may be a beneficial source of evidence in ranking articles;
- the documents are particularly volatile and become out of date very quickly. The article date is therefore critical for some topics. Moreover, it is important to choose appropriate topics for the evaluation;

- the retrieval effectiveness and efficiency of the news search engines vary widely from one server to another. It would be beneficial to take into account these differences when merging results lists.

Moreover when setting up our news metasearcher, we faced the following additional problems:

- although a server may not return any document for a given query at a given time, it does not always mean that it does not contain any documents related to the query. The reason could be transient server or network overload. This problem is not serious for our evaluation because our main goal is to find effective ways to merge the results lists from servers;
- it is important to write wrappers[3] that are able to collect all required information which might be of value. We concentrated our efforts on extracting URL, title, summary and document date when such information was available;
- each search engine included in news servers has its own query language. For our evaluation, we used the "AND" Boolean operator or equivalent;
- some engines return front pages containing only headlines. As far as possible, we tried not to include them in the final results list of the corresponding server.

2.2. Queries and current news sites

People from different backgrounds (Australian, African and Swiss) and professions (computer scientists, psychologists, research assistants in sociology, students in philosophy and art-history) were asked to generate short "bag of words" queries that they might write to a search interface of a current news service on the Web. Queries covered world news, science & technology, business, entertainment and sport. A total of 118 queries were collected with a query length varying from one to eight words (average length = 3.21) corresponding roughly to the mean length of search engine queries. From these requests, 114 search queries have at least one

[3] A wrapper provides a common programming interface to a search service. It translates a searcher's request into the query language of a primary search service and extracts information from returned results.

document returned by a server and a total of 107 queries were evaluated, corresponding to queries having at least one relevant document returned. Some of the queries are:

- Lockerbie trial
- dotcom stocks
- ELF corruption case
- MIR space station
- Madagascar eclipse 2001
- Tom Cruise Nicole Kidman separation
- George W. Bush Administration
- Arnaud Clement

Queries were submitted to news sites on February 9th and 26th, 2001. We used a script to remove stop words from queries and submit the remaining words to news servers using the "AND" Boolean operator or equivalent. The news sites included in our evaluation were selected among sites from around the world (east and west Europe, USA, Africa, Australia and Asia) and are:

- ABCNEWS: http://abcnews.go.com
- BBC Online: http://www.bbc.co.uk
- CNET: http://news.cnet.com
- CNN Financial Network: http://cnnfn.cnn.com
- FINANTIAL TIMES: http://www.ft.com
- MSNBC: http://www.msnbc.com
- NEWS.COM:AU: http://www.news.com.au
- THESTAR.COM: http://www.thestar.ca
- THE TIMES: http://www.thetimes.co.uk
- THE TOKYO WEEKLY: http://www.tokyo-weekly.ne.jp
- ALLAFRICA.COM: http://allafrica.com
- THE ST. PETERSBURG TIMES: http://www.sptimes.ru
- USATODAY: http://www.usatoday.com
- DISPATCH Online: http://www.dispatch.co.za
- WORLDNEWS.COM: http://www.worldnews.com

When building our test collection, we considered only the top ten documents returned by each news server, excluding front pages (going past ten if some pages are front pages). Limiting the number of documents per server and request to ten documents, we may potentially find a total of $114 \cdot 10 \cdot 15 = 17{,}100$ documents. However, for many queries (see Table 1), some servers do not return any documents, and others return less than ten articles. Therefore, a total of only 5,544 documents were returned, including 236 broken links. This corresponds to a mean of approximately 46 live documents per query.

As shown in Table 1, from a total of 114 submitted requests, in average only about half of them (57.60) have at least one document returned a news server. With a standard deviation of 30.07, we can clearly see that there is a great variation across news servers answers. Servers fail to return document for various reasons: some servers may be temporary unavailable during query submission, servers failing to respond in a limited delay are timed out (e.g. THESTAR) many times, and some servers like TOKYO-WEEKLY are only updated weekly or are devoted mainly to local news.

Table 2 depicts the number of queries over a total of 114 having at least one relevant item for each of the selected servers. For all news sites, there is a clear decrease (relative to Table 1) in the number of queries, showing that servers tend to send an answer to a user's request even if its database does not contain any relevant information. In other words, the underlying search engine does not know when it knows nothing on a given subject.

Table 1: Number of queries having at least one document returned (among 114 queries)

Server	# queries	%
WORLDNEWS	111	97.4%
THETIMES	95	83.3%
ALLAFRICA	90	78.9%
DISPATCH	90	78.9%
SPTIMES	71	62.3%
MSNBC	63	55.3%
FT	56	49.1%
BBC	56	49.1%
CNNFN	54	47.4%
USATODAY	50	43.9%
ABCNEWS	41	36.0%
NEWS.COM.AU	36	31.6%
CNET	32	28.1%
TOKYO-WEEKLY	10	8.8%
THESTAR	9	7.9%
Total number of queries	114	100%
Average	57.60	50.53%
Standard deviation	30.07	

Table 2: Number of queries having at least one relevant doc returned (among 114 queries)

Server	# queries	%
WORLDNEWS	91	79.8%
DISPATCH	71	62.3%
THETIMES	66	57.9%
BBC	53	46.5%
FT	47	41.2%
ALLAFRICA	45	39.5%
SPTIMES	43	37.7%
ABCNEWS	40	35.1%
MSNBC	36	31.6%
USATODAY	31	27.2%
CNNFN	24	21.1%
CNET	24	21.1%
NEWS.COM.AU	23	20.2%
TOKYO-WEEKLY	4	3.5%
THESTAR	1	0.9%
Total number of queries	107	93.9%
Average	39.93	35.03%
Standard deviation	24.21	

2.3. Broken links

When documents were not available due to broken links, they were removed from the test collection because assessors did not have enough information to judge them. As shown in Table 3, most of the selected news sites keep the number of broken links as low as possible (ten sites out of fifteen have less than 4% of broken links). For two cases however, MSNBC (17.35% of the retrieved articles were not available) and THESTAR (53.57% of broken links), we can see that these online services seem not to update their indexes sufficiently often.

Table 3: Broken links across the news servers

Server	Total links	Broken links	%
NEWS.COM.AU	115	0	0.00%
TOKYO-WEEKLY	28	0	0.00%
BBC	443	1	0.23%
FT	303	1	0.33%
ALLAFRICA	618	3	0.49%
USATODAY	353	2	0.57%
ABCNEWS	161	1	0.62%
DISPATCH	425	4	0.94%
CNNFN	289	4	1.38%
CNET	253	9	3.56%
THETIMES	728	32	4.40%
WORLDNEWS	1,108	66	5.96%
SPTIMES	277	26	9.39%
MSNBC	415	72	17.35%
THESTAR	28	15	53.57%
Sum	5,544	236	4.26%

2.4. Relevance assessments

The relevance assessments were carried out in Canberra, Australia. Five research assistants were recruited for this purpose - four Australians and one U.S. citizen. Each had completed or very nearly completed a university degree. All were familiar with searching and browsing via the Web but none were information technology professionals. None of them were connected with the companies operating the current news sites being evaluated.

Judging was carried out using a Web browser. We implemented a Perl CGI script which presented a merged list of all the results pages for a particular query, interleaved by source. Unfortunately, judging could not be blind as results pages were usually heavily branded with the identity of the source. Clicking on a link opened a new window displaying the target page from the news site. Relevant / Irrelevant (binary) judgments were recorded using radio buttons. The choice of binary judgments is consistent with previous studies in the area of distributed information retrieval and also with TREC ad hoc evaluations. Given the recent finding by Voorhees (2001) that the ranking of systems was significantly perturbed when only "highly

relevant" documents were used, it would be wise to employ at least three levels of relevance in future work. However, since queries in the present study usually referred to quite specific events or entities, there is greater likelihood that two judges will agree on the binary decision than on degrees of relevance.

Judges were asked to imagine that the queries were theirs and to evaluate on that basis. All the documents retrieved for a query were judged by the same person, ensuring consistency of the results. The assessment was performed from February 14th, 2001 to March 13th, 2001.

2.5. Evaluation measures

In order to obtain an overall measure of performance, we used TREC average precision, a relatively stable single-number measure which combines elements of both precision and recall. This measure is used for results merging comparison (Voorhees and Harman, 2000).

A decision rule is required to determine whether or not a given search strategy is better than another. We considered statistical inference methods such as Wilcoxon's signed rank test or the Sign test (Salton and McGill, 1983), (Hull, 1993) applied to average precision. In these cases, the null hypothesis H_0 states that both retrieval schemes produce similar retrieval performance. Such a null hypothesis plays the role of a devil's advocate, and this assumption will be accepted if two retrieval schemes return statistically similar average performance on a query-by-query basis, and rejected if not. Thus, in the tables found in this paper we will use the Sign test, based on the average precision with a significance level fixed at 5%. However, a decision to accept H_0 is not equivalent to the opinion that the null hypothesis H_0 is true, but instead represents the fact that "H_0 has not been shown to be false" resulting in insufficient evidence against H_0. When reading the Sign test results, the decision ">" or "<" means that the corresponding hypothesis "<=" or ">=" is rejected.

As our experimental news metasearcher displays only one page of merged results, we were interested to know the improvement that could be obtained on the average precision after ten and after twenty retrieved documents. These values were also included in our evaluations.

3. Individual news server evaluation

Servers vary considerably in the number of queries for which they return any answer. As shown in column 2 of table 4, out of a total of 118 queries, the selected servers provide answers for an average of 57.6 queries. There is a large difference between the best server, WORLDNEWS which provided 111 answers, and THESTAR with only 9 answers.

In order to obtain a first overview of the retrieval effectiveness of our fifteen news servers, we computed the average precision on an individual basis for each of the selected news services. This mean precision is calculated by averaging the precision achieved by each query returning at least one document (see Table 4).

When analyzing the mean performance provided by these servers (third column of Table 4), the average retrieval performance is 40.67 (median: 39.33) and the standard deviation around this mean is relatively high (30.07). On the other hand, the standard deviation associated with each server's performance shown in the last column indicates that we may encounter a relatively large variation across queries, varying from 25 (THESTAR) to 43.35 (NEWS.COM.AU).

Table 4: Average precision over queries for which the server returned at least one document

Server	# queries	Mean precision (%)	Standard deviation
ABCNEWS	41	81.43	27.68
BBC	56	68.07	31.63
FT	56	63.92	37.47
NEWS.COM.AU	36	50.57	43.35
DISPATCH	90	44.97	36.24
WORLDNEWS	111	43.57	31.18
THETIMES	95	40.78	36.52
CNET	32	39.33	33.10
MSNBC	63	37.01	38.67
SPTIMES	71	35.71	37.90
TOKYO-WEEKLY	10	27.08	41.07
CNNFN	54	25.30	34.66
USATODAY	50	22.01	25.39
ALLAFRICA	90	21.94	29.56
THESTAR	9	8.33	25.00
Average	57.60	40.67	33.96
Standard deviation	19.42	30.07	

In Table 5, we do the same computation except that we exclude requests having a zero precision, resulting in performance increases. From this table, one can see that the average precision for a news server is 62.32%. The performance difference between the best server (ABCNEWS, 83.46) and the worst (USATODAY, 35.51) reflects clearly that the retrieval performance varies to a greater extent in this kind of online service. Individually, the variation in mean performance is shown in the last column of Table 5. In this case, the difference between the largest variation (TOKYO-WEEKLY, 37.33) and the smallest (USATODAY, 23.63) is smaller than in Table 4.

Table 5: Average precision over queries for which the server returned at least one relevant document

Server	# queries	Mean precision (%)	Standard deviation
ABCNEWS	40	83.46	24.73
NEWS.COM.AU	23	79.15	24.97
FT	47	76.17	26.95
THESTAR	1	75.00	N/A
BBC	53	71.92	27.85
TOKYO-WEEKLY	4	67.71	37.33
MSNBC	36	64.77	28.29
SPTIMES	43	58.96	31.46
THETIMES	66	58.70	29.34
DISPATCH	71	57.00	31.21
CNNFN	24	56.92	29.84
WORLDNEWS	91	53.15	25.95
CNET	24	52.44	27.56
ALLAFRICA	45	43.88	27.99
USATODAY	31	35.51	23.63
Average	39.93	62.32	28.36
Standard deviation	24.21	13.45	

When studying the differences between Table 4 and 5, one can see that some servers may contain relevant articles for a subset of the selected topics, and the average precision is high only for those queries. On the other hand, for other requests, the retrieval performance tend to be 0. As an example, Table 5 shows that NEWS.COM.AU is better than BBC and FT news servers when computing only queries that return at least one relevant document. However, the ranking is reversed when looking at Table 4.

4. Merging strategies

In the previous section, we have evaluated individually each news server. However, our purpose is to evaluate metasearching and to achieve this purpose, we have to send the user's request to our fifteen selected servers and to merge the results lists of retrieved documents into a single list to be presented to the user. To achieve this, our broker may adopt various merging strategies that will be analyzed in this section.

As a first approach, we may assume that each news server contains approximately the same number of relevant documents and that they are equally distributed across results lists from servers (Voorhees *et al.*, 1995). Under this assumption, we can set up the final results list in a round-robin manner. Such a round-robin merging strategy (RR) is already used by earlier metasearch engines on the Web. This merging scheme will serve as a baseline for further comparisons in the following sections.

Moreover, we might also consider merging the results lists by sorting retrieved articles according to the score computed by the server they are originated from. Such a merging approach will be called "raw-score merging". However, it was not practical because the document scores were seldom reported. In any case scores would not have been comparable due to differences in indexing and retrieval strategies used by the servers.

In this section, we will thus consider various merging schemes that may produce comparable scores based on the information available from the servers, such as document title. To this purpose, Section 4.1 presents a general scoring function employed by our broker to compute a score to each retrieved article. Section 4.2 explains how we may exploit the document title in defining a document score. In Section 4.3, we explore the possibility to take account of document summaries in our merging scheme. In Section 4.4, we develop a merging approach that utilizes the title and the summary of the retrieved item in order to improve our merging strategy. As another merging approach, Section 4.5 suggests using the document date while Section 4.6 presents how we may obtain various estimated collection statistics in order to propose another merging model. Section 4.7 describes how we may take into account the server usefulness. Finally, the last section proposes to inspect the article contents in order to define a more effective merging strategy.

4.1. Generic document scoring function

Since document scores were seldom reported by the news servers and are not comparable, we had to define a general scoring function that returns comparable scores based on various document fields (e.g., document title, article summary or date) in order to define an effective merging strategy. Thus, for each document i belonging to collection j for the query Q, we will compute a weight, denoted w_{ij} as follows:

$$w_{ij} = \frac{NQW_i}{\sqrt{L_q^2 + LF_i^2}} \qquad (1)$$

within which NQW_i is the number of query words appearing in the processed field of the document i, L_q is the length (number of words) of the query, and LF_i is the length of the processed field of the document i.

This function returns a value of 0 when the intersection between the request and the selected document field is empty. On the other hand, when all search terms and only those appear in the processed field, our function returns the maximum value of $1/\sqrt{2}$ (or 0.7071).

This suggested formula is based on intuition that the more the search keywords appear in the processed document field(s) the greater the probability that the corresponding article is relevant. However, a weighting expression should not be a simple proportion of the number of query words appearing in the document field(s), it is also important to consider at the same time the length of both the request and the document field(s) as suggested in the previous formula.

Some of the algorithms described below may use the rank score, denoted RS_{ij}, and defined as $RS_{ij} = 1000 - SR_{ij}$ where SR_{ij} is the rank given by the server j to the document i. Since we consider a maximum of 10 items per server, this rank score function returns values between 999 for the best ranked document to 990 for the last retrieved article.

When we are able to provide a comparable score for each retrieved document, our broker must merge the various results lists to form a unique list of retrieved articles. To achieve this, we may adopt the RR merging scheme corresponding to the round-robin merging strategy based only on the ranks defined by servers. As an alternative, we may compute for each document a score using the XX field with our generic document score function. Based on these scores, we may adopt the raw-score merging approach that will be denoted SM-XX. However, instead of directly using the document score, we may also consider using only the rank obtained after applying our generic document scoring function. In this case, we will denote such a merging model as RR-XX.

The difference between SM-XX and RR-XX merging approaches is the following. In both cases, we compute a new document score based on the defined scoring function. When using the SM-XX approach, we merge all the results lists together and we sort it according to the document

score. Ties, if any, are broken according to document rank and in favor of the article appearing in the lower rank (for details on notations used in this paper, see appendix 2). In the RR-XX scheme, we independently sort the results list for each server. After these sorts, we apply the round-robin strategy on the new results lists. In this case, we expect to have roughly the same number of documents extracted from each server (the number of extracted items from each server will be the same if all results lists have the same length). On the other hand, using the SM-XX merging strategy, a server providing articles with higher document score will furnish more items in the final merged list.

4.2. Document title scoring (TS)

As a first merging strategy, we may consider using the document title of the retrieved records. Scoring the article using its title is done by assigning to the document the weight $w_{ij} \cdot 100{,}000$ in which w_{ij} is computed according to our generic function described in the previous section and using the title as processed field. However, the value w_{ij} (eq. 1) can be zero for some documents (e.g., article having no title or when no search keyword appears in the title). In this case, instead of assigning a value of 0 to such document, we attach to it its rank score $RS_{ij.}$ After all, such a document cannot be viewed as irrelevant simply because its title does not share any common word with the request. Knowing that the rank score value varies from 990 to 999 on the one hand, and, on the other, that the document title score varies from 0 to 70,710, we are sure when multiplying w_{ij} by 10,000 that the retrieved documents having a document score greater than 0 will be ranked before articles having no title or having a title without any word in common with the request.

In order to compare this merging strategy with the round-robin approach, we have reported in Tables 6 and in Figure 2 the retrieval effectiveness achieved using RR-TS (round-robin merging scheme based on the rank obtained by the document title scoring), SM-TS (raw-score merging approach based on document title scoring) and the baseline solution RR (round-robin merging strategy based on the original ranked lists given by the news servers).

Table 6a: Document title scoring

Merging strategies	Average precision	% change	Precision @10	% change	Precision @20	% change
RR (baseline)	48.97		42.52		40.42	
RR-TS	55.00	12.31%	49.16	15.62%	44.95	11.21%
SM-TS	63.76	30.20%	61.03	43.53%	50.56	25.09%

Table 6b: Sign test results

RR < RR-TS
RR < SM-TS
RR-TS < SM-TS

From data depicted in Tables 6, one can see that both RR-TS and SM-TS merging strategies increase the retrieval effectiveness over the round-robin approach. As shown in Table 6b, this improvement is statistically significant. When comparing the relative merit of our two suggested merging schemes, we may see that the improvement is greater for the SM-TS compared to the RR-TS scheme and this variation is also statistically significant (last row of Table 6b). In fact, the round robin hypothesis is invalidated by the fact that servers hold different numbers of relevant documents (see Section 2.1)

To have an overview of the retrieval performance of these three merging approaches, Figure 2 depicts the precision computed after retrieving different numbers of document. From this figure, it is clear that the SM-TS approach shows better performance at low cut-off values and seems to perform well when considering the first five or the first ten retrieved items.

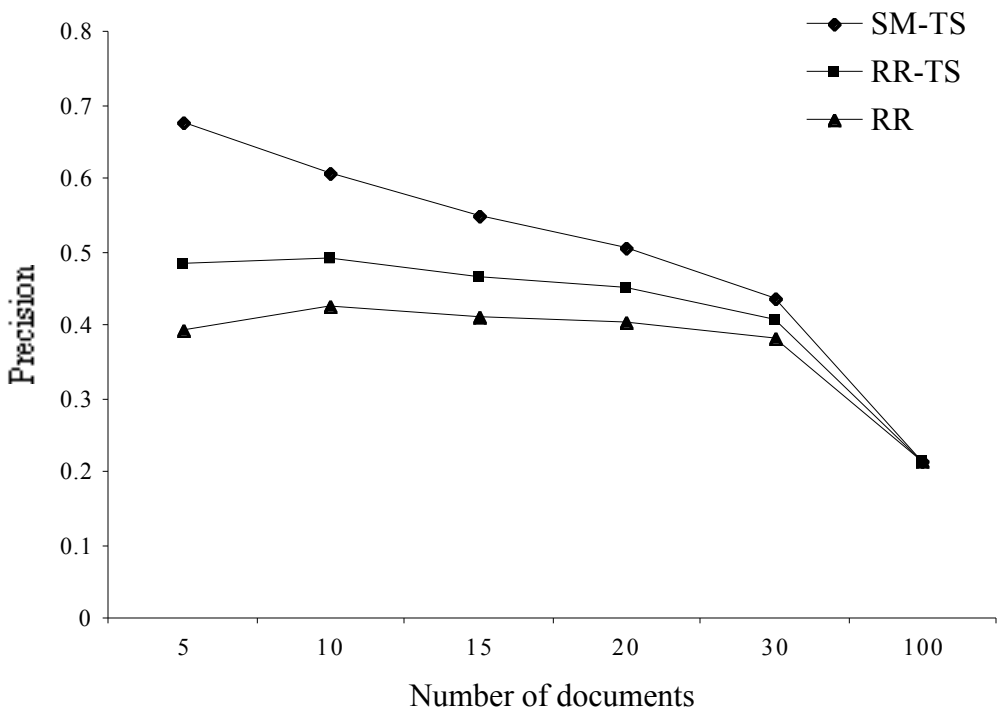

Figure 2: Precision of top results of SM-TS, RR-TS and RR

4.3. Document summary scoring (SS)

When retrieving a document from a current news server, we usually find a document title and an article summary. Such a summary consists of an abstract or sometimes corresponds to the head of the article (e.g., the first 200 words of the news). However, we must indicate that the title is more frequently associated with the document than a summary field.

In defining a merging scheme for our news services, we may exploit our generic document scoring function with this summary field. The retrieval performances achieved by the RR-SS round-robin merging strategy and the SM-SS raw-score merging approach based on the summary field are reported in Table 7a. As shown in Table 7b, these two schemes improve significantly the retrieval effectiveness over the simple round-robin model (RR). As before, the SM-SS raw-score approach presents a better performance that is also statistically significantly better than both round-robin models. Comparing the data of Table 7a with those of Table 6a, we can see that document title scoring seems to produce better retrieval performance than document summary scoring.

Table 7a: Document summary scoring

Merging strategies	Average precision	% change	Precision @10	% change	Precision @20	% change
RR (baseline)	48.97		42.52		40.42	
RR-SS	53.45	9.15%	47.57	11.88%	43.50	7.62%
SM-SS	62.21	27.04%	60.47	42.22%	50.00	23.70%

Table 7b: Sign test results

RR < RR-SS
RR < SM-SS
RR-SS < SM-SS

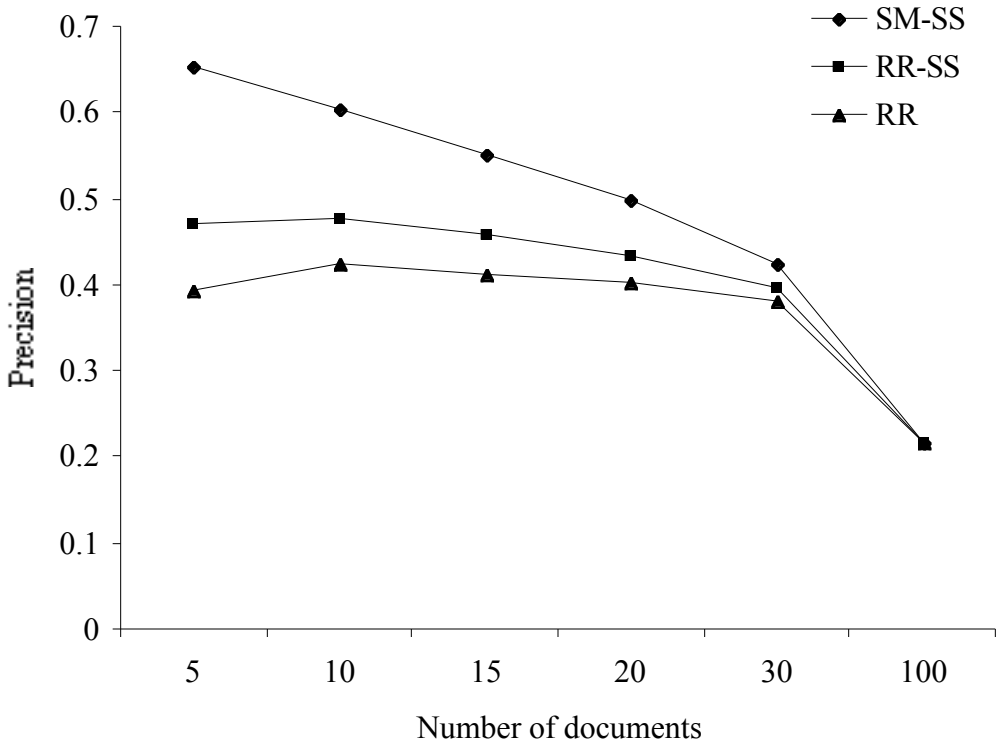

Figure 3: Precision of top results of SM-SS, RR-SS and RR

4.4. Combining title score and summary score (TSS)

In order to improve our merging process, we may take account for both the document title and the article summary in our generic document scoring. To achieve this, a first combined scoring approach for document i belonging to collection j is used as follows:

```
W_ij = TS_ij  /* first use the title field */
if W_ij == 0
     W_ij = SS_ij  /* if none, use the summary field */
endif
if W_ij == 0
     W_ij = RS_ij /* if none, use the rank */
endif
```

where RS_{ij} is the rank score described in Section 4.1. Recall that this latter value is always lesser than both the TS_{ij} or SS_{ij} score value as long as they are greater than zero. This combined scoring approach is grounded on our observation that the document title seems to reflect more accurately the article content than does the associated document summary. Therefore, we only considered the summary score when the title score was null.

The evaluation of this combined merging approach is depicted in Table 8a under the label "RR-TSS1" when we adopt a round-robin merging procedure or "SM-TSS1" when we chose the raw-score merging strategy. The average precision or the precision achieved after retrieving ten or twenty documents represents the highest retrieval performance seen so far. As shown in Table 8b, these two merging schemes are statistically better than the simple round-robin merging strategy (RR). Moreover, the Sign test indicates that the SM-TSS1 approach significantly improves the retrieval effectiveness over both the document title scoring (SM-TS) and the document summary scoring approach (SM-SS).

Table 8a: Title and summary combined scoring

Merging strategies	Average precision	% change	Precision @10	% change	Precision @20	% change
RR (baseline)	48.97		42.52		40.42	
RR-TSS1	56.94	16.28%	52.24	22.86%	46.59	15.26%
SM-TSS1	67.14	37.10%	63.74	49.91%	53.88	33.30%

Table 8b: Sign test results

RR < RR-TSS1
RR < SM-TSS1
RR-TSS1 < SM-TSS1
SM-TS < SM-TSS1
SM-SS < SM-TSS1

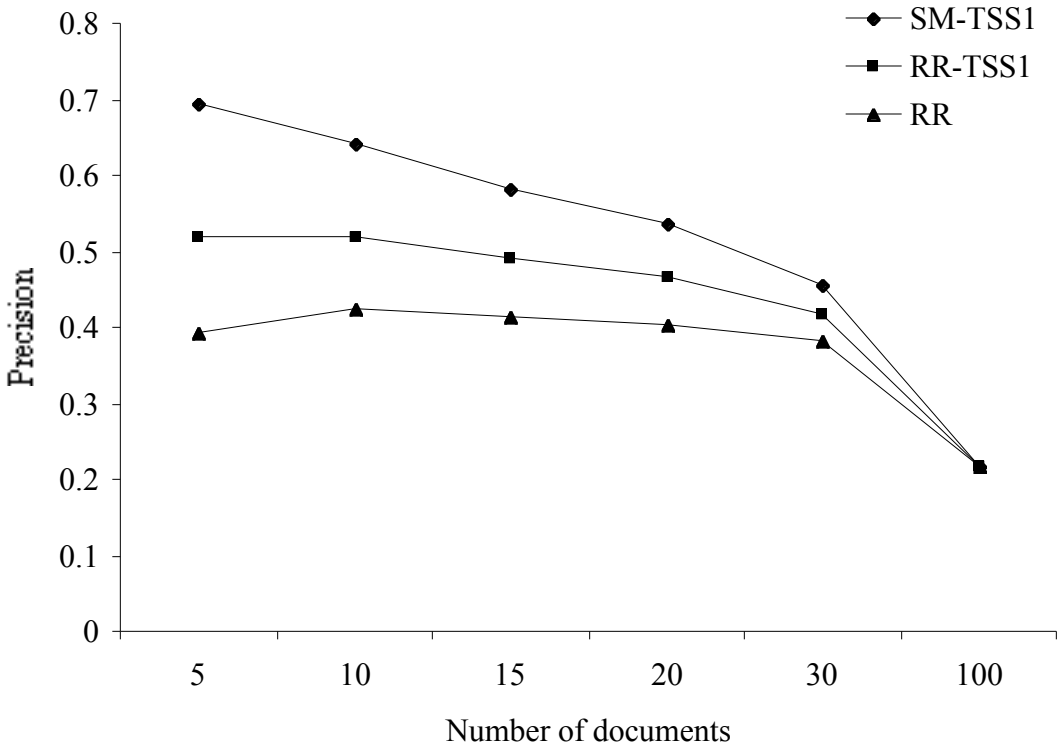

Figure 4: Precision of top results of SM-TSS1, RR-TSS1 and RR

As a second combined approach, we compute a new document score as a linear combination of the document title and article summary scoring as follows:

$$w_{ij} = k \cdot TS_{ij} + (1 - k) \cdot SS_{ij} \qquad (2)$$

where k is a constant between 0 and 1. It is used to give importance to title score (TS) or summary score (SS). After some tuning process by measuring the retrieval performance (average precision) achieved while varying k, we set k = 0.9 which suggests to give higher weight to title score (0.9) than to summary score (0.1). Our tuning shows that setting $0 < k < 1$ improves

performance compared to that obtained with TS (when k is close to 0) or SS (when k is close to 1). But the improvement is more substantial when setting a higher value to k[4]. This improvement is expected as we stated in section 2.1: "in general, the titles of documents returned by current news services are more accurate and reliable. Thus, title field seems to be a beneficial source of evidence for ranking articles". Table 9 depicts the retrieval effectiveness of this second combined merging strategy. As one can see, the resulting performances are close to those achieved by our first merging approach (see Table 8a). When using the Sign test (results not shown in this paper), we obtain the same conclusions than those presented in Table 8b. However, the drawback of this second merging procedure is the need to set the underlying constant k.

Table 9: Linear combination of the title and summary scoring function

Merging strategies	Average Precision	% change	Precision @10	% change	Precision @20	% change
RR (baseline)	48.97		42.52		40.42	
RR-TSS2	56.59	15.56	51.96	22.20	46.50	15.04
SM-TSS2	67.06	36.94	63.08	48.35	53.79	33.08

4.5. Document date

In the previous raw-score merging schemes, ties were broken according to document rank and in favor of the article appearing in the lower rank. Such ties appear when two articles have very similar titles (or summaries) or both titles have the same length and share the same number of keywords with the request. Instead of breaking ties according to the document rank, we suggest breaking ties according to document date and in favor of the article owning a more recent date. Such an approach is already used in Boolean search engines and this scheme seems to be even more appropriate when dealing with current news services. Formally, the date score is calculated as follows:

$$DS = 1000 \times (TodayDate - DocumentDate) \qquad (3)$$

[4] Note that the value of k is not particularly critical. The effectiveness measure changes only gradually as k is varied

where TodayDate was set to the assessment date. In case of a date difference greater than 1,000, SS is set to 0. This breaking scheme can be incorporated in the various raw-score merging strategies described previously, namely the document title (SM-TS-D), the document summary (SM-SS-D) or the combined merging approach (SM-TSS1-D).

Table 10a: Including document date for breaking ties

Merging strategies	Average precision	% change	Precision @10	% change	Precision @20	% change
SM-TS	63.76		61.03		50.56	
SM-TS-D	65.21	2.35%	61.31	0.46%	50.93	0.83%
SM-SS	62.21		60.47		50.00	
SM-SS-D	62.94	1.88%	60.84	0.61%	49.63	-0.74%
SM-TSS1	67.14		64.11		53.83	
SM-TSS1-D	68.01	1.22%	64.11	0.73%	53.41	-0.61%

Table 10b: Sign test results

SM-TS	<	SM-TS-D
SM-SS	<	SM-SS-D
SM-TSS1	<	SM-TSS1-D

Table 10a depicts the retrieval effectiveness of this strategy compared to the scheme based on the document rank. All the results based on the document date seem to be slightly better than those achieved by the scoring function using the document rank computed by the news servers. When using the Sign test as shown in Table 10b, the document date brought some improvement. When studying our second combined approach, we obtained the same result as with our first combined scheme.

4.6. Estimated collection statistics (ECS)

Well-known and effective merging techniques like CORI (Callan *et al.*, 2000) are based on the knowledge of various collection statistics like document frequency. However, in our context, it is impossible to obtain cooperation among news servers in order to obtain the required statistics. Moreover, we cannot download all documents containing a given search term from all servers to get the needed information. As an approximation to obtain an estimate of the required statistics, we use the title and summary fields provided within the list of the top 10 retrieved

documents in order to calculate document frequency. Such information will be used to define a collection score.

With this objective in mind, we adopt the Okapi probabilistic model (Robertson *et al.*, 2000) which will be applied not to rank retrieved documents but to define a collection score for each selected server. Thus, for each incoming query, our broker will consider each search keyword. For each such term and each news server, we compute a weight denoted wt_{ij} for the term t of the collection j as follows:

$$wt_{ij} = \frac{(k_1 + 1) \cdot df_{tj}}{K + df_{tj}} \cdot \log\left(\frac{C}{cf_t}\right) \quad \text{with } K = k_1 \cdot \left((1 - b) + b \cdot l_j / avl\right) \qquad (4)$$

where

- df_{tj} indicates the number of top documents of collection j containing the term t within its title and summary fields;
- cf_t denotes the number of collection returning at least one document containing the term t in its title and summary fields;
- C indicates the number of collections;
- l_j denotes the number of documents returned by the collection j for the current query;
- avl means the average number of documents returned by the collections for the current request;
- k_1 and b are constants and are set respectively to 1.5 and 0.50.

To define a collection score denoted p_j for the jth server, we simply sum over all weights wt_{tj}. However, what is really important is not the absolute value of the collection score p_j but the relative deviation from the mean collection score denoted $\bar{p}$ (computed as the arithmetic mean over all p_j values). Finally, we may exploit this collection score in a new document score. Thus for each document i belonging to the collection j, this new document score value (denoted nw_{ij}) is computed as follows:

$$nw_{ij} = w_{ij} \cdot \left(1 + 0.4 \cdot \left(p_j - \bar{p}\right) / \bar{p}\right) \qquad (5)$$

where w_{ij} indicates the document score computed according to one of our previous document scoring function like document title scoring (SM-TS, see Section 4.2), document summary scoring

(SM-SS, see Section 4.3) or one of our combined document scoring approaches (SM-TSS1 or SM-TSS2, see Section 4.4).

In Table 11a, we evaluate four different merging approaches based on the estimated collections statistics, namely SM-ECS-TS (based on the document title), SM-ECS-SS (using the article summary), and our two combined merging models SM-ECS-TSS1 and SM-ECS-TSS2. The retrieval performance depicted in Table 11a shows that we obtain positive improvement for all measures, and the results of the Sign tests exhibited in Table 11b reveal that these differences are always significant.

Table 11a: Performances using estimated collection statistics

Merging strategies	Average precision	% change	Precision @10	% change	Precision @20	% change
SM-TS	63.76		61.03		50.56	
SM-ECS-TS	67.15	5.32%	63.55	4.13%	52.66	4.15%
SM-SS	62.21		60.47		50.00	
SM-ECS-SS	64.25	3.28%	62.52	3.39%	50.51	1.02%
SM-TSS1	67.14		64.11		53.83	
SM-ECS-TSS1	68.52	2.06%	65.42	2.04%	54.25	0.78%
SM-TSS2	67.08		63.46		53.74	
SM-ECS-TSS2	68.24	1.73%	64.67	1.91%	53.93	0.35%

Table 11b: Sign test results

SM-TS	<	SM-ECS-TS
SM-SS	<	SM-ECS-SS
SM-TSS1	<	SM-ECS-TSS1
SM-TSS2	<	SM-ECS-TSS2

4.7. Server usefulness

In order to improve the merging process, we may take account of the servers retrieval effectiveness or, more precisely, of their retrieval performance compared to the server mean retrieval performance. Our underlying hypothesis is that we may extract more documents from servers presenting a precision better than the mean precision. We thus take account not only of the presence of pertinent items in the corresponding news collection but of the server's capability to extract relevant items and to present them among the top retrieved documents. When having

such a usefulness measure for all servers, we can define a new score for document i belonging to collection j (denoted nw_{ij}) as follows:

$$nw_{ij} = w_{ij} \cdot \left[1 + (0.8 \cdot (u_j - \bar{u})/\bar{u})\right] \qquad (6)$$

where u_j indicates the usefulness of server j, $\bar{u}$ corresponds to the mean server usefulness, and w_{ij} the score of document i of collection j. This suggested weighting expression takes account of the document score (w_{ij}) on the one hand, and on the other, of the server capability, compared to the servers mean precision, to extract and present pertinent documents in the top of the retrieved items. Finally, the weighting constants were chosen by tuning experiments.

As a first approach to measure server usefulness, we may consider the server precision as described in Table 5. To define the document score w_{ij}, we may use the document title (SM-Uprec-TS), the document summary (SM-Uprec-SS), and our combined merging model (SM-Uprec-TSS1). The retrieval performances depicted in Table 12a show that we obtain positive improvement (around +4.5%) for all measures when comparing the raw-score merging based on various document logical sections (SM-TS, SM-SS or SM-TSS1) with their corresponding counterpart using the server usefulness (SM-Uprec-TS, SM-Uprec-SS or SM-Uprec-TSS1).

Table 12a: Performances using server usefulness

Merging strategies	Average precision	% change	Precision @10	% change	Precision @20	% change
SM-TS	63.76		61.03		50.56	
SM-Uprec-TS	67.28	5.52%	63.74	4.44%	54.25	7.30%
SM-Uprec.ntc-TS	65.00	1.94%	61.78	1.23%	50.70	0.28%
SM-Uprec.oka-TS	65.39	2.56%	61.68	1.07%	51.07	1.01%
SM-SS	62.21		60.47		50.00	
SM-Uprec-SS	65.28	4.93%	62.99	4.17%	51.78	3.56%
SM-Uprec.ntc-SS	63.37	1.86%	61.12	1.07%	49.25	-1.50%
SM-Uprec.oka-SS	63.50	2.07%	61.03	0.93%	49.58	-0.84%
SM-TSS1	67.14		64.11		53.83	
SM-Uprec-TSS1	69.83	4.01%	65.51	2.18%	54.95	2.08%
SM-Uprec.ntc-TSS1	68.59	2.16%	65.89	2.78%	53.04	-1.47%
SM-Uprec.oka-TSS1	68.66	2.26%	65.98	2.92%	53.08	-1.39%

In order to apply this approach, we must have a set of queries with their relevance assessments to define the average precision of each server. However such information is not

simple to obtain and requires the definition of the relevance information for all retrieved items. Thus, when not all relevance judgment is available, we may estimate the server usefulness according to Craswell's *et al.* (2000) study. In this case, we will inspect only the first twenty retrieved items to define the relevance assessments. To achieve this, we have downloaded the documents corresponding to the first twenty retrieved items and indexed whole articles using the tf-idf (cosine normalization, denoted CBS-ntc.ntc) on the one hand, and on the other, the Okapi probabilistic model (denoted CBS-oka.bnn, see Section 4.8). Based on this limited relevance information, we may estimate for each server its usefulness. To achieve this objective, we consider the top twenty documents of each results list and the usefulness of server j having two documents among the top twenty will be fixed to 2 / 20 = 0.1. In our evaluations shown in Table 12a, we used the classical tf-idf approach (denoted SM-Uprec.ntc) and the Okapi probabilistic model (SM-Uprec.oka).

Table 12b: Sign test results

SM-TS < SM-Uprec-TS
SM-TS < SM-Uprec.ntc-TS
SM-TS < SM-Uprec.oka-TS
SM-SS < SM-Uprec-SS
SM-SS < SM-Uprec.ntc-SS
SM-SS = SM-Uprec.oka-SS
SM-TSS1 < SM-Uprec-TSS1
SM-TSS1 < SM-Uprec.ntc-TSS1
SM-TSS1 < SM-Uprec.oka-TSS1

In this table, one can see that even with limited relevance information, the retrieval models based on the server usefulness result in better retrieval performance than their corresponding counterpart ignoring server usefulness (e.g., SM-Uprec.ntc-TS vs. SM-TS, SM-Uprec.oka-SS vs. SM-SS or SM-Uprec.ntc-TSS1 vs. SM-TSS1). As described in Table 12b, these difference are always significant with the exception of SM-SS (0.6221) vs. SM-Uprec.oka-SS (0.6350).

Of course, as indicated in Table 12a, having all relevance information to measure the server usefulness presents a better solution (SM-Uprec) than our estimation based on the first twenty retrieved documents (SM-Uprec.ntc or SM-Uprec.oka).

4.8. Content based scoring (CBS)

Instead of using only one logical part of a document to build a document surrogate, we may download the whole article and index it using the SMART system. In this case, we build a single index for each query using all retrieved items and avoid the merging process. To achieve this, we may choose various vector-processing models or a probabilistic approach.

As a first approach, we adopted a binary indexing scheme within which each document or request is represented by a set of keywords without any weight. To measure the similarity between documents and requests, we count the number of common terms, computed according to the inner product (retrieval model denoted CBS-bnn.bnn in which the three code letters (e.g., bnn) indicates the weighting scheme applied to documents and the last three letters the model uses for indexing requests; see Appendix 1 for details).

Binary logical restrictions are often too limiting for document and query indexing. It is not always clear whether or not a document should be indexed by any given term, meaning a simple "yes" nor "no" is insufficient. In order to create something in between, the use of term weighting allows for better term distinction and increases indexing flexibility. As noted previously, the similarity between a document and the request is based on the number of terms they have in common, weighted by the component tf (retrieval model notation: CBS-nnn.nnn).

In a third IR model (Salton, 1989), those terms that do occur very frequently in the collection are not believed to be too helpful in discriminating between relevant and non-relevant items. Thus we might count their frequency in the collection, or more precisely the inverse document frequency (denoted by idf), resulting in a larger weight for sparse words and a smaller weight for more frequent ones. In this case, higher weights are given to terms appearing more often in a document (tf component) and rarely in other articles (idf component). As such, each term does not have an equivalent discrimination power, and a match on a less widely used keyword must therefore be treated as being more valuable than a match on a more common word. Moreover, using a cosine normalization (retrieval model notation: CBS-ntc.ntc), may prove beneficial and each indexing weight may vary within the range of 0 to 1.

Other variants may also be created, especially when considering that the occurrence of a given term in a document is a rare event. Thus, it may be a good practice to give more importance

to the first occurrence of this word as compared to any successive, repeating occurrences. Therefore, the tf component may be computed as the log(tf) + 1.0 (retrieval model notation: CBS-ltc.ltc) or as $0.5 + 0.5 \cdot$ [tf / max tf in a document]. In this latter case, the normalization procedure is obtained by dividing tf by the maximum tf value for any term in the document (retrieval model denoted "atn"). Different weighting formulae may of course be used for documents and requests, leading to other different weighting combinations (e.g., CBS-atn.ntc or CBS-lnc.ltc).

Finally we should consider that a term's presence in a shorter article provides stronger evidence than it does in a longer document. To account for this, we integrate document length within the weighting formula, leading to a more complex IR model; for example, the IR model denoted by CBS-Lnu.ltc (Buckley *et al.*, 1996), CBS-dnu.dtn (Singhal *et al.*, 1999) or the Okapi probabilistic search model (Robertson *et al.*, 2000) denoted CBS-oka.bnn. In these schemes a match on a small document will be treated as more valuable than a match on a longer document. The question that then arises is: How will these retrieval models behave when used with our test collection?

Table 13a: Content based scoring

Merging strategies	Average precision	% change	Precision @10	% change	Precision @20	% change
SM-Uprec-TSS1	69.83		65.51		54.95	
CBS-ntc.ntc	73.76	5.63%	70.47	7.57%	57.80	5.19%
CBS-dnu.dtn	73.15	4.75%	69.72	6.43%	57.90	5.37%
CBS-oka.bnn	73.11	4.70%	69.91	6.72%	57.99	5.53%
CBS-Lnu.ltc	72.57	3.92%	69.53	6.14%	57.80	5.19%
CBS-ltc.ltc	72.36	3.62%	69.81	6.56%	58.55	6.55%
CBS-lnc.ltc	72.15	3.32%	68.79	5.01%	57.48	4.60%
CBS-nnn.nnn	68.92	-1.30%	66.64	1.72%	55.61	1.20%
CBS-atn.ntc	63.81	-8.62%	61.78	-5.69%	53.04	-3.48%
CBS-bnn.bnn	52.51	-24.80%	46.45	-29.09%	42.99	-21.77%

Table 13b: Sign test results

SM-Uprec-TSS1 < CBS-ntc.ntc
SM-Uprec-TSS1 < CBS-dnu.dtn
SM-Uprec-TSS1 < CBS-oka.bnn
SM-Uprec-TSS1 < CBS-Lnu.ltc
SM-Uprec-TSS1 < CBS-ltc.ltc
SM-Uprec-TSS1 < CBS-lnc.ltc
SM-Uprec-TSS1 = CBS-nnn.nnn
SM-Uprec-TSS1 > CBS-atn.ntc
SM-Uprec-TSS1 > CBS-bnn.bnn

The retrieval performances shown in Table 13a indicate that when used with an appropriate weighting models, the effectiveness of this approach having a low efficiency (due to downloading and indexing) is better than the effectiveness of the best merging approach presented so far (SM-Uprec-TSS1) which is based only on available data (title, summary, rank and server usefulness). However, all "CBS-" approaches used only one inverted file and thus we do not need to merge various result lists into a single ranked list of retrieved items.

5. Discussion

From our study, the following observations can be noted:

- In current online news services, the mean percentage of unavailable documents (broken links) is around 4% (see Table 3);
- The average precision of various current online news services varies considerably (mean value of 40.67) and the associated standard deviation is also large (33.96) indicating that there is a large variation among requests (see Table 4 and 5);
- The information available from the servers which could be used in merging / selection is:
  - document rank;
  - document score (sometimes available but these values not comparable across servers);
  - usefulness of the server (calculated according to the retrieval effectiveness of the server);
  - document title (often available but not always present);
  - document summary (rarely available);
  - document date (almost always available);
  - document text (but it requires time to download).

- The raw-score merging strategy presents a better retrieval performance than the round-robin merging approach based on our generic document scoring function when using the document title (+30%, Table 6a), the document summary (+27%, Table 7a) or combining the title and summary of the document (+37%, Table 8a);
- Using the document date when breaking ties (and in favor of more recent articles) presents a small (+2%) but significant improvement in average precision (Tables 10);
- Estimating document frequency based on title and summary fields of the top retrieved items to define a collection weight seem to be enough to obtain better results (+3% improvement on average over the raw-score merging approach, see Table 11a);
- Taking account of the server's usefulness in defining the document score may improve the retrieval effectiveness (around +3%) compared to ranking schemes ignoring this feature (see Table 12);
- Downloading the documents and indexing them with the classical tf-idf vector-space model gives the highest quality results (see Tables 13). However, the main drawback of this strategy is the underlying cost of downloading and indexing the required documents.

These results suggest that current news search is an application in which there is a justifiable case for the use of metasearching techniques and in which it is possible to obtain high quality merged results using only the rudimentary information provided by online news services. Of course, metasearch is not the only way to achieve up-to-date search. Crawler-based search engines can invoke special high-frequency crawling regimes targeted at specific sites to ensure that current news indexes are up-to-date or, even better, enter into commercial partnerships with the news sources in which incoming news is immediately notified to the search engine.

There would be commercial and legal issues in operating the current news metasearcher as a production service. Current news services would no doubt block queries forwarded by a metasearcher if they felt that they were being deprived of advertising revenue derived from visits to their site. On the other hand, the metasearcher might bring additional traffic through greater visibility. In any case, these issues are not specific to metasearch and must also be faced by crawler-based systems.

The experimental metasearcher we have constructed accesses only fifteen of the hundreds of online news sources. To build a production metasearcher would necessitate consideration of

how many more news services to add to the pool. Each addition would require work to build and maintain the "wrapper" script by which queries are forwarded to the new engine and results are extracted. This work would need to be balanced against the additional coverage, recency or quality of news available from the new site.

In this study we have not yet addressed the issue of server selection. Selection could be used to reduce network costs and potentially to improve result quality by forwarding queries only to the subset of news sources likely to provide good answers. If hundreds of news sources were covered by the metasearcher, selection would be essential.

6. Conclusion

Our research shows that high retrieval effectiveness can be achieved in a realistic metasearch application using low cost merging methods, even when the primary servers provide very little information. Our approach merges result lists on the basis of new document scores calculated by the metasearcher. The scores are based on a combination of evidence: information easily extracted from results lists (rank, title, summary and date) and estimated metadata (server usefulness and estimated collection statistics). The proposed approach is simple and efficient as well as effective. We expect that it would be easily implemented in similar contexts, such as a metasearcher combining search engines for scientific articles.

As far as future work is concerned, an obvious next step would be to study server selection in the context of our test collection. Using the technique of dividing the collection into training and test sets in a large number of different ways, the current news test collection may be used to evaluate in a realistic setting, selection methods based on estimated collection statistics.

Acknowledgments

This research was supported, in part, by the SNSF (Swiss National Science Foundation) under grant 21-58 813.99 (Y. Rasolofo & J. Savoy).

References

Buckley, C., Singhal, A., Mitra, M. & Salton, G. (1996). New retrieval approaches using SMART. In Proceedings of TREC'4, (pp. 25-48). Gaithersburg: NIST Publication #500-236.

Callan, J. P., Lu, Z. & Croft, W. B. (1995). Searching distributed collections with inference networks. In Proceeding of the ACM-SIGIR'95, (pp. 21-28). New York: The ACM Press.

Callan, J. P., Connell, M. & A. Du, A. (1999). Automatic discovery of language models for text databases. In Proceedings of ACM-SIGMOD'99, (pp. 479-490). New York: The ACM Press.

Callan, J. P. (2000). Distributed information retrieval. In W. B. Croft (Ed.), *Advances in information retrieval*, (pp. 127-150). New York: Kluwer Academic Publishers.

Craswell, N., Hawking, D. & Thistlewaite, P. (1999). Merging results from isolated search engines. In Proceedings of the 10th Australian Database Conference, (pp. 189-200). New York: Springer-Verlag.

Craswell, N., Bailey, P. & Hawking, D. (2000). Server selection on the World Wide Web. In Proceedings of the ACM-DL'2000, (pp. 37-46). New York: The ACM Press.

Craswell, N., Hawking, D. & Griffiths, K. (2001). Which search engine is best at finding airline site home pages? Technical report, CSIRO Mathematical and Information Sciences, http://www.ted.cmis.csiro.au/~nickc/pubs/airlines.pdf.

French, J. C., Powell, A. L., Viles, C. L., Emmitt, T. & Prey, K. J. (1998). Evaluating database selection techniques: A testbed and experiment. In Proceedings of ACM-SIGIR'98, (pp. 121-129). New York: The ACM Press.

Gravano, L., Garcia-Molina, H. & Tomasic, A. (1994). The effectiveness of GlOSS for the text-database discovery problem. In Proceedings of the ACM-SIGMOD'94, (pp. 126-137). New York: The ACM Press.

Gravano, L., Chang, K., Garcia-Molina, H., Lagoze, C. & Paepcke, A. (1997). STARTS - Stanford Protocol Proposal for Internet Retrieval and Search, http://www-db.stanford.edu/~gravano/starts.html.

Hawking, D. & Thistlewaite, P. (1999). Methods for information server selection. *ACM Transactions on Information Systems*, 17(1), 40-76.

Hawking, D., Craswell, N., Bailey, P. & Griffiths, K. (2001a). Measuring search engine quality. *Information Retrieval*, 4(1), 33-59.

Hawking, D., Craswell, N. & Griffiths, K. (2001b). Which search engine is best at finding online services? In Poster Proceedings of the Tenth International World Wide Web Conference, http://www.ted.cmis.csiro.au/~dave/www10poster.pdf.

Hull, D. (1993). Using statistical testing in the evaluation of retrieval experiments. In Proceedings of the ACM-SIGIR'93, (pp. 329-338). New York: The ACM Press.

Kirsch S. T. (1997). Distributed search patent. U.S. Patent 5,659,732. http://software.infoseek.com/patents/dist_search/patents.htm.

Lawrence, S. & Lee Giles, C. (1998). Inquirus, the NECI meta search engine. Proceedings of WWW'7, 95-105.

Lawrence, S. & Lee Giles, C. (1999). Accessibility of information on the Web. *Nature*, 400, 107-109.

Le Calvé, A. & Savoy, J. (2000). Database merging strategy based on logistic regression. *Information Processing & Management*, 36(3), 341-359.

Nielsen, J. (2000). *Designing Web usability: The practice of simplicity*. Indianapolis: New Riders.

Rasolofo Y., Abbaci F., & Savoy, J. (2001). Distributed information retrieval: Approaches to collection selecting and results merging. In Proceedings ACM-CIKM'2001, (pp. 191-198). New York: The ACM Press.

Robertson, S. E., Walker, S. & Beaulieu, M. (2000). Experimentation as a way of life: Okapi at TREC. *Information Processing & Management*, 36(1), 95-108.

Salton, G. & McGill, M. J. (1983). *Introduction to modern information retrieval.* New-York: McGraw-Hill.

Salton, G. (1989). *Automatic text processing: The transformation, analysis, and retrieval of information by computer*. Reading: Addison-Wesley.

Savoy, J. (1997). Statistical inference in retrieval effectiveness evaluation. *Information Processing & Management*, 33(4), 495-512.

Savoy, J. & Picard, J. (2001). Retrieval effectiveness on the Web. *Information Processing & Management*, 37(4), 543-569.

Schwartz, M., & Task Force on Bias-Free Language (1995). *Guidelines for bias-free writing*. Bloomington: Indiana University Press.

Selberg, E. & Etzioni, O. (1995). Multi-service search and comparison using the Meta-Crawler. In Proceedings of the Fourth International World Wide Web Conference, http://www.w3.org/Conferences/WWW4/Papers/169.

Selberg, E. & Etzioni, O. (1997). The MetaCrawler architecture for resource aggregation on the Web. *IEEE Expert: Intelligent Systems and their Applications*, 12(1), 11-14.

Singhal, A., Choi, J., Hindle, D., Lewis, D. D. & Pereira, F. (1999). AT&T at TREC-7. In Proceedings TREC-7, (pp. 239-251). Gaithersburg, MD: NIST Publication #500-242.

Voorhees, E. M., Gupta, N. K. & Johnson-Laird, B. (1995). Learning collection fusion strategies. In Proceedings of the ACM-SIGIR'95, (pp. 172-179). New York: The ACM Press.

Voorhees, E. M. & Harman, D. (2000). Overview of the sixth text retrieval conference (TREC-6). *Information Processing & Management*, 36(1), 3-35.

Voorhees, E.M. (2001). Evaluation by highly relevant documents. In Proceedings of the ACM-SIGIR'2001, (pp. 74-82). New York: The ACM Press.

Appendix 1. Weighting schemes

To assign an indexing weight w_{ij} that reflects the importance of each single-term j in a document i, we may take three different factors into account. They are represented by the following three code letters respectively:

- within-document term frequency, denoted by tf_{ij} (first letter);
- collection-wide term frequency, denoted by df_j (second letter);
- normalization scheme (third letter).

bnn	$w_{ij} = 1$	nnn	$w_{ij} = tf_{ij}$
ltn	$w_{ij} = (\log(tf_{ij}) + 1) \cdot idf_j$	atn	$w_{ij} = idf_j \cdot [0.5 + 0.5 \cdot tf_{ij} / \max tf_{i.}]$
oka	$w_{ij} = \left((k_1 + 1) \cdot tf_{ij}\right) \Big/ \left(K + tf_{ij}\right)$	dtn	$w_{ij} = \left(\log\left(\log(tf_{ij}) + 1\right) + 1\right) \cdot idf_j$
lnc	$w_{ij} = \dfrac{\log(tf_{ij}) + 1}{\sqrt{\sum_{k=1}^{t} \left(\left(\log(tf_{ik}) + 1\right)\right)^2}}$	ntc	$w_{ij} = \dfrac{tf_{ij} \cdot idf_j}{\sqrt{\sum_{k=1}^{t} (tf_{ik} \cdot idf_k)^2}}$
ltc	$w_{ij} = \left(\log(tf_{ij}) + 1\right) \cdot idf_j \Big/ \sqrt{\sum_{k=1}^{t} \left(\left(\log(tf_{ik}) + 1\right) \cdot idf_k\right)^2}$		
dtu	$w_{ij} = \dfrac{\left(1 + \log\left(1 + \log(tf_{ij})\right)\right) \cdot idf_j \Big/ \left(1 + pivot\right)}{(1 - slope) \cdot pivot + slope \cdot nt_i}$		
Lnu	$w_{ij} = \dfrac{\left(1 + \ln(tf_{ij})\right) \Big/ \left(1 + pivot\right)}{(1 - slope) \cdot pivot + slope \cdot nt_i}$		

Table A.1: Weighting schemes

In Table A.1, document length (the number of indexing terms) of document i is denoted by nt_i, the constant advl is set at 900, the constant b at 0.75, the constant k_1 at 2, the constant pivot at 125 and the constant slope at 0.1. For the Okapi weighting scheme, K represents the ratio

between the length of document i measured by l_i (sum of tf_{ij}) and the collection mean noted by advl.

Appendix 2. Notations used in this paper

Symbol	Meaning
	Document scoring
CBS	Content based scoring
SS	summary score using eq. 1
TS	title score using eq. 1
TSS1	score obtained by combining title score and summary score (heuristic)
TSS2	score obtained by combining title score and summary score (linear combination eq. 2)
	Estimated collection statistics
ECS	Estimated collection statistics (eq. 4 and 5)
	Server usefulness
Uprec	Server usefulness estimated using average precision (Table 5)
Uprec.ntc	Server usefulness based on top 20 ntc (see appendix 1) ranking
Uprec.oka	Server usefulness based on top 20 OKAPI (see appendix 1) ranking
	Basic merging strategies
RR	round robin merging based on the original ranked lists given by the news servers
SM	raw score merging
	Merging strategies using only eq. 1
RR-SS	round-robin merging scheme based on the rank obtained by the document sumary scoring
RR-TS	round-robin merging scheme based on the rank obtained by the document title scoring
SM-SS	raw-score merging approach based on the summary scoring
SM-TS	raw-score merging approach based on document title scoring
SM-SS-D	SM-SS using date score (eq. 3) to break tie
SM-TS-D	SM-TS using date score (eq. 3) to break tie
	Merging strategies combining TS and SS
RR-TSS1	round-robin merging based on rank obtained by using TSS1
RR-TSS2	round-robin merging based on rank obtained by using TSS2
SM-TSS1	raw-score merging based on score obtained by using TSS1
SM-TSS1-D	SM-TSS1 using date score (eq. 3) to break tie
SM-TSS2	raw-score merging based on score obtained by using TSS2
	Combined merging strategies using ECS
SM-ECS-SS	raw-score merging based on scores obtained by combining ECS and TS
SM-ECS-TS	raw-score merging based on scores obtained by combining ECS and TS
SM-ECS-TSS1	raw-score merging based on scores obtained by combining ECS and TSS1
SM-ECS-TSS2	raw-score merging based on scores obtained by combining ECS and TSS2
	Combined merging strategies using server usefulness
SM-Uprec-SS	raw-score merging based on scores obtained by combining Uprec and SS
SM-Uprec-TS	raw-score merging based on scores obtained by combining Uprec and TS
SM-Uprec-TSS1	raw-score merging based on scores obtained by combining Uprec and TSS1
SM-Uprec.ntc-SS	raw-score merging based on scores obtained by combining Uprec.ntc and SS
SM-Uprec.ntc-TS	raw-score merging based on scores obtained by combining Uprec.ntc and TS
SM-Uprec.ntc-TSS1	raw-score merging based on scores obtained by combining Uprec.ntc and TSS1
SM-Uprec.oka-SS	raw-score merging based on scores obtained by combining Uprec.oka and SS
SM-Uprec.oka-TS	raw-score merging based on scores obtained by combining Uprec.oka and TS
SM-Uprec.oka-TSS1	raw-score merging based on scores obtained by combining Uprec.oka and TSS1