Server Selection Methods in Hybrid Portal Search

David Hawking
CSIRO ICT Centre
Canberra, Australia
david.hawking@acm.org

Paul Thomas
Department of Computer Science
Australian National University
paul.thomas@anu.edu.au

ABSTRACT

The TREC .GOV collection makes a valuable web testbed for distributed information retrieval methods because it is naturally partitioned and includes 725 web-oriented queries with judged answers. It can usefully model aspects of government and large corporate portals. Analysis of the .gov data shows that a purely distributed approach would not be feasible for providing search on a .gov portal because of the large number (17,000+) of web sites and the high proportion that do not provide a search interface. An alternative hybrid approach, combining both distributed and centralized techniques, is proposed and server selection methods are evaluated within this framework using web-oriented evaluation methodology. A number of well-known algorithms are compared against representatives (highest anchor ranked page (HARP) and anchor weighted sum (AWSUM)) of a family of new selection methods which use link anchortext extracted from an auxiliary crawl to provide descriptions of sites which are not themselves crawled. Of the previously published methods, ReDDE substantially outperformed three variants of CORI and also outperformed a method based on Kullback-Leibler Divergence (extended) except on topic distillation. HARP and AWSUM performed best overall but were outperformed on the topic distillation task by extended KL Divergence.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*selection process*; H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*performance evaluation (efficiency and effectiveness); distributed systems*

General Terms

Experimentation, Measurement, Performance

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.
SIGIR'05, August 15–19, 2005, Salvador, Brazil.
Copyright 2005 ACM 1-59593-034-5/05/0008 ...$5.00.

1. INTRODUCTION

Distributed information retrieval (DIR) systems, also known as metasearchers, have long been proposed as an alternative to large centralized indexes. They have been claimed to offer potential solutions to problems associated with enormous data scale, with rapidly changing data, and with the need to provide integrated search over multiple sources including those subject to access controls.

When the number of primary search services is large or when there is a cost associated with querying servers, a process of query-specific *server selection* is needed to maintain rapid query response and to avoid unnecessary network or search charges. A number of algorithms for server selection have been presented, including CORI [2], ReDDE [20], KL [21], and the GlOSS family [11]. In general, the performance of these techniques has been evaluated using TREC Ad Hoc resources[1].

The availability of extensive relevance judgments for the TREC Ad Hoc corpus has made it the basis for most distributed information retrieval (DIR) testbeds. However, arbitrary partitionings of a small collection of newspaper and government documents don't seem to model likely real-world applications of DIR methods. For example, the Ad Hoc collections lack the link structure, URLs, anchortext and click-through data which underpin successful Web[2] search engines.

In contrast, the .GOV and .GOV2 corpora used in recent TREC Web and Terabyte Track evaluations model a web environment in which an effective distributed search system might offer advantages. A scheme in which government agencies (or each government web site) indexed their own data and a government metasearcher provided whole-of-government search could potentially replace the current centralized service at the US government portal `http://www.firstgov.gov`. Such an alternative might lead to greater coverage, reduced network traffic and more rapid response to web site updates, as well as offering the potential for unified search access to a subset of resources defined by the searcher's access rights.

A further advantage of .GOV over other TREC collections in web applications is the availability of queries and judgments for search modes more characteristic of web search: homepage finding, named page finding, and topic distillation.

Searching the US government domain is a specific instance

[1]See `http://trec.nist.gov/`.
[2]We follow the convention of capitalizing Web when referring to the World Wide Web as opposed to generic webs.

Domain	Size	Domain	Size
.gov	33.0M	.gc.ca	3.8M
.gov.cn	7.6M	.gouv.fr	1.3M
.gov.uk	6.4M	.govt.nz	1.0M
.go.jp	5.5M	.dep.no	1.0M
.gov.au	4.5M	.nic.in	0.7M

Table 1: Sizes in pages of a selection of national government domains as estimated by Google (www.google.com) on 17 Jan 2005. The TREC .GOV2 corpus contains 25M pages; .GOV contains 1.25M.

of a general class of portal search applications. Apart from the possibility of many similar government portals around the world (see Table 1), there are many large-enterprise portals and many multi-source topic portals covering domains such as health, chemistry and travel. The attractiveness of distributed methods for building portals is greater when the sites to be included are widely distributed and where network bandwidth is limited or expensive.

In what follows, we will focus on the problem of searching the .gov domain and use the .gov related resources from TREC. Our experiments are conducted on the .GOV corpus because of the web-specific queries and judgments available for it, but we use the much deeper crawl represented by .GOV2 as a reference for characteristics of the full .gov domain.

We address a number of questions: First, is there a feasible model for DIR in the .gov domain? Second, how well do existing algorithms for server selection work in this domain? Third, can we devise methods which work better?

2. RELATED WORK

Many individual server selection algorithms have been proposed in the literature and evaluated using testbeds derived by partitioning TREC Ad Hoc corpora — generally into one or two hundred collections, and with an eye to producing collections of approximately equal size. Examples include [20, 11, 25]. CORI (Callan et al. [2]) is probably the best known. Nottelmann and Fuhr [16] have recently extended it with an estimation technique based on a general cost function, able to take account of retrieval cost and time as well as estimated relevance.

Similarly, Ipeirotis and Gravano [14] extended the language modelling techniques used by CORI, bGlOSS, and KL Divergence, and were able to show significant improvements in recall when tested with TREC Ad Hoc queries. Neither group were able to make use of web-style queries and judgements.

Past work has often used long queries; for example, Xu and Croft [25] report a series of experiments with a mean 34.5 terms per query. This is evidently much larger than the 2.35 words reportedly typical of queries to web search services. [22]

In a series of papers [10, 9, 17], French, Powell *et al.* have developed a set of testbeds based on the TREC Ad Hoc corpus and used it to evaluate CVV, gGlOSS, and CORI for server selection. The testbeds developed are based on data available at TREC-4 in 1995, and include SYM-236 (a division into 236 collections, arranged by source and publication date); UBC-100 (100 collections, arranged to have approximately equal byte counts); and UDC-236 (236 collections, arranged to have approximately equal numbers of documents). Other testbeds included up to 921 databases, but again arranged to keep size approximately uniform. Queries in all these testbeds were based on TREC Ad Hoc topics and divided into "short" and "long" forms, with mean lengths of 3.5 and 21 terms respectively, and in one investigation [9] the queries were first fed through an automatic expansion process.

Relatively little work has been done to confirm the applicability to web environments of server selection results obtained on TREC Ad Hoc data.

Rasolofo *et al.* [18] compared CORI and their own method for server selection and for results merging using an eight-way division of the TREC WT10g collection of web data. The authors expressly considered the nature of web queries and used either a very short form of the TREC topics (two terms on average) or queries garnered from logs of the Excite search engine (2.4 terms on average).

In [23], Singhal and Kaszkiel evaluated the performance of an algorithm optimized for the TREC Ad Hoc task in a web environment. Their study does not consider server selection, but the differences between TREC Ad Hoc data and the Web are acknowledged and the authors use an 18 million page crawl of the Web as a testbed. Using a web-specific search task, they found ample room for improvement in the document rankings produced by the TREC Ad Hoc algorithm, which suggests that published server selection algorithms may benefit from similar scrutiny.

Craswell *et al.* [5] evaluated CORI, vGlOSS, and CVV in a testbed based on the 2GB, 956 server WT2g crawl of the Web. They concluded that CORI, and a modified version of the CORI algorithm, performed reasonably effectively at the server selection task. This is similar to the present work, but with some differences: the testbed in [5] featured many medium-sized servers and few small ones (the reverse of the case in the web generally) and retrieval tasks typical of ad hoc rather than web search. They also considered every server to be searchable, which we do not (see Section 3.3).

3. .GOV AND .GOV2 TEST COLLECTIONS

The .GOV2 corpus is believed to be a good snapshot of the .gov domain. Its page count of 25M is of similar magnitude to the Google estimate (Table 1) and is within 10% of the Yahoo! estimate of 27.4M obtained on 17 Jan 2005.

Unfortunately, at the time of writing, there were no judgments for web-style queries against the .GOV2 corpus. Accordingly we used the .GOV corpus after comparing its properties against those of .GOV2. Table 1 shows that the .GOV corpus is representative of the size of a number of national government domains.

The 18GB .GOV corpus includes 1.25 million documents, drawn from 7792 servers in an early 2002 crawl of the .gov domain conducted by NIST[3].

3.1 Distribution of documents across servers

The approximately equal size distribution in many previous experiments is not typical of the .gov domain or of the Web, where documents are typically distributed between servers according to a power law (as described by eg. Huberman and Adamic [13]). For some search applications, such as the .gov portal considered here, past partitionings

[3]See http://es.csiro.au/TRECWeb/govinfo.html

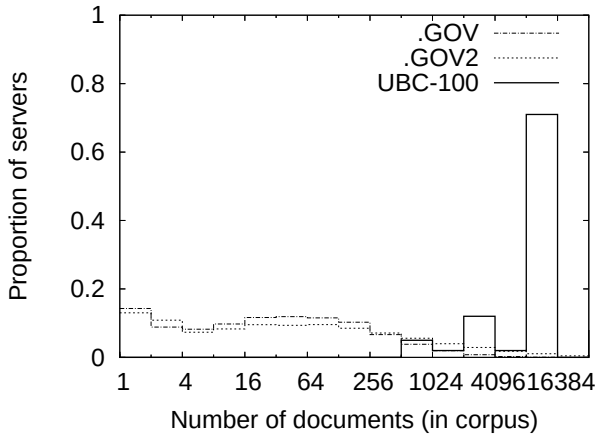

Figure 1: Distribution of documents amongst servers in the .GOV, .GOV2, and UBC-100 corpora.

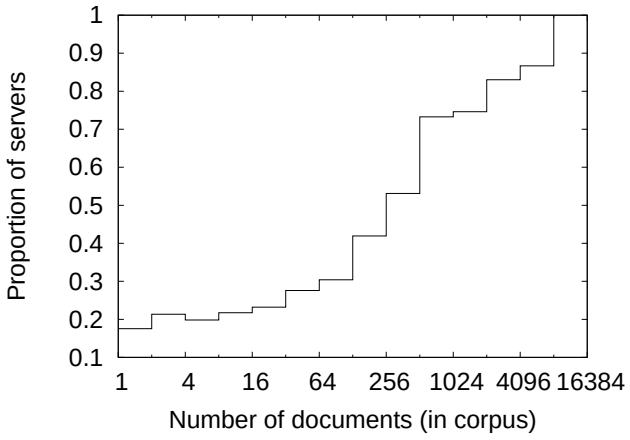

Figure 2: Proportion of servers in the .GOV corpus which have a search interface.

Testbed	Size (GB)	Coll'ns	Avg docs / coll'n	Avg MB / coll'n
UBC-100	3	100	10,800	32
SYM-236	3	236	2,900	11
UDC-236	3	236	2,900	11
kmeans	2	100	5,700	20
10col	3	10	107,800	320
.GOV	18	7792	200	3

Table 2: Characteristics of some popular testbeds, and the .GOV corpus.

also produce too few collections by an order of magnitude.

The .GOV corpus contains documents from 7792 servers, compared with 17,186 for .GOV2. Figure 1 summarizes the distribution of documents across servers, for these collections and for the commonly-used UBC-100 TREC Ad Hoc testbed. There are differences between the .GOV and .GOV2 distributions (presumably due to incomplete crawling of many sites in .GOV) but the difference between these collections and UBC-100 is overwhelming.

Summary statistics for distribution of documents in the .GOV corpus, and several other testbeds used in the literature, are given in Table 2.

3.2 Queries and relevance judgments

Queries and relevance judgments available for the .GOV corpus are of three kinds, which are considered relevant to web searches generally [8].

In the first, "topic distillation", a broad query is interpreted as a request for key homepages. For example, relevant documents for the topic "science" may include the homepages of government science agencies. The same topic would be interpreted in the context of the TREC Ad Hoc judgments as meaning "find me all the pages you can about science", which would return many more relevant documents, each counted as having equal value. Topic distillation is considered more like a typical web search task since web users tend to browse from an entry page to find information they need (see, for example, Teevan *et al.* [24]).

The second type of search task considered, "homepage finding", and third, "named page finding", are similar. In both cases a query is interpreted as a request for a single web page, which may or may not be known to exist. Homepage queries are taken to be explicit requests for a homepage. Named page queries are for a page which is known to exist, but which may not always be a homepage. Both types of query have exactly one relevant document, the page in question or duplicates of it. For this reason, standard evaluation measures for the Ad Hoc task, such as precision and recall at n, are not useful.

3.3 Search interfaces

Not all servers in the .gov domain provide a public search interface; evidently, if a search interface is not available for a site then that site cannot be selected for metasearch.

To discover which servers in the .GOV corpus provide a search facility, we implemented the classifier described in Cope *et al.* [4]. Given the HTML for a form, this classifier can determine, fairly reliably, whether the form is an interface to a search engine. We first extracted all the server homepages from the corpus: the homepage for server `x.gov` is either the page at `http://x.gov/`, or the page at one of eighteen alternatives such as `http://x.gov/index.html`, `http://x.gov/home.htm` or `http://x.gov/start.asp`. We found homepages for 6294 servers. Each form on each homepage was then passed to the classifier, and if a form appeared to be a search interface the server was marked as searchable.

A number of servers do not make a search facility available on the homepage, but instead link to a separate search page. As a second phase, we also examined the HTML of those homepages with no apparent search interface for links to pages with "search" in the name, or links with "search" in the anchor text, and recorded those servers as searchable.

Those servers with no search interface on the homepage and no search link from the homepage were marked as non-searchable.

Overall, 31% of servers appear searchable, but there is a great deal of variability. Searchability in the .GOV corpus seems to correlate well with server size, as illustrated in Figure 2: while only 20% of servers with between four and seven documents in the corpus provide a search interface, 75% of those with between 1024 and 2047 documents provide

an interface as do all of those with 8192 documents or more. Since larger collections are more likely to be searchable, 65% of documents in the corpus belong to searchable servers.

4. SEARCHING THE .GOV DOMAIN

In this section, we consider four possible models for searching the .gov domain, and outline possible implications for quality, speed, and cost of the search. As extreme cases, we consider a central index and "pure" metasearch (the terminology in this section is from Craswell *et al.* [6]); we also consider "selective" metasearch. Finally, we present a hybrid model which we believe represents the most plausible alternative.

4.1 Central index

An obvious way to search is to crawl all of the .gov domain ahead of time, building a local index, and use this index to respond to queries. This is what most current web search engines do, including the service provided on the FirstGov portal. However, the cost of crawling is significant, local optimizations may be lost, and crawling may result in a slow response to site changes. If crawling is carried out frequently to maintain freshness, significant load may be imposed on the web servers.

4.1.1 Cost of crawling

A major problem faced by a crawling server is the network traffic generated by a large-scale crawl. Since some overhead is incurred in crawling (in HTTP headers, duplicates eliminated, etc), the traffic generated will exceed the total size of real data in .gov. Craswell *et al.* suggest an overhead of 70%[4], which means we might reasonably expect each full crawl resulting in 426GB of text (as in the case of .GOV2) to require as much as 724GB network traffic.

Incremental crawling would reduce network load considerably but the traffic generated by frequent incremental crawls is still substantial. A number of other methods are available for reducing traffic still further when cooperation and trust exists between website operators and the search engine. For example, on request, a local site could transmit a compressed form of changed pages only. Trust is more likely in the .gov domain than on the web at large.

4.1.2 Local optimizations

Since individual servers can be expected to know something about their documents, they can offer appropriate local optimizations. For example, they may translate terms ("law" to "act" for a legislation search, or "exhaust" to "emissions" for a search of environmental information) or rank results according to frequency of use. This local knowledge is lost if the site is searched via a crawl.

4.2 Pure metasearch

In the pure metasearch model, each query is submitted to each .gov server, and the results are merged. This presents significant difficulties to do with scale and availability.

4.2.1 Network bandwidth, availability, and response time

An obvious problem is the cost, in network traffic, of forwarding a user's query to each server and receiving the replies. Following [6], the network traffic generated is $(S_q + S_r)(|C|+1)$, where S_q and S_r are the size of each query and response respectively and $|C|$ is the number of servers (collections). (The extra query-and-response is the traffic from and to the user.) Using figures of 1KB for S_q and 20KB for S_r, with the 7792 servers of the .GOV corpus the cost per query is 160MB. This traffic could be reduced by reducing the number of results fetched from each server [15] but the approach is still infeasible for even the best-connected hosts.

Queries can only be forwarded after their receipt by the metasearch agent, and the final results can be returned to the user only after every search engine has returned its result list (or been timed out, with consequent loss of effectiveness). This means the user is likely to experience significant delays, and the system as a whole will often be faced with unavailability of one or more servers.

4.2.2 Interface wrappers

Since we cannot in general assume that each server provides an identical search interface, we will need to provide a wrapper for each interface — potentially each server — which converts the user's query, feeds it to the search interface, and extracts search results. Generating and maintaining these wrappers for a large number of servers represents a good deal of work, which could be dramatically reduced by adoption of standards.

4.2.3 Availability of search interfaces

Queries can only be forwarded to servers which make a search service available. As noted above, this is only 31% of servers in the case of .GOV. Without change of practice by the web servers, pure metasearch is currently not feasible.

4.3 Selective metasearch

A well-studied, although little-used, alternative to pure metasearch is a more selective model. In this model, a searcher forwards a query to a subset of the available servers, chosen by a server selection mechanism such as CORI. The central searcher must maintain adequate data to inform the selection, but in exchange can dramatically reduce the time and traffic needed to respond to a query. For example, if twenty servers are chosen instead of all 7792 in the .GOV corpus, the network traffic needed to reply to a query is reduced from 160MB to 440KB.

This is the model generally assumed in studies of server selection. However, as it stands it is not appropriate for searching the .gov domain: to work as required, every server must be searchable, and an appropriate wrapper must be maintained for every server. We suggest incorporating selective metasearch into a hybrid model, described below, which allows for this.

4.4 Hybrid models

Both pure and selective metasearch seem to be impractical for searching the .gov domain as it stands. As an alternative, we consider a hybrid system of the type suggested by [6], whereby some or all of the servers with a search interface are considered candidates for metasearch and the remainder of servers are crawled. To answer a user's query, a number of searchable servers are selected (as in selective metasearch) for search along with the local index. Since the largest servers in .gov are the most likely to provide a search interface, the volume of traffic for a crawl can be substan-

[4]We suspect that this may be an under-estimate.

tially reduced; freshness of the index is also less of a concern.

There are a number of ways to choose from the .GOV corpus which servers are candidates for metasearch. In the simplest baseline case, we can consider all and only those servers which seem to have a search interface according to our classifier. Alternatively, we can assume that by policy servers over a certain size will be searchable, and are candidates. The chosen size determines the cost of a crawl and of a query in our hybrid system; it also determines the size of index and (in conjunction with a decision on crawl frequency) the possible staleness in our results.

In our experiments we assumed that the crawled servers would be indexed collectively and treated by the metasearcher as a single server.

5. SERVER SELECTION METHODS

We consider ten methods for server selection. Modified KL Divergence, CORI (three variants), and ReDDE represent state-of-the-art server selection algorithms, which have been well tested in other scenarios. We also introduce two ranking methods (HARP and AWSUM, see Section 5.4) suitable for the hybrid model discussed for .gov. Finally, random ranking, size-based ranking, and crawl-first ranking are presented as controls.

5.1 Extended KL Divergence

Kullback-Leibler Divergence was suggested by Xu and Croft [25], and interpreted in a language modelling context by Si *et al.* [21]. We find the servers C_i with the highest probability P given query Q, where:

$$P(C_i|Q) = \frac{P(Q|C_i)P(C_i)}{P(Q)}$$

$P(Q)$ depends only on the query, not the server, so can be ignored. $P(Q|C_i)$ can be estimated with a unigram language model as

$$P(Q|C_i) = \prod_{q \in Q} (\lambda P(q|C_i) + (1-\lambda)P(q|G))$$

where G is the global language model and $\lambda = 0.5$. In the present work we consider an extension described in [19], and assign $P(C_i)$ according to estimated server sizes:

$$P(C_i) = \frac{|\hat{C_i}|}{\sum_j |\hat{C_j}|}$$

5.2 CORI and extensions

The CORI algorithm [2] is an adaptation of the INQUERY document ranking algorithm, treating each server as a compound "document" and using document frequency *df* instead of term frequency.

The score p for a server C_i, for each query term t is given by $p = b + (1-b)TI$, where:

$$I = \frac{\log|C| + 0.5}{|C| + 1};\ T = \frac{df_t}{df_t + tf\_base + tf\_factor \frac{cw}{avcw}}$$

Here df_t is the number of documents in C_i containing the query term, cw is the number of words in C_i, and $avcw$ is the mean cw across all servers. The other terms are constants: $b = 0.4$, $tf\_base = 50$, and $tf\_factor = 150$.

We also consider two extensions to CORI suggested by [19], which make use of the documents C_{isamp} sampled from C_i in the course of building the language model. In the first, CORI_ext1, we scale *cw* and *df* by $\frac{|\hat{C_i}|}{|C_{isamp}|}$. In the second, CORI_ext2, we also scale tf_base and tf_factor. Since these variants expressly adjust for estimated server size, it may be expected that they would perform better over the large variety of server sizes in the .GOV corpus.

5.3 ReDDE

Si and Callan's ReDDE algorithm [20] estimates the distribution of relevant documents between servers by reference to sampled documents. The number of relevant documents in each server is estimated with

$$\hat{\text{rel}_q}(i) = \sum_{d_j \in C_i} P(\text{rel}|d_j) \frac{|\hat{C_i}|}{|C_{isamp}|}$$

$P(\text{rel}|d_j)$ is in turn estimated by ranking the sampled documents with relation to the query terms, and scoring a small constant for each server each time one of its documents appears in the ranked list. The list is cut off at the point where the accumulated server sizes, $\sum_{d_j \in \text{ranked}} \frac{|\hat{C_{d_j}}|}{|C_{d_j samp}|}$, exceeds a constant.

5.4 Anchortext methods

Because the portal applications considered in this paper are wholly in the web domain, it is possible to make use of web-specific information, such as link anchortext, in server selection and to rely on the fact that web document names (URLs) include the full hostname of the web server which published the document.

In a fully crawled approach to portal search, anchortext can be used to identify the most valuable answers to queries and is known to contribute heavily to retrieval effectiveness in web-style evaluations [7]. We hypothesized that anchortext derived from something less than a full crawl might also be useful in server selection, given that the anchortext of links to a web page allows us to know something about that page (and the web server which published it) without accessing its content.

We propose a family of web server selection methods based on anchortext extracted from an auxiliary crawl of web sites related in some way to the content of the portal. In the context of the hybrid approach explored in this paper, the obvious auxiliary crawl to use is that of the servers without search interfaces, since these sites have to be crawled anyway in order to provide search of their content. To reduce potential bias, we have excluded nepotistic (within-server) links as these are only available for sites in the central crawled collection.

We report results for two methods from this family using anchortext derived from the central crawled collection. It would be interesting to apply the same methods to different auxiliary collections such as a shallow crawl over all the sites in the portal or indeed a full but long out-of-date crawl of those sites, but this remains for future work. The following descriptions of the methods are slightly simplified for clarity.

5.4.1 HARP (Highest Anchor Ranked Page)

The anchor text for all the non-nepotistic, within-domain links in the central crawled collection is formed into surrogate documents according to the target URL of the link. Some of the targets correspond to real documents in the central crawled collection, others to documents on other servers

and a proportion to documents which do not currently exist. Surrogates for documents with many incoming links will in general be much longer than surrogates for those with few incoming links. Note that document content and other metadata is completely ignored.

The collection of surrogate documents is indexed and queries are processed against the index using the AF1 scoring formula [12] which includes no length normalization and in which the contribution of high term frequencies (tf_d) does not flatten off as is appropriate for normal text documents.

$$w_t = \alpha \log(tf_d + 1) \times \log\left(\frac{|C_i| - n + 0.5}{n + 0.5}\right) \qquad \text{(AF1)}$$

where w_t is the weight contribution of (Porter stemmed) query term t and n is the number of documents whose incoming anchortext contains t. Term coordination is imposed — fully matching documents are always ranked ahead of partial matches. Since only anchor text is used, the value of α doesn't affect the ranking.

A ranking of up to 1000 web page results is returned and converted to a ranking of servers using simple syntactic processing of the URLs and lookups of a list of search server hostnames. Example: if a the URL of a page in the results is `http://ks.water.usgs.gov/Kansas/pubs/fact-sheets/fs.022-98.html` and `ks.water.usgs.gov` is in the the list of search servers, then `ks.water.usgs.gov` will be added to the tail of the ranked list of servers unless it is already higher in the list. If the server (from .gov) does not appear in the list of search interfaces, then the central crawled collection will be appended to the server list, if not already present.

In this way, search servers (and the central crawled collection) are ranked by the score of their highest ranked page.

Note that the central crawled collection is added to the tail of the list if not already present, since the cost of searching it is low. This approach also avoids returning an empty list.

5.4.2 AWSUM (Anchor Weighted Sum)

This is a variant of HARP in which each page in the original ranking contributes to the score of its server. In order to prevent a server with many low-value pages overwhelming another with fewer but higher-value answers, each page's contribution is its score divided by its rank. Servers are ranked by descending score, with the central crawled collection at the tail of the list if not otherwise present.

5.5 Other methods

For comparison baselines, we also consider three simple server selection methods: ranking servers randomly ("random"), ranking servers by their estimated size ("size"), and always selecting the central crawled server first and others randomly ("crawled").

5.6 Validation of implementations

When comparing new methods with those previously published, it is often difficult to be sure that the older methods have been correctly implemented. We have endeavoured to do so by evaluating our implementations of CORI and extensions, extended KL and ReDDE on a testbed for which results have previously been published. We used the UBC-100 collection and TREC Ad Hoc queries 51–100, and compared R_n scores at three key points with those read from the graphs in Figure 1 of [19]. The values we obtained were often higher than the read-off values and were always within the margin of difference to be expected given uncontrolled differences in indexing, stemming etc.

6. EXPERIMENTS

We compared the ten methods using 725 queries and related relevance judgments from the TREC Web Track 2002–2004. 125 of these were topic distillation queries, 375 homepage finding queries, and 225 named page queries.

6.1 Estimating language models and server size

Several of the methods presented need an estimate of each server's size, and the CORI family and extended KL need an estimate of each server's language model.

To estimate language models, we used single-term probe queries in the manner of [3] until we had processed 300 unique documents or 150 queries. The first query term was selected at random from a list of common English words, then terms were chosen at random from the learned language model (or from the list of common words if the model was exhausted). The learned models were filtered with the SMART stoplist.

Server sizes were estimated using the sample-resample method described in [20], choosing query terms at random from the estimated language model and using five queries.

We used full language model and accurate size for the central crawled sever as this information would be available in the modeled setting.

6.2 Measures

Modified average precision (MAP), a standard measure from the TREC Web Track, was used to compare the server selection methods. MAP is defined by

$$\text{MAP} = \frac{\sum_{i=1}^{\text{num\_rel\_ret}} i/\text{rank}(i)}{R}$$

Where num_rel_ret is the number of relevant servers in the ranking; rank(i) is the rank of the ith relevant server; and R is the total number of relevant servers, or the requested length of the ranked list if this is smaller. In each experiment we ranked up to 100 servers.

In the case where there is only one relevant server (all named page and homepage queries, and some of the topic distillation queries), MAP is identical to mean reciprocal rank (MRR).

7. RESULTS AND DISCUSSION

Table 3 shows results obtained for the most natural hybrid division — rely on local search interfaces where they exist and crawl the rest, treating the crawl as a single "server".[5]

The number of servers (1972) in this partitioning is so large that *random* server selection scores very close to zero. Because the central crawled server is by far the largest, *crawled* and *size* methods always agree in their first choice server. After that, choosing servers on the basis of size rather than randomly gives only small benefit. Given the homepage-dominated, tiny-relevant-set evaluation we used, it may be that the reason the central crawled server is so valuable may be more due to the fact that it covers a large

[5]To enable replication of this work by others, the authors are happy to provide on request their list of servers providing search interfaces.

	Home-page (375)	Named page (225)	Topic dist. (125)	Mean (725)
AWSUM	0.6710	**0.5417**	0.3227	**0.5708**
HARP	**0.6964**	0.4984	0.3103	0.5684
ReDDE	0.5203	0.5187	0.3131	0.4841
KL_ext1	0.5094	0.3354	**0.3366**	0.4256
CORI_ext1	0.4462	0.2956	0.2357	0.3632
CORI	0.4241	0.3163	0.2331	0.3577
Size	0.3298	0.3599	0.1873	0.3146
Crawled	0.3153	0.3554	0.1669	0.3022
CORI_ext2	0.2909	0.2190	0.1837	0.2501
Random	0.0050	0.0007	0.0054	0.0037

Table 3: MAP@100 results, for the hybrid model metasearching all .GOV servers known to provide a search interface plus a central crawl of the rest.

number of sites rather than the fact that it includes a large number of pages. Here, selection methods are not rewarded for identifying servers with large numbers of documents which match the query but only for choosing servers holding one of a small set of answers.

The differences between the performance of *CORI* and *size* are not significant at the 0.05 level (Wilcoxon signed ranks text). *CORI* is solidly outperformed overall by *extended KL* (Wilcoxon $p < 0.01$, except for the named page task where the difference was not significant) which is itself outperformed by ReDDE (Wilcoxon $p < 0.05$), although the advantage on the topic distillation task is reversed (Wilcoxon $p < 10^{-8}$).

Overall, AWSUM was the best performing method, though it only slightly outperforms HARP. Across all 725 queries AWSUM outperformed ReDDE by 18% on MAP@100 (Wilcoxon $p < 10^{-3}$). Extended KL Divergence performed best on the topic distillation queries, but AWSUM was within 5% on MAP@100 with ReDDE only slightly further behind.

The performance of the anchortext based selection methods is encouraging as the space of possible variants has only lightly been explored. It would be interesting to investigate the performance of this family of methods with other auxiliary collections which might be available in practice, such as a long-out-of-date full crawl of the same domain or a recent shallow crawl of the domain covered by the portal (limited number of pages fetched per server).[6]

It is not clear why scores for the named page task are substantially lower than those for the homepage task for some methods but not others since both are navigational tasks in the sense of [1]. For HARP and AWSUM the difference may be due to a higher prevalence of inter-server anchortext referring to homepages. However, the relative difference for the non-anchortext CORI and KL methods, is even greater, while for ReDDE the difference is very small. In all the sampling based methods, ranking of results for probe queries made use of link counts and anchortext.

We also compared the server selection methods on artificial partitionings of .GOV in which it is assumed that all servers publishing more than a threshold t were assumed to provide a search interface, corresponding to the hypothetical enforcement of an edict. Space does not permit presentation of full results, but for $t = 256$ (1053 searchable servers in .GOV) the MAP@100 scores of the various methods were similar to those presented in Table 3.

Another natural partitioning of the .gov data would be by agency or jurisdiction rather than by server. For example NASA and the state of California might provide a search service covering all their web servers.[7] This would reduce the number of collections to be searched and the work required to implement wrappers but would require maintenance of a list of which servers correspond to which agencies and the URL of the appropriate search interface — otherwise results from a single server might be included more than once. The absence of such a list made it difficult for us to investigate agency-based partitioning. Partitioning by agency raises the interesting possibility of multi-level metasearch.

A limitation of this study is that although the queries used were specifically oriented toward web search and the evaluation reflected the sort of behaviour web searchers expect of web search engines, real queries and user relevance/utility judgments were not available. At the time of writing, we did not have access to query logs from FirstGov. Even if we had such logs, there remains the difficulty of reconstructing the coresponding searcher's need and the value they would assign to candidate answers. In production use of a hybrid metasearcher, pseudo-judgments against real queries could be obtained using click-through data collected in normal operation.

Click-through data could also possibly be exploited in an enhanced server selection method.

8. CONCLUSION

Using a generally available web-oriented testbed, which models a range of plausible portal search applications, we have shown that ReDDE is superior to several previously published server selection methods in this application, although Extended KL Divergence worked best on the topic distillation queries. The testbed features a real partitioning into servers and a range of web query types.

ReDDE and KL Divergence do not exploit the web-specific information available in web search environments (and available in this testbed). We propose a family of methods based on link anchortext derived from an auxiliary collection (such as the central crawled collection in a hybrid implementation) and show that representatives of this family (HARP and AWSUM) are capable of out-performing ReDDE on the web-oriented tasks we have studied.

A further advantage of the web-oriented methods using the central crawled server as auxiliary collection is that, by virtue of outgoing links, they contribute usefully to solving the problem of identifying which servers operate within the domain covered by the portal. Server selection information is also automatically kept up to date. By contrast, server selection methods relying on probe queries would not automatically discover the commissioning of new servers or the decommissioning of old ones and would need to regularly fetch sample documents to maintain up-to-date models.

In the absence of an end-to-end effectiveness evaluation combining server selection with results merging on real

[6] As previously suggested by Craswell, such collections could also be used for estimating language models and sizes.

[7] Note that it is possible that the search interfaces we identified within .GOV actually cover content of more than one web server.

queries, no conclusion can be reached about whether the hybrid model we have discussed could achieve results competitive with those of a fully crawled service. In the absence of a full analysis of network costs, update frequency requirements and other portal characteristics, it is not certain what economic or quality benefits (eg. responsiveness to content change, provision of appropriate security model etc.) the hybrid model might bring. However, the fact that the best server selection methods studied are on average capable of finding the server containing the desired page (or one of the key resources on a broad topic) within the first two or three servers in the ranking provides encouragement to further investigation.

Despite the good performance of the anchor text methods, the practical implementation of a hybrid centralized/ distributed replacement for the current FirstGov government portal search would face formidable challenges, particularly in the accurate identification and characterization of available search interfaces and the development and maintenance of wrappers for them. Because of the huge number of sites to be included, the success of a hybrid implementation would depend on adoption of one or a small set of standard search interfaces across metasearched sites.

In the US, where network bandwidth is plentiful and traffic costs are low, distributed or hybrid models are unlikely to be adopted for providing government portal search to the general public. These methods are likely to be much more attractive when network costs are higher and bandwidth is more restricted and/or when included sites are subject to individual or group based access restrictions.

Investigation of alternative auxiliary collections and refinement of anchortext based selection methods remains for future work as does the evaluation of results merging methods on a web-oriented testbed.

9. REFERENCES

[1] Andrei Broder. A taxonomy of web search. *SIGIR Forum*, 36(2), 2002.

[2] J. P. Callan, Z. Lu, and W. Bruce Croft. Searching distributed collections with inference networks. In *Proc. ACM SIGIR 1995*, 1995.

[3] Jamie Callan, Margaret Connell, and Aiqun Du. Automatic discovery of language models for text databases. In *Proc. ACM SIGMOD 99*, 1999.

[4] Jared Cope, Nick Craswell, and David Hawking. Automated discovery of search interfaces on the web. In *Proc. 14th Australasian Database Conference*, 2003.

[5] Nick Craswell, Peter Bailey, and David Hawking. Server selection on the world wide web. In *Proc. ACM Digital Libraries Conference*, June 2000.

[6] Nick Craswell, Francis Crimmins, David Hawking, and Alistair Moffat. Performance and cost tradeoffs in web search. In *Proc. ADC 2004*, January 2004.

[7] Nick Craswell, David Hawking, and Stephen Robertson. Effective site finding using link anchor information. In *Proc. ACM SIGIR 2001*, 2001.

[8] Nick Craswell, David Hawking, Ross Wilkinson, and Mingfang Wu. Overview of the TREC-2003 web track. In *Proc. TREC 2003*, November 2003.

[9] James C. French, Alison L. Powell, Jamie Callan, Charles L. Viles, Travis Emmitt, Kevin J. Prey, and Yun Mou. Comparing the performance of database selection algorithms. In *Proc. ACM SIGIR 1999*, August 1999.

[10] James C. French, Allison L. Powell, Charles L. Viles, Travis Emmitt, and Kevin J. Prey. Evaluating database selection techniques: A testbed and experiment. In *Proc. ACM SIGIR 1998*, 1998.

[11] Luis Gravano, Héctor García-Molena, and Anthony Tomasic. GlOSS: Text-source discovery over the internet. *ACM Transactions on Database Systems*, 24(2), June 1999.

[12] David Hawking, Trystan Upstill, and Nick Craswell. Toward better weighting of anchors. In *Proc. ACM SIGIR 2004*, July 2004.

[13] Bernado A. Huberman and Lada A. Adamic. Evolutionary dynamics of the world wide web. Technical report, Xerox Palo Alto Research Center, February 1999. `http://www.hpl.hp.com/research/idl/papers/webgrowth/`.

[14] Panaglotis G. Ipeirotis and Luis Gravano. When one sample is not enough: Improving text database selection using shrinkage. In *Proc. ACM SIGMOD 2004*, Paris, June 2004. ACM Press.

[15] Ronny Lempel and Shlomo Moran. Optimizing result prefetching in web search engines with segmented indices. In *VLDB*, pages 370–381, 2002.

[16] Henrik Nottelmann and Norbert Fuhr. Combining CORI and the decision-theoretic approach for advanced resource selection. In *Proc. ECIC 2004*. Springer, 2004.

[17] Allison L. Powell and James C. French. Comparing the performance of collection selection algorithms. *ACM Transactions on Information Systems*, 21(4), October 2003.

[18] Yves Rasolofo, Faïza Abbaci, and Jacques Savoy. Approaches to collection selection and results merging for distributed information retrieval. In *Proc. CIKM 2001*, November 2001.

[19] Luo Si and Jamie Callan. The effect of database size distribution on resource selection algorithms. In *Proc. SIGIR 2003 Workshop on Distributed Information Retrieval*, August 2003.

[20] Luo Si and Jamie Callan. Relevant document distribution estimation method for resource selection. In *Proc. ACM SIGIR 2003*, July–August 2003.

[21] Luo Si, Rong Jin, Jamie Callan, and Paul Ogilvie. A language modeling framework for resource selection and results merging. In *Proc. CIKM 2002*, 2002.

[22] Craig Silverstein, Monika Henzinger, Hannes Marais, and Michael Moricz. Analysis of a very large web search engine query log. *SIGIR Forum*, 33(1), 1999.

[23] Amit Singhal and Marcin Kaszkiel. A case study in web search using TREC algorithms. In *Proc. WWW10*, May 2001.

[24] Jaime Teevan, Christine Alvarado, Mark S. Ackerman, and David R. Karger. The perfect search engine is not enough: A study of orienteering behaviour in directed search. In *Proc. 2004 Conference on Human Factors in Computing Systems*, April 2004.

[25] Jinxi Xu and W. Bruce Croft. Cluster-based language models for distributed retrieval. In *Proc. ACM SIGIR 1999*, August 1999.