Quality of language models for distributed information retrieval

Paul Thomas
paul.thomas@csiro.au

17 March 2009

Abstract

Collections used in distributed information retrieval (DIR) are often described by unigram language models, composed of simple term-probability statistics. In most cases, this information is not directly available from constituent collections and must be estimated by the DIR tool itself from a sample of documents. Factors affecting the quality of such estimates are not well understood, and nor is the impact of estimate quality.

Several measures of quality for unigram language models have been described, and three are used here to investigate how the quality of a model changes given document samples of differing size or quality. I show that although all models improve given larger samples, those built with more biased samples are of significantly lower quality; and that one of the three measures, Kullback-Leibler divergence, best describes model quality. Finally, it is shown that model quality has an impact on the effectiveness of standard server selection algorithms.

1 Introduction

Distributed information retrieval (DIR) systems use a single tool, commonly called a *broker*, to provide a unified interface to any number of independent search engines. Given a user's query, the broker will select one or more search engines likely to have relevant documents ("server selection"), translate and forward the query, collate all results ("result merging"), and present a single result list to the user.

Many DIR algorithms make use of characterisations of each of the available servers. As well as size, overlap, query language features, and other characteristics, an important part of these characterisations is the subject matter of each collection: the types of document[1] which are available and the types of queries for which it would be a good choice. This information has been used for to produce summaries of collections (Callan and Connell, 2001) and is most commonly used to inform server selection.

In a "cooperative" environment, search servers cooperate with brokers by providing information helpful for DIR: for example, by providing document counts or term information. The much more common "uncooperative" servers, by contrast, provide only a basic search interface; brokers working with these servers must provide their own methods for extracting information. Past work has proposed several techniques for summarising subject matter in both of these environments. Section 2 describes these, as well as variants and related work.

Unigram language models, outlined in Section 2.2, can be estimated with no cooperation from servers and are commonly used in server selection algorithms. These models are examined in experiments in Sections 3 and 4, where the quality of a model is shown to depend on the quality of the sampling mechanism used. While previous work has generally used TREC or Web data, these experiments use collections representing a wide variety of data types and sizes. Across a range of collections, models built with multiple queries sampling (Thomas and Hawking, 2007) are largely indistinguishable from those built with true random samples; those built with query-based sampling (Callan et al., 1999) are of lower quality. Examination of the correlation and relative power of three measures suggests that one measure, Kullback-Leibler divergence, captures model quality more usefully than other proposed alternatives.

Further experiments in Section 3 demonstrate a strong correlation between the quality of language models, on all three measures, and the performance of two server selection algorithms. This suggests that selection methods, and other DIR algorithms, are dependent on the quality of models used as input.

[1] "Document" is used here to describe the unit of retrieval. This may be, for example, a Web page; a piece of email; or a local file.

2 Related work

A large body of work has addressed the problem of characterising the subjects covered by collections. There have been two major approaches: classifications in a pre-determined taxonomy, such as a cataloguing scheme (Section 2.1), and models of the terms used in documents (Section 2.2). In each case, some proposed techniques require server cooperation; some require the text of documents; and others assume that only a query interface is available.

2.1 Classifications

A number of systems have been developed to classify collections according to their subject matter using a pre-determined taxonomy. If queries can also be classified accurately, such schemes simplify server selection: the only operation is to select for each query those servers which cover a relevant part of the topic hierarchy.

Sheldon et al. (1994) described a hierarchical DIR system which used "content labels" to describe cooperating servers. These labels, which were manually assigned, included information on servers (name, location, administrator, etc.) and collections (searchable fields and possible query completions). Labels were exposed to the user through a search tool and could be used for manual server selection and for search.

The more involved Pharos system (Dolin et al., 1996) used several pre-defined taxonomies to represent the contents of each collection and to suggest appropriate servers for a given query. Collections could be on- or off-line. Metadata provided by each server represented that server's holdings according to four parallel hierarchies: subject matter, according to Library of Congress classifications (LCC);[2] geographical coverage of the collection; historical coverage of the collection; and the publication dates of included documents.

This metadata was collated by each server from per-document information. The classification of each document could be manual, for example for library holdings already classified with the LCC, or automatic; Dolin et al. reported using latent semantic indexing for automatic classification. Once collated, the metadata was distributed using a scheme similar to Usenet news. "Mid-level" servers collected and replicated metadata regarding particular classifications or time periods, and an overview of the whole distributed system was collected at each of a number of "high-level" servers.

The hierarchical classification of servers was used by Pharos to suggest a set of servers for a user's query. Using latent semantic indexing or some equivalent method, the query could be mapped to a node, or set of nodes, in a relevant hierarchy; or the user could navigate the hierarchy and choose a node directly. With the vocabulary controlled in this way, queries to high-level servers would return a coarse-grained view of the

[2] `http://www.loc.gov/cataloging/classification/`

available resources, and a user could indicate relevant mid-level servers and eventually individual collections.

Simulation experiments with random collections and queries (Dolin et al., 1997) and with newsgroup data (Dolin et al., 1999) suggested that the classification scheme enabled fairly precise selection of collections, although the authors noted that it was not clear how it would scale to larger holdings. Although Pharos was expected to scale to millions of servers, it appears that it has not been used in full-scale experiments.

Pharos relied on cooperating servers and a shared classification scheme. Alternatives have assumed uncooperative servers, and used a query interface to classify collection contents.

Profusion (Gauch et al., 1996) used a set of 13 categories chosen to be representative of users' interests, including "science and engineering", "computer science", and "travel". Forty-eight queries, or about four queries per category, were issued to each of the six servers in use, and each returned document was manually judged for relevance. The servers' performance across each category was later used to influence server selection and result merging.

The Profusion strategy, although appropriate for a small number of servers and classifications, could not scale well to large numbers of either. The more scalable "probe and count" algorithm (Ipeirotis et al., 2001) uses the size of result sets, not document text, to classify collections in a hierarchical taxonomy. In a first step, a rule-based classifier such as RIPPER (Cohen, 1995, 1996) is trained with a set of documents, each in a known category. The classifier outputs rules of the form "if the document contains $term_i$ and $term_j$ and ..., then it is in $category_k$".

Each learned rule is then transformed into a Boolean "probe query" of the form "$term_i$ AND $term_j$ AND ..." and forwarded to the server; the size of the result set is used to approximate the number of documents about $category_k$ in the collection. It is not clear how the probe and count algorithm deals with engine-imposed limits on the size of result sets, inaccurate size estimates from servers, or servers which do not have a Boolean query language.

Document counts for each category are adjusted according to the classifier's accuracy on the training data, and the adjusted counts are finally used to find the node in the hierarchy which best represents the collection. The algorithm can be adapted to prefer specificity — classifications which cover only those topics in the collection — or coverage — classifications which cover as much of the collection as possible. Experiments with two sets of data, from newsgroups and Web-based databases respectively, demonstrated that the probe and count algorithm required fewer interactions with the server than alternatives and that its performance, on analogues of precision and recall, was reasonable.

The probe and count algorithm, renamed "QProber", was extended by Gravano and Ipeirotis (2003). While probe and count relied on classifiers themselves producing rules of the form "if $term_i$ and $term_j$ and ...", QProber included an algorithm for generating

such rules from the output of other classifiers including support vector machines and naïve Bayesian classifiers. A further set of experiments confirmed that these alternative classifiers produced server characterisations of similar accuracy.

Meng et al. (2002) described a similar system, also using probe queries to assign collections to a hierarchical taxonomy. Probe queries were created from the names of each category and all of its sub-categories, and the similarity between each query and documents in the collection was used to inform the assignment.

Meng et al. introduced three algorithms for calculating this similarity. The first, "high similarity with database centroid", uses information on term occurrences in each document to calculate an overall centroid for the collection. This centroid is compared, using a cosine similarity measure, with the description of each classification and the most similar classifications are retained. Although this technique requires access to accurate term frequency data for each document in the collection, and is therefore not useful for uncooperative servers, Meng et al. suggest that a sample of documents could be used instead. This was however not considered further.

The second algorithm, "high average similarity over retrieved documents", is a variant on the first. Rather than calculate the similarity between the entire collection and each description, this variant translates each description to a query, takes the text of the top few documents, and uses the similarity between these documents and the description as the ranking criteria. This algorithm requires access to the text of returned documents, and assumes servers are effective, but can otherwise work with an uncooperative server.

The third and final algorithm of Meng et al. (2002) uses singular value decomposition (SVD). Again, term information is assembled for each of a number of training collections. This, plus a manual classification of each collection, is given as input to an SVD classifier, which finds correlations between terms and classifications. These correlations, with information on term frequencies, can be used to classify a new collection.

Experiments using classifications from Yahoo![3] and the Open Directory Project[4] compared automatic classifications of 24 collections — 18 newsgroups and six Web collections from two universities — to a manual classification. An analogue of 11-point average precision indicated that the retrieved documents variant outperformed the other two algorithms; in a case study, a prototype which used this algorithm correctly classified a single Web-based database.

2.2 Language models

A second line of research has described collections by building *language models*. These models consider documents as the result of a stochastic process of language generation, and describe a probability distribution which captures this. In principle, these distribu-

[3] `http://dir.yahoo.com/`
[4] `http://dmoz.org/`

tions can be arbitarily complex but models have most commonly been simple: a set of individual terms with associated frequency information (Ponte and Croft, 1998). Tables 1 and 2 have examples with, respectively, term frequency and document frequency. Such language models have also been referred to as "representatives" of collections (Liu et al., 2001; Shokouhi et al., 2007) and as "content summaries" (Ipeirotis et al., 2001; Ru and Horowitz, 2005).

Language models were introduced to information retrieval by Ponte and Croft (1998), who described a use of models to rank documents in response to a query. A model is first inferred from the text of each document. Assuming each term occurence is independent of all others, the probability of generating the query q from the process underlying document d is the product of the probabilities of generating each term in the query and the probabilities of *not* generating every other term:

$$\Pr(q|d) = \prod_{t \in q} \Pr(t|d) \prod_{t \in \mathscr{T}_d \setminus q} (1 - \Pr(t|d)), \tag{1}$$

where $\mathscr{T}_d$ is the set of terms in the document.

These per-document language models are smoothed in two ways. A naïve calculation of $\Pr(t|d)$ is to use the relative frequency of t in d and assign $\Pr(t|d) = \mathrm{tf}_d(t)/|d|$, where $\mathrm{tf}_d(t)$ is the term frequency of t in d — the number of times t occurs in d — and $|d|$ is the number of terms in the document. If a term t does not occur at all in d, this leads to the conclusion that $\Pr(t|d) = 0$ and that t is never a possible output from the underlying stochastic process. As Ponte and Croft observe, this is a strange conclusion: not seeing something should not mean it is impossible. It is also not useful, since $\Pr(q|d)$ will be zero in all such cases. Instead, where a term does not occur we can smooth the model by approximating

$$\Pr_{\mathrm{zero}}(t) = \frac{\sum_{d \in \mathscr{D}} \mathrm{tf}_d(t)}{\sum_{d \in \mathscr{D}} |d|},$$

so if the term does not appear in a document at all then the mean rate of occurrence over the whole collection is used instead.

Models are also smoothed by including, for each term t, a component based on the mean probability of t in those documents where it appears: this is

$$\Pr_{\mathrm{avg}}(t) = \frac{\sum_{d \in \mathscr{D}} (\mathrm{tf}_d(t)/|d|)}{\mathrm{df}(t)}.$$

Given a mixing parameter $\hat{R}$, this and the previous correction finally give

$$\Pr(t|d) = \begin{cases} (\mathrm{tf}_d(t)/|d|)^{1-\hat{R}} \times \Pr_{\mathrm{avg}}(t)^{\hat{R}} & \text{if } \mathrm{tf}_d(t) > 0; \\ \Pr_{\mathrm{zero}}(t) & \text{otherwise.} \end{cases}$$

This definition is substituted into Equation 1, and documents are ranked according to $\Pr(q|d)$. Experiments by Ponte and Croft (1998), using TREC ad hoc data, showed

significant improvements in recall and precision using this model-based ranking instead of a standard ranking from INQUERY (Callan et al., 1992).

Similar smoothing can be seen in the Kullback-Leibler method for server selection (Si et al., 2002; Xu and Croft, 1999). Ipeirotis and Gravano (2004) have suggested a further smoothing technique, which uses mixtures of models according to a collection's place in a given topic hierarchy; this topic-based method increased the performance of the CORI server selection algorithm (Callan et al., 1995) in their experiments.

Ponte and Croft described unigram language models for individual documents; more commonly, models are used to summarise entire collections. Language models for collections are used in a number of server selection techniques including CORI, CVV (Yuwono and Lee, 1997), GlOSS (Gravano et al., 1999), and Kullback-Leibler divergence, and are considered in the experiments below.

Gravano et al. (1997) describe STARTS, a protocol for communication between a broker and cooperating servers which includes this sort of collection-scale model. Each STARTS-compatible server makes available metadata on each collection it indexes; this metadata includes a term list with associated frequency information. This could be used by a broker to inform server selection and result merging. Harvest (Bowman et al., 1994) similarly distributed metadata, in this case at the document level, between cooperating components. SearchDB-ML (Powell and Fox, 1998), a simpler alternative, did not include term data but did include manually-assigned descriptions of each server which could be used to inform server selection.

In DIR applications with uncooperative servers, language models for collections can be built from a sample of documents. Query-based sampling, for example, was originally proposed as a means to build language models (Callan et al., 1999), and recent work has measured the quality of this technique by the quality of the models produced (see for example Baillie et al. (2006b); Shokouhi et al. (2007); and Section 3 below). Since these models are built from a sample — a subset of a collection — they are constrained to reporting relative term frequencies, such as the mean number of term occurrences per document sampled. Absolute counts can be estimated with the addition of a collection size estimate; Ipeirotis and Gravano (2002) also suggest an algorithm which estimates total term counts while sampling as long as the overall shape of the term distribution is known. This alternative could for example be used to estimate df in the CORI algorithm, although it does not appear to have been considered.

2.3 Model quality

Several metrics have been proposed to measure the quality of a unigram language model. Most commonly, collection term frequency ratio and differences in term rankings have been used; alternatives include Kullback-Leibler divergence, predictive likelihood, and rates of change of document frequencies.

	Collection		Model	
Term t	$\text{tf}(t)$	$\Pr(t\|c)$	$\text{tf}_m(t)$	$\Pr(t\|m)$
cat	15	0.3191	8	0.3071
a	8	0.1702	2	0.0771
the	6	0.1277	5	0.1921
on	6	0.1277	3	0.1154
mat	4	0.0851	3	0.1154
sat	3	0.0638	3	0.1154
rat	3	0.0638	1	0.0387
ate	2	0.0426	1	0.0387
Total	47		26	

Table 1: A subset of term frequencies, extracted from a fictional collection. See the text for calculations.

2.3.1 Collection term frequency ratio

Collection term frequency (ctf) ratio was introduced by Callan et al. (1999) to measure the quality of unigram models built by query-based sampling. It measures the correspondence between the vocabulary of a model m and that of a collection, weighted to give more emphasis to more common terms.

The *term frequency* of a term t, $\text{tf}(t)$, is the total number of times it occurs in the collection. If $\mathscr{T}$ denotes the set of terms in a collection, ctf ratio measures the proportion of total term occurrences accounted for in the model:

$$\text{ctf ratio} = \frac{\sum_{t \in m} \text{tf}(t)}{\sum_{u \in \mathscr{T}} \text{tf}(u)}.$$

The ratio falls in the range $[0, 1]$, and higher values are assumed to represent a better quality model.

For example, consider Table 1, which extracts from a fictional collection. In this collection there are a total of 47 term occurrences. A model which included only the terms "cat" and "a would have a collection term frequency ratio of $(15+8)/47$, or 0.4893, regardless of the frequency of these terms in the model. If it also included "the" and "on", it would produce a ctf ratio of $(15+8+6+6)/47 = 0.7447$, despite including only four of eight terms.

2.3.2 Term rankings

The ctf ratio considers the proportion of terms included in the model, but does not distinguish between a model with accurate term frequency information and a model

	Collection			Model			
Term t	$\mathrm{df}(t)$	Rank	g_k	$\mathrm{df}_m(t)$	Rank	f_j	d_i^2
cat	6	1	1	5	1	1	0
on	5	2	1	3	4	3	4
a	4	4	3	2	6	1	4
the	4	4	3	4	2	1	4
mat	4	4	3	3	4	3	0
rat	3	6.5	2	1	7.5	2	1
sat	3	6.5	2	3	4	3	6.25
ate	2	8	1	1	7.5	2	0.25
Total							19.5

Table 2: A subset of document frequencies, extracted from a fictional collection. See the text for calculations.

without. For example, a single document such as a dictionary may contribute greatly to ctf ratio but without information on relative frequencies such a document would not be particularly helpful in selecting servers (Baillie et al., 2006b). A second measure introduced by Callan and Connell (2001) addresses this by comparing the ranking of terms in the model and the collection.

The Spearman rank correlation coefficient r_s is used to measure the correlation between two lists of ranks. This is applied by Callan and Connell to measure the quality of a model by first ordering terms in the collection according to document frequency, $\mathrm{df}(t)$; doing the same for terms in the model with $\mathrm{df}_m(t)$; and calculating the correlation

$$r_s = \frac{1 - \frac{6}{n^3-n}\left(\sum_i d_i^2 + \frac{1}{12}\sum_j (f_j^3 - f_j) + \frac{1}{12}\sum_k (g_k^3 - g_k)\right)}{\sqrt{\left(1 - \frac{\sum_j (f_j^3 - f_j)}{n^3-n}\right)}\sqrt{\left(1 - \frac{\sum_k (g_k^3 - g_k)}{n^3-n}\right)}}.$$

Here n is the number of terms which are in both the model and the collection (so $n = |\mathscr{T} \cap m|$) and d_i the difference, for the ith term, between its rank in the collection and its rank in the model. f_j is the number of terms tied for the jth rank in the model, and g_k is the number of ties in the kth rank for the collection; an example is in Table 2. Note that if two or more terms have the same frequency, such as "rat" and "sat" in Table 2, each has a rank adjusted for the number of tied terms.

The coefficient takes values in $[-1, 1]$, where 1 indicates a perfect positive correlation (so the terms are in exactly the same order), -1 indicates a perfect negative correlation (so the terms are in exactly the opposite order), and 0 indicates no correlation. Since the terms in the model may only be a subset of the terms in the collection, r_s is only

calculated over the terms in the model. Therefore, if terms — even common ones — are missing from the model, r_s will not reflect this.

Results reported here use a simpler calculation (Sheskin, 2004):

$$\begin{aligned} r_s &= \frac{T_f + T_g - \sum_i d_i^2}{2\sqrt{T_f T_g}}, \\ \text{where } T_f &= \frac{1}{12}\left(n^3 - n - \sum_j (f_j^3 - f_j)\right) \\ \text{and } T_g &= \frac{1}{12}\left(n^3 - n - \sum_k (g_k^3 - g_k)\right). \end{aligned}$$

Table 2 includes data from an example collection and from a sample model. We can calculate

$$\begin{aligned} T_f &= \tfrac{1}{12}\left(n^3 - n - \textstyle\sum_j (f_j^3 - f_j)\right) \\ &= \tfrac{1}{12}\left(8^3 - 8 - [3(1^3 - 1) + 2(2^3 - 2) + 3(3^3 - 3)]\right) \\ &= 35. \\ \text{Similarly, } T_g &= 35, \\ \text{and } r_s &= \frac{T_f + T_g - \sum_i d_i^2}{2\sqrt{T_f T_g}} \\ &= \frac{35 + 35 - 19.5}{2\sqrt{35 \times 35}} \\ &= 0.7214. \end{aligned}$$

Baillie et al. (2006b) suggest that r_s may be a superior measure to ctf ratio, since if term frequencies follow a known distribution, such as hyperbolic rank-frequency Zipf (1949) or Waring (Wolfram, 1992), then it may be possible to reconstruct frequency information from a ranking. If so, an accurate ranking should be sufficient to inform DIR algorithms.

2.3.3 Kullback-Leibler divergence

The ctf ratio considers the proportion of terms included in a model, and r_s the relative ranking of each. An alternative measure considers both, as well as the frequency of each term.

Kullback-Leibler divergence (Kullback and Leibler, 1951) was introduced to this context by Xu and Croft (1999) as a means to compare a query to a collection; it is used here to compare a model to a collection, and was independently suggested for the

same purpose by Baillie et al. (2006b). The collection and model are each considered as discrete probability distributions over $\mathscr{T}$, the terms in the collection, and the divergence is then the relative entropy between the two. (Other information theoretic measures, such as Jensen-Shannon divergence or Jeffreys divergence, also describe the difference between two distributions but have been little used in information retrieval.) The divergence between a collection c and its model m is defined by[5]

$$\mathrm{D}_{KL}(c\|m) = \sum_{t\in\mathscr{T}} \Pr(t|c) \log_2 \frac{\Pr(t|c)}{\Pr(t|m)}, \tag{2}$$

where $\Pr(t|c)$ is the probability distribution over the collection c

$$\Pr(t|c) = \frac{\mathrm{tf}(t)}{\sum_{u\in\mathscr{T}} \mathrm{tf}(u)}$$

and $\Pr(t|m)$, the probability distribution over the model m, uses Laplace smoothing in case terms are missing or the sample is small:

$$\Pr(t|m) = \frac{\mathrm{tf}_m(t) + \alpha}{\sum_{u\in\mathscr{T}} \mathrm{tf}_m(u) + |\mathscr{T}|\alpha}.$$

The experiments here follow Xu and Croft (1999) in assigning $\alpha = 0.01$. (An earlier use by Ipeirotis and Gravano (2004) did not smooth $\Pr(t|m)$, and was therefore constrained to considering only terms in both the model and the collection.)

Kullback-Leibler divergence falls in $[0,\infty)$. A value of zero indicates complete concordance; larger values indicate more divergence between the distributions. This is not a true distance metric since it is not symmetric (it is possible that $\mathrm{D}_{KL}(a\|b) \neq \mathrm{D}_{KL}(b\|a)$ for some a and b) and does not obey the triangle inequality (it is possible that $\mathrm{D}_{KL}(a\|c) > \mathrm{D}_{KL}(a\|b) + \mathrm{D}_{KL}(b\|c)$)(Kullback, 1959).[6]

For example, consider again the model in Table 1. Using $\alpha = 0.01$ and with $|\mathscr{T}| = 8$, the contribution for the term "cat" is

$$\Pr(\text{“cat”}|c) \log_2 \frac{\Pr(\text{“cat”}|c)}{\Pr(\text{“cat”}|m)}.$$

In this case $\Pr(\text{“cat”}|c) = 15/47 = 0.3191$ and $\Pr(\text{“cat”}|m) = (8 + 0.01)/(26 + 8 \times 0.01) = 0.3071$, so the contribution is $0.3191 \log_2(0.3191/0.3071) = 0.0177$. Repeating

[5]The choice of base for the logarithm is not critical. The data presented here uses base two, and measures divergence in bits.

[6]As originally described by Kullback and Leibler (1951), Kullback-Leibler divergence is symmetric: $\mathrm{D}_{KL}(c\|m) = \sum_{t\in\mathscr{T}} \left(\Pr(t|c) \log_2 \frac{\Pr(t|c)}{\Pr(t|m)} + \Pr(t|m) \log_2 \frac{\Pr(t|m)}{\Pr(t|c)} \right)$. The form given above, a "directed divergence" measure (Kullback, 1959), is typical of current use. Jensen-Shannon and Jeffreys divergence are symmetric.

this for every other term in the collection and summing the contributions gives an overall divergence.

Unlike ctf ratio and r_s, Kullback-Liebler divergence directly measures the difference between two discrete distributions, and considers both vocabulary and absolute frequency. This suggests it will be a more useful measure of model quality, and the results of the experiments below confirm this.

2.3.4 Other measures

Baillie et al. (2006a) have suggested a fourth measure of model quality, the likelihood of a model generating a pool of queries $\mathscr{P}$. By letting $\mathscr{P}$ be different for each user, model quality can be calculated relative to a that user's information needs; pools can also be collection independent. Experiments by Baillie et al. showed a moderate to strong correlation between likelihood and Kullback-Leibler divergence, suggesting this could measure quality without any knowledge of true term distributions; however a pool must be defined before this measure can be calculated, and since there is no pool of likely queries in this case the likelihood measure is not considered in the experiments below.

Monroe et al. (2000) considered two further metrics, which like predictive likelihood can be calculated without reference to the correct collection statistics. Their "df1 ratio" is the fraction of terms in the model which have been seen in only one document: experiments suggested that when this reaches a minimum the model is "good enough". Monroe et al. also suggest the rate of change of document frequencies as a measure of model stability: when the change from one iteration to the next is sufficiently small, the model can be considered good enough. Neither of these metrics seemed promising in early experiments (Monroe et al., 2000, 2002) and neither is considered here.

3 Language modelling experiments

Unigram language models as described above are used in a number of server selection techniques including CORI, CVV, GlOSS, and Kullback-Leibler divergence, and are therefore considered here. Experiments address three questions: how good are language models built with sampled documents? How can this quality be measured? And can models be improved with more effort?

These experiments do not consider the classification strategies of Section 2.1. Most successful server selection techniques to date have used unigram language models, and it is not presently clear what form of topic hierarchy or query classification technique would be appropriate for brokers generally. Classification-based methods would be a worthwhile area for future work.

		Size (in terms)			
Collection	No. docs	Range	Mean	Std dev	Topics
calendar	1k	1–20	4	2	Mixed
zsh-list	9k	2–59k	176	179	Narrow
procmail	24k	2–14k	207	215	Narrow
email	25k	1–26k	199	295	Mixed
WSJ	99k	9–10k	462	450	Broad
.GOV	1.2M	0–43k	6803	5720	Broad

Table 3: Summary statistics of collections used in the experiments.

3.1 Collections

The experiments described here use six collections, summarised in Table 3. These were used in experiments on "personal metasearch" (Thomas, 2008) and are representative of the range of resources which are likely to be used across DIR applications most generally: sizes range over three orders of magnitude, data types are varied, and topic areas range from the very focussed (development of the zsh shell) to the very broad (several years' worth of archived email). Each collection is mostly English-language. None are on the scale of the largest collections a broker may access, such as the Web or Dialog.[7] However, rather than ask each broker to sample (for example) the public Web, it is likely that a it would use hard-coded or pre-computed characteristics for larger collections. The collections used here span the likely size range of local, private, or workgroup collections, where pre-computed estimates are infeasible.

3.2 Samples

Models were built from documents returned by each of three different samplers. As a baseline, models were built using documents selected entirely at random; this is not possible in practice but provides a best case for comparison.

A second set of models was built from documents sampled by the multiple queries technique (Thomas and Hawking, 2007). This technique issues several queries, using a set of common English-language words, and ignores those with no results and those where the number of results appears to be constrained by some server-imposed limit. After a number of queries are issued (100 in these experiments), pages are selected randomly from the union of all results. The multiple queries sampler was the best performing across a rangle of collections in earlier experiments (Thomas and Hawking,

[7] Dialog (`http://www.dialog.com/`), a commercial search tool for publications, was claimed to include 1,400,000,000 documents in February 2008.

2007).

The third set of models was built from documents sampled by the query-based sampling technique of Callan et al. (1999). At each iteration, this sampler selects a term at random from the existing model and sends this as a query to the search engine; each document in the result set is then downloaded and the model updated. In these experiments, the first query was taken from a list of common English words. Although this sampler is more biased, it has been commonly used in previous work (Baillie et al., 2006b; Callan and Connell, 2001; Hawking and Thomas, 2005; Shokouhi et al., 2007; Si and Callan, 2003).

3.3 Model quality

Each experiment considers the quality of language models built from each set of samples. Given access to the full collection, and hence accurate term frequency information, the experiments consider ctf ratio, r_s, and Kullback-Leibler divergence.

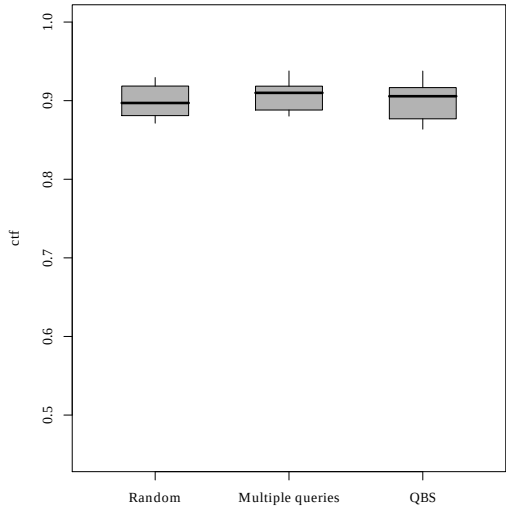
4 Results

Figure 1 summarises the ctf ratio, r_s, and D_{KL} measures over ten models for each of the six collections. In all but one case, a model was built using 300 documents selected randomly, by the multiple queries sampler, or by query-based sampling. The multiple queries sampler was unable to sample a full 300 documents from the calendar collection using the default query pool, so models used 200 documents in this case. No stemming or stopping was carried out.

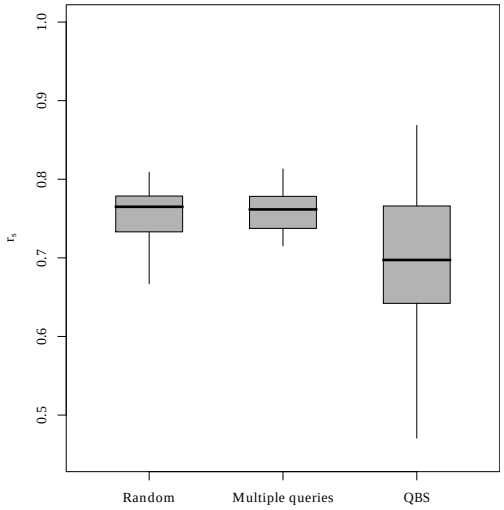
On all three measures, models built by random sampling and multiple queries samples are remarkably similar; there are no significant differences (two-sided t test, $\alpha = 0.05$). Models built by the query-based sampler differ significantly from both the others on the r_s measure ($p < 0.0007$ in both cases) and on the D_{KL} measure ($p < 0.0006$ against random sampling, $p < 0.02$ against multiple queries sampling).

As seen in earlier work (Thomas and Hawking, 2007), these results suggest both that a better quality sample makes an appreciable difference to DIR algorithms and that collection characterisations based on the multiple queries sampler are essentially indistinguishable from those based on true random samples.

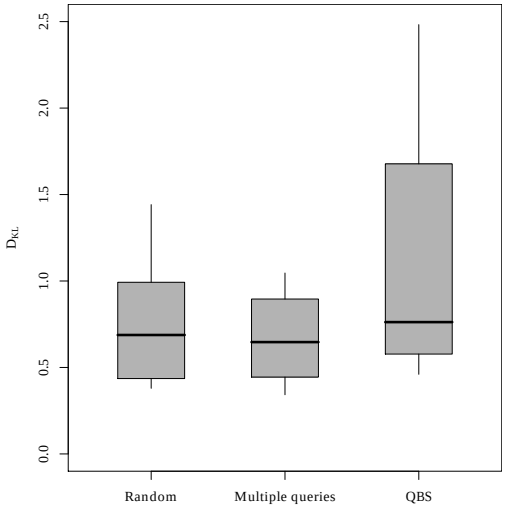
These results also suggest that ctf ratio may not have much discriminative power, since it scores models built from a biased sample as highly as it does those from a truly random sample. This is not surprising: a large fraction of the ctf ratio will be gained from seeing the most common words, even once, and these words are likely to be represented in even a biased sample of documents. This in turn suggests that ctf ratio is not likely to be a useful measure of model quality.



(a) ctf ratio (1 is best). No significant differences in model quality.



(b) r_s (1 is best). Both random samples and multiple queries samples produce significantly better models than query based samples.



(c) D_{KL} (0 is best). Both random samples and multiple queries samples produce significantly better models than query based samples.

Figure 1: Quality of models built from different samples. Ten models were built from 300 documents for each of six collections. The thick line marks the median, boxes show the interquartile range, and thin lines show the range. For clarity, outliers more than $1\frac{1}{2}$ times the interquartile range from the box are not plotted.

4.1 Evolution of models

The models used in the first experiment were based on samples of 300 documents, following Callan et al. (1999). Two hypotheses seem reasonable:

1. Models built from more documents may be of higher quality; however,
2. The quality of models may be constrained by any bias in the document samples themselves.

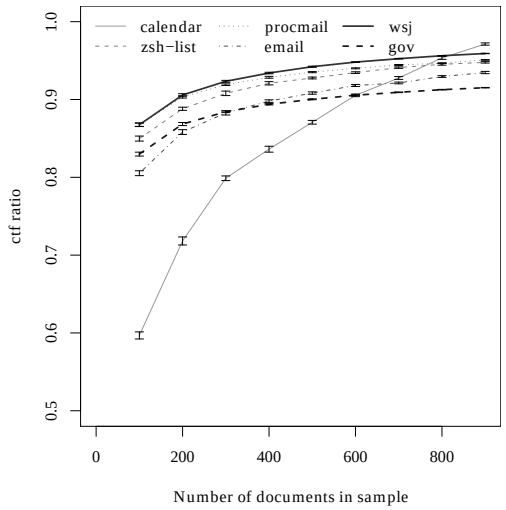
Figures 2 to 4 show improvements in ctf ratio, r_s, and D_{KL} for models built with increasing numbers of documents sampled randomly (Figure 2), with the multiple queries sampler (Figure 3), and with the query-based sampler (Figure 4). In each case, the lines plotted are the mean measure of each of ten models; the bars give one standard error to either side. As noted earlier, the calendar collection is particularly difficult to sample and neither the multiple queries nor the query-based method were able to produce large samples. The data for this collection is correspondingly cut off at 200 and 400 documents, respectively.

There are clear trends in each case. Models tend to improve on all metrics, and with all samplers, as more documents are included. This improvement does however slow down, and when models are built from random documents there is no significant improvement in ctf ratio across all collections with more than 700 documents (t test, $\alpha = 0.05$). Spearman's coefficient r_s does not improve significantly after 400 documents are used, and D_{KL} does not improve significantly after 500 documents. Using documents from the multiple queries sampler, there is no significant improvement in ctf ratio after 700 documents, in r_s after 200, or in D_{KL} after 400; for the query-based sampler there is no significant improvement in ctf ratio after 700, in r_s after 100, or in D_{KL} after 400 documents have been included in the model.

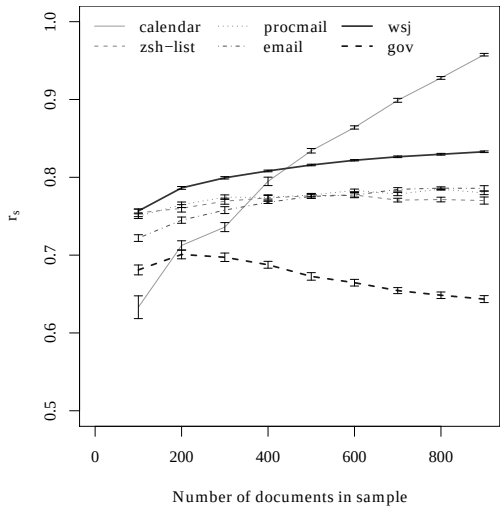
Models of the calendar collection improve fastest in all cases, although from a relatively poor base. This is likely due to the nature of the documents in this collection: they are very short, with a mean of only four terms, and there is little overlap between terms in any two documents. With such a collection, there are very few common terms, so models built from a small number of documents will be of low quality; however, since there is little overlap between documents, each additional document will contribute to the terms seen and hence to the quality of the model.

Models of the .GOV collection are also of relatively low quality, although the ctf ratio is similar to other collections. This can be explained by the size of the collection: with many more documents than other collections, a fixed size sample will cover less of the collection.

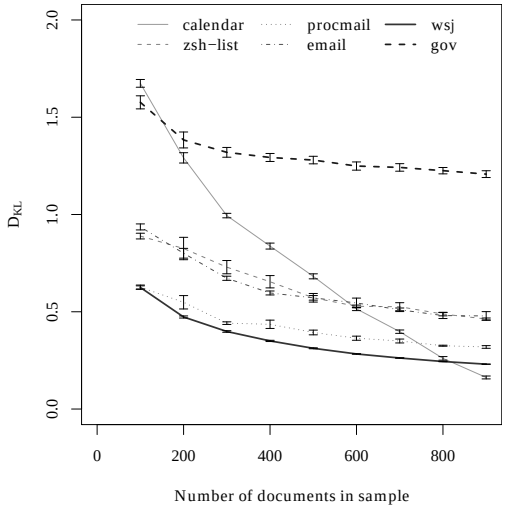
The effect of biased samples, seen elsewhere (Thomas and Hawking, 2007), is confirmed in these experiments as models built with documents from the query-based sample continue to have lower ctf ratio, lower r_s, and higher D_{KL}. With 900 documents



(a) ctf ratio (1 is best). No significant improvement overall after 700 documents are included.

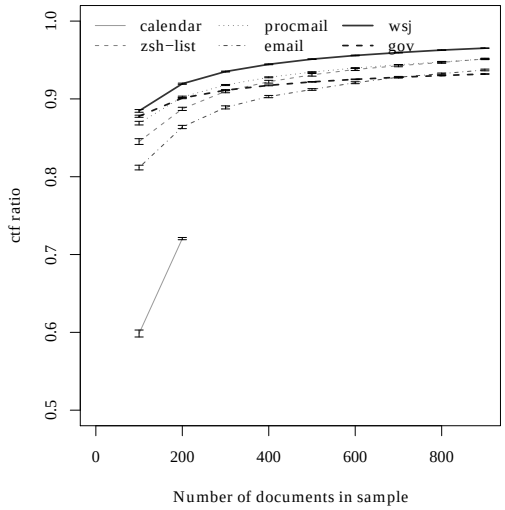


(b) r_s (1 is best). No significant improvement overall after 400 documents are included.

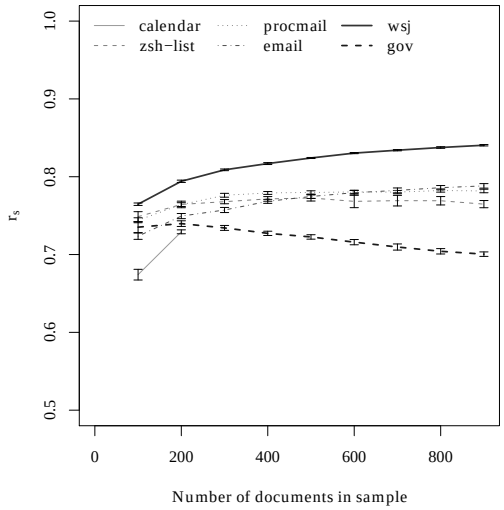


(c) D_{KL} (0 is best). No significant improvement overall after 500 documents are included.

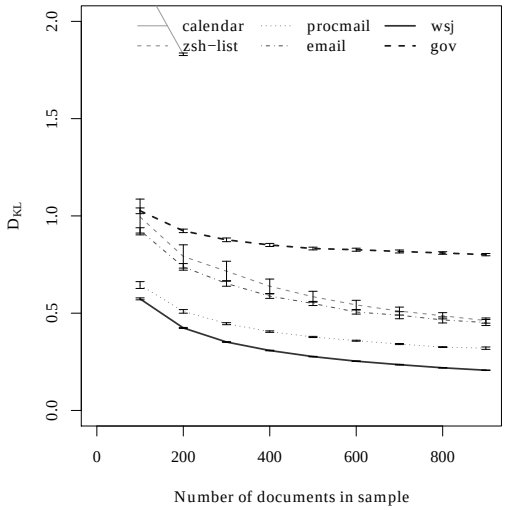
Figure 2: Improvement in quality measures with more documents from random samples. Plotted figures are the means from ten models each built from the specified number of documents, selected at random. Bars are ± 1 standard error.



(a) ctf ratio (1 is best). No significant improvement overall after 700 documents are included.

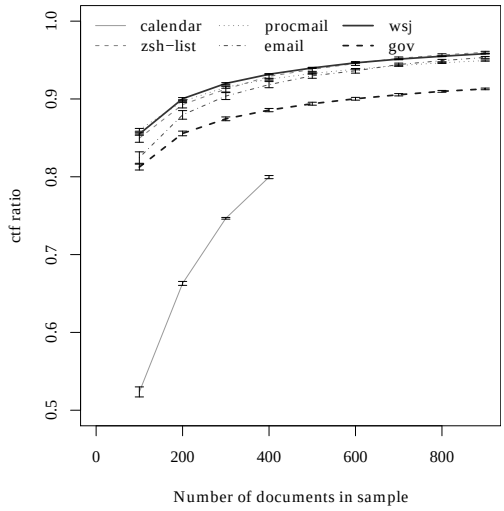


(b) r_s (1 is best). No significant improvement overall after 200 documents are included.

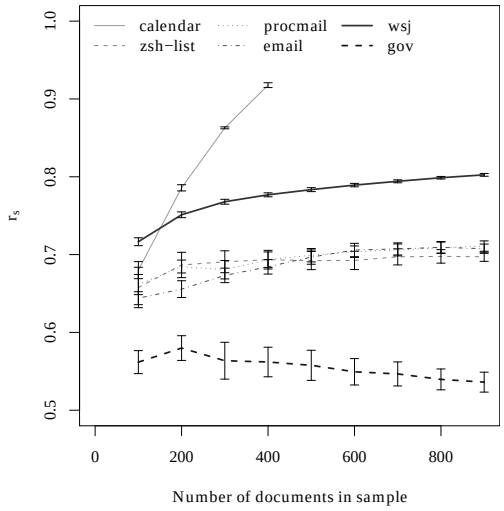


(c) D_{KL} (0 is best). No significant improvement overall after 400 documents are included. Divergence is very high (2.2 ± 0.03) for the calendar collection at 100 documents sampled.

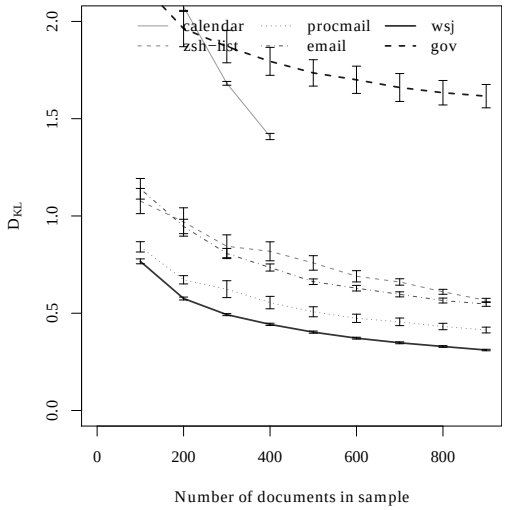
Figure 3: Improvement in quality measures with more documents from multiple queries samples. Plotted figures are the means from ten models each built from the specified number of documents, selected by the multiple queries sampler. Bars are ± 1 standard error.



(a) ctf ratio (1 is best). No significant improvement overall after 700 documents are included.



(b) r_s (1 is best). No significant improvement overall after 100 documents are included.



(c) D_{KL} (0 is best). No significant improvement overall after 400 documents are included. Divergence is very high for the calendar collection at 100 (2.7 ± 0.06) and 200 (2.1 ± 0.02) documents sampled.

Figure 4: Improvement in quality measures with more documents from query-based samples. Plotted figures are the means from ten models each built from the specified number of documents, selected by the query-based sampler. Bars are ± 1 standard error.

per model, there is again no significant difference in overall ctf ratio between the three samplers, but a significant difference remains between models built with the query-based sampler and models built with either of the other samplers if the r_s or D_{KL} measures are considered ($p < 0.02$ in each case).

Overall, these observations support the two hypotheses on p. 15. First, models do improve on all metrics as the number of documents used increases, although improvements slow down and are not significant past around 700 documents. Second, the quality of the samples used is a constraint on the quality of the models which are built: with the more biased samples of documents from query-based sampling, all three measures stop improving earlier and are significantly worse than alternatives.

4.2 Correlation between measures

From the results in Figures 2 to 4, it seems that the three measures are in agreement: as the ctf ratio improves, so does r_s and Kullback-Leibler divergence. To investigate further, Pearson's product-moment correlation was computed between all three measures across all 1500 models built for the experiments above.

Over all models, ctf ratio and D_{KL} are moderately negatively correlated (Pearson's $r = -0.75$). Since a lower D_{KL} score represents a better model and a higher ctf ratio represents the same, this indicates that the two measures broadly agree on the quality of each model. r_s is less correlated with the other two measures: $r = 0.23$ with ctf ratio and $r = -0.65$ with D_{KL}.

In similar experiments Baillie et al. (2006b), using query-based sampling of TREC data, also observed a negative correlation between ctf ratio and D_{KL}. Unlike the experiments reported above, however, their results indicated a negative correlation also existed beteween ctf ratio and r_s, and a positive correlation (meaning in this case disagreement) between r_s and D_{KL}. It is not clear why this should be the case: seeing terms with a higher document frequency, and hence improving ctf ratio, need not result in terms being ranked poorly. Baillie et al. suggest that with more data, they may have seen a positive correlation between ctf ratio and r_s, and presumably a negative correlation between r_s and D_{KL}.

4.3 Correlation with selection performance

A final set of experiments investigated the correlation between model quality, as reported by the three measures above, and the performance of the common CORI and Kullback-Leibler algorithms for server selection. This provides an indication of the practical importance of model quality; if there is no or only a weak correlation between quality and (for example) selection performance, there is little use measuring quality to begin with. On the other hand, if they do predict performance, then any technique which

improves models on these measures should have an impact in improved selection as well as in any other applications.

Figure 5 plots the performance of the two selection algorithms — measured with $\mathscr{R}_1$, the mean recall at one collection selected (Gravano and García-Molina, 1995) — as a function of model quality. Models here were generated from ninety sets of samples, ten each of $100, 200, \ldots 900$ documents chosen randomly from each of the six collections; the mean measure over all six collections was used to indicate the overall quality of the sample. In each case, the mean measure varied from set to set but improved as observed in Section 4.1 as more documents were included. 120 queries were used, 20 based on each of the six collections, and relevance judgements conducted over pools of depth five. For each query, relevant documents were found in one or two collections; each collection contributed at least 45 relevant documents in total.

Correlation between model quality and $\mathscr{R}_1$ was high for both CORI and Kullback-Leibler divergence, with absolute coefficients of correlation (Spearman's r_s) of between 0.64 and 0.82 ($p \ll 0.05$ in each case). Selection performance, at least for these two algorithms, does seem to depend upon the quality of the language models available; they will improve if models improve and degrade if they are made worse. Further, all three measures discussed in Section 2.3 are useful; all three are good predictors of how well selection algorithms will fare.

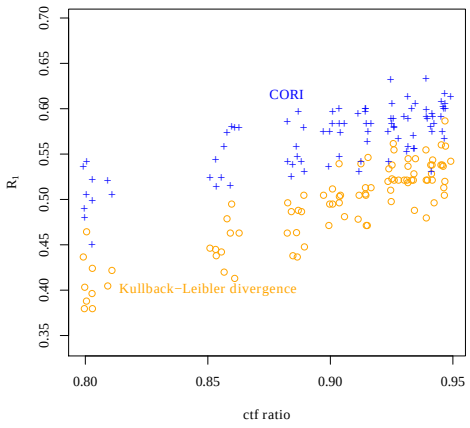
5 Conclusions

Alongside attributes such as collection size, an important aspect of characterisation is the subject matter and language used at each server. This can inform server selection by suggesting what types of documents a server makes available, and the types of queries for which it would be a good choice. In particular, unigram language models which record term frequency information are used by many server selection algorithms.

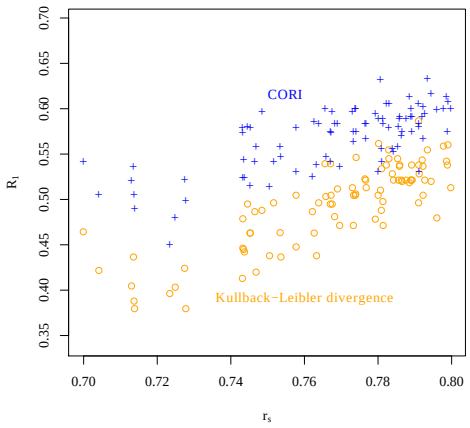
Experiments with such models show that the quality of the document sample used makes an appreciable difference to the quality of the eventual model. Models built with the multiple queries sampler are largely indistinguishable from those based on truly random documents; for example, there is no significant difference in Kullback-Leibler divergence when 300 documents are included. Models built with query-based sampling are of lower quality.

Of the three quality measures considered, ctf ratio and Kullback-Leibler divergence appear to be in agreement over the quality of models, across a range of sizes and sampling methods, while r_s does not agree with either.

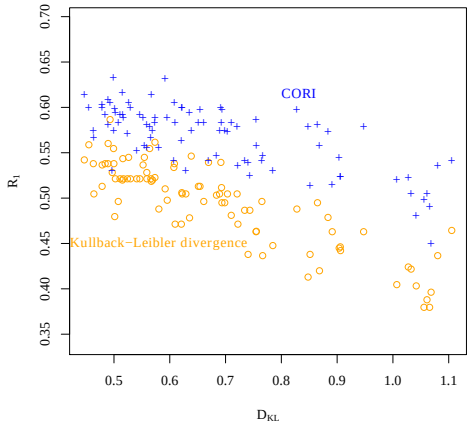
Finally, it makes sense to care: these experiments have demonstrated a strong correlation between the quality of a unigram language model, as reported by any of the three measures, and the performance of two representative server selection algorithms. The quality of a model does have an appreciable impact on the performance of a working broker.



(a) $\mathscr{R}_1$ and ctf ratio. Correlation $r_s = 0.66$ for CORI, 0.82 for Kullback-Leibler divergence



(b) $\mathscr{R}_1$ and r_s. Correlation $r_s = 0.66$ for CORI, 0.79 for Kullback-Leibler divergence



(c) $\mathscr{R}_1$ and D_{KL}. Correlation $r_s = -0.64$ for CORI, -0.82 for Kullback-Leibler divergence

Figure 5: Correlation between measures of model quality and selection performance at one collection selected, for CORI and Kullback-Leibler divergence. Each plotted point is the mean $\mathscr{R}_1$ over 120 queries. $p \ll 0.05$ in each instance.

6 Acknowledgements

I thank David Hawking for helpful feedback on earlier versions of this paper and on the experiments it describes.

References

Baillie, M., Azzopardi, L., Crestani, F., 2006a. Adaptive query-based sampling of distributed collections. In: Proc. SPIRE. No. 4209 in Lecture Notes in Computer Science.

Baillie, M., Azzopardi, L., Crestani, F., 2006b. Towards better measures: Evaluation of estimated resource description quality for distributed IR. In: Proc. First International Conference on Scalable Information Systems.

Bowman, C. M., Danzig, P. B., Hardy, D. R., Manber, U., Schwartz, M. F., 1994. Harvest: A scalable, customizable discovery and access system. Tech. Rep. CU-CS-732-94, University of Colorado at Boulder Department of Computer Science.

Callan, J., Connell, M., 2001. Query-based sampling of text databases. ACM Trans. Info. Systems 19 (2).

Callan, J., Connell, M., Du, A., 1999. Automatic discovery of language models for text databases. In: Proc. ACM SIGMOD.

Callan, J. P., Croft, W. B., Harding, S. M., 1992. The INQUERY retrieval system. In: Proc. Third International Conference on Database and Expert Systems Applications.

Callan, J. P., Lu, Z., Croft, W. B., 1995. Searching distributed collections with inference networks. In: Proc. ACM SIGIR.

Cohen, W. W., 1995. Fast effective rules induction. In: Proc. Twelfth International Conference on Machine Learning.

Cohen, W. W., 1996. Learning trees and rules with set-valued features. In: Proc. Thirteenth National Conference on Artificial Intelligence and Eighth Innovative Applications of Artificial Intelligence Conference. Vol. 1.

Dolin, R., Agrawal, D., Dillon, L., El Abbadi, A., 1996. Pharos: A scalable distributed architecture for locating heterogeneous information sources. Tech. Rep. TRCS96-05, Department of Computer Science, University of California at Santa Barbara.

Dolin, R., Agrawal, D., El Abbadi, A., 1999. Scalable collection summarization and selection. In: Proc. ACM International Conference on Digital Libraries.

Dolin, R., Agrawal, D., El Abbadi, A., Dillon, L., 1997. Pharos: A scalable distributed architecture for locating heterogeneous information sources. In: Proc. CIKM.

Gauch, S., Wang, G., Gomez, M., 1996. ProFusion: Intelligent fusion from multiple, distributed search engines. Journal of Universal Computer Science 2 (9).

Gravano, L., Chang, K., García-Molina, H., Lagoze, C., Paepcke, A., 1997. STARTS: Stanford protocol proposal for internet retrieval and search. In: Proc. ACM SIGMOD.

Gravano, L., García-Molina, H., 1995. Generalizing GlOSS to vector-space databases and broker hierarchies. In: Proc. VLDB.

Gravano, L., García-Molina, H., Tomasic, A., 1999. GlOSS: Text-source discovery over the internet. ACM Trans. Database Systems 24 (2).

Gravano, L., Ipeirotis, P. G., 2003. QProber: A system for automatic classification of hidden-web databases. ACM Trans. Info. Systems 21 (1).

Hawking, D., Thomas, P., 2005. Server selection methods in hybrid portal search. In: Proc. ACM SIGIR.

Ipeirotis, P. G., Gravano, L., 2002. Distributed search over the hidden web: Hierarchical database sampling and selection. In: Proc. VLDB.

Ipeirotis, P. G., Gravano, L., 2004. When one sample is not enough: Improving text database selection using shrinkage. In: Proc. ACM SIGMOD.

Ipeirotis, P. G., Gravano, L., Sahami, M., 2001. Probe, count, and classify: Categorising hidden-web databases. In: Proc. ACM SIGMOD.

Kullback, S., 1959. Information Theory and Statistics. John Wiley & Sons, New York, NY, USA.

Kullback, S., Leibler, R. A., 1951. On information and sufficiency. Annals of Mathematical Statistics 22 (1).

Liu, K.-L., Santoso, A., Yu, C., Meng, W., Zhang, C., 2001. Discovering the representative of a search engine. In: Proc. CIKM. Poster.

Meng, W., Wang, W., Sun, H., Yu, C., 2002. Concept hierarchy based text database categorization. Knowledge and Information Systems 4 (2).

Monroe, G. A., French, J. C., Powell, A. L., Jan. 2002. Obtaining language models of web collections using query-based sampling techniques. In: Proc. Hawaii International Conf on System Sciences. IEEE Computer Society Press.

Monroe, G. A., Mikesell, D. R., French, J. C., 2000. Determining stopping criteria in the generation of web-derived language models. Tech. Rep. CS-2000-30, Department of Computer Science, University of Virginia.

Ponte, J. M., Croft, W. B., 1998. A language modeling approach to information retrieval. In: Proc. ACM SIGIR.

Powell, J., Fox, E. A., 1998. Multilingual federated searching across heterogenous collections. D-Lib Magazine 4 (9).

Ru, Y., Horowitz, E., 2005. Indexing the invisible web: A survey. Online Information Review 29 (3).

Sheldon, M. A., Duda, A., Weiss, R., O'Toole, J., Gifford, D. K., 1994. Content routing for distributed information servers. In: Proc. Int. Conf. on Extending Database Technology.

Sheskin, D. J., 2004. Handbook of Parametric and Nonparametric Statistical Procedures, 3rd Edition. Chapman & Hall/CRC, Boca Raton, Florida, USA.

Shokouhi, M., Baillie, M., Azzopardi, L., 2007. Updating collection representations for federated search. In: Proc. ACM SIGIR.

Si, L., Callan, J., 2003. Relevant document distribution estimation method for resource selection. In: Proc. ACM SIGIR.

Si, L., Jin, R., Callan, J., Ogilvie, P., 2002. A language modeling framework for resource selection and results merging. In: Proc. CIKM.

Thomas, P., 2008. Server characterisation and selection for personal metasearch. Ph.D. thesis, Australian National Univeristy, under examination.

Thomas, P., Hawking, D., 2007. Evaluating sampling methods for uncooperative collections. In: Proc. ACM SIGIR.

Wolfram, D., 1992. Applying informetric characteristics of databases to IR system file design, part I: Informetric models. Information Processing and Management 28 (1).

Xu, J., Croft, W. B., 1999. Cluster-based language models for distributed retrieval. In: Proc. ACM SIGIR.

Yuwono, B., Lee, D. L., 1997. Server ranking for distributed text retrieval systems on the internet. In: Proc. 5th Int. Conf. on Database Systems for Advanced Applications.

Zipf, G. K., 1949. Human Behaviour and the Principle of Least Effort: An Introduction to Human Ecology. Addison-Wesley, Reading, MA, USA.

Contact Us
Phone: 1300 363 400
+61 3 9545 2176
Email: enquiries@csiro.au
Web: www.csiro.au

Your CSIRO

Australia is founding its future on science and innovation. Its national science agency, CSIRO, is a powerhouse of ideas, technologies and skills for building prosperity, growth, health and sustainability. It serves governments, industries, business and communities across the nation.