Document Allocation Policies for Selective Searching of Distributed Indexes

Anagha Kulkarni and Jamie Callan
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Avenue, Pittsburgh, PA 15213
anaghak,callan@cs.cmu.edu

ABSTRACT

Indexes for large collections are often divided into *shards* that are distributed across multiple computers and searched in parallel to provide rapid interactive search. Typically, *all* index shards are searched for each query. For organizations with modest computational resources the high query processing cost incurred in this exhaustive search setup can be a deterrent to working with large collections. This paper investigates document allocation policies that permit searching only a few shards for each query (*selective search*) without sacrificing search accuracy. Random, source-based and topic-based document-to-shard allocation policies are studied in the context of selective search.

A thorough study of the tradeoff between search cost and search accuracy in a sharded index environment is performed using three large TREC collections. The experimental results demonstrate that selective search using topic-based shards cuts the search cost to less than 1/5th of that of the exhaustive search without reducing search accuracy across all the three datasets. *Stability* analysis shows that 90% of the queries do as well or improve with selective search. An *overlap*-based evaluation with an additional 1000 queries for each dataset tests and confirms the conclusions drawn using the smaller TREC query sets.

Categories and Subject Descriptors

H.3 [**INFORMATION STORAGE AND RETRIEVAL**]: Information Search and Retrieval

General Terms

Algorithms, Experimentation

1. INTRODUCTION

It is common for search engines, in both, research and commercial systems, to partition large document collection into multiple tiers and *shards* (disjoint indexes) [4, 3, 17]. Distribution of documents across multiple indexes enables parallelization of the search process which in turn improves the query response time. However, even in this setup the cost associated with searching a large collection (or tier) remains high because the entire collection is searched for every query processed by the system. For organizations with modest resources the high cost associated with searching large collections can be a deterrent to experimenting with such datasets. Our goal is to enable organizations with modest computer resources to work with large collections without compromising search accuracy.

For most search tasks only a small portion of the collection is relevant to any particular query. If one could identify and search only this subset of potentially relevant documents (*selective search*) then the search cost could be reduced without compromising search accuracy. Thus our goal is to facilitate selective searching of shards by allocating documents to shards such that documents relevant to the same query are concentrated in a few shards. To achieve this we investigate two types of document allocation policies – source-based and topic-based which are described in detail in Section 3. Earlier work [21] investigated these policies using relatively small datasets. In this paper, we study the issues such as scale and efficiency that arise when working with large document collections. We adapt the techniques used for source-based and topic-based allocation to be efficient at partitioning a large collection into shards and also be effective in selective searching of these shards.

Given a set of shards, deciding which index shards to search for a given query, is a type of *resource selection* problem [6]. In most prior research on resource selection, the resources were usually independent search engines that might be uncooperative. Selectively searching the shards of a large index is however an especially *cooperative* federated search problem where the federated system can define the resources (shards) and expect complete support from them. We employ one of the well-established resource selection algorithms (described in Section 4) to select the set of shards that are estimated to contain all or most of the potentially relevant documents for a given query.

Since our goal is to perform search on large collections with a workstation-equivalent computer, we conducted a detailed analysis of the tradeoff between the average search cost and the average search accuracy in a sharded index environment. In addition to good average-case effectiveness, a second goal is to minimize the number of queries that are harmed by selective search (*stability*).

CIKM'10, October 26–30, 2010, Toronto, Ontario, Canada.

The empirical results demonstrate that the adapted topic-based allocation policy can provide search accuracy that is statistically indistinguishable from that of complete search while the search cost is cut to less than 1/5th of that of the exhaustive search for all the datasets. The stability analysis reveals that 10% or fewer queries are adversely affected when the search cost is maintained at about 10-15% of the complete search cost. An additional evaluation using a larger set of queries (1000 queries) confirms these results using a different metric, *overlap*, which measures the percentage of documents that are common to the complete search and selective search results at specified document rank cutoffs.

2. RELATED WORK

The two tasks that are central to this work are *document allocation* and *resource selection*. We provide a brief summary of prior research on these topics.

2.1 Resource Selection

Federated search *resource selection* algorithms identify the resources (typically distinct search engines) or shards that are most likely to return relevant documents for a query. There is a large body of prior research on this topic.

CORI [7] represents the contents of each resource as one large 'document' and then adapts INQUERY's document ranking algorithm to rank resources. ReDDE and SUSHI [18, 19] sample documents from each resource to create a *central sample index* which is then queried to estimate the distribution of relevant documents across the resources. Arguello et al. [1] take a classification based approach to resource ranking. For each resource they learn a binary classifier that estimates the relevance probability of the resource to the given query. Document contents, query category and click-through logs are used to define classifier features.

Any of these algorithms could be employed for shard selection. In our preliminary experiments with CORI and ReDDE we observed that ReDDE outperformed CORI even in this co-operative setup. We thus use ReDDE in this work.

2.2 Document Allocation

Most prior research treats the assignment of documents to resources or shards as a given ([7, 18, 19, 2]), but there are a few exceptions.

The work that is most closely related to ours was done by Xu and Croft [21]. (We highlight the difference between the two approaches in Section 3.3) They used a two-pass K-means clustering algorithm and a KL-divergence based distance metric to organize document collection into 100 clusters. Three organization policies that catered to different levels of cooperation were proposed. *Global clustering*, the most cooperative setup partitioned the entire collection into 100 clusters, while the *local clustering* re-organized only the documents within a resource into topical clusters. The least cooperative setup, *multi-topic representation* did not physically re-organize any documents but simple estimated the topic models for each of the resources which were then used for improving resource selection. The experimental results showed that selective search using global clustering organization performed about as well as exhaustive search when 10 shards were searched. Local clustering did worse than global clustering. Multi-representation however did much worse than global clustering. All three organization policies did better than the baseline of source-based organization.

Larkey et al. [10] studied selective search on a dataset composed of over a million US Patents documents. The dataset was divided into 401 topical units using manually assigned patent categories, and into 401 chronological units using dates. Selective search using 10 partitions was found to be more effective using the topical organization than the chronological organization.

Puppin et al. [15] used query logs to organize document collection into multiple shards. The query log covered a period of time when exhaustive search was used for each query. These training queries and the documents that they retrieved were co-clustered to generate a set of *query clusters* and a set of corresponding *document clusters*. Sixteen clusters were created in this way. Documents that were not retrieved by any of the training queries cannot be clustered using the proposed technique and thus were put in a separate (fall-back) cluster that consisted of 52% of the collection. Selective search using shards defined by these clusters was found to be more effective than selective search using shards that were defined randomly.

Our work aims at reducing the average search cost required to process queries against large-scale collections while maintaining the search quality. Other studies have looked at load-balancing, caching and query-latency issues in the context of distributed search systems. For example, Puppin et al. [16] used the approach in [15] (described above) to organize a collection into clusters and to rank the clusters for each new query. However, now the top T clusters were assigned a maximum search priority and the rest of the clusters receive a linearly decreasing priority. A cluster's search priority along with its current load status determined if the query would be executed against the cluster. This effectively promoted load-balancing across all clusters. They also experiment with *incremental caching* which processed as follows. Run each query against the set of next best clusters that have not been searched for this query before. Thus incremental caching can build-up a good result set for frequent queries over time. They observe the maximum overlap (87.1%) between the exhaustive search results and the selective search results when using incremental caching and load-balancing with peak load of 55.5%, that is, on an average each query is sent to 50% of the clusters in the system.

It is important to note that techniques proposed for load-balancing, caching and query-latency, such as the above, and the work proposed in this paper are not mutually exclusive. These techniques can very well be applied in our setup to further reduce the search cost.

3. DOCUMENT ALLOCATION POLICIES

Our goal is to investigate document allocation policies that are effective, scalable, and applicable in both research and commercial environments. As a result, although we recognize the considerable value of query logs and well-defined categories, in this work we focus on allocation policies that are not dependent on availability of such resources. The three policies studied in this work, random, source-based, and topic-based are described in the following sub-sections.

3.1 Random Document Allocation

The random allocation policy assigns each document to one of the shards at random with equal probability. One might not expect a random policy to be effective for selective search since it might spread the relevant documents across

multiple shards. However, we experiment with this policy because it is one of the most efficient and generally applicable policies and was also a baseline in prior research [15].

3.2 Source-based Document Allocation

The datasets used in our work are all from the Web. The source-based policy uses document URLs to define shards. The document collection is sorted based on document URLs, which arranges documents from the same website consecutively. Groups of M/K consecutive documents are assigned to each shard (M: total number of documents in the collection, K: number of shards). The key assumption behind source-based allocation is that documents from the same website are typically related or similar to each other. According to *cluster hypothesis* which states that *closely associated documents tend to be relevant to the same request* [20], this could have an effect of concentrating relevant documents in a few shards which in turn could facilitate selective searching of shards. Source-based allocation was also used as a baseline in prior research [21].

3.3 Topic-based Document Allocation

Cluster-based and category-based document allocation policies that appeal to the *cluster hypothesis* to define topic-based shards were effective in prior research [21, 10]. Xu and Croft [21] showed that the K-means clustering algorithm can be used to partition small collections into distributed indexes that support effective resource selection. The K-means algorithm is simple and effective, thus it is a good candidate for implementing a topic-based document allocation policy. However, problems of scale and accuracy emerge when applying Xu and Croft's method to much larger datasets. We address both these problems by adapting the K-means algorithm and the similarity metric used by the clustering algorithm. Our approach to topic-based allocation policy is described in the following paragraphs.

Typically, a clustering algorithm is applied to the entire dataset in order to generate clusters. Although the computational complexity of the K-means algorithm [11] is linear in the number of documents (M), applying this algorithm to large collections is still computationally expensive. Thus, we sample a subset (S) of documents from the collection ($|S| << |M|$), using uniform sampling without replacement. The standard K-means clustering algorithm is applied to S and a set of K clusters is generated after five passes of re-assignments on the set S. The remaining documents in the collection ($M - S$) are then projected onto the space defined by the K clusters. This process of assigning the remaining documents to the K clusters can be parallelized, thus even massive collections can be efficiently partitioned into shards.

K-means algorithm requires a similarity or a distance metric to estimate the best centroid assignment for a document during each of the passes through the collection. We compute the similarity between a document D and a centroid C^i using the symmetric version of negative Kullback-Liebler divergence (Equation 1). The first component in the equation computes $KL(C^i||D)$ which puts emphasis on the terms that are important to the centroid while the contribution of the terms in the document is dampened by the logarithm function. However, the second component compensates for this bias of the first component by emphasizing the terms that are important to the document.

$$sim(C^i, D) = \sum_{w \in C^i \bigcap D} p_{c^i}(w)\ log \frac{p_d(w)}{\lambda\ p_B(w)} + \sum_{w \in C^i \bigcap D} p_d(w)\ log \frac{p_{c^i}(w)}{\lambda\ p_B(w)} \quad (1)$$

$p^i_c(w)$ and $p_d(w)$ are the unigram language models of the cluster centroid C^i and the document D, respectively. Using maximum likelihood estimation (MLE), the cluster centroid language model computes to, $p^i_c(w) = c(w, C^i)/\sum_{w'} c(w', C^i)$ where $c(w, C^i)$ is the occurrence count of w in C^i. Following Zhai and Lafferty [22], we estimate $p_d(w)$ using MLE with Jelinek-Mercer smoothing which gives

$$p_d(w) = (1 - \lambda)\ c(w, D)/\sum_{w'} c(w', D) + \lambda\ p_B(w) \quad (2)$$

$p_B(w)$ is the probability of the term w in the background model which is an average of the K centroid models. As such, the presence of $p_B(w)$ in the denominator plays an important role of incorporating the inverse collection frequency of the term into the metric which Zhai and Lafferty [22] found to behave similar to the traditional inverse document frequency (IDF) statistic. (Please refer to [14] for the transformation of negative KL-divergence with smoothing into a similarity metric as used in Equation 1.)

We found the above similarity metric to be more effective than the variant used by Xu and Croft [21] due to mainly three reasons. Firstly, the distance metric used by Xu and Croft is heavily biased towards shorter centroids. Secondly, the non-symmetric nature of the metric does not allow the centroid's language model to make much contribution, thus terms that are important only to the document but not to the centroid can make a large contribution towards the distance value. Lastly, the term weighting that is provided by inverse collection or inverse document frequency is not available in the metric used by Xu and Croft.

In addition to K-means we also conducted preliminary experiments using another technique for creating topic-based shards – latent dirichlet allocation (LDA) [5]. However, our results consistently demonstrated both K-means and LDA to be equally effective in defining topical shards. We chose K-means over LDA due to its lower computational cost.

3.4 Sample size and OOV terms

Using a subset instead of the entire collection to learn the clusters reduces the computational cost and makes the approach efficient. However, it also introduces the issue of out-of-vocabulary (OOV) terms during inference. Depending upon the size of the subset (S) that was used for learning, the remaining documents in the collection are bound to contain terms that were not observed in S and thus are absent from the learned clusters models. In such a situation, inference must proceed using the seen terms and ignore the OOV terms. However, the inference quality can potentially degrade because of the discounting of the OOV terms. It is important to select a sample size that leads to a small percentage of OOV terms per document. In Section 6.1 we present an analysis of sample size versus percentage of OOV terms per document for the datasets used in this work.

Table 1: Datasets and Query Sets

Dataset	Number of Documents	Number of Words (billion)	Vocabulary Size (million)	Average Document Length	Query Set	Average Query Length	Average Number of Relevant Documents/Query
Gov2	25,205,179	23.9	39.2	949	701-850	3.1	179 (+/- 149)
Clue-CatB	50,220,423	46.1	96.1	918	TREC09:1-50	2.1	80 (+/- 49)
Clue-CatA-Eng	503,903,810	381.3	1,226.3	757	TREC09:1-50	2.1	114 (+/- 64)

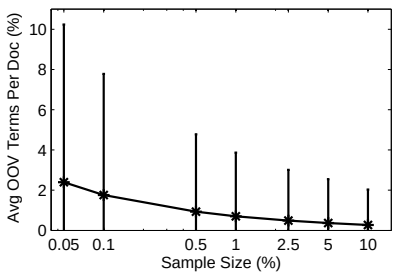

Figure 1: Sample size vs. percentage of OOV terms per document, on average, for the Gov2 dataset.

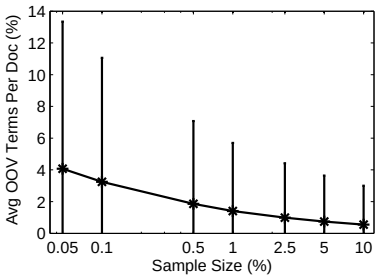

Figure 2: Sample size vs. percentage of OOV terms per document, on average, for the Clue-CatB Dataset.

4. SHARD SELECTION

After index shards are defined, a resource selection algorithm is used to determine which shards to search for each query. Our research used ReDDE [18], a widely used algorithm that selects shards by estimating a query specific distribution of relevant documents across shards. To this end, a *centralized sample index*, CS, is created, one that combines samples from every shard R. For each query, a retrieval from the central sample index is performed and the top N documents are assumed to be relevant. If n_R is the number of documents in N that are mapped to shard R then a score s_R for each R is computed as $s_R = n_R * w_R$, where the shard weight w_R is the ratio of size of the shard $|R|$ and the size of its sample. The shard scores s_R are then normalized to obtain a valid probability distribution which is used to rank the shards.

In this work, we used a variation of ReDDE, which produced better results in preliminary experiments. Rather than weight each retrieved sampled document equally, we use the document score assigned by the retrieval algorithm to weight the document.

Recall that selectively searching the shards of a large collection is a cooperative task in which global statistics are readily available, thus the document scores computed by each shard are comparable. As a result, merging the document rankings generated by searching the top ranked shards is straightforward.

4.1 Dependence Model Queries

Modeling dependencies among the query terms has been shown to improve adhoc retrieval performance [13]. We investigate whether this holds for selective search as well. The query representation proposed by Metzler and Croft [13] expresses term dependence using proximity constraints constructed from the query terms. We use the full-dependence model where each query term is assumed to be dependent on all the other query terms.

5. DATASETS

Three large datasets were used in this work: Gov2, the CategoryB portion of ClueWeb09 (Clue-CatB) and the English portion of ClueWeb09 (Clue-CatA-Eng). The summary statistics of these datasets are given in Table 1.

The Gov2 TREC corpus [8] consists of 25 million documents from the US government domains, such as .gov and .us, and also from government related websites, such as, www.ncgov.com and www.youroklahoma.com [1]. TREC topics 701-850 were used for evaluation with this dataset. The statistics for these queries are provided in the Table 1.

The ClueWeb09 is a newer dataset that consists of 1 billion web pages crawled in January and February 2009. Out of the 10 languages present in the dataset we use the English portion in this work. The Clue-CatB dataset consists of the first 50 million English pages and the Clue-CatA-Eng consists of all the English pages in the dataset (over 500 million). For evaluation with both Clue-CatB and Clue-CatA-Eng datasets we use the 50 queries that were used in the Web track at TREC 2009.

6. EXPERIMENTAL METHODOLOGY

The three datasets were converted to Indri[2] indexes after stoplisting and stemming with the Krovetz stemmer.

6.1 Sample size and OOV terms

We begin by addressing the issue of OOV terms described in Section 3.4. Figures 1 and 2 (x-axis in log domain) demonstrate that the average percentage of OOV terms per document is low even for small sample sizes. The Clue-CatA-Eng dataset shows trend very similar to those of Clue-CatB dataset. Note that the drop in the average values isn't linear in the sample size; as more documents are seen, the percentage of unseen terms does not decrease proportion-

[1]http://www.mccurley.org/trec/
[2]http://www.lemurproject.org/indri/

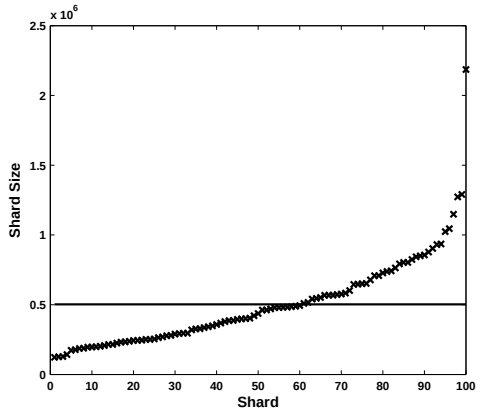

Figure 3: Distribution of shard sizes for the topic-based shards of the Clue-CatB dataset.

ately. Heaps' law [9] offers an explanation for this trend – when examining a corpus, the rate at which vocabulary is discovered tapers off as the examination continues. Thus after a certain point increasing the sample size has little effect on the percentage of OOV terms per document.

We leverage these observations to make our experimental methodology efficient. For Gov2 and Clue-CatB datasets we sample 1% (250K and 500K documents) and for Clue-CatA-Eng dataset we sample 0.1% (500K documents) of the entire collection using uniform sampling. These samples are used by K-means for defining the shards and the remaining documents in the collection are projected onto these shards, as described in Section 3.3.

6.2 Setup

The number of shards for each of the three datasets was chosen such that each shard would contain about 0.5M documents. The Gov2 dataset which contains 25M documents was partitioned into 50 shards, Clue-CatB (50M documents) was partitioned 100 shards and Clue-CatA-Eng (500M documents) was organized into 1000 shards using each of the document allocation policies.

A language modeling and inference network based retrieval model, Indri [12], was used for our experiments. Document retrieval was performed using the simple bag-of-words query representation and also with the full-dependence model query representation. The Indri query language, which supports structured queries, was used for the dependence model queries. For each query the set of shards was ranked using the variant of ReDDE algorithm described in Section 4 and the top T shards were searched to generate the merged ranked list of documents.

The precision at rank 10, 20 and 30 metrics (P@10, P@20, and P@30) were used to compare the search accuracy of exhaustive search with that of selective search. Significance testing of these results was performed using the paired t-test. We also use search cost to compare selective search and exhaustive search. Earlier work has typically used the number of shards or resources that were searched for a query as the search cost. However, when shards are of varying sizes, as is often the case, it is more appropriate to re-define search cost as the percentage of documents in the collections that were searched for the query, as we do in this work. For example, we see that the distribution of shard sizes for the topical shards of Clue-CatB dataset (Figure 3) is far from uniform. The Gov2 and Clue-CatA-Eng datasets exhibit similar non-uniform distributions for topical shards. These non-uniform distribution are an artifact of the topical distribution of the documents in the collections and thus unlikely to be uniform for most collections. However, the shard size distributions obtained with the random and source-based policies are uniform due to the definitions of these policies. In such a situation the new definition of search cost enables a fair comparison across these policies. Thus for selective search the cost depends on the number of shards that were searched and the fraction of documents that were present in these shards. For exhaustive search the cost is 100%.

Table 2: Selective search on Gov2 with bag-of-words query. ▼ denotes significantly worse P@r than exhaustive search ($p < 0.05$).
Exhaustive search: P@10=0.53, P@20=0.50, P@30=0.48, Cost=100%

	P@10	P@10		P@20		P@30	
	Rnd	Src	Top	Src	Top	Src	Top
1 Shrd	▼0.21	▼0.32	▼0.42	▼0.26	▼0.36	▼0.23	▼0.34
Cst(%)	2.0	2.0	3.5	2.0	3.5	2.0	3.5
3 Shrds	▼0.33	▼0.47	0.51	▼0.42	▼0.48	▼0.38	▼0.45
Cst (%)	6.0	6.0	10.3	6.0	10.3	6.0	10.3
5 Shrds	▼0.38	0.52	0.53	▼0.47	0.50	▼0.44	0.47
Cst (%)	10.0	10.0	16.6	10.0	16.6	10.0	16.6
10 Shrds	▼0.44	0.53	0.53	0.49	0.50	▼0.47	0.48
Cst (%)	20.0	20.0	29.9	20.0	29.9	20.0	29.9
15 Shrds	▼0.46	0.53	0.53	0.49	0.50	0.47	0.48
Cst (%)	30.0	30.0	39.6	30.0	39.6	30.0	39.6

7. RESULTS AND DISCUSSION

The selective search results for the Gov2 dataset with bag-of-words query representation and the three allocation policies are provided in Table 2. For P@10, selective search using topic-based shards and source-based shards, both provide search accuracy that is statistically indistinguishable from that of exhaustive search when the average search cost is about 10% of that of exhaustive search. However, for the higher document rank cutoffs (P@20 and P@30), the topical shards provide an edge over the source-based shards. In the case of the topical shards it would be sufficient to search the top 5 shards to achieve search accuracy that is statistically indistinguishable from that of exhaustive search for all the three metrics and the search cost would be about 17%, while for source-based one would have to search the top 15 shards incurring search cost of 30% to reach the same goal.

As expected, the random allocation policy does much worse than exhaustive search, even when searching 30% of the collection. The results for P@20 and P@30 with random allocation shards are equally or more worse and do not provide any further insights. We thus report only P@10 results for this allocation policy to conserve space.

Table 3 provides selective search results for the Gov2 dataset with dependence model queries. As observed by Metzler and Croft in [13], the dependence model queries lead to better search performance than bag-of-words queries – an improvement of 10% is obtained for exhaustive search

Table 3: Selective search on Gov2 with dependence model query. ▼ denotes significantly worse P@r than exhaustive search ($p < 0.05$).
Exhaustive search: P@10=0.58, P@20=0.54, P@30=0.52, Cost=100%

	P@10	P@10		P@20		P@30	
	Rnd	Src	Top	Src	Top	Src	Top
1 Shrd	▼0.22	▼0.38	▼0.44	▼0.31	▼0.39	▼0.27	▼0.36
Cst(%)	2.0	2.0	3.5	2.0	3.5	2.0	3.5
3 Shrds	▼0.36	▼0.52	0.56	▼0.45	▼0.52	▼0.41	▼0.50
Cst (%)	6.0	6.0	10.4	6.0	10.4	6.0	10.4
5 Shrds	▼0.42	▼0.55	0.58	▼0.50	0.54	▼0.47	0.51
Cst (%)	10.0	10.0	16.4	10.0	16.4	10.0	16.4
10 Shrds	▼0.48	0.57	0.59	0.53	0.54	▼0.50	0.52
Cst (%)	20.0	20.0	30.0	20.0	30.0	20.0	30.0
15 Shrds	▼0.50	0.58	0.58	0.54	0.54	0.51	0.52
Cst (%)	30.0	30.0	39.7	30.0	39.7	30.0	39.7

Table 4: Selective search on Clue-CatB with dependence model query. ▼ denotes significantly worse P@r than exhaustive search ($p < 0.05$).
Exhaustive search: P@10=0.30, P@20=0.29, P@30=0.29, Cost=100%

	P@10	P@10		P@20		P@30	
	Rnd	Src	Top	Src	Top	Src	Top
1 Shrd	▼0.09	▼0.16	0.27	▼0.12	0.27	▼0.09	0.24
Cst (%)	1.0	1.0	1.3	1.0	1.3	1.0	1.3
3 Shrds	▼0.16	0.23	0.31	▼0.17	0.32	▼0.14	0.30
Cst (%)	3.0	3.0	4.1	3.0	4.1	3.0	4.1
5 Shrds	▼0.18	0.25	0.30	▼0.18	0.31	▼0.16	0.31
Cst (%)	5.0	5.0	6.6	5.0	6.6	5.0	6.6
10 Shrds	0.24	0.28	0.30	▼0.24	0.30	▼0.20	0.30
Cst (%)	10.0	9.8	13.2	9.8	13.2	9.8	13.2
15 Shrds	▼0.23	0.31	0.30	0.27	0.30	▼0.23	0.30
Cst (%)	15.0	15.0	19.3	15.0	19.3	15.0	19.3

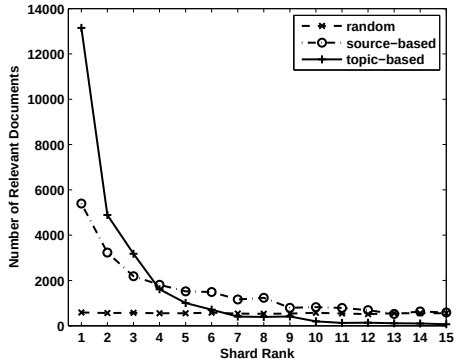

Figure 4: Distribution of relevant documents across top shards for the three allocation policies for the Gov2 dataset. Total number of relevant documents: 26917

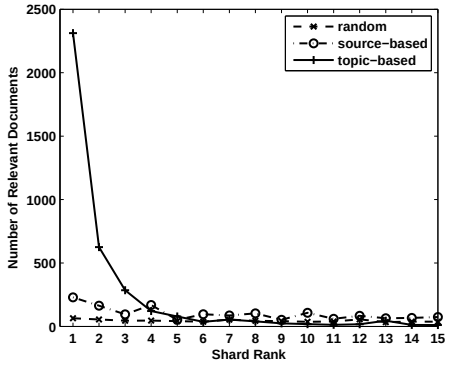

Figure 5: Distribution of relevant documents across top shards for the three allocation policies for the Clue-CatB dataset. Total number of relevant documents: 4002

and for some of the selective search settings as well. The topical shards exhibit more or less the same trends as with bag-of-words queries. For source-based however it seems to be harder to reach the higher complete search performance – 20% of the collection has to be searched to obtain comparable P@10 performance while for bag-of-words it was 10%. In the interest of space we report only dependence model results henceforth, due to their higher accuracy.

Figure 4 provides another perspective for comparing the allocation policies for the Gov2 dataset. The distribution of relevant documents using the three allocation policies demonstrates that the topical shards exhibit maximum skew in the distribution while random allocation leads to nearly uniform distribution of relevant documents across shards. 89% of the total number of relevant documents in the collection are concentrated in the top 5 topical shards. Note that these are not the same 5 shards across all the queries; for each query this would be a different set of 5 shards. The best topic-based shards contain more than twice as many relevant documents as the best source-based shards. In short, K-means does a much better job of concentrating relevant documents into small number of shards than a source-based document allocation policy.

Results for the Clue-CatB dataset and the Clue-CatA-Eng datasets are provided in Tables 4 and 5 respectively. For both these datasets, selective search using topical shards leads to reduction in the search cost by orders of magnitude as compared to exhaustive search. Selective search performance becomes statistically indistinguishable from that of exhaustive search by searching just the top ranked shard which is less than 1.5% and 0.5% of the documents for Clue-CatB and Clue-CatA-Eng, respectively. For source-based shards we see similar trends for Clue-CatB dataset as were seen for the Gov2 dataset – the complete search performance is reached relatively early for P@10 but for P@20 and P@30 the search cost is much higher. For Clue-CatA-Eng dataset the search cost for selective search to become comparable to exhaustive search is similar across the three metrics. Figures 5 and 6 show similar trends in the distribution of relevant documents as was exhibited by the Gov2 shards. 85% and 78% of the relevant documents are concentrated in the top 5 topical shards for the Clue-CatB and the Clue-CatA-Eng datasets, respectively.

For the Clue-CatA-Eng dataset using the single best shard for selective retrieval gives an average-case performance that is 29% higher than the exhaustive search performance for the

Table 5: Selective search on Clue-CatA-Eng with dependence model query. ▼ denotes significantly worse P@r than exhaustive search and ▲ denotes significantly better P@r than exhaustive search ($p < 0.05$). Exhaustive search: P@10=0.14, P@20=0.14, P@30=0.14, Cost=100%

	P@10	P@10		P@20		P@30	
	Rnd	Src	Top	Src	Top	Src	Top
1 Shrd	▼0.03	▼0.06	0.18	▼0.05	0.16	▼0.04	0.15
Cst (%)	0.1	0.1	0.2	0.1	0.2	0.1	0.2
3 Shrds	▼0.05	0.11	0.17	▼0.08	0.17	▼0.07	0.17
Cst (%)	0.3	0.3	0.6	0.3	0.6	0.3	0.6
5 Shrds	▼0.07	0.14	0.17	0.11	0.16	0.09	0.16
Cst (%)	0.5	0.5	1.0	0.5	1.0	0.5	1.0
10 Shrds	▼0.08	0.16	▲0.16	0.12	▲0.17	0.10	▲0.17
Cst (%)	1.0	1.0	2.6	1.0	2.6	1.0	2.6
15 Shrds	▼0.10	0.16	▲0.15	0.13	▲0.15	0.11	▲0.16
Cst (%)	1.5	1.5	4.1	1.5	4.1	1.5	4.1

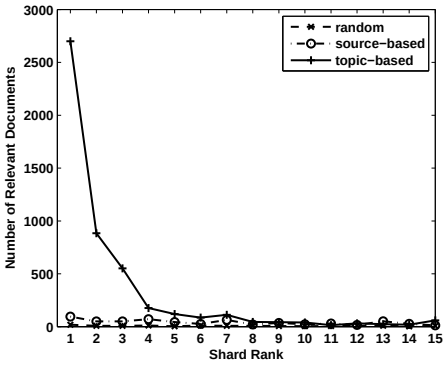

Figure 6: Distribution of relevant documents across top shards for the three allocation policies for the Clue-CatA-Eng dataset. Total number of relevant documents: 5684

P@10 metric. However, this improvement is not statistically significant. The smaller improvement of 7% obtained with top 15 shards is significant, however, at a search cost of 4%. A query-level analysis reveals that for some queries using just the topmost shards provides large improvements in performance but hurts a few other queries. Searching the top 15 shards provides small improvements for many queries and does not hurt any query. This pattern points towards an interesting future direction of estimating optimal shard-cutoffs that are specific to each query.

Random allocation leads to poor selective search performance across all the three datasets. It is not uncommon for large-scale search engines to adopt this allocation policy for parallelization of the search process [4]. These results underscore the necessity of performing exhaustive search when using random allocation which is what these search systems employ. Selective search and the search cost benefits provided by it are difficult to exploit using this allocation policy.

The selective search performance for the Clue datasets becomes comparable to that of exhaustive search much earlier in terms of shard cutoff and search cost than that for the Gov2 dataset. This could be an artifact of the differences in the topical diversity of the datasets. The ClueWeb-09 dataset was intended to be representative of the 'useful' (high PageRank) part of the Web. The Gov2 TREC dataset was however restricted to a small and relatively focused subset of the Web (the .gov and other government agency domains in the USA). As a result, the topic-based shards for the Clue datasets are topically more diverse than those for the Gov2 dataset. This could have an effect of concentrating similar documents in fewer shards for Clue datasets. Thus searching the top ranked shard is sufficient to retrieve most of the relevant documents. The topical diversity could also help in reducing the errors during shard ranking. The number of shards used to partition each of these collections might also be influencing the topical diversity of the shards. This analysis is one of the future directions for this work.

The Clue datasets and the Gov2 dataset are also different in terms of the level of noise that is present in these datasets. Clue datasets have high percentage of noise while Gov2 does not. This could also be one of the reasons why selective search is able to provide a significant improvement over exhaustive search for the Clue datasets. Selective searching of shards provides a natural way to eliminate noise from the search space which could improve the search accuracy by reducing the false positives.

Recall that the samples that were used to define the K-means clusters were a very small subset of the dataset, 1% or 0.1% of the collection. These results demonstrate that an exact clustering solution that uses the entire collection is not necessary for selective search to perform at par with the exhaustive search. An efficient approximation to topic-based techniques can partition large collection effectively and facilitate selective search.

These results reveal that each of the document allocation policies, more or less, converges to the exhaustive search performance, however, at very different rates. Topic-based converges the fastest and random is the slowest.

7.1 Stability analysis

The results in the previous section demonstrate that the *average-case* performance of selective search is comparable to that of exhaustive search for the source-based and topic-based policies at different search costs. In this section we study the *stability* of selective search – we compare the source-based and topic-based allocation policies in terms of their capability to replicate exhaustive search accuracy for queries of different potentials.

Specifically, the queries are categorized into three types using the P@10 value obtained with exhaustive search – poorly performing ($P@10 <= 0.2$), moderate ($0.2 < P@10 <= 0.6$) and good ($P@10 > 0.6$). Figures 7 through 11 and Figure 13 plot the stability results for the three datasets organized using source-based and topic-based policies. The x-axis specifies the percentage of queries that degraded, did as well and improved over exhaustive search. The queries that degraded and the one that improved are further divided using the above query categories. The y-axis specifies the average percentage difference in P@10 with selective search and P@10 with exhaustive search.

For the Gov2 dataset, selective search recreates exhaustive search accuracy for 39% of the queries when using source-based shards. With the topic-based shards, selective search replicates exhaustive search accuracy for twice as many queries (82%). 27% of the queries improve with se-

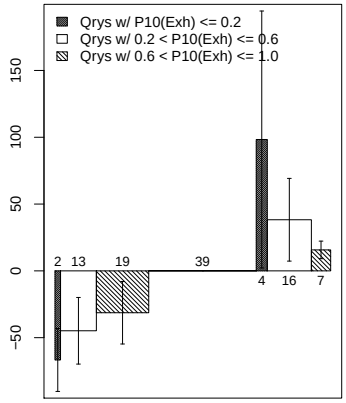

Figure 7: Stability analysis for Gov2 dataset with source-based policy. 5 shards selected.

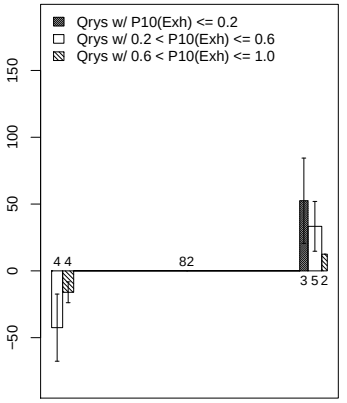

Figure 9: Stability analysis for Gov2 dataset with topic-based policy. 5 shards selected.

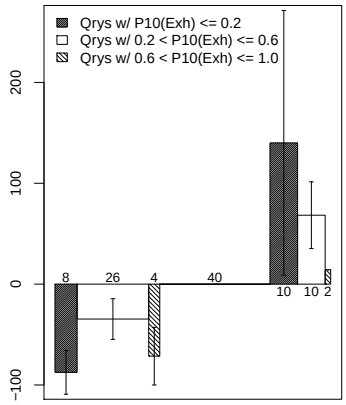

Figure 8: Stability analysis for Clue-CatB dataset with source-based policy. 10 shards selected.

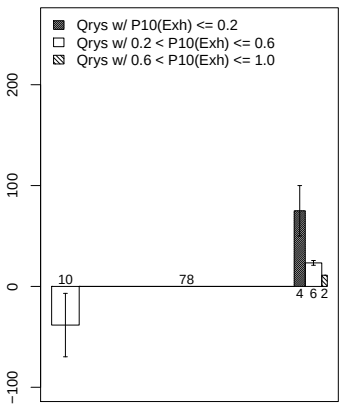

Figure 10: Stability analysis for Clue-CatB dataset with topic-based policy. 10 shards selected.

lective search on source-based shards for the Gov2 dataset, however, 34% of the queries also degrade indicating a high-variance in the query-level performance and thus an unstable setup. The topic-based shards provide a more stable performance. Similar trends are seen for the Clue datasets.

For all the three datasets the source-based shards degrade queries that were already poorly performing even with exhaustive search while the topical shards do not degrade any of these queries except for a small percentage of queries (2%) with the Clue-CatA-Eng dataset. 90% or more queries either recreate or outperform exhaustive search accuracy with selective search on topic-based shards across the board.

We acknowledge that the evaluation query sets for the Clue datasets are relatively small at 50 queries. In the following section, we thus compare performance of exhaustive search with selective search on source-based and topic-based shards using 1000 additional queries.

7.2 Overlap-based evaluation

The *overlap*-based evaluation assumes that the top n documents retrieved by exhaustive search are relevant and computes the precision-at-rank-n ($P@n_{overlap}$) for the selective search results using the n "relevant" documents. One might also say that the overlap evaluation assumes that the goal of selective search is to *replicate* (but not improve upon) exhaustive search. Overlap analysis provides a method for comparing the two sharding approaches that does not require human judgments. We measure Precision at rank 10 ($P@10_{overlap}$) for the analysis presented in this section.

For each of the Clue datasets we sampled 1000 queries from the TREC 2009 Million Query Track[3] and for the Gov2 dataset we sampled 1000 queries from the AOL query log.Exhaustive search was performed for each of these sets of 1000 queries to obtain the 10 "relevant" documents for each query. Selective search using both the source-based shards and the topical shards was performed. Top 10 shards were searched in both the setups and $P@10_{overlap}$ was computed using the top 10 "relevant" documents.

The histograms of $P@10_{overlap}$ values for the 1000 queries using the source-based and topic-based shards are provided in Figures 12, 14 and 15 for the three datasets. The topic-based shards lead to precision of 1.0 for far more queries than does source-based.

We do not put much faith in the exact values obtained

[3]http://ir.cis.udel.edu/million/

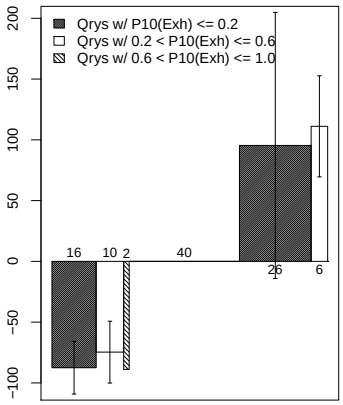

Figure 11: Stability analysis for Clue-CatA-Eng dataset with source-based policy. 10 shards selected.

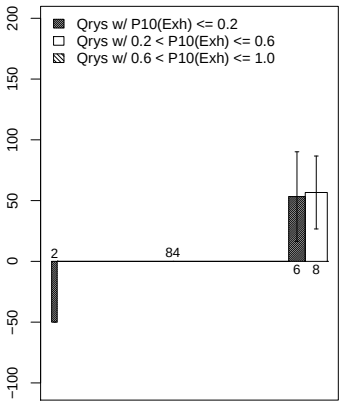

Figure 13: Stability analysis for Clue-CatA-Eng dataset and topic-based policy. 10 shards selected.

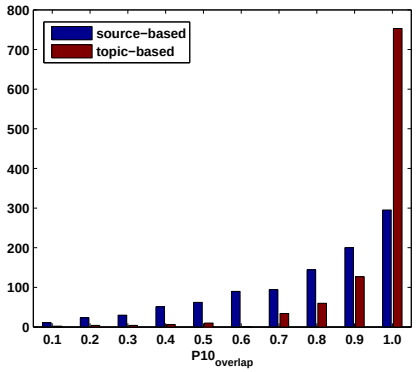

Figure 12: Overlap-based evaluation for Gov2 dataset. 5 shards selected.

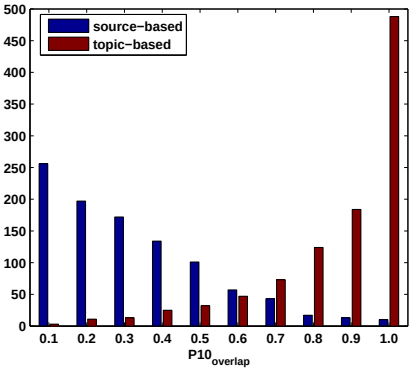

Figure 14: Overlap-based evaluation for Clue-CatB dataset. 10 shards selected.

using the overlap-based measure since in reality it is rarely the case that all the top n documents retrieved by exhaustive search are truly relevant. However, these results are reliable indicators of the patterns, such as, the topic-based shards provide a better setup for selective search than source-based. This seems to hold true for datasets with topically diverse (Clue datasets) as well as for datasets with relatively focused documents (Gov2 dataset).

8. CONCLUSIONS

This work demonstrated that exhaustive search of document collection is not always necessary to obtain competitive search accuracy. Large datasets can be partitioned into topic-based distributed indexes or shards, and then searched selectively. An important step in this process is the allocation of documents to various shards. We investigated three types of document allocation policies: random, source-based and topic-based (using K-means clustering).

Experimental results on three large datasets demonstrated that selective search of topical shards cuts the search cost to less than 1/5th of that of the exhaustive search with no loss of accuracy, on average. A stability analysis investigated the effectiveness of different methods at minimizing the number of queries harmed by selective search. It revealed that 90% or more queries did as well or improved over the exhaustive search accuracy when searching a subset of topical shards.

The topic-based techniques studied in this work have two useful properties – scalability and generality. Scalability is achieved by using a sampling-based approximation of the topic-based allocation technique that can efficiently partition a large collection into topical shards. Our experiments show that even relatively small samples provide good coverage and statistics of corpus vocabulary. Generality is provided by the clustering method used to define topics, because it does not require any specific resource such as training data, query logs, or predefined categories. Existing techniques such as caching that make use of resources like query-logs and click-through data to reduce search cost, can be used in combination with the techniques studied in this paper to further lower the search cost.

To the best of our knowledge, this work is the first to perform a thorough comparison of central index search performance and selective search performance at different levels of search cost using large-scale datasets. We establish that an efficient document allocation technique can be effective at reducing the search cost without hurting the accuracy.

The research presented here used static thresholds to determine how many shards to search for each query. A static threshold was effective for about 90% of the queries. An in-

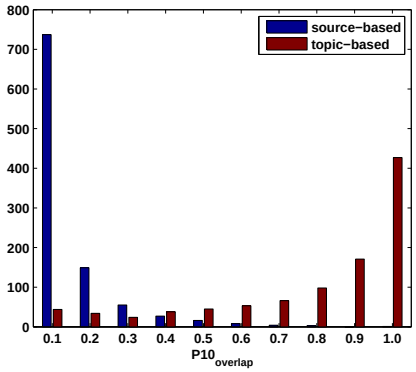

Figure 15: Overlap-based evaluation for Clue-CatA-Eng dataset. 10 shards selected.

teresting question for future research is whether a dynamic threshold (dynamic variation of search costs) might further reduce the number of queries harmed by selective search.

9. ACKNOWLEDGMENTS

This work was in part supported by the NSF grants IIS-0841275 and IIS-0916553. Any opinions, findings, conclusions and recommendations expressed in this paper are the authors' and do not necessarily reflect those of the sponsors.

10. REFERENCES

[1] J. Arguello, J. Callan, and F. Diaz. Classification-based resource selection. In *Proceeding of the 18th ACM conference on Information and knowledge management*, pages 1277–1286, New York, NY, USA, 2009. ACM.

[2] J. Arguello, F. Diaz, J. Callan, and J.-F. Crespo. Sources of evidence for vertical selection. In *Proceedings of the ACM SIGIR conference on Research and development in information retrieval*, pages 315–322, New York, NY, USA, 2009. ACM.

[3] R. Baeza-Yates, V. Murdock, and C. Hauff. Efficiency trade-offs in two-tier web search systems. In *Proceedings of the ACM SIGIR conference on Research and development in information retrieval*, pages 163–170, Boston, MA, USA, 2009. ACM.

[4] L. A. Barroso, J. Dean, and U. Hölzle. Web search for a planet: The google cluster architecture. *IEEE Micro*, 23(2):22–28, 2003.

[5] D. Blei, A. Ng, and M. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3:993–1022, January 2003.

[6] J. Callan. Distributed information retrieval. In *Advances in Information Retrieval*, pages 127–150. Kluwer Academic Publishers, 2000.

[7] J. P. Callan, Z. Lu, and W. B. Croft. Searching distributed collections with inference networks. In *Proceedings of the ACM SIGIR conference on Research and development in information retrieval*, pages 21–28, New York, NY, USA, 1995. ACM.

[8] C. Clarke, N. Craswell, and I. Soboroff. Overview of the TREC 2004 Terabyte track. In *Proceedings of the 2004 Text Retrieval Conference*, 2004.

[9] J. Heaps. *Information Retrieval – Computational and Theoretical Aspects*. Academic Press Inc., New York, NY, 1978.

[10] L. S. Larkey, M. E. Connell, and J. Callan. Collection selection and results merging with topically organized U.S. patents and TREC data. In *Proceedings of the conference on Information and knowledge management*, pages 282–289, New York, NY, USA, 2000. ACM.

[11] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297. University of California Press, 1967.

[12] D. Metzler and W. B. Croft. Combining the language model and inference network approaches to retrieval. *Information Processing and Management*, 40(5):735–750, 2004.

[13] D. Metzler and W. B. Croft. A markov random field model for term dependencies. In *Proceedings of the ACM SIGIR conference on Research and development in information retrieval*, pages 472–479, New York, NY, USA, 2005. ACM.

[14] P. Ogilvie and J. Callan. Experiments using the lemur toolkit. In *Proceedings of the 2001 Text Retrieval Conference*, pages 103–108, 2001.

[15] D. Puppin, F. Silvestri, and D. Laforenza. Query-driven document partitioning and collection selection. In *Proceedings of the 1st international conference on Scalable information systems*, page 34, New York, NY, USA, 2006. ACM.

[16] D. Puppin, F. Silvestri, R. Perego, and R. Baeza-Yates. Load-balancing and caching for collection selection architectures. In *Proceedings of the 2nd international conference on Scalable information systems*, pages 1–10, Suzhou, China, 2007. ICST.

[17] K. M. Risvik, Y. Aasheim, and M. Lidal. Multi-tier architecture for web search engines. *Web Congress, Latin American*, 0:132, 2003.

[18] L. Si and J. Callan. Relevant document distribution estimation method for resource selection. In *Proceedings of the ACM SIGIR conference on Research and development in information retrieval*, pages 298–305, New York, NY, USA, 2003. ACM.

[19] P. Thomas and M. Shokouhi. Sushi: Scoring scaled samples for server selection. In *Proceedings of the ACM SIGIR conference on Research and development in information retrieval*, pages 419–426, New York, NY, USA, 2009. ACM.

[20] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, 1979.

[21] J. Xu and W. B. Croft. Cluster-based language models for distributed retrieval. In *Proceedings of the ACM SIGIR conference on Research and development in information retrieval*, pages 254–261, New York, NY, USA, 1999. ACM.

[22] C. Zhai and J. Lafferty. A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems*, 22(2):179–214, 2004.