Topic-based Index Partitions for Efficient and Effective Selective Search

Anagha Kulkarni and Jamie Callan
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
5000 Forbes Ave, 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. This paper investigates document allocation policies that permit searching only a few shards for each query (*selective search*) without sacrificing search quality. Three types of allocation policies (random, source-based and topic-based) are studied. K-means clustering is used to create topic-based shards. We manage the computational cost of applying these techniques to large datasets by defining topics on a subset of the collection. Experiments with three large collections demonstrate that selective search using topic-based shards reduces search costs by at least an order of magnitude without reducing search accuracy.

Categories and Subject Descriptors

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

Keywords

selective searching, federated search, document clustering

1. INTRODUCTION

Traditionally, searching a collection of documents was a serial task accomplished using a single central index. However, as the document collections increased in size, it became necessary and a common practice to partition collections into multiple disjoint indexes (*shards*) [2, 1]. These distributed indexes facilitate parallelization of search which in turn brings down the query processing time. However, even in this architecture the cost associated with searching large-scale collections is high. For organizations with modest resources this becomes a challenge and potentially limits the scale of the collections that they can experiment with.

LSDS-IR Workshop, July 2010. Geneva, Switzerland.

Our goal is to organize large collections into shards such that the shards facilitate a search setup where only a subset of the shards are searched for any query (*selective search*) and yet provide a performance that is at par with that provided by exhaustive search. The amount of work required per query is naturally much lower in the selective search setup and thus it does not necessitate availability of large computing clusters to work with large document collections.

We experiment with three document allocation policies random, source-based and topic-based to partition the collections into shards (Section 3). One of the main challenges that we tackle in this work is to scale the organization policies to be able to process large collections. Some of the above policies are naturally efficient but for others we propose an approximation technique that is efficient and can parallelize the partitioning process. We also establish that the approximation would not lead to any significant loss in effectiveness. The other contribution of this work is the introduction of a simple yet more accurate metric for measuring the search cost incurred for each query (Section 6.2).

2. RELATED WORK

There have been few other studies that have looked at partitioning of collections into shards. Xu and Croft [16] used a two-pass K-means clustering algorithm and a KL-divergence distance metric to organize a collection into 100 topical clusters. They also experiment with source-based organization and demonstrate that selective search performed as well as exhaustive search, and much better than a source-based organization. The datasets used in this work are small and thus it not clear whether the document organization algorithms employed in this work would scale and be effective for large-scale datasets such as the ones used in our work. Secondly, it has been a common practice in previous work to compute search cost by looking at the number of shards searched for a query which is what is used by Xu and Croft. However, in most setups the shards are of non-uniform sizes and thus this formulation of search cost does not enable an accurate analysis of the trade-off between search cost and accuracy. We remedy this by factoring in the individual shard sizes into the search cost formulation.

Larkey et al. [7] 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 was more effective using the topical organization than the chronological organization.

Puppin et al. [12] used query logs to organize a 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 16 *document clusters*. Documents that could not be clustered because they were not retrieved by any query (50% of the dataset) were put in a 17th (fallback) cluster. Selective search using shards defined by these clusters was found to be more effective than selective search using shards that were defined randomly. The number of shards is relatively very small for a large dataset and the distribution of documents across the shards using this approach is skewed. The inability of the algorithm to partition documents that have not appeared in the query log make this technique's performance highly dependent on the query-log used for the partitioning.

Once the collection has been organized into shards, deciding which index shards to search from the given set of shards is a type of *resource selection* problem [3]. In prior research [4, 14, 13] , 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.

3. DOCUMENT ALLOCATION POLICIES

Our goal is to investigate document allocation policies that are effective, scalable, and applicable in both research and commercial environments. Although we recognize the considerable value of query logs and well-defined categories, they are not available in all environments, thus our research assume access only to the document contents to develop the allocation techniques. This work studies random, source-based, and topic-based allocation policies.

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, but it was a baseline in prior research [12]. Our experimental results show that for some of the datasets random allocation is more effective than one might expect.

3.2 Source-based Document Allocation

Our datasets 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). Source-based allocation was used as a baseline in prior research [16].

3.3 Topic-based Document Allocation

The *Cluster Hypothesis* states that *closely associated documents tend to be relevant to the same request* [15]. Thus if the collection is organized such that each shard contains a similar set of documents, then it is likely that the relevant documents for any given query will be concentrated in just a few shards. Cluster-based and category-based document allocation policies were effective in prior research [16, 12, 7].

We adapt K-means clustering [8] such that it would scale to large collections and thus provide an efficient approach to topical sharding of datasets.

Typically, a clustering algorithm is applied to the entire dataset in order to generate clusters. Although the computational complexity of the K-means algorithm [8] is only 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. The remaining documents in the collection ($M - S$) are then projected onto the space defined by the K clusters. Note that the process of assigning the remaining documents in the collection to the clusters is parallelizable. Using this methodology large collections can be efficiently partitioned into shards.

We use the negative Kullback-Liebler divergence (Equation 1) to compute the similarity between the unigram language model of a document D ($p_d(w)$), and that of a cluster centroid C^i ($p^i_c(w)$). (Please refer to [11] for the derivation.) 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 [17], we estimate $p_d(w)$ using MLE with Jelinek-Mercer smoothing which gives $p_d(w) = (1 - \lambda)\ c(w, D)/\sum_{w'} c(w', D) + \lambda\ p_B(w)$. The term $p_B(w)$ is the probability of the term w in the background model. The background model is an average of the K centroid models. Note that the background model plays the role of inverse document frequency for the term w.

$$KL(C^i||D) = \sum_{w \in C^i \bigcap D} p^i_c(w)\ log \frac{p_d(w)}{\lambda\ p_B(w)} \qquad (1)$$

We found this version of KL-divergence to be more effective than the variant used by Xu and Croft [16].

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 [14], a widely used algorithm that prioritizes 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.

Selective search of index shards is a cooperative environment where global statistics of each shard are readily available. Thus merging the document rankings generated by searching the top ranked shards is straightforward.

Table 1: Datasets and Query Sets

Dataset	Number of Documents	Number of Words (billion)	Vocabulary Size (million)	Avg Doc Len	Query Set	Avg Qry Len	Avg Number of Rel Docs Per Qry
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)

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 [5] 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 that were crawled between 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

Using a subset instead of the entire collection to learn the clusters reduces the computational cost 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 or topic 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.

Figure 1 (x-axis in log domain) demonstrates that the average percentage of OOV terms per document is low even for small sample sizes. 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 proportionately. Heaps' law [6] 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 0.1% (25K and 50K documents) and for Clue-CatA-Eng dataset we sample 0.01% (50K documents) of the entire collection using uniform sampling. These samples are used by K-means for cluster learning.

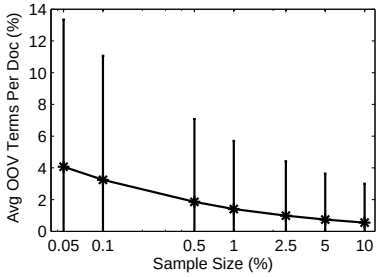

Figure 1: Sample size vs. percentage of OOV terms per document, on average, for the Clue-CatB Dataset.

6.2 Setup

The Gov2 and Clue-CatB datasets were each partitioned into 100 shards while the Clue-CatA-Eng dataset was organized into 500 shards using each of the document allocation techniques. The top 10 terms for nine of the 100 topical shards of the Clue-CatB dataset are given in Table 2. These are the terms that explain the majority of the probability mass in the language models for each of these topical clusters. For these nine shards and for most of the other 91 shards a semantically coherent topic emerges from these terms.

A language modeling and inference network based retrieval model, Indri [9], was used for our experiments. Modeling dependencies among the query terms has been shown to improve adhoc retrieval performance [10]. We investigate if this holds for selective search as well. Thus 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 metric (P10) was used to compare the search accuracy of exhaustive search with that of selective search. We define the search cost of a query to be the percentage of documents that were searched. For exhaustive search the cost is 100% while 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.

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

Table 2: Top terms from topical shards of the Clue-CatB dataset.

Topic A	Topic B	Topic C	Topic D	Topic E	Topic F	Topic G	Topic H	Topic I
state	policy	recipe	music	game	law	entertain	price	health
politics	privacy	food	record	play	patent	com	accessory	care
election	information	cook	song	casino	attorney	news	com	center
party	terms	com	album	free	com	sports	size	service
war	service	home	wikipedia	online	legal	advertise	product	school
america	site	new	edit	com	lawyer	home	clothing	child
government	rights	make	rock	puzzle	www	blog	item	home
vote	copyright	make	com	download	california	list	ship	program
new	return	oil	band	poker	home	business	home	educate
president	com	cup	video	arcade	case	search	costume	parent

Table 3: P10 values for selective search on Gov2 with bag-of-words query. ▼ denotes significantly worse P10 than exhaustive search ($p < 0.05$). Exhaustive search: P10=0.530, Cost=100%

	Rand	Source	K-means
1 Shard	▼0.169	▼0.236	0.491
Cost (%)	1.00	1.00	1.24
3 Shards	▼0.302	▼0.419	0.511
Cost (%)	3.00	3.00	3.62
5 Shards	▼0.338	▼0.456	0.520
Cost (%)	5.00	5.00	6.00
10 Shards	▼0.384	▼0.492	0.533
Cost (%)	10.00	10.00	11.38
15 Shards	▼0.411	0.507	0.530
Cost (%)	15.00	15.00	15.40

Table 4: P10 values for selective search on Gov2 with dependence model query. ▼ denotes significantly worse P10 than exhaustive search ($p < 0.05$). Exhaustive search: P10=0.580, Cost=100%

	Rand	Source	K-means
1 Shard	▼0.165	▼0.255	▼0.504
Cost(%)	1.00	1.00	1.26
3 Shards	▼0.304	▼0.443	▼0.552
Cost (%)	3.00	3.00	3.62
5 Shards	▼0.357	▼0.491	0.575
Cost (%)	5.00	5.00	6.00
10 Shards	▼0.419	0.556	0.583
Cost (%)	10.00	10.00	11.38
15 Shards	▼0.442	0.560	0.584
Cost (%)	15.00	15.00	15.63

7. RESULTS AND DISCUSSION

The selective search results for the Gov2 dataset with bag-of-words query representation are provided in Table 3.

Selective search on shards defined by K-means provides search accuracy that is statistically indistinguishable from that of exhaustive search when the search cost is 1.24% of that of exhaustive search. For source-based shards the top 15 shards have to be searched to obtain comparable search accuracy, however, even this leads to an order of magnitude reduction in search cost.

Recall that the samples that were used to define the K-means clusters were quite small, 0.1% and 0.01% of the collection. These results show 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.

Table 4 provides selective search results for the Gov2 dataset with dependence model queries. As observed by Metzler and Croft in [10], the dependence model queries lead to better search performance than bag-of-words queries – an improvement of 10% is obtained for exhaustive search and for many of the selective search settings as well. Selective search proves to be as capable as exhaustive search in leveraging the information about query term dependence. The trends observed in Table 4 are similar to those observed in Table 3 – topic-based shards provide the cheapest setup for obtaining selective search accuracies that are comparable to those of exhaustive search. However, the absolute search cost for the selective search to be statistically indistinguishable from exhaustive search goes up from 1.24% (bag-of-words) to 6%. Nevertheless, the search cost (6%) is still an order of magnitude smaller than the cost for exhaustive search. In the interest of space we report only dependence model results henceforth, due to their higher accuracy.

Results for the Clue-CatB dataset and the Clue-CatA-Eng datasets are provided in Tables 5 and 6. The topic-based technique perform as well as the exhaustive search by searching only the top ranked shard which is less than 2% and 0.5% of the documents for Clue-CatB and Clue-CatA-Eng, respectively. Searching the top 3 shards provides nearly 10% and 30% improvement over exhaustive search for Clue-CatB and Clue-CatA-Eng, respectively, and the latter is found to be statistically significant. To the best of our knowledge these results provide an evidence for the first time that selective search can consistently improve over exhaustive search while searching a small fraction of the collection.

For both the datasets, selective search loses this advantage over the exhaustive search by searching more shards. This indicates that a smaller but tightly focused search space can be better than a larger search space. This also implies that searching a fixed number of shards for each query might not be ideal. This is an interesting topic for future research in selective search. The source-based shards continue to provide a competitive baseline for both the datasets.

A query-level analysis of the effectiveness of different methods at minimizing the number of queries harmed by selective search revealed that 86% or more queries did as well or improved over exhaustive search accuracy when performing selective search using topic-based shards. While for

Table 5: P10 values for selective search on ClueCatB with dependence model query. ▼ denotes significantly worse P10 than exhaustive search and ▲ denotes significantly better P10 than exhaustive search ($p < 0.05$).
Exhaustive search: P10=0.300, Cost=100%

	Rand	Source	K-means
1 Shard	▼0.080	▼0.156	0.302
Cost (%)	1.00	1.00	1.63
3 Shards	▼0.180	0.244	0.330
Cost (%)	3.00	3.00	4.99
5 Shards	▼0.212	0.278	0.314
Cost (%)	5.00	5.00	7.85
10 Shards	0.252	0.304	0.292
Cost (%)	10.00	9.80	14.69
15 Shards	0.254	0.306	0.294
Cost (%)	15.00	15.00	21.84

Table 6: P10 values for selective search on ClueCatA-Eng with dependence model query. ▼ denotes significantly worse P10 than exhaustive search and ▲ denotes significantly better P10 than exhaustive search ($p < 0.05$).
Exhaustive search: P10=0.142, Cost=100%

	Rand	Source	K-means
1 Shard	▼0.024	▼0.056	0.152
Cost (%)	0.20	0.20	0.32
3 Shards	▼0.046	0.112	▲0.182
Cost (%)	0.60	0.60	1.12
5 Shards	▼0.066	0.120	0.174
Cost (%)	1.00	1.00	2.08
10 Shards	▼0.088	0.168	0.160
Cost (%)	2.00	2.01	4.81
15 Shards	0.114	0.174	0.146
Cost (%)	3.00	3.00	7.40

source-based 60% or more queries were found to perform well with selective search.

The selective search performance for the Clue datasets becomes comparable to that of exhaustive search much earlier in terms of shard cutoff than that for the Gov2 dataset. We believe this could be an artifact of the differences in the topical diversity of the datasets – the ClueWeb-09 dataset is much more diverse than the Gov2 dataset. As a result the topical shards of the ClueWeb-09 dataset are more dissimilar to each other 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 and the topically focused shards must also help reduce the errors during shard ranking.

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 is relatively clean. This could 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 some of the noise from the search space which improves the search accuracy by reducing the false positives from the final results.

More generally, 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 converges the slowest.

8. CONCLUSIONS

This work demonstrated that exhaustive search of document collection is not always necessary to obtain competitive search accuracy. To enable this we partitioned the dataset into distributed indexes or shards, and then selectively searched a small subset of these shards. 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.

Empirical results on three large datasets demonstrated that selective search of topic-based shards provides at least an order of magnitude reduction in search costs with no loss of accuracy, on average. 86% or more queries did as well or improved over exhaustive search accuracy when performing selective search using topic-based shards for all the three datasets. Although previous work hasn't reported this number anecdotal results suggest that this is much more stable than prior research. The results also demonstrate for the first time that selective search can consistently improve over exhaustive search while searching only a small fraction of the collection if a good document allocation policy has been employed to create the shards.

The topic-based document allocation technique studied in this work has two useful properties – scalability and generality. Scalability is achieved by using sampling-based approximation of K-means clustering to 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 K-means clustering used to define topics, because it does not require any specific resources such as training data, query logs, click-through data, 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.

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] R. Baeza-Yates, V. Murdock, and C. Hauff. Efficiency trade-offs in two-tier web search systems. In *Special Interest Group on Information Retrieval*, pages 163–170, Boston, MA, USA, 2009. ACM.

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

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

[4] J. P. Callan, Z. Lu, and W. B. Croft. Searching distributed collections with inference networks. In

Special Interest Group on Information Retrieval, pages 21–28, New York, NY, USA, 1995. ACM.

[5] C. Clarke, N. Craswell, and I. Soboroff. Overview of the TREC 2004 Terabyte track. In *TREC*, 2004.

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

[7] L. S. Larkey, M. E. Connell, and J. Callan. Collection selection and results merging with topically organized U.S. patents and TREC data. In *Conference on Information and Knowledge Mangement*, pages 282–289, New York, NY, USA, 2000. ACM.

[8] 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.

[9] D. Metzler and W. B. Croft. Combining the language model and inference network approaches to retrieval. *Inf. Process. Manage.*, 40(5):735–750, 2004.

[10] D. Metzler and W. B. Croft. A markov random field model for term dependencies. In *Special Interest Group on Information Retrieval*, pages 472–479, New York, NY, USA, 2005. ACM.

[11] P. Ogilvie and J. Callan. Experiments using the lemur toolkit. In *TREC*, pages 103–108, 2001.

[12] D. Puppin, F. Silvestri, and D. Laforenza. Query-driven document partitioning and collection selection. In *InfoScale*, page 34, New York, NY, USA, 2006. ACM.

[13] M. Shokouhi. Central-rank-based collection selection in uncooperative distributed information retrieval. In *The 29th European Conference on Information Retrieval*, Rome, Italy, 2007.

[14] L. Si and J. Callan. Relevant document distribution estimation method for resource selection. In *Special Interest Group on Information Retrieval*, pages 298–305, New York, NY, USA, 2003. ACM.

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

[16] J. Xu and W. B. Croft. Cluster-based language models for distributed retrieval. In *Special Interest Group on Information Retrieval*, pages 254–261, New York, NY, USA, 1999. ACM.

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