

On the Relationship Between Query Characteristics and IR Functions Retrieval Bias

Shariq Bashir and Andreas Rauber

Institute of Software Technology and Interactive Systems, Vienna University of Technology, Austria.

E-mail: {bashir, rauber}@ifs.tuwien.ac.at

Bias quantification of retrieval functions with the help of document retrievability scores has recently evolved as an important evaluation measure for recall-oriented retrieval applications. While numerous studies have evaluated retrieval bias of retrieval functions, solid validation of its impact on realistic types of queries is still limited. This is due to the lack of well-accepted criteria for query generation for estimating retrievability. Commonly, random queries are used for approximating documents retrievability due to the prohibitively large query space and time involved in processing all queries. Additionally, a cumulative retrievability score of documents over all queries is used for analyzing retrieval functions (retrieval) bias. However, this approach does not consider the difference between different query characteristics (QCs) and their influence on retrieval functions' bias quantification. This article provides an in-depth study of retrievability over different QCs. It analyzes the correlation of lower/higher retrieval bias with different query characteristics. The presence of strong correlation between retrieval bias and query characteristics in experiments indicates the possibility of determining retrieval bias of retrieval functions without processing an exhaustive query set. Experiments are validated on TREC Chemical Retrieval Track consisting of 1.2 million patent documents.

Introduction

The main objective of information retrieval (IR) systems is to maximize effectiveness. In order to do so, IR systems attempt to discriminate between relevant and non-relevant documents with the help of different ranking functions. To measure effectiveness metrics such as Average Precision, Q -measure (Normalized Discounted) Cumulative Gain, Rank-Based Precision, Binary Preference (bref) are commonly used (Sakai, 2008). The main limitation of these measures is that they focus almost exclusively on precision,

i.e., the fact that the (most) relevant documents are returned on top of a ranked list, as this constitutes the primary criterion of interest in most standard IR settings. With evaluation measures such as recall and F_β , aspects of the completeness of the result set are being brought into consideration. While for most standard application domains, the retrieval of a small number of most relevant information items is sufficient, some domains are highly recall oriented such as legal or patent retrieval. In these settings, it is essential that all documents relevant to a specific query are returned (Arampatzis, Kamps, Kooken, & Nussbaum, 2007; Magdy & Jones, 2010). These domains are more concerned with ensuring that everything relevant has been found and often seek to demonstrate that something (e.g., a document that invalidates a new patent application) does not exist. This is different from the prototypical IR task, where a user seeks to find a set of relevant documents for satisfying his information need. This gives rise to a new evaluation measure namely *retrievability, accessibility, or document findability* (Azzopardi & Vinay, 2008), highlighting the two points of view, i.e., retrievability from the retrieval function perspective, and findability from a user's perspective. Retrievability provides an indication of how easily a document can be retrieved using a given retrieval function, whereas findability provides an indication of how easily a document can be found by a user with the given retrieval function. Essentially, retrievability is the ease with which it is possible for any document to be retrieved, and so the retrievability of a document depends upon the document collection and the IR model. A document with high retrievability means that a user has a high probability of finding that document by querying. Conversely, a document with low retrievability in a particular retrieval model is likely to be difficult to find by the user, up to impossible for documents showing a retrievability of 0. Clearly, if a document is difficult or impossible to retrieve, in general, then it will be difficult or impossible to retrieve when relevant. It is the inability to retrieve certain relevant documents that will lead to low recall.

Recently, a number of studies on document corpora have shown that retrieval functions significantly and substantially

Received January 18, 2011; revised March 22, 2011; accepted March 23, 2011

© 2011 ASIS&T • Published online in Wiley Online Library (wileyonlinelibrary.com). DOI: 10.1002/asi.21549

differ in terms of the retrieval bias that they impose on (individual and groups of) documents in a collection. For example, in Azzopardi and Vinay, (2008) it was shown that different best match retrieval functions provide substantially different levels of retrieval bias across the collection, whereas in Bashir and Rauber (2010, 2009b) it was shown that pseudo-relevance feedback with query expansion could be used for increasing the retrievability of documents.

The retrievability estimation framework proposed by Azzopardi and Vinay (2008) is divided into two phases, namely (1) query list generation (if no query list is known in advance) and (2) the processing of these queries on different retrieval functions for analyzing their bias. Among both, query generation is the most fundamental and important phase. Up to now there is no single criterion exists that helps in defining how to generate queries for retrievability estimation. In most studies, a queries subset approach is used for approximating documents' retrievability as processing an exhaustive set of queries would be prohibitively expensive. This approach has several shortcomings.

First, in related studies there is no consistent approach used for generating queries subset to analyze retrieval bias. Almost every study uses its own approach for generating a queries subset. For instance, Azzopardi et al. generate this subset by taking only the top 2 million (Azzopardi & Vinay, 2008) or top 1 million (Azzopardi & Bache, 2010) 2-terms queries, that also most occurred in the collection. Bashir and Rauber (2009b, 2010) generate this subset by issuing (max 200) or (max 90) 2-, 3-, 4-terms queries per document with the help of language modeling and documents relatedness concepts. The main drawback of all these studies is that there is no comparison performed on large-scale queries that allow us to determine which approach provides most accurate retrievability ranking of documents similar to exhaustive set of queries (the universe of all possible queries). Specifically, we want to verify how far a queries subset-based approach provides an accurate approximation of retrieval bias if it is compared with exhaustive queries. The other shortcoming of previous studies is that retrieval bias is analyzed without considering any difference between different QCs. Retrievability scores of documents are cumulated over all queries, regardless of whether these are higher or lower quality queries, or potentially retrieving many or few documents. In this article we argue that, similar to other IR tasks where different QCs play a critical part on the effectiveness of retrieval functions (Cronen-Townsend, Zhou, & Croft, 2002), retrievability analysis with different QCs subsets also allows us to better determine access behavior of retrieval functions. Without analyzing retrieval bias with respect to different QCs it is difficult to understand how far the obtained values are representative and useful for a given user setting.

We thus need to identify the relationship between retrieval bias of retrieval functions and different QCs. In IR research, simulation with different QCs always provides an inexpensive avenue for testing, training, and evaluating retrieval algorithms with the ability to precisely control the experimental conditions. For instance, the length (i.e., long

vs. short), style (highly discriminative terms vs. popular terms), quality (noisy query terms, translations, etc.), and number of queries that can be produced for a given topic can be greatly varied to define a specific scenario. This enables selective evaluation of particular query types. By understanding the relationship between retrieval bias of retrieval functions and different QCs, retrieval functions performance can be better understood and help to guide the development (Azzopardi, de Rijke, & Balog, 2007; Jordan, Watters, & Gao, 2006; Tague, Nelson, & Wu, 1981). To address these, we partition the exhaustive set of all queries of 3- or 4-terms length according to their different characteristics, and analyze the relationship of each partition with different levels of retrieval bias. Our experiments show a strong correlation of QCs with different levels of retrieval bias. This correlation indicates that the bias of retrieval functions depends on three characteristics of queries: (a) size of answer set, (b) quality of queries, and (c) query terms frequency in the documents. Large answer sets, low query term frequencies, and low quality of queries increase the retrieval bias. We further test how far query subsets provide accurate retrievability rankings of documents similar to exhaustive queries. Our results suggest that estimates are still good when the retrieval bias is estimated from query subsets rather than from exhaustive queries.

We furthermore analyze the effect of retrievability on different corpus characteristics, considering specifically the homogeneity of a corpus with respect to document length and vocabulary richness. Experiments reveal that retrieval functions show specific bias on the extreme ends of these scales, i.e., for very long/short or vocabulary poor/rich documents. This may lead to the definition of specific strategies for handling retrieval in diverse corpora if equal accessibility of documents is a desired feature.

The remainder of this article is structured as follows. The next section reviews related work on the methods of discovering retrieval functions/search engines bias. First the concept of retrievability measurement is introduced and then a modified score considering a potential bias introduced by the way the queries are generated is presented in the section Retrievability Measurement. The TREC-CRT (TREC Chemical Retrieval Track) benchmark corpus that is used for experiments in this article and retrieval bias results for several configurations of this corpus and different retrieval models are described in the section Experiment Setup. The next section then presents a framework for analyzing the relationship between the retrieval bias of retrieval functions and different QCs. Finally, the final section briefly summarizes the key lessons learned from this study.

Related Work

The bias analysis of retrieval functions has always received high attention in the IR research community (Bar-Ilan, Keenoy, Levene, & Yaari, 2009; Lawrence & Giles, 1999; Mowshowitz & Kawaguchi, 2002; Vaughan & Thelwall, 2004). Web coverage and documents retrieval bias are two well-known measures to analyze a potential search engine

bias. Here we provide an overview of the main works in both areas.

Bias analysis based on Web coverage

Lawrence and Giles (1999) performed a study to analyze the coverage bias of Web search engines. For this purpose, they used six search engines and a large query log from a scientific organization. The queries should have returned the same set of pages for all six engines, with duplicate URLs or pages removed. Their experiments revealed that no single search engine covers more than 57.5% of the estimated full Web. They also showed that some large search engines cover less than 5% of the Web. Finally, the authors concluded that the solution to the problem of search engines not indexing the whole Web is to use meta-search engines or to define goal-driven search engines that have a specific focus, e.g., sports or scientific literature.

Vaughan and Thelwall (2004) performed a study on the coverage of Web pages from 42 countries to discover the index bias of three major search engines. For this purpose they used their own research crawler, and crawled domains from 42 countries. Large numbers of queries were submitted to three search engines and their research crawler. The bias quantification was on the basis of site coverage ratio, and it was computed as the number of pages covered by the search engines divided by the number of pages covered by their research crawler. The main limitation of their study was that it did not consider the constantly changing nature of the Web, as their developed crawler could remain behind the indexes of search engines since the researchers did not have similar number of resources available as major search engines.

Moshowitz and Kawaguchi (2002) undertook a study for discovering bias in 15 major commercial search engines. For generating queries, they used the ACM computing classification system as queries, and the top 30 results for each search engine were recorded. Their large experiments' results confirmed that there was some bias in all search engines. Their proposed bias measurement function involved the numbers of unique domains as a ranked array based on the combination of all Web search results returned by the queries. However, this measurement could itself have introduced bias into the experiments as it is not based on all possible results from the Web but only on the combinations of the Web pages returned for every search engine. Second, their measurement cannot show whether there is a bias against particular results if all of the included search engines are biased against similar results.

Azzopardi and Owen (2009) and Owens (2009) conducted a study on bias analysis of search engines. A major concern of their study was to discover whether search engines unfairly lead users to particular sites over other sites. For this purpose, they discovered the relative news bias of three search engines. They report this relative bias among search engines in the form of political bias and predilection for specific sites. They performed an experiment over 9 weeks, by posing a large number of realistic and currently topical queries to the news

sections of the three search engines and stored the resulting list of URLs. From their different results they showed that there are significant biases toward predilections for certain news sources in all of the engines.

Lauw, Lim, and Wang (2006) found that deviation (controversy) in the evaluation scores of objects in reviewer-object models can also be used for discovering bias. They observed that bias and controversy of reviewers to objects are mutually dependent to each other. This dependency indicates that there will be more bias if there is higher deviation to less controversial object. To identify this controversy and bias they used a reinforcement model. Their approach of discovering bias can also be applied in a Web search setting. In this case the reviewers can be regarded as Web search engines and the objects that they are reviewing (ranking) are Web pages. Based on this approach, search engines will be more biased if they give high ranks to less controversial (ranks) Web pages of other search engines.

All these studies revealed a range of possible biases, for example, if one site has more coverage than another. These studies are usually motivated by the view that search engines may be providing biased content, and these measures are aimed at being regulatory in nature, whether sites in particular geographical locations are favored, or whether search engines are biased given a particular topic. As opposed to coverage-based measures of sites, our work focuses on individual documents' retrievability scores, which can also be used to detect such biases. However, our main objective is to more precisely understand the effect of such biases.

Bias analysis based on documents retrievability

Azzopardi and Vinay (2008) introduced a measure for quantification of a retrieval functions' bias on the basis of findability (accessibility) of individual documents. It measures how likely a document can be found at all by a specific retrieval function. The analysis of the individual retrievability scores of documents is performed using Lorenz curves and Gini-Coefficients. Their experiments with AQUAINT and .GOV datasets reveal that with a TREC-style evaluation a proportion of the documents with very low retrievability scores (sometimes more than 80% of the documents with the higher bias retrieval functions) can be removed without significantly degrading performance. This is because the retrieval functions are unlikely to ever retrieve these documents due to the bias they exhibit over the collection of documents.

Similar to Azzopardi and Vinay's (2008) experiments, Bashir and Rauber (2009a) analyzed retrievability of documents specifically with respect to relevant and irrelevant queries to identify whether highly retrievable documents are really highly retrievable, or whether they are simply more accessible from many irrelevant queries rather than from relevant queries. However, the evaluation is based on a rather limited set of queries. Experiments revealed that 90% of patent documents which are highly retrievable across all types of queries, are not highly retrievable on their relevant query sets.

The effect of query expansion-based approaches for improving document retrievability is thoroughly investigated in Bashir and Rauber (2009b, 2010). These studies concluded that short queries are not efficient for correctly capturing and interpreting the context of a search. Therefore, noisy documents at higher rank position shift the retrievability results to fewer documents, thus creating higher retrieval bias. To overcome this limitation, their approach expands the queries on the basis of pseudo-relevance feedback documents. Furthermore, pseudo-relevance feedback documents are identified using cluster-based (Bashir & Rauber, 2009b) and terms-proximity-based methods (Bashir & Rauber, 2010). Experiments with different collections of patent documents suggest that query expansion with pseudo-relevance feedback can be used as an effective approach for increasing the findability of individual documents and decreasing the retrieval bias.

Another study by Azzopardi and Bache (2010) analyzed the relationship between retrievability- and effectiveness-based measures (Precision, Mean Average Precision). Their results show that the two goals of maximizing access and maximizing performance are quite compatible. They further conclude that reasonably good retrieval performance can be obtained by selecting parameters that maximize retrievability (i.e., when there is the least inequality between documents according to Gini-Coefficient given the retrievability values). Their results support the hypothesis that retrieval functions can be effectively tuned using retrievability-based measure without recourse to relevance judgments, making it an attractive alternative for automatic evaluation.

The main limitation of all the results published so far is that retrieval bias of retrieval functions is approximated without analyzing how far different query subset generation approaches provide accurate approximation of retrieval bias, if their approximated retrieval bias is compared with the retrieval bias that is approximated from an exhaustive query set. In this article, we present a series of experiments on a large-scale document corpus in the same application domain as the previous studies on retrievability. The benchmark corpus of 1.2 million TREC Chemical Retrieval Track (TREC-CRT) patents is used to validate the hypothesis of uneven retrievability in a large corpus (Lupu, Huang, Zhu, & Tait, 2009). Specifically, we want to verify the relationship between retrievability for different QCs, and further the possibility of predicting retrieval bias with the help of fewer queries.

Retrievability Measurement

Given a retrieval function RS , a collection of documents D , and a large set of queries Q . Retrievability measures how far each document $d \in D$ is retrievable within the top- c rank results of all queries $q \in Q$. Retrievability of a document is essentially a cumulative score that is proportional to the number of times the document can be retrieved within that cut-off c over the set Q . A retrieval function is called best performing, if each document d has a similar retrievability score,

i.e., is equally likely to be found. More formally, retrievability $r(d)$ of $d \in D$ can be defined as follows:

$$r(d) = \sum_{q \in Q} f(k_{dq}, c) \quad (1)$$

$f(k_{dq}, c)$ is a generalized utility/cost function, where k_{dq} is the rank of d in the result set of query q , c denotes the maximum rank that a user is willing to proceed down the ranked list. In most studies, the function $f(k_{dq}, c)$ returns a value of 1, if $k_{dq} \leq c$, and 0 otherwise. Retrievability inequality can further be analyzed using the Lorenz curve (Gastwirth, 1972). In economics and the social sciences, the Lorenz curve is used to visualize the inequality of wealth distribution in a population. It is created by first sorting the individuals in the population in ascending order of their wealth and then plotting a cumulative wealth distribution. If the wealth in the population was distributed equally, then we would expect this cumulative distribution to be linear. The extent to which a given distribution deviates from equality is reflected by the amount of skew in the distribution. Azzopardi and Vinay (2008) employ this idea in the context of a population of documents, where their wealth is represented by $r(d)$ and plot the result. The more skewed the plot, the greater, the amount of inequality, or bias within the population. The Gini-Coefficient G is used to summarize the amount of retrieval bias in the Lorenz curve and provides a bird's eye view. It is computed as

$$G = \frac{\sum_{i=1}^n (2i - n - 1)r(d_i)}{(n - 1) \sum_{j=1}^n r(d_j)} \quad (2)$$

where $n = |D|$ is the number of documents in the collection sorted by $r(d)$. If $G = 0$, then no bias is present because all documents are equally retrievable. If $G = 1$, then only one document is retrievable and all other documents have $r(d) = 0$. By comparing the Gini-Coefficients, we can analyze the retrieval bias imposed by the underlying retrieval functions on a given document collection.

The retrievability measure that is defined above cumulates $r(d)$ scores of documents over all queries. Thus, long documents that contain larger vocabulary potentially have a higher number of query combinations possible than short documents in case of exhaustive query generation process. This may favor long documents that are retrievable from only a smaller fraction of all their possible queries than short documents that are potentially retrievable from a larger fraction of their queries. To understand this phenomenon, let us consider the example presented in Table 1 with six documents and their estimated $r(d)$ scores with three different retrieval functions (A,B,C). Doc1, Doc2, and Doc4 are long documents than Doc3, Doc5, and Doc6; therefore, have much larger possible number of query combinations. (For the context of this example, we are assuming only 3-terms queries). From Table1 it can be easily inferred that in terms of percentage of documents retrievable from their all possible queries combinations, retrieval function B is far better than retrieval function A, and retrieval function C

TABLE 1. Retrieval bias representation with $r(d)$ and $\hat{r}(d)$. G refers to Gini-Coefficient value.

Docs.	Unique terms	Total queries	IR function A	IR function B	IR function C
Retrieval bias with $r(d)$					
Doc1	40	9,880	791	5,928	9,880
Doc2	35	6,545	851	3,600	6,545
Doc3	8	56	55	40	56
Doc4	28	3,276	525	2,130	3,276
Doc5	10	120	118	90	120
Doc6	12	220	187	176	220
	Overall Bias		$G = 0.50$	$G = 0.70$	$G = 0.71$
Retrieval bias with $\hat{r}(d)$					
Doc1	40	9,880	0.08	0.60	1
Doc2	35	6,545	0.13	0.55	1
Doc3	8	56	0.98	0.70	1
Doc4	28	3,276	0.16	0.65	1
Doc5	10	120	0.98	0.75	1
Doc6	12	220	0.85	0.80	1
	Overall bias		$G = 0.48$	$G = 0.08$	$G = 0$

is better than retrieval function B. Therefore, after retrieval bias computation, retrieval function C, and retrieval function B should showing a lower bias than retrieval function A. However, using standard retrieval bias analysis—retrieval function A is wrongly showing a lower Gini-Coefficient than retrieval function B, and similarly retrieval function B is wrongly showing a lower Gini-Coefficient than retrieval function C. This happens due to the fact that the difference between vocabulary richness is not considered while computing retrieval bias. To solve this problem, Equation 3 normalizes the cumulative retrievability scores (Normalized Retrievability) of documents by the number of queries they were created from, and thus potentially can retrieve, a particular document, it is defined as:

$$\hat{r}(d) = \frac{\sum_{q \in Q} f(k_{dq}, c)}{|\hat{Q}(d)|} \quad (3)$$

Cumulative $r(d)$ scores of documents are normalized with $\hat{Q}(d)$, the set of queries that can retrieve d when not considering any rank cut-off factor, i.e., those queries that contain at least one term that is also present in d . This accounts for differences in vocabulary richness across different documents. Documents with a large vocabulary size produce many more queries. Such documents are thus theoretically retrievable via a much larger set of queries. The standard $r(d)$ score would thus penalize a retrieval function that provides perfectly balanced retrievability to all documents just because some documents are rather vocabulary-poor and cannot be retrieved by more than the few queries that can be created from their vocabulary. This is where a normalized retrievability score accounting for different vocabulary sizes per document provides an unbiased representation of bias without automatically inflicting a penalty on retrieval functions that favor or disfavor long documents. Table 1 shows how normalized retrievability provides a more realistic estimate for retrieval functions retrieval bias. Now retrieval function

C is correctly showing less retrieval bias than retrieval function B, and accordingly retrieval function B is showing less bias than retrieval function A.

Experiment Setup

Benchmark dataset and queries generation

To analyze the relationship between QCs and retrievability bias, we use the 1.2 million patents from the TREC Chemical Retrieval Track (TREC-CRT),¹ allowing validation of retrieval bias analysis on a large-sale corpus within a recall-oriented domain (Lupu, Huang, Zhu, & Tait, 2009). The subset of 34,200 documents for which relevance assessments are available as part of TREC-CRT serves as seed for query generation. Rank cut-off is set to $c = 100$. Queries are generated with combinations of those terms that appear more than one time in the document. For these terms, all 3- and 4-terms combinations are used to create the set of queries Q . We consider only those queries that have a termset document frequency of more than the rank cut-off $c = 100$ (otherwise, the queries become too specific and do not assist in capturing retrieval bias between different retrieval functions as all retrieval functions would return them somewhere under the top- c documents.) We generate and process around 38 million 3-terms and 118 million 4-terms queries. These queries are posed against the complete corpus of 1.2 million documents as boolean AND queries with subsequent ranking according to the chosen retrieval model to determine retrievability scores as defined in Equation (3). Furthermore, queries are categorized and partitioned into smaller subsets based on their different characteristics, and retrieval bias of retrieval functions is analyzed individually with each partitioned subset for discovering correlations between different retrieval functions retrieval bias levels and different QCs.

¹Available at <http://www.ir-facility.org/research/evaluation/trec-chem-09>

Retrieval functions

Two standard IR models and three different variations of language models with term smoothing are used for retrieval bias analysis. These are TF-IDF the OKAPI retrieval function BM25, Jelinek-Mercer language model JM, Dirichlet (Bayesian) language model DirS, and Absolute Discounting language model TwoStage.

TF-IDF & BM25. TF-IDF (Equation (4)) and OKAPI BM25 (Robertson & Walker, 1994) (Equation (5)) as de facto standard retrieval functions are used as a baseline that the other retrieval models are compared to

$$TFIDF(d, q) = \sum_{w \in q} \frac{f(d, q_w)}{|d|} \log \frac{|D|}{df(q_w)} \quad (4)$$

$$BM25(d, q) = \sum_{w \in q} \log \frac{|D| - df(q_w) + 0.5}{df(q_w) + 0.5} \times \frac{f(d, q_w)(k + 1)}{f(d, q_w) + k \left(1 - b + b \frac{|d|}{|\bar{d}|}\right)} \quad (5)$$

where $f(d, q_w)$ is the within-document frequency of query term q_w in d , and $|D|$ is the total number of documents in the collection. $|d|$ represents document length $df(q_w)$ is the number of documents containing q_w , and $|\bar{d}|$ is the average document length in the collection from which documents are drawn. k and b are parameters, usually chosen as $k=2.0$ and $b=0.75$.

Language models with terms smoothing. Language models try to estimate the probability for each document that the query q was generated by the underlying model. Here terms are assumed to occur independently, and the probability is the product of the individual query terms given the document model M_d of document d :

$$P(q|M_d) = \prod_{w \in q} P(q_w|M_d) \quad (6)$$

$$P(w|M_d) = \frac{f(d, q_w)}{|d|} \quad (7)$$

The overall similarity score for the query and the document could be zero if some of query terms do not occur in the document. However, it is not sensible to rule out a document just because a single query term is missing. For dealing with this, language models make use of smoothing to balance the probability mass between occurrences of terms in documents, and terms not found in the documents.

Jelinek-Mercer smoothing. Jelinek-Mercer smoothing language model (Zhai & Lafferty, 2004) combines the relative frequency of a query term $w \in q$ in the document d with the relative frequency of the term in the collection (D) as a whole. With this approach, the maximum likelihood estimate

is moved uniformly toward the collection model probability $P(w|D)$:

$$P(w|M_d) = (1 - \lambda) \frac{f(d, q_w)}{|d|} + \lambda P(q_w|D) \quad (8)$$

$f(d, q_w)$ represents the frequency of term w in document d . The value of λ is normally suggested as ($\lambda = 0.7$).

Dirichlet (Bayesian) smoothing (DirS). As long documents allow us to estimate the language model more accurately, Dirichlet smoothing (Zhai & Lafferty, 2004) smoothes them less. If we use the multinomial distribution to represent a language model, the conjugate prior of this distribution is the Dirichlet distribution. This gives:

$$P(w|M_d) = \frac{f(d, q_w) + \mu P(q_w|D)}{|d| + \mu} \quad (9)$$

As μ gets smaller, the contribution from the collection model also becomes smaller, and more emphasis is given to the relative term weighting. According to Zhai and Lafferty (2004), the optimal value of μ is around 2,000.

Two-stage smoothing (two-stage). In this model (Zhai, 2002), the retrieval function first smoothes the document language model using the Dirichlet prior. Then, the retrieval function mixes the document language model with a query background model using Jelinek-Mercer smoothing. The smoothing function is therefore:

$$P(w|M_d) = (1 - \lambda) \frac{f(d, q_w) + \mu P(q_w|D)}{|d| + \mu} + \lambda P(q_w|D) \quad (10)$$

where μ is the Dirichlet prior parameter and λ is the Jelinek-Mercer parameter. In our experimentation setting, we set the parameters $\mu = 2000$ and $\lambda = 0.7$, respectively.

Standard retrievability analysis

Figure 1 plots retrieval bias of different retrieval functions using Lorenz curves with a rank cut-off factor $c = 100$ with $\hat{r}(d)$. Bias is reflected by the amount of skew in the curves. The more skewed the curve, the greater the amount of inequality or bias within the documents. The curves of *BM25*, *TF-IDF*, and *JM* are less skewed than the curves of *DirS* and *TwoStage*, reflecting less retrieval bias. Similarly, the curves with 4-term queries are more skewed than 3-terms queries; therefore, indicating long queries add more bias than short queries. Language modeling approaches, particularly *DirS* and *TwoStage*, in their standard parameter setting are usually optimized towards short documents. In these settings, they are heavily biasing in terms of accessibility, resulting in a rather skewed Lorenz curve. Table 2 lists the retrievability inequality providing Gini-Coefficients for a range of other rank cut-off factors. Note the high bias experienced when limiting oneself to short result lists of 10 or 50 documents. As expected, the Gini-Coefficient tends to decrease slowly for all query sets and models as the rank cut-off factor increases.

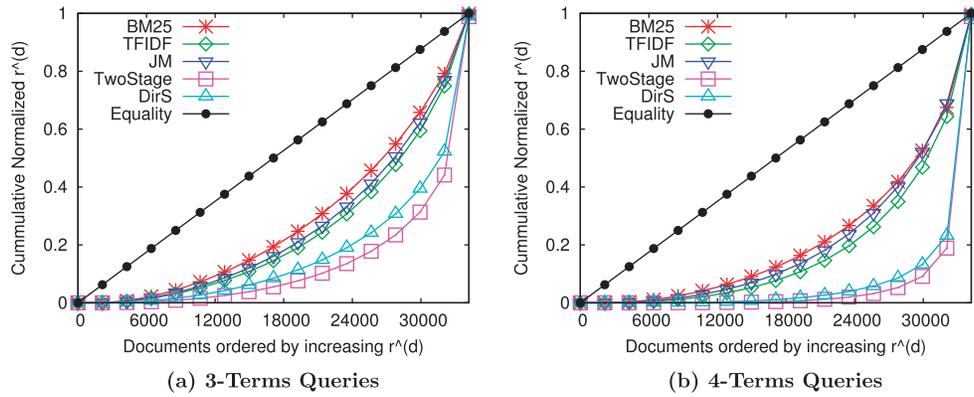


FIG. 1. Lorenz curves for representing retrieval bias of retrieval functions with cumulative $\hat{r}(d)$ scores approach and rank cut-off factor $c = 100$. Figure (a) shows retrievability inequality on 3-Terms queries and (b) shows retrievability inequality on 4-terms queries. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

TABLE 2. Gini-coefficient scores representing retrieval bias of different retrieval functions on various rank cut-off factors (c) with $\hat{r}(d)$ retrievability scores calculation approach on 3- and 4-terms queries.

Retrieval function	3-Terms queries					4-Terms queries				
	$c = 10$	$c = 50$	$c = 100$	$c = 200$	$c = 300$	$c = 10$	$c = 50$	$c = 100$	$c = 200$	$c = 300$
BM25	0.83	0.68	0.56	0.51	0.48	0.95	0.80	0.68	0.64	0.59
TFIDF	0.93	0.77	0.64	0.58	0.52	0.97	0.87	0.74	0.70	0.61
JM	0.86	0.73	0.61	0.56	0.50	0.93	0.81	0.71	0.66	0.61
TwoStage	0.90	0.89	0.82	0.75	0.72	0.97	0.97	0.94	0.88	0.85
DirS	0.89	0.85	0.77	0.73	0.68	0.95	0.93	0.90	0.82	0.76

As c increases, bias steadily decreases indicating that lower bias is experienced when considering longer ranked lists.

This indicates that retrievability inequality within the collection is mitigated by the willingness of the user to search deeper down into the ranking. If users examine only the top documents, they will face a greater degree of retrieval bias. In terms of the bias induced by the tested retrieval functions, we note that *TwoStage* has the greatest inequality between documents over both query sets while *BM25* appears to provide the least inequality.

Comparison between $r(d)$ and $\hat{r}(d)$

To analyze the difference between $r(d)$ and $\hat{r}(d)$, we compare the retrieval bias of both functions with respect to their dissimilarity on different subsets of documents. We analyze this factor by dividing the collection into a number of subsets based of their length and vocabulary size (number of unique terms per document).

Before discussing the results, it is important to mention that $\hat{r}(d)$ does not provide a different estimate for retrievability than $r(d)$. The only major difference is that $r(d)$ measures retrievability without considering diversity in document length or vocabulary size. $\hat{r}(d)$ implicitly accounts for this difference by considering the number of queries that a document can theoretically be retrieved by, which is obviously higher for vocabulary-rich documents. $\hat{r}(d)$ specifically pushes the retrievability rank of all those low retrievable

TABLE 3. Gini-coefficient values of different retrieval functions comparing $r(d)$ and $\hat{r}(d)$ with Rank cut-off factor $c = 100$ on 3-and 4-terms queries.

Retrieval function	3-Terms queries		4-Terms queries	
	$r(d)$	$\hat{r}(d)$	$r(d)$	$\hat{r}(d)$
BM25	0.53	0.56	0.63	0.68
TFIDF	0.49	0.64	0.53	0.74
JM	0.46	0.61	0.52	0.71
TwoStage	0.67	0.82	0.75	0.94
DirS	0.61	0.77	0.72	0.90

documents (according to their $r(d)$ value) that are only relevant to a rather small number of queries in the first place, even though they may be highly findable by these few queries.

Table 3 provides a comparison between Gini-Coefficients of all retrieval functions with both functions. Only with *BM25* a not very sharp increase is seen in the value of Gini-Coefficient when $\hat{r}(d)$ is used as compared with $r(d)$. Yet, relative to the other retrieval functions *BM25* has moved from third rank to first rank in terms of lowest retrievability bias, whereas according to the standard $r(d)$ *JM* exhibits the lowest bias. We analyze this factor further by plotting the distribution of $r(d)$ and $\hat{r}(d)$ over increasing documents

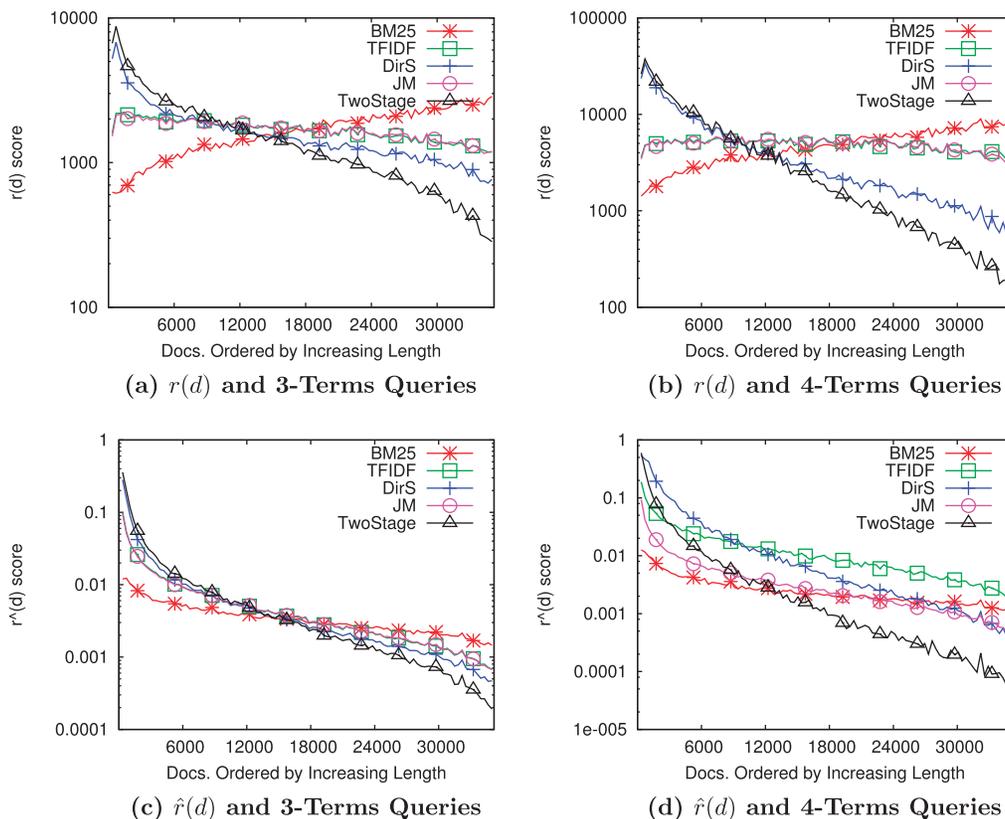


FIG. 2. Relationship between $r(d)$ and $\hat{r}(d)$ over document length. Figures (a) and (b) show retrievability scores with $r(d)$, where only with BM25 long documents have high $r(d)$ scores. Figures (c) and (d) show retrievability scores with $\hat{r}(d)$. The decreasing scores for long documents reveal that these cannot be retrievable for a large fraction of their vocabulary, though the difference between retrieval functions is less pronounced. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

length and vocabulary size. Figure 2 shows the relationship of $r(d)$ and $\hat{r}(d)$ with length, and Figure 3 shows their relationship with vocabulary size. The $r(d)$ and $\hat{r}(d)$ when computed with *TF-IDF*, *JM*, *DirS*, and *TwoStage* are negatively correlated with length and vocabulary size. Higher $\hat{r}(d)$ over short documents indicates that these retrieval functions are biased toward short documents, but due to the small number of total possible queries, $r(d)$ for these documents erroneously indicates lower retrievability. With *BM25*, the distribution of $r(d)$ when plotted over increasing length and vocabulary size shows significant positive correlation, but $\hat{r}(d)$ has a slight negative correlation. This indicates that *BM25* does not over penalize long documents as we see it with *TF-IDF*, *JM*, *DirS* or *TwoStage*.

Of particular interest are the surprisingly good retrievability scores for short documents with language modeling approaches for both the un-normalized as well as the normalized retrievability scores. Whereas these language modeling functions severely degrade in the performance for long documents, leading to an overall rather high bias as we have seen in Figure 1. This seems to be due to the fact that these models are mostly tuned on benchmark corpora consisting of predominantly short texts. Patent documents are rather long when compared with other traditional datasets. This is also why little improvement has been found with these models and

standard parameters settings on this domain (Kang, Na, Kim, & Lee, 2007). As pointed out in this study, we have started investigating the effect of the smoothing parameters. First results indicate that higher smoothing parameters μ in fact lead to better retrievability (and even higher effectiveness) in this setting, highlighting the need for further investigation of this issue.

Tables 4–7 provide a sample of documents on the extreme ends for *BM25* and *JM* on 4-Terms queries. Tables 4 and 6 show a set of documents that are low retrievable when $r(d)$ is used, but high retrievable when $\hat{r}(d)$ is used. In both retrieval functions, documents are mostly short and are well retrievable for the few queries that are generated from them. Similarly, Tables 5 and 7 show a sample of documents that are high retrievable when $r(d)$ is used, but low retrievable when $\hat{r}(d)$ is used. In contrast to before, these documents are mostly long, producing a large number of queries, out of which each document is only retrievable by a rather small fraction.

To verify the hypothesis about the correlation between document length/vocabulary size and the difference between absolute ($r(d)$) and relative ($\hat{r}(d)$) retrievability, we order the documents based on their length and vocabulary size, and subdivide the collection into 10 equal-sized subsets. After partitioning the collection, we compare the documents retrievability ranks relationship between both measures per

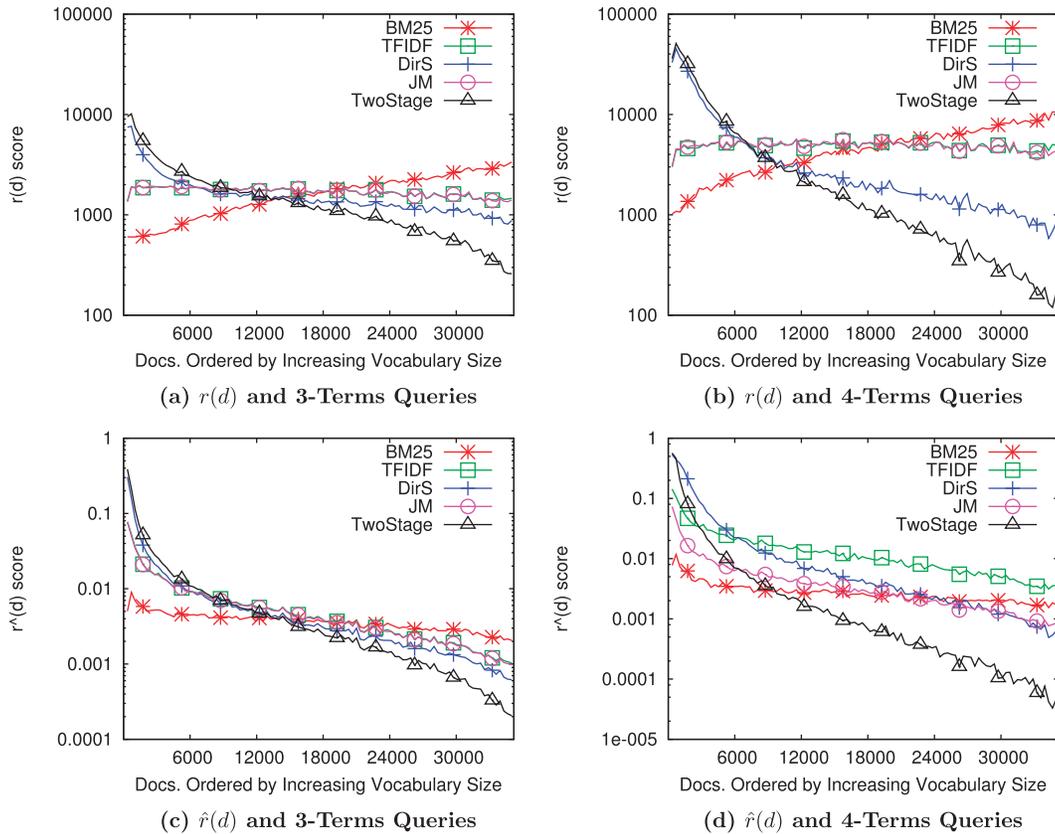


FIG. 3. Relationship between $r(d)$ and $\hat{r}(d)$ over collection when collection is ordered by increasing document vocabulary size. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

TABLE 4. A sample of Top 10 documents with *BM25*, representing that they are low retrievable when $r(d)$ was used, but high retrievable when $\hat{r}(d)$ was used.

Document ID	Length	Vocabulary size	$\hat{Q}(d)$	$r(d)$	$\hat{r}(d)$	Rank in $r(d)$	Rank in $\hat{r}(d)$
US-6265361	454	115	87,629	706	0.0081	29,224	969
US-4963269	1014	209	90,266	724	0.0080	29,119	985
US-6251505	827	148	86,956	598	0.0069	30,007	1,581
US-6172156	550	162	89,307	603	0.0068	29,977	1,667
US-5601715	1919	234	88,216	594	0.0067	30,031	1,682
US-6271181	297	95	90,808	608	0.0067	29,941	1,708
US-6706824	459	132	89,279	575	0.0064	30,162	1,895
US-4986901	1367	217	86,716	520	0.0060	30,557	2,288
US-6180660	1557	365	90,843	527	0.0058	30,506	2,451
US-6414110	508	116	87,611	491	0.0056	30,770	2,670

Mostly these documents are short but have higher percentage of retrievability out of their total queries.

subset using Spearman's rank correlation coefficient. Figure 4 shows the correlation between the two measures. Results show that major difference occurs for either very long or very short documents, or documents having very large or small vocabulary. For average documents, little difference between these two measures can be noted. This indicates that if a given corpus shows large diversity between documents in terms of length or vocabulary size, then we can expect that there would be also larger difference between $r(d)$ and $\hat{r}(d)$ scores. This, in turn, may hint at the need to handle the retrieval of

documents on the extreme ends separately if equal access probability should be provided.

Retrievability Analysis with Different QCs

The objective of retrievability measurement is to compare different retrieval functions in terms of access, and to identify those retrieval functions that are more effective having less retrieval bias. Estimating retrievability over all possible queries is important only if QCs are independent of retrieval

TABLE 5. A sample of the Bottom 10 documents with *BM25*, representing that they are high retrievable when $r(d)$ was used, but low retrievable when $\hat{r}(d)$ was used.

Document ID	Length	Vocabulary size	$\hat{Q}(d)$	$r(d)$	$\hat{r}(d)$	Rank in $r(d)$	Rank in $\hat{r}(d)$
US-5366737	11,128	1,864	11,130,050	10,536	0.0009	3,006	24,080
US-5545412	11,081	1,828	11,011,920	9,243	0.0008	4,055	25,352
US-5560862	6,547	1,231	10,702,164	8,951	0.0008	4,314	25,392
US-6117357	7,736	1,432	11,090,816	8,787	0.0008	4,476	25,961
US-5460747	6,609	1,230	10,705,499	8,045	0.0008	5,323	26,496
US-5550289	11,178	1,828	11,130,672	8,302	0.0007	4,983	26,571
US-6166218	10,419	1,566	11,112,632	7,963	0.0007	5,428	26,932
US-5846517	14,058	2,347	11,091,722	7,337	0.0007	6,253	27,632
US-6071494	14,494	2,362	11,093,443	7,249	0.0007	6,407	27,743
EP-1037673	21,298	3,270	11,089,055	6,619	0.0006	7,438	28,444

Mostly these documents are long, but have lower percentage of retrievability out of their total queries.

TABLE 6. A sample of the Top 10 documents with *JM*, representing that they are low retrievable when $r(d)$ was used, but high retrievable when $\hat{r}(d)$ was used.

Document ID	Length	Vocabulary size	$\hat{Q}(d)$	$r(d)$	$\hat{r}(d)$	Rank in $r(d)$	Rank in $\hat{r}(d)$
US-6328106	358	118	135,165	1,964	0.0145	23,175	1,374
US-6190581	515	142	172,730	1,780	0.0103	24,216	2,688
US-6464873	873	173	200,188	1,696	0.0085	24,692	3,665
US-6753288-B2	399	154	205,131	1,210	0.0059	27,573	5,971
US-6768017	504	160	213,925	1,207	0.0056	27,600	6,284
US-5667666	587	183	221,933	1,244	0.0056	27,360	6,338
US-5614330	634	168	203,199	1,135	0.0056	28,056	6,368
US-6753288	399	154	205,131	1,057	0.0052	28,505	7,009
EP-0540840	399	157	185,776	891	0.0048	29,521	7,590
US-6555596-B1	294	131	146,930	679	0.0046	30,742	7,892

Mostly these documents are short but have higher percentage of retrievability out of their total queries.

TABLE 7. A sample of the Bottom 10 documents with *JM*, representing that they are high retrievable when $r(d)$ was used, but low retrievable when $\hat{r}(d)$ was used.

Document ID	Length	Vocabulary size	$\hat{Q}(d)$	$r(d)$	$\hat{r}(d)$	Rank in $r(d)$	Rank in $\hat{r}(d)$
US-5554686	10,252	1,301	11,019,657	12807	0.0012	1,451	21,589
US-6046295	10,524	1,277	11,068,556	12,434	0.0011	1,660	21,902
EP-1194402	12,284	1,598	11,401,901	12,471	0.0011	1,638	22,128
EP-0900262	8,023	1,463	10,719,815	9,216	0.0009	4,111	24,186
US-5348621	13,904	2,403	10,653,869	9,033	0.0008	4,306	24,292
US-5413725	24,076	1,669	10,666,860	9,046	0.0008	4,292	24,296
EP-0551390	9,511	1,579	11,078,594	9,367	0.0008	3,952	24,317
US-5861366	10,667	1,835	11,500,663	9,210	0.0008	4,119	24,733
US-5858117	10,448	1,778	11,353,880	7,958	0.0007	5,733	25,789
US-5162445	21,630	1,845	12,436,970	6,916	0.0006	7,259	27,361

Mostly these documents are long, but have lower percentage of retrievability out of their total queries.

bias. That is, if QCs do not have any influence on decreasing or increasing retrieval bias. However, this is not the case. Our experiments reveal that retrieval bias is not independent of QCs. Certain features of queries have strong influence on increasing or decreasing retrieval bias measured for any retrieval function. These do not dramatically alter the relative estimate of retrieval bias, but only affect the bias magnitude of approximation. The main advantage of this correlation is that it creates the possibility of predicting retrieval bias of retrieval

functions by processing fewer queries. Additionally, this correlation is fruitful for identifying those queries that have strong influence on increasing or decreasing retrieval bias.

We analyze the retrieval bias of retrieval functions with the help of following QC factors.

- Query termset document frequency (QTDF)
- Cumulative query terms frequencies per document (QTTF)
- QCs based on query performance prediction methods

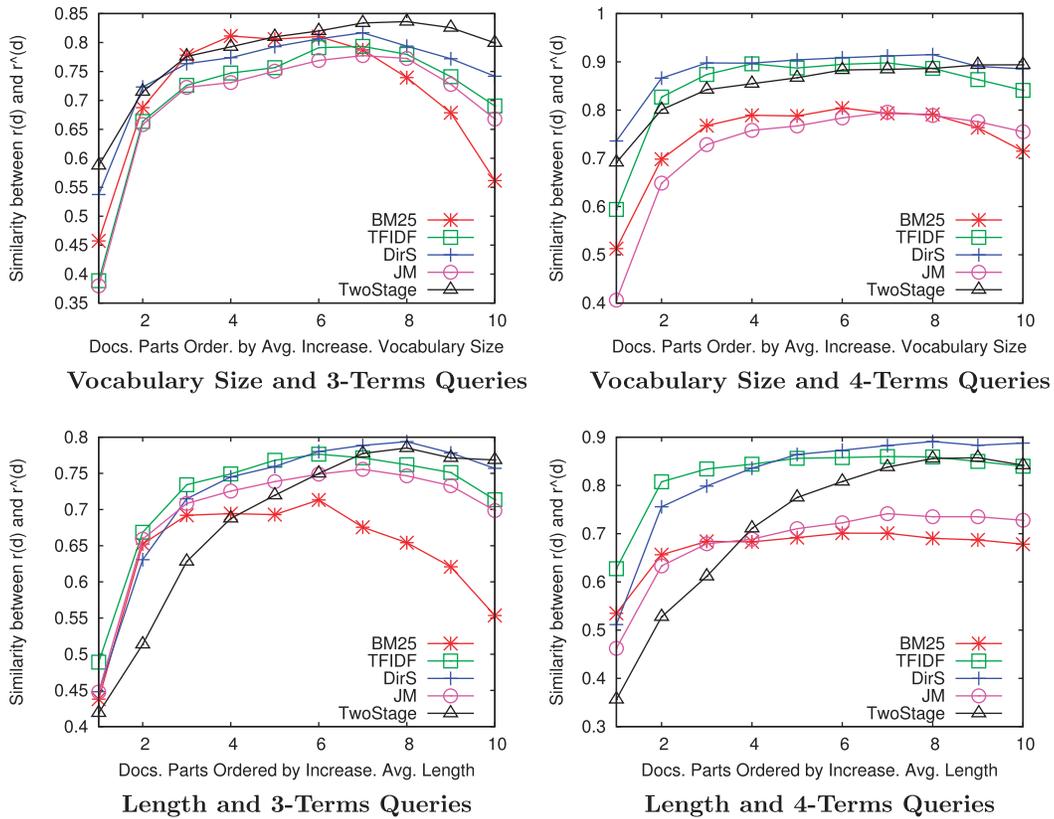


FIG. 4. Spearman's rank correlation between $r(d)$ and $\hat{r}(d)$ over different subsets of collection subdivided by length and vocabulary size. Correlation is low on lower and higher subsets of length and vocabulary, indicating large difference between two functions on these two extremes. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

For each QC factor, we partition the initial query set into $k = 100$ smaller query subsets using equal width interval binning method. This involves sorting the queries in ascending order of QC score, and distributing them into k buckets or bins. We use smoothing by bin medians for labeling the partitioned subsets. The following sections describe these analyses in more detail.

Correlation analysis with query termset document frequency (QTDF)

This QC is based on the combined query termset document frequency in the corpus, capturing how widely used or specific the termset of a query are. More formally,

$$QTDF(q) = df(q) \quad (11)$$

We partition the query set into 100 partitions on the basis on queries termset document frequency (QTDF) scores in the corpus.

Figure 5(a) and (b) shows a strong correlation between retrieval bias and $QTDF$ scores. From the results, it can be easily inferred that the amount of retrieval bias within documents of corpus is mitigated with lower $QTDF$ scores of queries. If users' queries have higher $QTDF$ scores then

they will experience a greater degree of retrieval bias, i.e., many documents will turn up for a large number of queries, whereas others will be completely missed. On higher $QTDF$ scores, for instance ≥ 1000 , *DirS* and *TwoStage* retrieval functions are not able to retrieve more than 50% of document via any query (Figure 5(c) and (d)), and *TF-IDF*, *JM*, and *BM25* are not able to retrieve around 20% of all seed documents. This ratio decreases further with all retrieval functions when query $QTDF$ scores decrease. Overall, the retrievability performance of all those functions that either normalize document length (*BM25*) or normalize query term frequency relative to document length (*TF-IDF*, *JM*) is better. However, in the case when queries show larger $QTDF$ scores then retrieval functions performing pure length normalization like *TF-IDF* show a negative affect on the access. It penalizes long documents as they are become less retrievable. This is the reason why the retrievability performance of *BM25* is better on these classes of queries, since it does partial normalization and avoids the over-penalization of long documents by a second global parameter b . Significant correlation between $QTDF$ scores and retrieval bias of retrieval functions indicates analyzing retrieval bias with different QCs is far more effective for examining the retrievability behavior of retrieval functions as compared with the cumulative bias analysis approach over all queries.

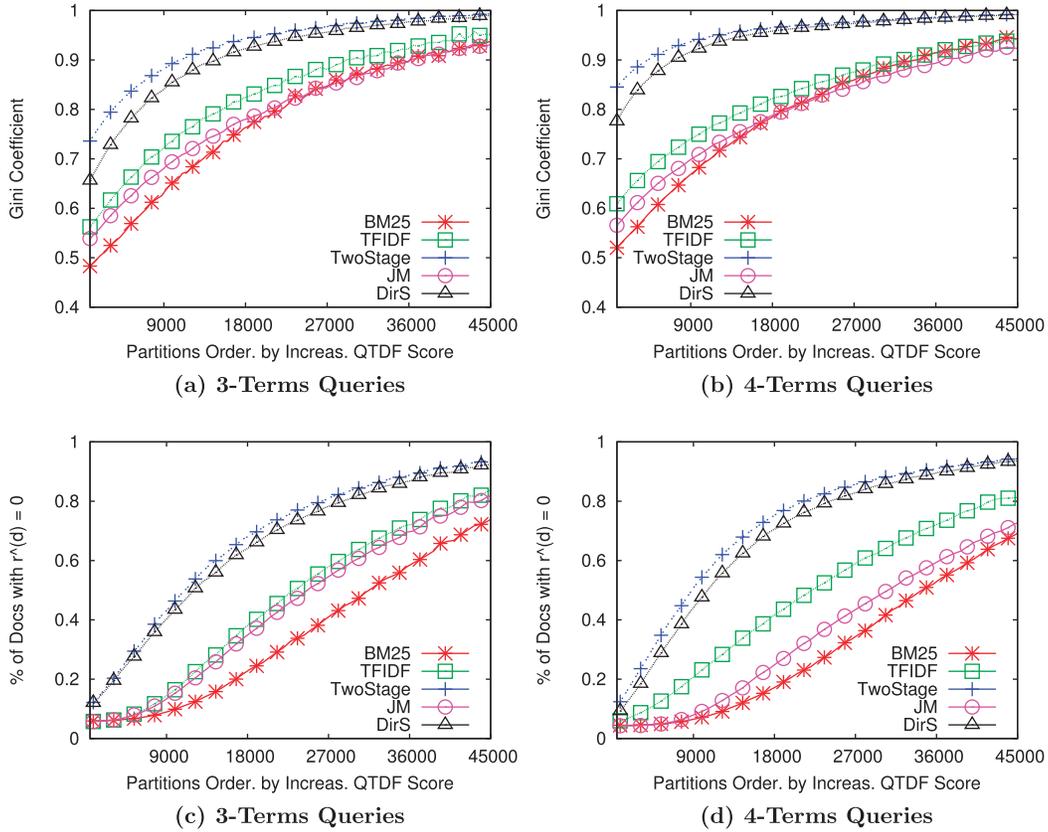


FIG. 5. Relationship of (retrieval functions) retrieval bias over increasing query termset document frequency (*QTFD*) partitions. Lower values of *QTFD* result in lower retrieval bias, whereas higher values of *QTFD* show higher retrieval bias. Figures (a) and (b) show relationship between retrieval bias and *QTFD* partitions, (c) and (d) show percentage of documents with $\hat{r}(d) = 0$ in each partition. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Correlation analysis with cumulative query terms frequencies per document (*QTFF*)

We further want to analyze the relationship between the cumulative query terms frequencies (*QTFF*) within individual documents and retrieval bias. *QTFF* scores are normalized by document length to remove the effect of short or long documents. To perform analysis, we subdivide the queries into 100 partitions with the help of following steps. The binning of all queries is obtained as follows:

For each document d and then for each query q per document, we sum up the normalized terms frequencies in the document for all terms w in the query. This results in a set of basically $d \times q(d)$ query scores, i.e., a score for each query for each document (ignoring queries with a score of 0, i.e., queries the terms of which do not appear in the document at all). More formally, *QTFF* is computed as follows:

$$QTFF(d, q) = \sum_{w \in q} f(d, q_w) \quad (12)$$

This set is sorted by score and subdivided into 100 partitions. After creating partitions of queries, we then individually analyze the retrieval bias of each retrieval function

within each partition. Here, it is important to mention that although the same query may be distributed into several partitions depending upon its *QTFF* score for different documents. We process each query only once with all documents in the corpus. We then aggregate the $\hat{r}(d)$ score of the top- $c = 100$ documents into the different partitions according to their *QTFF* scores.

The results in Figure 6 show that *QTFF* also has a strong correlation with different levels of retrieval bias. Larger cumulative query terms frequencies are more effective for increasing retrievability of documents. We note a considerable decrease in the retrieval bias of retrieval functions when the queries *QTFF* scores are greater than 0.15. This indicates that it is easier to access or find documents when documents have higher *QTFF* scores for many queries. However, as the *QTFF* scores within documents decrease, the retrievability inequality among documents also increases, with indicating larger fraction of documents in D are become hard to find within top- c rank positions. With queries *QTFF* scores < 0.10 all retrieval functions show Gini-Coefficient scores $> G = 0.70$ (higher retrieval bias). In terms of comparison between different retrieval functions *BM25*, *TF-IDF*, and *JM* are less biased than other retrieval functions.

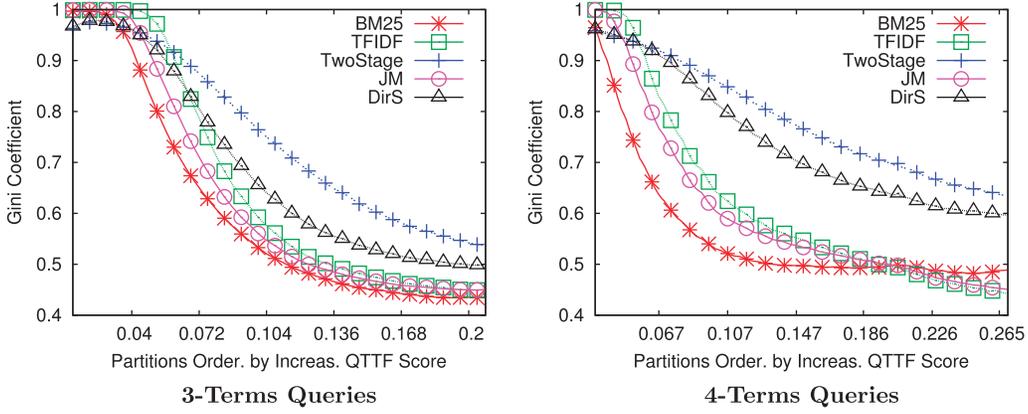


FIG. 6. Correlation analysis of (retrieval functions) retrieval bias over cumulative queries terms frequencies within documents (QTF). Lower *QTF* of documents generates higher retrieval bias and larger retrievability inequality among documents. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

Correlation analysis with query performance prediction methods

As we want to create plausible queries, we need to understand how far existing measures for query quality correlate with retrieval bias. This will allow us to get, first of all, a more realistic figure for the retrieval bias of retrieval functions under real-world conditions, specifically when we are able to compute the same query quality measures from an even small number of actual queries used by (professional) searchers on a specific corpus. We can then only select those queries from the exhaustive list that show similar query quality scores, allowing us to compare the retrieval bias of different retrieval functions on a much smaller number of queries, whereas at the same time obtaining more plausible results. In order to do so, we use several pre-retrieval predictors of predicting query quality (Cronen-Townsend et al., 2002). Pre-retrieval predictors solely rely on information that is available at indexing time, therefore, can be calculated more quickly than methods relying on the result list, causing less overhead to the search retrieval functions (He & Ounis, 2006; Zhao, Scholer, & Tsegay, 2008). The following QC factors are used for this analysis.

1. **AvIDF**: AvIDF determines the query difficulty on the basis of the specificity of a query, relying on the average of the inverse document frequency (idf) of the query terms. A term that occurs in many documents can be expected to have a high term frequency in the collection; thus, decreasing the specificity of a query (He & Ounis, 2006).

$$AvIDF = \frac{1}{|q|} \sum_{w \in q} \left[\log \frac{|D|}{df(q_w)} \right] \quad (13)$$

2. **AvICTF**: Instead of using idf, AvICTF relies on term frequencies of query for calculating the specificity of a query (He & Ounis, 2006).

$$AvICTF = \frac{1}{|q|} \sum_{w \in q} \left[\log_2 \frac{|t|}{tf(q_w)} \right] \quad (14)$$

$|t|$ refers to the total number of terms in the corpus, and $tf(q_w)$ represents the total term count of q_w in collection D .

3. **Simplified Query Clarity (SCS)**: Query clarity refers to the specialty/ambiguity of a query. According to Cronen-Townsend et al. (2002), the clarity (or on the contrary, the ambiguity) of a query is an intrinsic feature of a query, which has an important impact on the performance of retrieval function. The proposed clarity score is based on the sum of the Kullback Leibler divergence of the query model from the collection model, which involves computation of relevance scores for the query model, which is time-consuming. To avoid the expensive computation of query clarity, He and Ounis (2006) proposed a simplified clarity score as a comparable pre-retrieval performance predictor. It is calculated as:

$$SCS = \sum_{w \in q} \frac{1}{|q|} \log_2 \frac{\frac{1}{|q|}}{\frac{tf(q_w)}{|t|}} \quad (15)$$

4. **Collection Query Similarity (AvgSCQ)**: This query quality predictor is based on the similarity between collection and query (Zhao et al., 2008). The authors argue that a query that is similar to the collection as a whole is easier to retrieve documents for, since the similarity is an indicator of whether documents answering the information need are contained in the collection. As the score increases with increased collection term frequency and increased idf, terms that appear in few documents many times are favored. Those terms can be seen as highly specific, as they occur in relatively few documents, while at the same time, they occur often enough to be important to the query:

$$AvgSCQ = \sum_{w \in q} \left(1 \times \log(tf(q_w)) \times \log \left(1 + \frac{|D|}{df(q_w)} \right) \right) \quad (16)$$

5. **Term Weight Variability (AvgVAR)**: AvgVAR exploits the distribution of term weights across the collection (Zhao et al., 2008). If the term weights across all documents containing query term q_w are similar, there is little evidence for a retrieval function on how to rank those documents

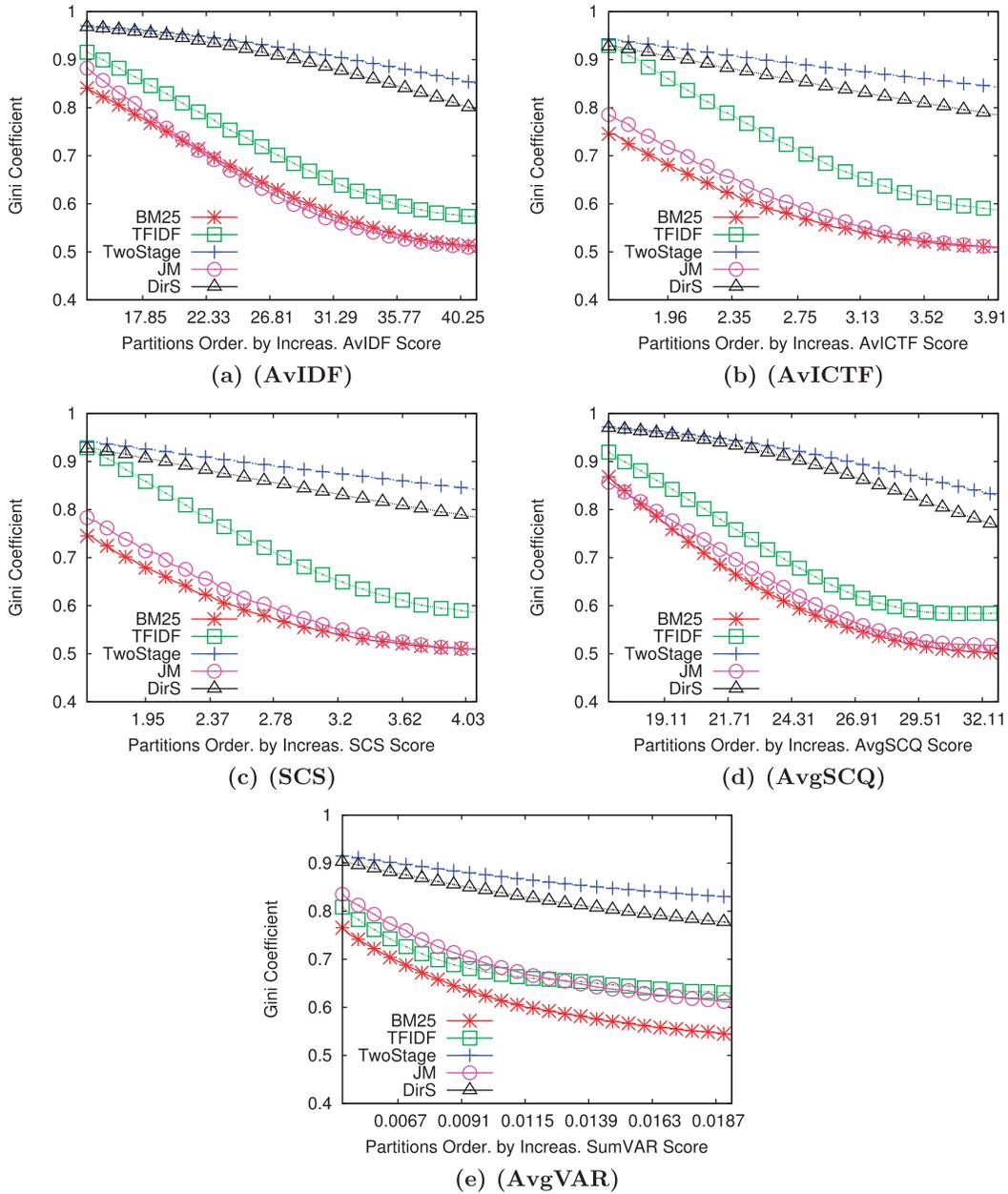


FIG. 7. Relationship between retrieval bias of retrieval functions and queries partitions subdivided with query quality scores on 4-terms queries. Higher quality queries show less retrieval bias, whereas lower quality queries show higher retrieval bias. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

given q_w , and thus different retrieval algorithms are likely to produce widely different rankings. Conversely, if the term weights differ widely across the collection, ranking becomes easier and different retrieval algorithms are expected to produce similar rankings. This predictor is calculated as follows:

$$AvgVAR = \sum_{w \in q} \sqrt{\frac{1}{df(q_w)} \sum_{d \in N_{q_w}} (\hat{t}(d, q_w) - \bar{t}_{q_w})^2} \quad (17)$$

N_{q_w} represents the set of all documents having q_w . $\hat{t}(d, q_w)$ is the term weight within document d and it is based on TF-IDF, \bar{t}_{q_w} is the average weight of \hat{t} over all documents containing q_w .

Figure 7 shows a strong correlation between retrieval functions retrieval bias and query performance prediction factors. (Due to space considerations, we show results only for 4-terms queries as 3-terms queries exhibit virtually identical behavior.) In all settings, higher quality query subsets exhibit lower retrieval bias than lower quality queries. This hypothesis is confirmed by the results in Figure 8 results, when looking at the percentage of documents with $\hat{r}(d) = 0$ (i.e., documents that cannot be retrieved via any query with $c = 100$). Lower quality queries result in a higher percentage of documents with $\hat{r}(d) = 0$, i.e., that never show up within the top-100 for any query. This percentage becomes worse when more than 50% of documents cannot be retrieved via any

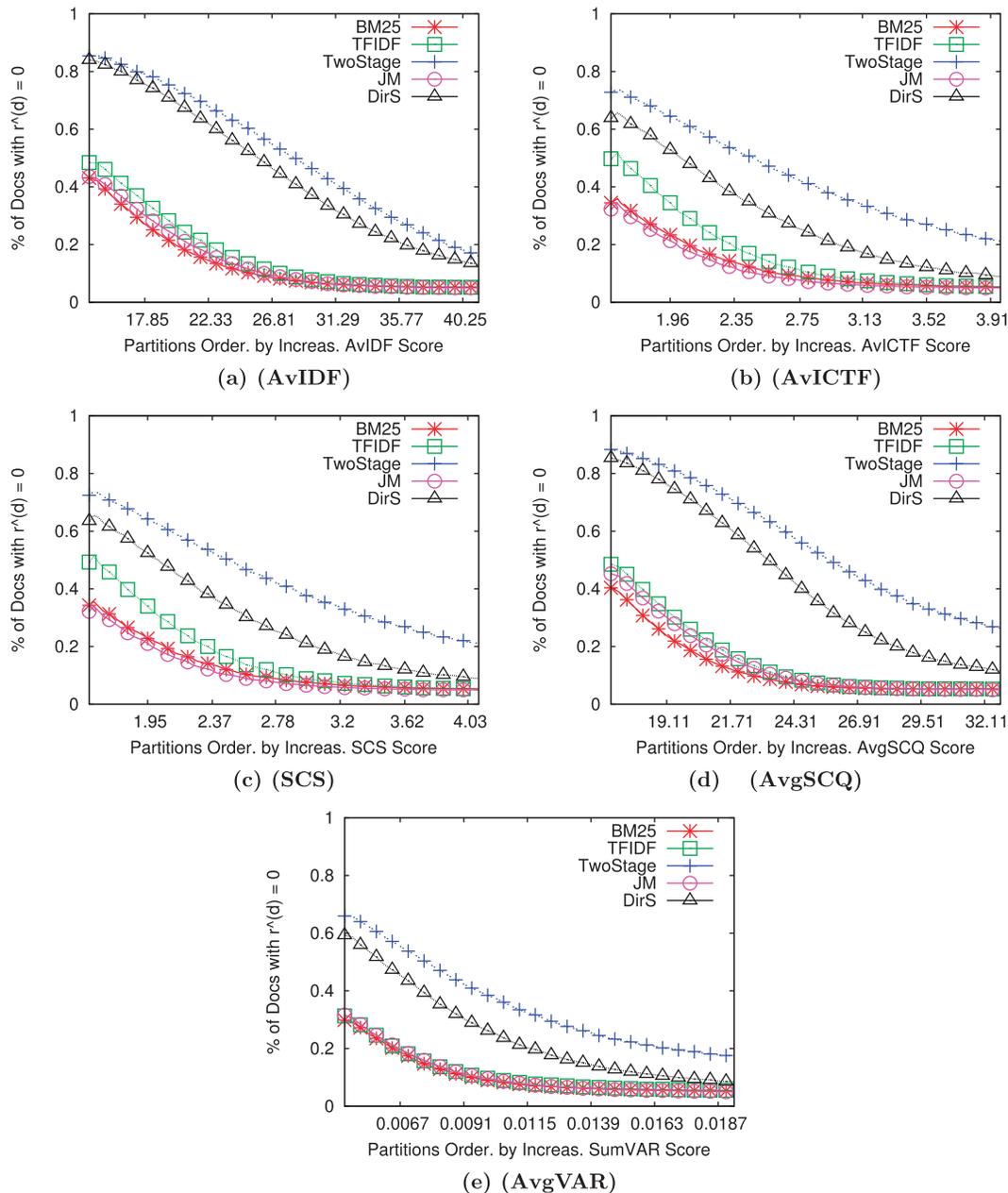


FIG. 8. Percentage of documents in different partitions with $\hat{r}(d) = 0$ out of 34,200 seed documents. This analysis is performed with 4-terms and queries are subdivided into different partitions on the basis of five query quality predictors. Lower quality queries have larger percentage of such documents due to higher retrieval bias. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

query with *TwoStage* and *DirS* retrieval functions on lower quality queries. This indicates that experienced users are expected to face lower retrieval bias. For inexperienced users, retrieval functions exhibit a high bias if they issue lower quality queries. Table 8 shows the correlation between different QC factors. Three QC factors *AvICTF*, *SCS*, and *AvIDF* are highly correlated: *AvIDF* has strong correlation with *AvgSCQ* and *AvICTF* has strong correlation with *SCS*. Therefore, it is not necessary to report both *AvIDF* and *AvgSCQ*, and *AvICTF* and *SCS*. Other QC factors have a moderate to strong relationship to one another. The retrieval functions curves that represent relationship between retrieval bias and QCs are

TABLE 8. The correlation of query characteristics factors with other query characteristics factors on 4-terms queries.

	QTDF	AvIDF	AvICTF	SCS	AvgSCQ	AvgVAR
QTDF		-0.69	-0.59	-0.60	-0.69	0.09
AvIDF			0.67	0.66	0.85	0.06
AvICTF				0.86	0.49	-0.59
SCS					0.58	-0.60
AvgSCQ						0.23
AvgVAR						

Correlations are calculated on the basis of Spearman rank correlation coefficient.

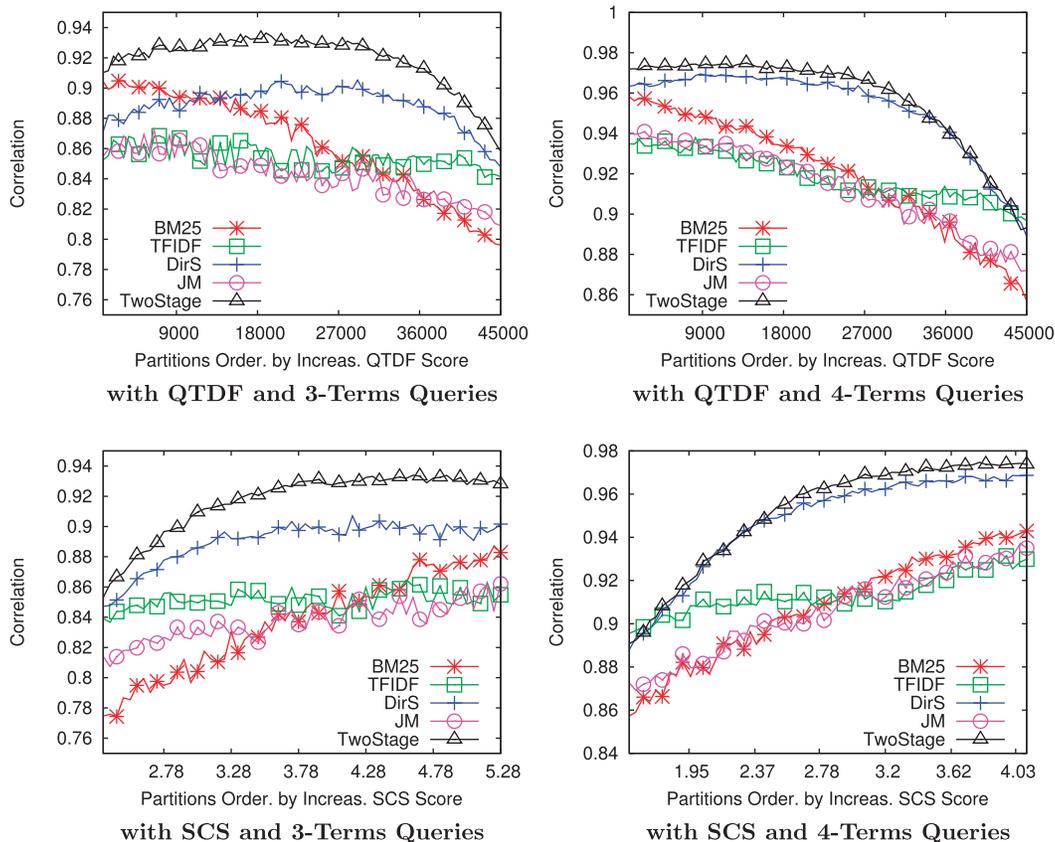


FIG. 9. Correlation between partitioned queries subsets and all queries subset of 3- and 4-terms length on the basis of individual documents retrievability ranking. Correlation is calculated with Spearman's rank correlation coefficient. Results show significant correlation between two sets. This indicates that query characteristics-based subset generation approach provides significant accurate approximation of retrieval bias. [Color figure can be viewed in the online issue, which is available at wileyonlinelibrary.com.]

more skewed on *AvIDF*- and *AvgSCQ*-based query subsets than other prediction methods.

Furthermore, it is interesting to note that all query performance prediction-based QC factors have only a moderate relationship with the query itemset combination frequency *QTDF* factor. This indicates that queries quality factors are not highly related to *QTDF* factor, and have their own influence on retrieval bias of retrieval functions. *AvgVAR* and *QTDF* are highly uncorrelated. This is because in *AvgVAR* query quality is predicted on the basis of query terms frequency weights across collection not on the basis of document frequency. The access performance of *BM25* and *JM* is better than other retrieval approaches. The results of *JM* are somewhat better than *BM25* on high-quality queries when queries are predicted with *AvIDF* and *AvgSCQ* factors. *TF-IDF* has moderate performance, while two language modeling approaches *TwoStage* and *DirS* show the worst performance.

Correlation analysis

To generate (smaller sets of) representative queries for retrieval bias analysis, we need to analyze how far documents in individual partitions provide accurate retrievability

rankings of individual documents similar to exhaustive queries (all queries, without partitioning them). In Figure 9, we provide this comparison using Spearman's rank correlation coefficient over *QTDF* and *SCS* QC factors. The results of Figure 9 show only small difference between retrievability ranking of documents when the retrieval bias is estimated from the exhaustive set of queries or only from query partitions. This suggests that query partitioning by different QC factors can be used for approximating documents $\hat{r}(d)$ scores and retrieval bias analysis.

Conclusions

Document retrievability is a measurement in IR for the analysis of retrieval bias of retrieval functions. In recent years, several attempts have been made to quantify retrieval functions bias using this concept. The main limitation of these studies is that they do not follow predefined criteria for query generation. Mostly queries are generated using a random selection approach, a rather small subset from the huge number of potential queries. To understand how to generate queries for retrieval bias estimation and the influence of different characteristics of queries on different levels

of retrieval bias in this work, we analyzed the relationship between retrieval bias and different QCs. Our findings yield the following two main points:

- Cumulative retrieval bias estimation approach over all available queries does not correctly represent retrieval bias of retrieval functions for all different QCs. We show several different characteristics of queries that have their own influence on increasing or decreasing retrieval bias of retrieval functions.
- Retrieval bias of retrieval functions has a strong correlation with different QCs. This relationship can be utilized for predicting retrieval bias without processing all queries.

In terms of retrieval functions performance, the two language modeling approaches Dirichlet Bayesian Smoothing (*DirS*) and Two-Stage Smoothing (*TwoStage*) performed rather poorly compared with *BM25*, *TF-IDF*, and Jelinek-Mercer Smoothing (*JM*). This may most likely be attributed to the fact that the parameter settings for these (otherwise very well-performing) models are commonly tuned on corpora consisting of rather short and potentially more length-balanced documents. First experiments indicate that by increasing the smoothing parameter the retrieval bias becomes less pronounced and the overall performance in term of effectiveness measures such as precision and recall increases. This effect will require further investigation to establish a clear correlation between parameter tuning, corpora characteristics and a retrieval functions' performance.

It can be shown that all retrieval functions exhibit a certain bias. However, rather than simply considering the overall bias of a retrieval function, more specific analysis allows us to fine-tune retrieval function selection and impact estimation for different application settings. Criteria influencing the selection and tuning are, on the one hand, corpus-specific, considering the diversity of documents with respect to length and vocabulary size. Others are query, and thus rather user specific, impacting on the average query length and query quality/specificity that users will be able to provide to the retrieval function.

Future work will concentrate specifically on two issues. On the other hand, we want to obtain a better formal understanding of the underlying causes for low retrievability. On the one hand, this definitely affects outlier documents. Yet, it is not only these more easily identifiable documents that may exhibit low retrievability. It may also affect documents that are too far from the subspace reached by query term combinations in the entire term \times document space, with too far potentially being rather close in dense areas, whereas in other areas the quasi-random results delivered by conventional distance metrics in very high-dimensional and very sparse feature spaces may be at the root of problems. We want to obtain a better understanding of the consequences of the behavior of different retrieval functions, allowing us to potentially identify combinations of query retrieval functions that are more suitable for processing either specific types of queries or optimized for certain sub-parts of a document corpus. This will also involve extensive studies on different

document corpora to better understand the relationship between retrieval function bias and corpus characteristics.

References

- Arampatzis, A., Kamps, J., Kooen, M., & Nussbaum, N. (2007). Access to legal documents: Exact match, best match, and combinations. Proceedings of the Sixteenth Text Retrieval Conference (TREC 2007), Gaithersburg, Maryland.
- Azzopardi, L., & Bache, R. (2010). On the relationship between effectiveness and accessibility. In Proceedings of the 33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '10) (pp. 889–890). New York: ACM Press.
- Azzopardi, L., de Rijke, M., & Balog, K. (2007). Building simulated queries for known-item topics: An analysis using six European languages. In Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '07) (pp. 455–462). New York: ACM Press.
- Azzopardi, L., & Owens, C. (2009). Search engine predilection towards news media providers. In Proceedings of the 32nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '09) (pp. 774–775). New York: ACM Press.
- Azzopardi, L., & Vinay, V. (2008). Retrievability: An evaluation measure for higher order information access tasks. In Proceeding of the 17th ACM Conference on Information and Knowledge Management (CIKM '08) (pp. 561–570). New York: ACM Press.
- Bar-Ilan, J., Keenoy, K., Levene, M., & Yaari, E. (2009). Presentation bias is significant in determining user preference for search results a user study. Journal of the American Society for Information Science and Technology, 60(1), 135–149.
- Bashir, S., & Rauber, A. (2009a). Analyzing document retrievability in patent retrieval settings. In Proceedings of the 20th International Conference on Database and Expert Systems Applications (DEXA '09). Lecture Notes in Computer Science, 5690, 753–760.
- Bashir, S., & Rauber, A. (2009b). Improving retrievability of patents with cluster-based pseudo-relevance feedback documents selection. In Proceedings of the 18th ACM Conference on Information and Knowledge Management (CIKM '09) (pp. 1863–1866). New York: ACM Press.
- Bashir, S., & Rauber, A. (2010). Improving retrievability of patents in prior-art search. In Proceedings of the 32nd European Conference on Information Retrieval Research (ECIR '10). Lecture Notes in Computer Science, 5993, 457–470.
- Cronen-Townsend, S., Zhou, Y., & Croft, W.B. (2002). Predicting query performance. In Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '02) (pp. 299–306). New York: ACM Press.
- Gastwirth, J.L. (1972). The estimation of the Lorenz curve and Gini index. The Review of Economics and Statistics, 54(3), 306–316.
- He, B., & Ounis, I. (2006). Query performance prediction. Information System Journal, 31(7), 585–594.
- Jordan, C., Watters, C., & Gao, Q. (2006). Using controlled query generation to evaluate blind relevance feedback algorithms. In Proceedings of the Sixth ACM/IEEE-CS Joint Conference on Digital Libraries (JCDL '06) (pp. 286–295). New York: ACM Press.
- Kang, I.-S., Na, S.-H., Kim, J., & Lee, J.-H. (2007). Cluster-based patent retrieval. Information Processing and Management Journal, 43(5), 1173–1182.
- Lauw, H.W., Lim, E.-P., & Wang, K. (2006). Bias and controversy: Beyond the statistical deviation. In Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 625–630). New York: ACM Press.
- Lawrence, S., & Giles, C.L. (1999). Accessibility of information on the web. Nature, 400, 107–109.
- Lupu, M., Huang, J., Zhu, J., & Tait, J. (2009). TREC-CHEM: Large scale chemical information retrieval evaluation at trec. SIGIR Forum, 43(2), 63–70.
- Magdy, W., & Jones, G.J. (2010). Pres: A score metric for evaluating recall-oriented information retrieval applications. In Proceedings of the 33rd

- Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '10) (611–618). New York: ACM Press.
- Mowshowitz, A., & Kawaguchi, A. (2002). Bias on the web. *Communications of the ACM*, 45(9), 56–60.
- Owens, C. (2009). A study of the relative bias of web search engines toward news media providers (master's thesis.), University of Glasgow.
- Robertson, S.E., & Walker, S. (1994). Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '94)* (pp. 232–241). New York: ACM Press.
- Sakai, T. (2008). Comparing metrics across TREC and NTCIR: The robustness to system bias. In *Proceedings of the 17th ACM Conference on Information and Knowledge Management (CIKM '08)* (pp. 581–590). New York: ACM Press.
- Tague, J., Nelson, M., & Wu, H. (1981). Problems in the simulation of bibliographic retrieval systems. In *Proceedings of the Third Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (pp. 236–255). New York: ACM Press.
- Vaughan, L., & Thelwall, M. (2004). Search engine coverage bias: evidence and possible causes. *Information Processing & Management Journal*, 40(4), 693–707.
- Zhai, C. (2002). Risk minimization and language modeling in text retrieval. (PhD thesis.) Carnegie Mellon University.
- Zhai, C., & Lafferty, J. (2004). A study of smoothing methods for language models applied to information retrieval. In *ACM Transactions on Information Systems*, 22(2), 179–214.
- Zhao, Y., Scholer, F., & Tsegay, Y. (2008). Effective pre-retrieval query performance prediction using similarity and variability evidence. In *Proceedings of the 30th European Conference on Advances in Information Retrieval (ECIR'08)*. *Lecture Notes in Computer Science*, 5478, 52–64.