J Intell Inf Syst (2014) 42:457–484
DOI 10.1007/s10844-013-0274-3

Producing efficient retrievability ranks of documents using normalized retrievability scoring function

Shariq Bashir · Akmal Saeed Khattak

Received: 9 April 2013 / Revised: 6 August 2013 / Accepted: 12 August 2013 /
Published online: 6 September 2013

Abstract In this paper, we perform a number of experiments with large scale queries to analyze the retrieval bias of standard retrieval models. These experiments analyze how far different retrieval models differ in terms of retrieval bias that they imposed on the collection. Along with the retrieval bias analysis, we also exploit a limitation of standard retrievability scoring function and propose a normalized retrievability scoring function. Results of retrieval bias experiments show us that when a collection contains highly skewed distribution, then the standard retrievability calculation function does not take into account the differences in vocabulary richness across documents of collection. In such case, documents having large vocabulary produce many more queries and such documents thus have theoretically large probability of retrievability via a much large number of queries. We thus propose a normalized retrievability scoring function that tries to mitigate this effect by normalizing the retrievability scores of documents relative to their total number of queries. This provides an unbiased representation of the retrieval bias that could occurred due to vocabulary differences between the documents of collection without automatically inflicting a penalty on the retrieval models that favor or disfavor long documents. Finally, in order to examine, which retrievability scoring function has better effectiveness than other for correctly producing the retrievability ranks of documents, we perform a comparison between the both functions on the basis of known-items search method. Experiments on known-items search show that normalized retrievability

S. Bashir (✉)
Center for Science and Engineering, New York University Abu Dhabi,
Musaffah, Abu Dhabi, United Arab Emirates
e-mail: shariq.bashir@nyu.edu

A. S. Khattak
Natural Language Processing Research Group,
Department of Computer Science,
University of Leipzig, Leipzig, Germany
e-mail: akhattak@informatik.uni-leipzig.de

scoring function has better effectiveness than the standard retrievability scoring function.

Keywords Information systems evaluation · Documents accessibility · Documents findability · Known-items search · Patent retrieval · Recall-oriented retrieval

1 Introduction

Access to large information with the help of web and internet is playing an important part in the transformation of society. One important part of this overall process are information retrieval (IR) systems. IR systems deal with the storage (indexing), organization, management and retrieval of information (Baeza-Yates and Ribeiro-Neto 1999; Chowdhury 2004; Manning et al. 2008). After indexing, one important factor that shapes the access to information is the role of retrieval strategy (retrieval model) (Singhal 2001). It acts as a middleware between the users' required information and the users' effort to access the information. The main role of a retrieval model is to first discriminate between the relevant and irrelevant information, and then to display the relevant results to the users according to descending order of their relevance, so that the users' can view the most relevant information at the top most ranked positions. In past several years, a large number of retrieval models are proposed for the various kinds of retrieval tasks. One main problem that always remain in the IR researchers' attention is how to choose the right model according to given retrieval task. It is a tedious task, and falls under the research domain of evaluation of retrieval models (Harter and Hert 1997; Voorhees 2002; Voorhees and Harman 2005; Sanderson and Zobel 2005). Historically, research on the evaluation of retrieval models is always focused on either the effectiveness or the efficiency (speed). These are the only two measures that mostly remain in focus in the core research of IR community for determining the quality of retrieval models. The main limitation of these measures is that they focussed almost exclusively on only the set of few documents, i.e., the fact that the (most) relevant documents are returned at the top of ranked lists, as this constitutes the primary criterion of interest for most of standard retrieval tasks (web retrieval, question answering (Voorhees 2001), opinion retrieval (Ounis et al. 2006, etc). With evaluation measures such as recall, aspects of the completeness of information are being brought into the consideration. Recently based on the accessibility (retrievability, how easily the information can be accessed), a complementary and so-called higher order evaluation has been proposed. Instead of analyzing how well the system performs in terms of speed or effectiveness, the retrievability measure provides more fine grained indication of how easily the different information within the collection can be reached or access with the given retrieval models (Azzopardi and Vinay 2008). This offers a higher level and abstract level of view for understanding that what influence the given IR systems or retrieval models provide for accessing all relevant information in the collection, but not just the set of information that are given in the form of judged relevant documents by a group of few people. This is particularly important for the recall-oriented retrieval domains like patent or legal retrieval, where focus of retrieval is more given towards ensuring that everything relevant has been found and often seeks to demonstrate

that something (e.g. a document which invalidates a new patent application) does not exist (Arampatzis et al. 2007; Magdy and Jones 2010). Furthermore, it specifically examines whether the lack of access to information actually impedes one's ability to access the required information within the collection.

In this paper, we examine the retrieval bias of different retrieval models for different collections. The collections that we use for experiments contain patent and news documents. For these collections, we first determine the retrievability of documents with different retrieval models, and then we analyze how far these retrieval models are different in terms of retrieval bias that they imposed on the documents of collections. The overall retrievability of documents provides an indication of how easily the documents are accessible with different retrieval models. The overall retrievability inequality between the documents of collection shows retrieval bias of retrieval models.

The second major contribution of this paper is the introduction of normalized retrievability of documents relative to the total number of queries (query's set) of documents. The standard retrievability calculation function (Azzopardi and Vinay 2008) does not consider the differences in vocabulary richness across different documents. Documents having large vocabulary produce a large number of queries. Such documents thus have theoretically a large probability of retrievability via a much large number of queries. Due to this problem, the standard retrievability scoring function may favors long documents than short documents. We thus propose a normalized retrievability scoring function that considers this difference. This provides an unbiased representation of retrieval bias that arises due to vocabulary difference between the documents of collection without automatically inflicting a penalty on the retrieval models that favor or disfavor long documents. At the end a comparison is performed between the two retrievability scoring functions on the basis of known-item search method (Azzopardi et al. 2007). This comparison helps in analyzing that which function has better effectiveness than the other for producing better retrievability ranks of documents.

The remainder of this paper is structured as follow. Section 2 reviews related work on bias analysis of search systems. Section 3 provides a comprehensive introduction about retrievability measurement and its application to retrieval bias analysis of retrieval models. Section 4 explains datasets, retrieval models and mechanism for queries generation that we used for experiments. Section 5 starts with first describing a limitation of standard retrievability calculation function and then it introduces a normalized retrievability calculation function. Detailed retrievability analysis of both functions on all collections is presented in Section 6. In Section 7 we compare the effectiveness of both functions using known-items search method. Finally, Section 8 briefly summarizes key lessons learned from this study.

2 Related work

Due to novel and recently proposed domain, there is no extensive research done on the retrievability measure. However, in past there exist a number of studies on the web coverage of search engines, and these are somewhat related to this domain. In the following section, we provide a overview of the major works of both domains: (a) Web coverage based bias analysis, and (b) Retrievability based bias analysis.

2.1 Web coverage based bias analysis

Lawrence and Giles (1999) performed a study to analyze the coverage bias of web search engines. For this purpose they used 6 search engines and a large query log from a scientific organization. These queries should return the same set of pages for all 6 engines, as they thought that these engines have similar coverage since they are indexing the same set of documents. To express the coverage of the engines with respect to the size of the web, they used 128 million pages from Northern Light search engine at the time of their experiments as an absolute value. 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 only 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. A large number of queries were submitted to three search engines and their developed research crawler. The bias quantification was on the basis of site coverage ratio, and it was computed on 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 they did not have similar number of resources available as major search engines have.

Mowshowitz and Kawaguchi (2002) undertook a study to discover bias in fifteen major commercial search engines. In order to generate queries, they used the ACM computing classification system as queries, and the top 30 results of each search engine were recorded. Their large experiments results confirmed that there was some bias in all search engines. Their proposed bias measurement uses the number 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 introduce bias into the experiments as it is not based on all possible results of the web but only on the combinations of the web pages returned from the search engines. Secondly, their measurement cannot show if there is a bias against particular results if all of the included search engines are biased against similar results.

Lauw et al. (2006) found that deviation (controversy) in the evaluation scores of objects in the 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 biased if there is high deviation towards less controversial object. To identify this controversy and bias they proposed a reinforcement model. Their approach of discovering bias can also be applied in the web search setting. In this case the reviewers can be considered as web search engines and the objects that they are reviewing (ranking) are web pages. According to this approach, search engines will be more biased if they give high ranks to low ranked web pages of other search engines.

Owens (2009) conducted a recent study on the bias analysis of search engines. One major concern of their study was to discover whether the search engines unfairly lead users to particular sites over other sites. For this purpose they discovered the relative news bias of 3 search engines. They reported this relative bias amongst search engines in the forms of political bias and predilection for specific sites. They performed the experiments over 9 weeks, and posed a large number of realistic and currently topical queries to the news sections of 3 search engines. On the basis of their experiments results they showed that there are significant biases towards predilections for a certain news sources in all search engines.

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

2.2 Retrievability based bias analysis

Azzopardi and Vinay (2008) introduced a measure for the quantification of retrieval bias of retrieval models on the basis of accessibility of documents. It measures, how likely a document can be found at all by a specific retrieval model. Their retrievability experiments on AQUAINT and .GOV datasets revealed that with a TREC-style evaluation a proportion of the documents that have very low retrievability scores (sometimes more than 80 % of the documents in case of high retrieval bias models) can be removed without significantly degrading the effectiveness of retrieval models. This is because the retrieval models are unlikely to ever retrieve these documents due to the bias they exhibit on the documents of collection.

Bache and Azzopardi (2010) performed retrievability experiments on a patent collection. Their results confirmed the presence of a large amount of retrieval bias on the patent collection. In order to reduce this retrieval bias, they used a series of hybrid retrieval models. The features of these hybrid models were based on the term frequency sensitivity, length normalization and the convexity. Their results showed that the hybrid models provide greater access to the documents than the standard retrieval techniques (BM25 and TFIDF).

Similar to Azzopardi and Vinay (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, their evaluation is based on using a rather limited set of queries. Their experiments revealed that 90 % of documents that are highly retrievable across all types of queries are not highly retrievable when they are searched from relevant queries.

Experiments on query expansion based approaches for improving documents retrievability are thoroughly investigated in Bashir and Rauber (2009b, 2010b). In these studies, authors concluded that short queries are not efficient for correctly

capturing and interpreting the context of required search. Therefore, noisy documents at higher rank positions drift the retrievability results to fewer documents, creating a higher retrieval bias. To overcome this limitation, they proposed techniques to select relevant documents for pseudo-relevance feedback on the basis of documents clustering (Bashir and Rauber 2009b) and term-proximity based methods (Bashir and Rauber 2010b). Their 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 retrieval bias.

Bashir and Rauber (2010a) proposed an approach for improving the retrievability of documents on the basis of low and high retrievable corpus partitioning. In this approach, rather than retrieving and ranking documents from a single corpus they first split the two categories of documents *low* and *high* retrievable documents into two partitions. Having splitting the corpus into these two categories they then perform retrieval by treating these classes as independent partitions, and process queries independently for each partition and subsequently combining the result sets. Their results showed that this helps in increasing overall retrievability, reducing the dominance of certain documents in query processing and thus reducing the bias of retrieval models.

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 motivate 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.

3 Retrievability measurement

The following description of retrievability measurement as introduced by Azzopardi and Vinay (2008) (adopted from Bashir and Rauber 2010a) provides a quick introduction of how it is measured.

Given a collection D, a retrieval models accepts a user query q and returns a ranked list of documents, which are deemed to be relevant to q. We can thus consider the retrievability of a document as a two system dependent factors, (a) how retrievable it is, with respect to the collection D, and (b) the effectiveness of the ranking strategy of the retrieval model. In order to derive an estimate of this quantity, Azzopardi and Vinay (2008) in their experiments used query-set based sampling approach (Callan and Connell 2001). Q the query set could be either a historical sample of queries or a artificial simulated substitute similar to users' queries. Then, each $q \in Q$ is issued to retrieval model, and the retrieved documents along with their positions in the ranked list are recorded. Intuitively, retrievability of a document d to be high when:

1. There are many probable queries in Q which can be expressed in order to retrieve d, and

2. when retrieved, the rank r of the document d is low than a rank cutoff (threshold) c. This is the point at which the user would stop examining the ranked list. This is a user dependent factor, and thus reflects a particular retrieval scenario for obtaining a more accurate estimate of this measure. For instance, in web-search scenario a low c would be more accurate as users are unlikely to go beyond the first page of the results, while in the context of recall-oriented retrieval settings (for instance, legal or patent retrieval), a high c would be more accurate.

Thus based on the Q, r and c, we formulate the following measure for the retrievability of d.

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

$f(k_{dq}, c)$ is a generalized utility/cost function, where k_{dq} is the rank of d in the result list of query q. c denotes the maximum rank that a user is willing to proceed down in the ranked list. The function $\hat{f}(k_{dq}, c)$ returns a value of 1, if $k_{dq} \leq c$, and 0 otherwise. Defined in this way, the retrievability of a document is essentially a cumulative score that is proportional to the number of times the document can be retrieved within that cutoff c over the set Q. This fulfills our aim, in that the value of $r(d)$ would be high when there are a large number of highly probable queries that can retrieve the document d at the rank less than c, and the value of $r(d)$ would be low when only a few number of queries retrieve the document. Furthermore, if a document is never returned at the top ranked c positions, possibly because it is difficult to retrieve by the retrieval model, then it has $r(d) = 0$.

The cumulative measure of retrievability on the basis of binary $f(k_{dq}, c)$ function ignores the ranking positions of documents in the ranked list, i.e. how accessible the documents are in the ranking. Gravity based measure can be used for this purpose by setting the function to reflect the effort of going further down in the ranked list, and it is defined as

$$\hat{f}(k_{dq}, \beta) = \frac{1}{(k_{dq})^{\beta}} \tag{2}$$

The rank cutoff factor is changed to β which is a dampening factor that adjusts how accessible the document is in the ranking. In our experiments we score the retrievability of documents only on the basis of cumulative measure.

Retrievability inequality between documents can be further analyzed using *Lorenz Curve* (Gastwirth 1972). In Economics and the Social Sciences, Lorenz Curve is used to visualize the inequality of wealth in a population. This is performed 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 the equality is reflected by the amount of skewness in the distribution. Azzopardi and Vinay (2008) employed similar idea in the context of a collection of documents, and the wealth of documents are represented by $r(d)$ function. The more skewed the plot, the greater the amount of inequality, or (retrieval) bias within the collection. The *Gini–Coefficient*

Table 1 The properties of document collections that are used for the retrieval bias analysis

Dataset	Total docs.	Seed docs.	Rank cutoff factors
TREC-CRT	1.2 million	34,205	50,100,150,200,250
ChemAppPat	36,998	36,998	5,10,15,20,25
DentPat	27,988	27,988	5,10,15,20,25
ATNews	47,693	47,693	5,10,15,20,25

Seed docs: = This is the set of documents that are used for query generation and retrievability analysis

(Gastwirth 1972) G is used to summarize the amount of retrieval bias in the Lorenz Curve and provides bird's eye view. It is computed as follows.

$$G = \frac{\sum_{i=1}^{|D|} \left(2 \cdot i - |D| - 1\right) \cdot r(d_i)}{(|D| - 1) \sum_{j=1}^{|D|} r(d_j)} \tag{3}$$

D represents the set of documents in the collection. 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 of different retrieval methods, we can analyze the retrieval bias imposed by the underlying retrieval systems on a given document collection.

4 Experimental set-up

4.1 Document collections

We use the following four collections (Table 1) for the retrieval bias analysis. Table 1 presents the basis properties of these collections. Seed documents represent the set of those documents that are used for query generation and retrievability analysis.

- **TREC 2009 Chemical Retrieval Track Collection:** This collection consists of 1.2 million patent documents from the TREC Chemical Retrieval Track (2009) (*TREC-CRT*)[1] (Lupu et al. 2009). Due to the large size of collection, determining the retrievability for all documents of collection requires large processing time and resources. Thus in order to complete the experiments in a reasonable time, a subset of *34,205* documents (judged documents) for which the relevance assessments are available as part of *TREC-CRT* serve as seed for query generation and retrievability analysis. As compared to other three collections, the documents in this collection are very long. The distributions of document length and vocabulary size are also highly skewed (see Fig. 1). For this collection, retrieval bias is analyzed with five rank cutoff factors $c = 50$, $c = 100$, $c = 150$, $c = 200$, and $c = 150$.

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

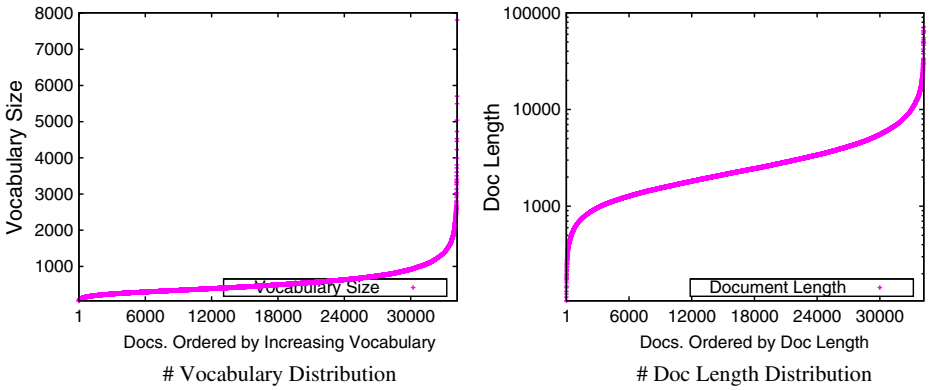


Fig. 1 Documents vocabulary and length distributions for the *TREC-CRT* collection

- **USPTO Patent Collections:** These collections are downloaded from the freely available US patent and trademark office website.[2] We collect all patents that are listed under the United State Patent Classification (USPC) classes *433 (Dentistry)*, and *422 (Chemical apparatus and process disinfecting, deodorizing, preserving, or sterilizing)*. These collections consist of 64,986 documents, with 36,998 documents in *USPC Class 422* and 27,988 documents in *USPC Class 433*. The *USPC Class 433* documents are called with *DentPat Collection*, and the *USPC Class 422* documents are called with *ChemAppPat Collection*. The patent numbers of these collections are available at.[3] Similar to the *TREC-CRT* collection, the documents in this collection are long, however, the distributions of documents length and vocabulary size are less skewed than the *TREC-CRT* collection (see Figs. 2 and 3). For both collections the retrieval bias is analyzed with the rank cutoff factors $c = 5$, $c = 10$, $c = 15$, $c = 20$ and $c = 25$.
- **Austrian News Dataset:** Our final collection consists of 47,693 Austrian news documents.[4] We call this collection *(ATNews Collection)*. As compared to above three collections, the documents in this collection are mostly short, however, the distributions of document length and vocabulary size are skewed similar to the *TREC-CRT* collection (see Fig. 4). For this collection we use the rank cutoff factors $c = 5$, $c = 10$, $c = 15$, $c = 20$ and $c = 25$ for the retrieval bias analysis.

4.2 Retrieval models

Four standard IR models and four different variations of language models with term smoothing are used for retrieval bias analysis. These are standard TFIDF, NormTFIDF, the OKAPI retrieval model BM25, SMART, Jelinek–Mercer language model JM, Dirichlet (Bayesian) language model DirS, Absolute Discounting language model, and TwoStage language model.

[2] Available at http://www.uspto.gov/.

[3] http://www.ifs.tuwien.ac.at/~bashir/Analyzing_Retrievability.htm

[4] http://www.ifs.tuwien.ac.at/~andi/tmp/STANDARD.tgz

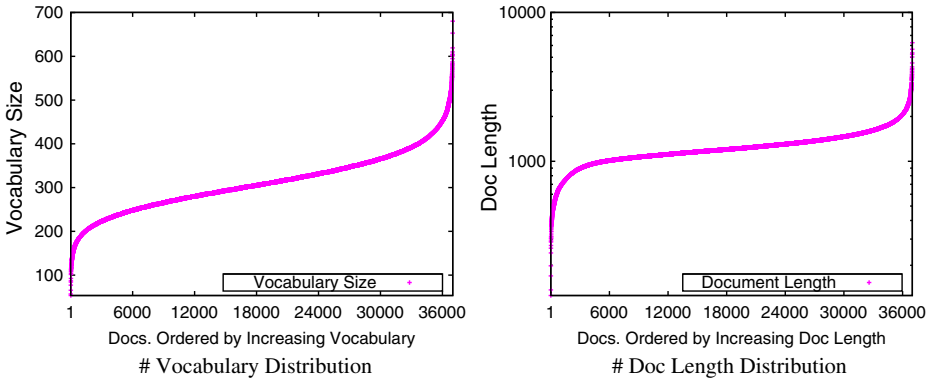


Fig. 2 Documents vocabulary and length distributions for the *ChemAppPat* collection

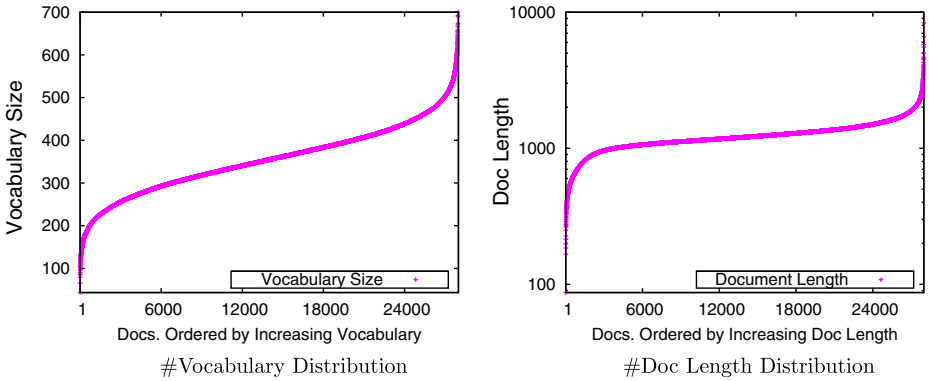


Fig. 3 Documents vocabulary and length distributions for the *DentPat* collection

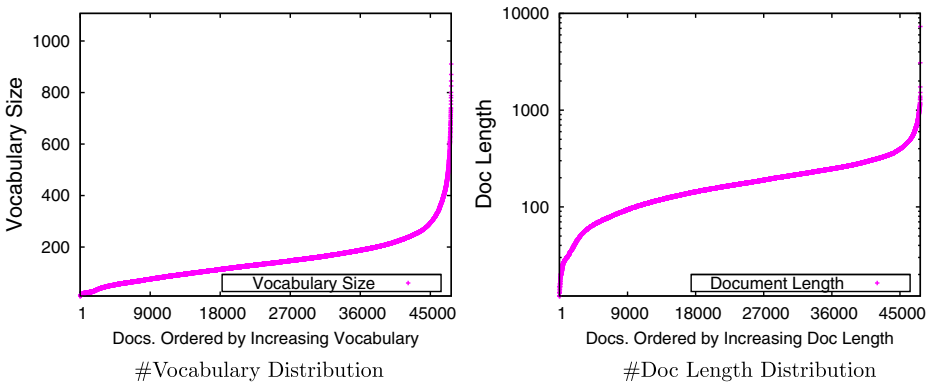


Fig. 4 Documents vocabulary and length distributions for the *ATNews* collection

4.2.1 Standard retrieval models

- **TFIDF:** The TFIDF (term frequency inverse document frequency) is a retrieval model often used in information retrieval. It is a statistical measure used to evaluate how important a query terms is to a document. The importance increases proportionally to the number of times a term appears in the document but is offset by the frequency of the term in the collection. The standard TFIDF retrieval model is described as follow:

$$TFIDF(d,q) = \sum_{t \in q} tf_{t,d} \log \frac{|D|}{df_t} \tag{4}$$

  $tf_{t,d}$ is the term frequency of query term t in d, and $|D|$ is the total number of documents in the collection. df_t represents the total number of documents containing t.
- **NormTFIDF:** The standard TFIDF does not normalize the term frequencies relative to document length, thus sensitive and bias towards large absolute term frequencies. It is possible to address the length bias by using document length $|d|$, and defied normalized TFIDF (NormTFIDF) as:

$$NormTFIDF(d,q) = \sum_{t \in q} \frac{t_{t,d}}{|d|} \log \frac{|D|}{df_t} \tag{5}$$

- **BM25:** Okapi BM25 arguably one of the most important and widely used information retrieval model. It is a probabilistic function and nonlinear combination of three key attributes of a document: term frequency $t_{t,d}$, document frequency df_t, and the document length $|d|$. The effectiveness of BM25 is controlled by two parameters k and b. These parameters control the contributions of term frequency and document length. If $k = 0$, the function reduces to 1 and the relevance scores of documents are calculated solely based on the occurrences of query terms across the collection only. The large value of k makes the function nearly linear in $tf_{t,d}$. Typically k is used with $k = 2.0$. This demonstrates the nonlinear contribution of $tf_{t,d}$ to the final document relevance scores. The parameter b controls the length normalization. It is set between 0 and 1. Large values of b (close to 1) simply make high normalization, thus short documents are more favored over long documents. While if the values are small or b approaches to zero, then the effect of normalization becomes small, and long documents are more favored over short documents due to the their large absolute term frequencies. We used the following standard function of BM25 proposed by Robertson and Walker (1994):

$$BM25(d,q) = \sum_{t \in q} \log \frac{|D| - df_t + 0.5}{df_t + 0.5} \frac{tf_{t,d}(k+1)}{tf_{t,d} + k\left(1 - b + b\frac{|d|}{|\overline{d}|}\right)} \tag{6}$$

  $|\overline{d}|$ is the average document length in the collection from which the documents are drawn. k and b are two parameters, and they are used with $k = 2.0$ and $b = 0.75$.

- **SMART:** The System for Manipulating and Retrieving Text (SMART) is a retrieval model in information retrieval. It is based on the Vector Space Model. We use the following variation of SMART developed by Singhal (1997) at AT&T Labs.

$$SMART(d, q) = \sum_{t \in q} \left(w_d * w_q\right) \tag{7}$$

$$w_d = \frac{1 + \log(tf_{t,d})}{1 + \log(avtf)} * \frac{1}{0.8 + 0.2\frac{utf}{pivot}} \tag{8}$$

$$w_q = (1 + \log(tf_{t,d})) * \log \frac{|D| + 1}{df_t} \tag{9}$$

  $avtf$ represents the average number of occurrences of each term in the d, utf is the number of unique terms in d, and $pivot$ represents the average number of unique terms per document.

4.2.2 Language models with term smoothing

Language model tries to estimate the relevance of document by estimating the probabilities of terms in the document. The terms are assumed to occur independently, and the probability is the product of the individual query's terms given the document model M_d of document d:

$$P(q|M_d) = \prod_{t \in q} P(t|M_d) \tag{10}$$

$$P(t|M_d) = \frac{tf_{t,d}}{|d|} \tag{11}$$

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 of missing only a few or single term. For dealing with this, language models make use of smoothing to balance the probability mass between the occurrences of terms present in documents, and the terms not found in the documents. We use the following four variations of terms smoothing in our experiments.

- **Jelinek–Mercer Smoothing (JM):** Jelinek–Mercer smoothing (Zhai 2002) combines the relative frequency of a query's term $t \in q$ in the document d with the relative frequency of the term in the collection (D). The amount of smoothing is controlled by the λ, and it is set between 0 and 1. Small smoothing values of λ close to 0 add only the contribution of term frequencies. Thus every single match receives a high boost. Note that the term frequencies are normalized by the document length and, therefore, the short documents in case of boolean AND queries might have high values of $\frac{f_{(d,q_w)}}{|d|}$ than the long documents. However, in case of long (boolean OR queries) the long documents might have high overall relevance scores than the short documents due to covering more query's material. Large values λ make large smoothing, and this reduces the effect of relative term frequencies within the documents, and more importance is given towards the relative frequencies of terms in the collection. This results in both

long and short document being favored. In case of $\lambda = 1$ the length of the query becomes less important and the frequencies of the terms across the collection dominate the retrieval model.

$$P(t|M_d) = (1-\lambda)\frac{tf_{t,d}}{|d|} + \lambda P(t|D) \tag{12}$$

$P(t|D)$ is the probability of term t occurring in the collection ($\sum_{d\in D} tf_{t,d}/\sum_{d\in D}|d|$). According to the suggested value of λ by Zhai (2002), we use (λ with 0.7).

- **Dirichlet (Bayesian) Smoothing (DirS):** This smoothing technique makes smoothing dependent on the document length. Since long documents allow us to estimate the language model more accurately, therefore this technique smoothes them less, and this is done with the help of a parameter μ. Since the value of μ is added in the document length, thus small values of μ retrieve less long documents. If the μ is used with large values, then the distinction for difference between document lengths becomes less extreme, and long documents are more favored over short documents. Again, this favoritism mostly occurs in case of long boolean OR queries.

$$P(t|M_d) = \frac{tf_{t,d} + \mu P(t|D)}{|d| + \mu} \tag{13}$$

According to Zhai (2002) suggestion, we use the μ with 2,000.

- **Two-Stage Smoothing (Two-Stage):** This smoothing technique first smoothes the document model using the Dirichlet prior probability with the parameter μ (as explained above), and then, it mixes the document model with the query background model using Jelinek–Mercer smoothing with the parameter λ. The query background model is based upon the term frequency in the collection. The smoothing function is therefore:

$$P(t|M_d) = (1-\lambda)\frac{tf_{t,d} + \mu P(t|D)}{|d| + \mu} + \lambda P(t|D) \tag{14}$$

Where μ is the Dirichlet prior probability and λ is the Jelinek–Mercer parameter. In our experiments, we use the parameter $\mu = 2{,}000$ and $\lambda = 0.7$ respectively.

- **Absolute Discount Smoothing (AbsDis):** This technique makes smoothing by subtracting a constant $\delta \in [0, 1]$ from the counts of each seen term. The effect of δ is similar to Jelinek–Mercer parameter λ, but differs in this sense that it discounts the seen terms probabilities by subtracting a constant δ instead of multiplying them by $(1-\lambda)$.

$$P(t|M_d) = \frac{\max(tf_{t,d} - \delta, 0)}{|d|} + \frac{\delta|T_d|}{|d|}P(t|D) \tag{15}$$

T_d is the set of all unique terms of d. We use the δ with 0.7.

4.3 Generating queries for retrievability analysis

We consider all sections (title, abstract, claims, description, background summary) of patent documents for both retrieval and query generation. Stop words are removed

Table 2 Properties of Q that are used for the retrieval bias analysis

Characteristics	TREC-CRT	ChemAppPat	DentPat	ATNews
$\|Q\|$	30 Million	30 Million	30 Million	30 Million
Minimum query result list size	100	45	45	45
Avg query result list size	1,636	156	161	87
Avg # of queries/document	531,442	127,032	173,516	54,551

prior to indexing and words stemming is performed with Porter stemming algorithm. Additionally, we do not use all those terms of the collection that have document frequency greater than 25 % relative to the total collection size. Next, queries for retrievability analysis are generated with the combinations of those terms that appear more than one time in the documents. For these terms, all 3-terms and 4-terms combinations are used in the form of boolean AND queries for creating the exhaustive set of queries Q, and duplicate queries are removed from the Q.

As we explained in Section 3, a third factor that effects the retrievability of documents along with the user ability to formulate the query and the retrieval bias of retrieval model is the difference between the result list size of the query and the user's ability that how much deeply he/she would check/read the retrieved documents of the query. In retrievability measurement this difference is controlled with a rank cutoff factor. The high difference implies that the user would go through only a small portion of the retrieved documents, and thus we can expect to this that the retrievability of documents would be highly depend upon the retrieval bias of retrieval model. Low bias of retrieval model would make a large number of documents high retrievable at top ranked positions. On the other hand, if this difference is small, or the size of query result lists become less than the rank cutoff factor, then the user would go through a large portions of documents and thus the bias of retrieval models would create low effect on the retrievability of documents.

Therefore, in order to precisely analyze the effect of retrieval bias of retrieval models, the size of query result lists nor should be too close to the user's rank cutoff neither should be too large. Large result lists of queries represent the frequent terms combinations of the collection, and the users would rarely use them for searching their information. Therefore, in order to reasonably approximating the retrieval bias, we remove all those queries from Q that either retrieve only a few number of documents or retrieve very large number of documents. Under this principle, for the *TREC-CRT* collection, we remove all those queries from the Q that retrieve less than 100 documents. Similarly for the *ChemAppPat*, *DentPat* and *ATNews* collections we remove all those queries from the Q that retrieve less than 45 documents. Next, we order all queries in Q on the basis of increasing result list sizes, and select only top 30 *million* queries (low frequent terms combinations) for the documents retrieval against the complete collection as boolean AND queries with subsequent ranking according to the chosen retrieval models to determine the retrievability scores of documents.[5] Table 2 shows the general characteristics of Q for different collections. Figures 5, 6, 7, 8 show the distributions of the total number of queries

[5]The complete query set for all collections are available at http://www.ifs.tuwien.ac.at/~bashir/Analyzing_Retrievability.htm.

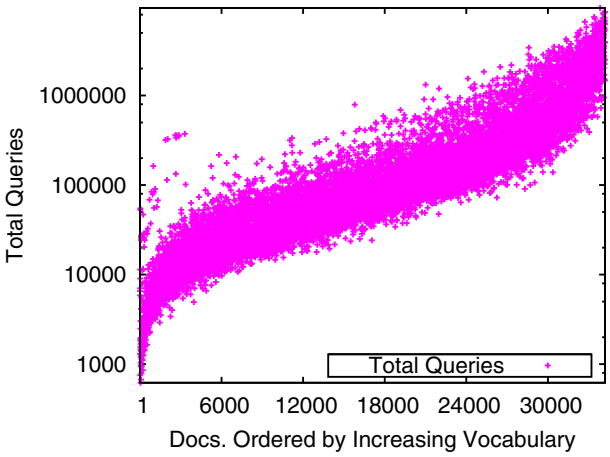


Fig. 5 The distribution of total number of queries per document for the *TREC-CRT* collection. Documents are ordered on the basis of increasing vocabulary size

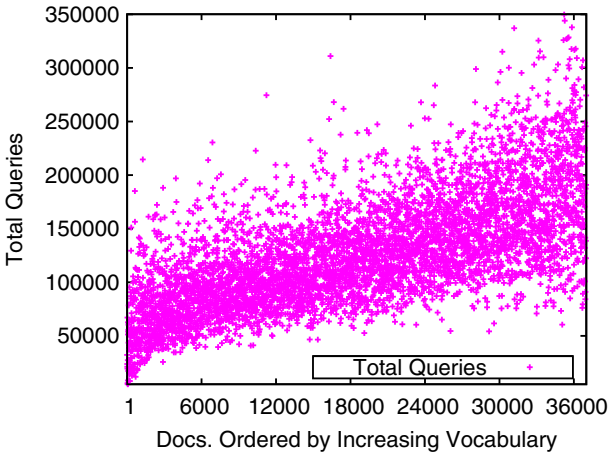


Fig. 6 The distribution of total number of queries per document for the *ChemAppPat* collection. Documents are ordered on the basis of increasing vocabulary size

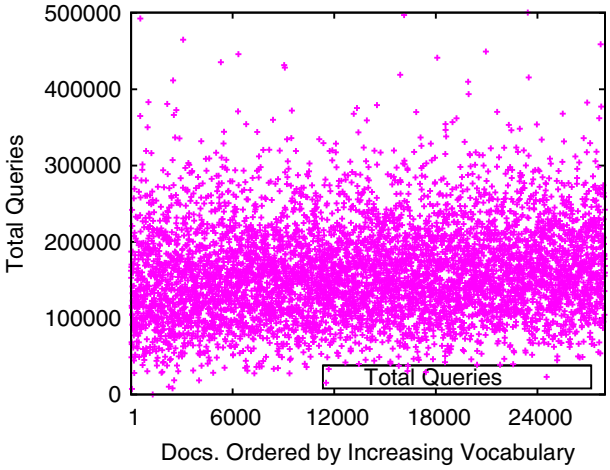


Fig. 7 The distribution of total number of queries per document for the *DentPat* collection. Documents are ordered on the basis of increasing vocabulary size

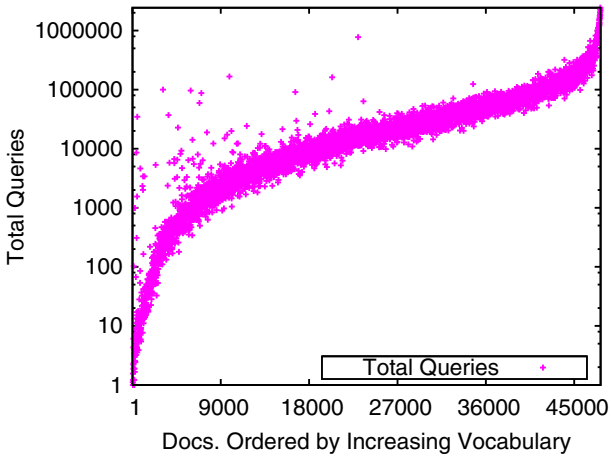


Fig. 8 The distribution of total number of queries per document for the *ATNews* collection. Documents are ordered on the basis of increasing vocabulary size

per document relative to the vocabulary size of documents. The *TREC-CRT* and *ATNews* collections have large difference between documents vocabulary sizes, thus for these collections the queries distributions are highly skewed. The *ChemAppPat* and *DentPat* collections have less difference between documents vocabulary sizes, thus for these collections the distributions of queries are less skewed.

5 Normalized retrievability scoring function

The retrievability measure that is defined in Section 3 cumulates the retrievability scores of documents over all queries. Thus, in case of exhaustive query generation, long documents potentially have high number of query combinations possible than short documents due to large vocabulary sizes. Figures 5–8 show the distribution of the total number of queries for different documents. Long documents have large query sets than short documents. This may favor long documents that are retrievable from only a small fraction of their all possible queries than the short documents that are potentially retrievable from a large faction of their queries. To understand this phenomena, let us consider the example presented in Table 3 with 6 documents and their estimated $r(d)$ scores from three different retrieval models (A,B,C). Doc1, Doc2 and Doc4 are long documents than Doc3, Doc5 and Doc6, therefore these documents have large query combinations. (For the context of this example, we are assuming only the 3-terms queries). From the Table 3 it can be easily inferred that in terms of percentage of documents retrievable from their all possible query combinations, the retrieval model B is better than the retrieval model A, and the retrieval model C is better than the retrieval model B. Therefore after retrieval bias computation, the retrieval model C and the retrieval model B should show low retrieval bias than retrieval model A. However, using the standard retrievability calculation function, the retrieval model A is wrongly showing low Gini–Coefficient (representing retrieval bias) than the retrieval model B, and accordingly the retrieval model B is wrongly showing low Gini–Coefficient than the retrieval model C. This happened due to ignoring the difference between the vocabulary richness of documents while computing the retrieval bias. We thus propose to normalize the

Table 3 Retrieval bias representation with $r(d)$ and $\hat{r}(d)$

Docs.	Unique terms	Total queries	IR model A	IR model B	IR model 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		$G = 0.48$	$G = 0.08$	$G = 0$

G refers to Gini–Coefficient value

cumulative retrievability scores (normalized retrievability) of documents with the total number of queries they were created from, and thus potentially can retrieve a particular document, and it is defined as:

$$\hat{r}(d) = \frac{\sum_{q \in Q} f(k_{dq}, c)}{|\hat{Q_{(d)}}|} \tag{16}$$

The cumulative $r(d)$ scores of documents are normalized with $\hat{Q_{(d)}}$. This is the set of all queries that can retrieve d when not considering any rank cutoff factor. This accounts for difference in the vocabulary richness across different documents of collection. Documents having large vocabulary size produce many more queries. Such documents are thus theoretically retrievable via a much large set of queries. The standard $r(d)$ score would thus penalize a retrieval model that provides perfectly balanced retrievability to all documents just because some documents are rather vocabulary-poor and cannot be retrieved by more than a few number of queries that can be created from their vocabulary. This is where a normalized retrievability score accounts for the different vocabulary sizes per document, and it provides an unbiased representation of the retrieval bias without automatically inflicting a penalty on the retrieval models that favor or disfavor long documents. Table 3 shows how the normalized retrievability provides a more realistic estimate of the retrieval bias of retrieval models. Now retrieval model C is correctly showing low retrieval bias than the retrieval model B, and accordingly retrieval model B is showing low retrieval bias than the retrieval model A.

6 Retrieval bias analysis

Tables 4, 5, 6 and 7 list the retrievability inequality providing Gini–Coefficients for a range of rank cutoff factors for different collections. Note that the high bias is

Table 4 Gini–Coefficient scores representing the retrieval bias of different retrieval models on various rank cutoff factors for the *TREC-CRT* collection

Retrieval model	$\hat{r}(d)$			$r(d)$		
	$c = 50$	$c = 100$	$c = 250$	$c = 50$	$c = 100$	$c = 250$
NormTFIDF	0.68	0.60	0.48	0.83	0.81	0.77
BM25	0.51	0.47	0.40	0.77	0.77	0.76
DirS	0.57	0.52	0.44	0.77	0.76	0.74
JM	0.65	0.58	0.48	0.77	0.76	0.74
AbsDis	0.61	0.55	0.46	0.76	0.75	0.74
TwoStage	0.58	0.52	0.42	0.80	0.78	0.76
TFIDF	0.89	0.84	0.74	0.97	0.96	0.94
SMART	0.95	0.92	0.85	0.98	0.97	0.95

As rank cutoff factor increases, bias steadily decreases indicating that the low retrieval bias is experienced when considering the long ranked lists

experienced when limiting oneself to short result lists of 5 or 50 documents. The Gini–Coefficient tends to decrease slowly for all query sets and for all retrieval models as the rank cutoff factor increases. This indicates that the retrievability inequality within the collection is mitigated by the willingness of the users to search deeper down into the result list. If user examines only a few portion of the result list, then he/she will face a greater degree of retrieval bias.

Overall, *BM25* on all collections exhibits low retrieval bias than all other retrieval models. The four language modeling approaches (*DirS*, *TwoStage*, *JM*, *AbsDis*) also exhibit low retrieval bias than *TFIDF*, *SMART* and *NormTFIDF*.

6.1 Comparing $r(d)$ and $\hat{r}(d)$

In order to analyze the difference between $r(d)$ and $\hat{r}(d)$, we examine the relationship between the document retrievability scores produced from both functions with respect to the document length and vocabulary size. This is done by dividing the collection into a number of subsets on the basis of document length and vocabulary size.

Table 5 Gini–Coefficient scores representing the retrieval bias of different retrieval models on various rank cutoff factors for the *ChemAppPat* collection

Retrieval model	$\hat{r}(d)$			$r(d)$		
	$c = 5$	$c = 10$	$c = 25$	$c = 5$	$c = 10$	$c = 25$
NormTFIDF	0.52	0.43	0.30	0.52	0.45	0.39
BM25	0.38	0.33	0.25	0.39	0.38	0.37
DirS	0.44	0.37	0.27	0.44	0.40	0.37
JM	0.48	0.40	0.29	0.41	0.37	0.36
AbsDis	0.42	0.35	0.26	0.39	0.38	0.38
TwoStage	0.46	0.38	0.27	0.48	0.42	0.38
TFIDF	0.61	0.50	0.35	0.67	0.59	0.49
SMART	0.46	0.38	0.27	0.93	0.89	0.79

As rank cutoff factor increases, bias steadily decreases indicating that the low retrieval bias is experienced when considering the long ranked lists

Table 6 Gini–Coefficient scores representing the retrieval bias of different retrieval models on various rank cutoff factors for the *DentPat* collection

Retrieval model	$\hat{r}(d)$			$r(d)$		
	$c = 5$	$c = 10$	$c = 25$	$c = 5$	$c = 10$	$c = 25$
NormTFIDF	0.54	0.45	0.31	0.53	0.46	0.40
BM25	0.40	0.34	0.26	0.40	0.38	0.37
DirS	0.45	0.38	0.28	0.46	0.42	0.38
JM	0.49	0.42	0.30	0.43	0.39	0.36
AbsDis	0.43	0.36	0.27	0.41	0.39	0.38
TwoStage	0.47	0.39	0.28	0.49	0.44	0.38
TFIDF	0.62	0.52	0.36	0.68	0.60	0.50
SMART	0.92	0.86	0.72	0.93	0.89	0.79

As rank cutoff factor increases, bias steadily decreases indicating that the low retrieval bias is experienced when considering the long ranked lists

Before going to analysis, it is important to mention here that the $\hat{r}(d)$ does not provide a different estimate for the retrievability. The only major difference between the both functions is that $r(d)$ measures retrievability without considering diversity in the 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 high for the vocabulary-rich documents. $\hat{r}(d)$ specifically pushes the retrievability ranks of all those low retrievable documents toward high ranks positions (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 high findable by these few queries.

Figures 9, 10, 11, 12, 13, 14, 15 and 16 show the graphical relationship between $r(d)$ and $\hat{r}(d)$ with respect to document length and document vocabulary size.Table 8 shows the correlation between the document retrievability scores of both functions using Spearman's rank correlation coefficient. The results indicate that when there exists a large diversity between the documents length and the vocabulary size, then the correlation between $r(d)$ and $\hat{r}(d)$ is low. Such example can be observed from the *TREC-CRT* and the *ATNews* collections. On both collections, the $r(d)$

Table 7 Gini–Coefficient scores representing the retrieval bias of different retrieval models on various rank cutoff factors for the *ATNews* collection

Retrieval model	$\hat{r}(d)$			$r(d)$		
	$c = 5$	$c = 10$	$c = 25$	$c = 5$	$c = 10$	$c = 25$
NormTFIDF	0.50	0.39	0.21	0.54	0.53	0.57
BM25	0.49	0.38	0.21	0.52	0.52	0.57
DirS	0.43	0.33	0.20	0.77	0.73	0.69
JM	0.50	0.38	0.20	0.53	0.52	0.56
AbsDis	0.51	0.40	0.23	0.56	0.57	0.61
TwoStage	0.42	0.32	0.20	0.78	0.75	0.70
TFIDF	0.68	0.56	0.36	0.95	0.92	0.87
SMART	0.72	0.59	0.33	0.87	0.83	0.73

As rank cutoff factor increases, bias steadily decreases indicating that the low retrieval bias is experienced when considering the long ranked lists

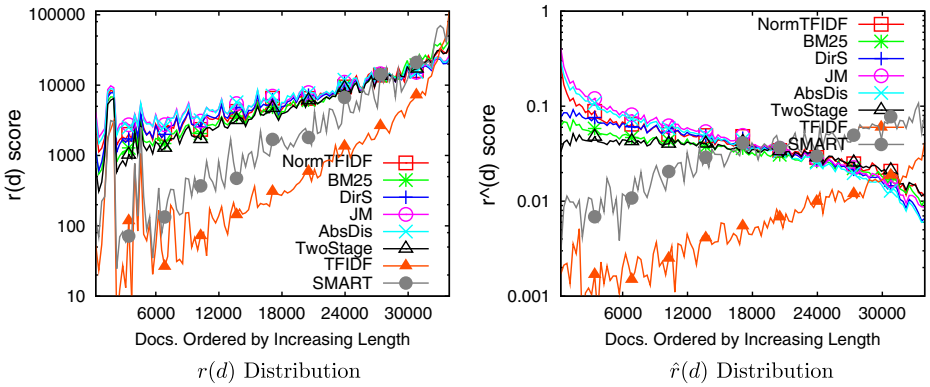


Fig. 9 Relationship between $r(d)$ and $\hat{r}(d)$ on the *TREC-CRT* collection on the basis of increasing document length

scores of documents with all retrieval models have positive correlation with the document length and the vocabulary size. This indicates that in the case of $r(d)$, long documents have high retrievability than the short documents. This happened due to generation of large number of queries in case of long documents than short documents. On the other side, $\hat{r}(d)$ has negative correlation with the document length and the vocabulary size. This observation is more visible on (*BM25*, *JM*, *DirS*, *TwoStage*, *AbsDis*, and *NormTFIDF*) models. This indicates that for these models long documents are not retrievable with a good percentage from their total queries than short documents. Thus most of long documents have low $\hat{r}(d)$ scores than short documents. However, in the case of *SMART* and *TFIDF*, $\hat{r}(d)$ also has positive correlation with the document length and the vocabulary size. This is because, these models do not normalize term frequencies relative to document length, and thus the large absolute term frequencies of query terms are preferred over small absolute term frequencies. This is the reason why the long documents using these models also have high percentage of retrievability out of their total queries.

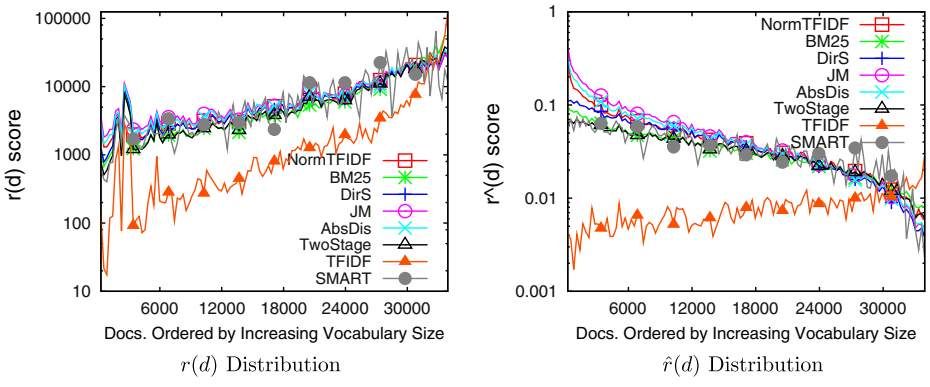
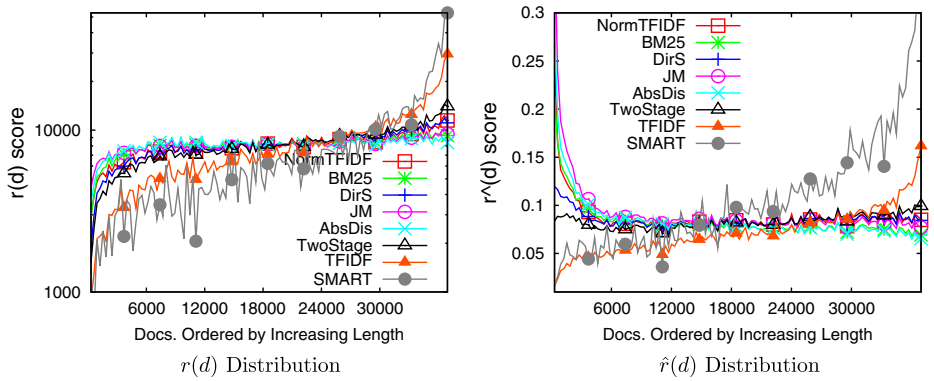
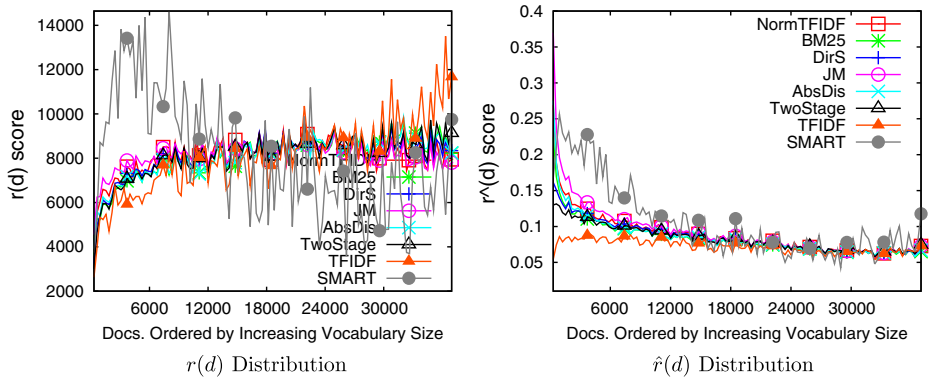


Fig. 10 Relationship between $r(d)$ and $\hat{r}(d)$ on the *TREC-CRT* collection on the basis of increasing document vocabulary size



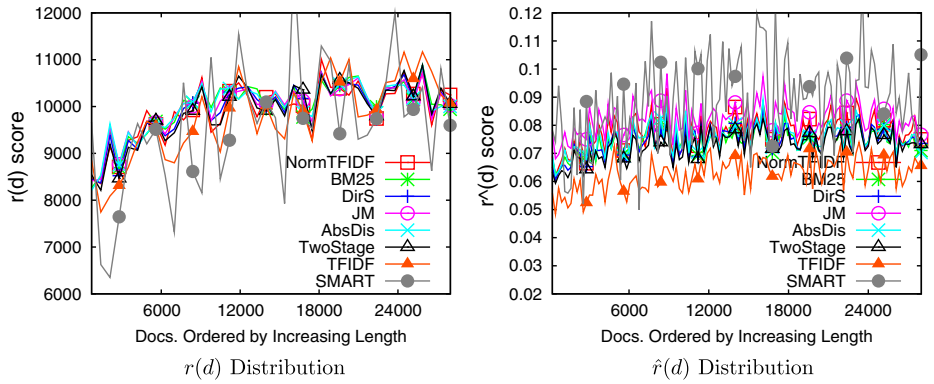
$r(d)$ Distribution $\hat{r}(d)$ Distribution

Fig. 11 Relationship between $r(d)$ and $\hat{r}(d)$ on the *ChemAppPat* collection on the basis of increasing document length



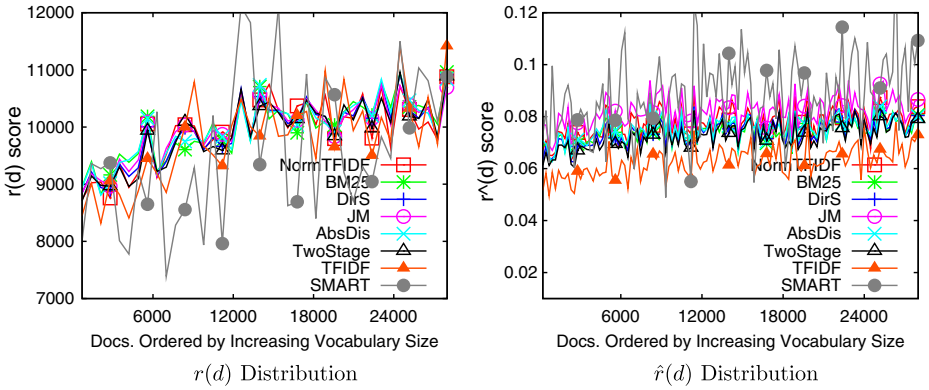
$r(d)$ Distribution $\hat{r}(d)$ Distribution

Fig. 12 Relationship between $r(d)$ and $\hat{r}(d)$ on the *ChemAppPat* collection on the basis of increasing document vocabulary size



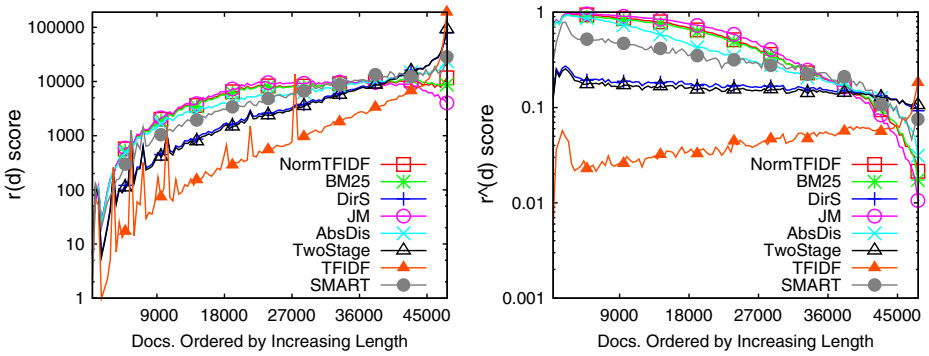
$r(d)$ Distribution $\hat{r}(d)$ Distribution

Fig. 13 Relationship between $r(d)$ and $\hat{r}(d)$ on the *DentPat* collection on the basis of increasing document length



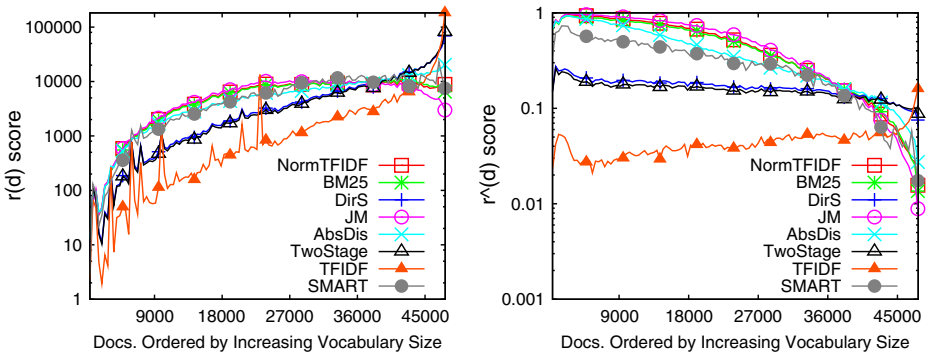
$r(d)$ Distribution $\hat{r}(d)$ Distribution

Fig. 14 Relationship between $r(d)$ and $\hat{r}(d)$ on the *DentPat* collection on the basis of increasing document vocabulary size



$r(d)$ Distribution $\hat{r}(d)$ Distribution

Fig. 15 Relationship between $r(d)$ and $\hat{r}(d)$ on the *ATNews* collection on the basis of increasing document length



$r(d)$ Distribution $\hat{r}(d)$ Distribution

Fig. 16 Relationship between $r(d)$ and $\hat{r}(d)$ on the *ATNews* collection on the basis of increasing document vocabulary size

Table 8 Correlation between $\hat{r}(d)$ and $r(d)$ on the basis of document retrievability scores

Retrieval model	TREC-CRT	ChemAppPat	DentPat	ATNews
NormTFIDF	0.40	0.51	0.64	−0.46
BM25	0.09	0.19	0.40	−0.51
DirS	0.17	0.35	0.54	0.03
JM	0.23	0.33	0.52	−0.41
AbsDis	0.14	0.25	0.45	−0.58
TwoStage	0.29	0.42	0.58	0.05
TFIDF	0.90	0.74	0.81	0.68
SMART	0.82	0.90	0.80	0.58

The above stated difference between the $r(d)$ and $\hat{r}(d)$ becomes small when either the diversity between the document length and the vocabulary sizes decreased or the total number of queries for short and long documents become equal. This observation can be seen by looking at the results of *DentPat* and *ChemAppPat* collections. This indicates that if the given collection contains a large diversity between documents in terms of their lengths or vocabulary sizes, then we can expect to that there would be a large difference between the $r(d)$ 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.

7 Analyzing $r(d)$ and $\hat{r}(d)$ effectiveness using known-items search method

In above section we analyze the retrieval biases of retrieval models using two retrievability scoring functions. If we compare both functions only on the basis of Gini–Coefficients, then it is not clear that which retrievability scoring function is more efficient than the other for correctly producing the retrievability ranks of documents. In order to examine their effectiveness, we use the known-item topics search method proposed in Azzopardi et al. (2007).

Our hypothesis behind performing these experiments is to analyze if a user tries to retrieve the documents of varying retrievability scores, then we can expect to that it would be more difficult to formulate queries for retrieving low retrievable documents than retrieving high retrievable documents. In order to perform experiments we need topic queries and their relevance judgments. Since we do not have explicit topics (queries) and relevance judgments for all collections, therefore we construct implicit topics and their relevance judgments on the basis of known-items search method (Azzopardi et al. 2007).

Known-items search assumes that a user knows a document (topic query) in the collection that he/she thinks that it is relevant for his/her need and he/she has already seen this document in the collection. This forms a topic and an implicit relevant judged document. Now there is some need arisen and the user wants to retrieve this document. In order to retrieve this document he/she will try to recall different terms of the document for constructing a query. Azzopardi et al. (2007) in their work assumed that the terms that the user could recall depend on the following two factors: (a) the popularity of terms in the document (term frequency, famous terms), and the (b) discriminative terms (mixture of term frequency and inverse document frequency). We used first factor in order to generate topic queries. By

Table 9 Correlation between the MRR and the retrievability scoring functions for the *TREC-CRT* collection

Negative correlation indicates that the high retrievable documents have low effectiveness, thus difficult to find through queries

Retrieval model	$r(d)$	$\hat{r}(d)$
NormTFIDF	0.02	0.60
BM25	−0.19	0.24
DirS	0.20	0.68
JM	0.02	0.65
AbsDis	−0.45	0.87
TwoStage	0.58	0.53
TFIDF	0.83	0.92
SMART	0.76	0.82

repeatedly performing this task it is possible to generate numerous queries for retrieving known-items, and this would help in constructing a cheap test bed for checking the effectiveness of retrieval models. To perform these experiments we use the following steps.

1. First of all we divide (partition) the collection into 30 equal sized buckets according to the retrievability ranks of documents. We create these bucked one for $r(d)$ function and one for $\hat{r}(d)$ function. After partitioning, the first buckets contains the 3.33 % documents of the collection that have high retrievability scores, while the last bucket contains the 3.33 % documents of the collection that have low retrievability scores.
2. From each bucket we randomly pick 40 documents as known-items topics (total $30 * 40 = 1{,}200$ topics). Next, the terms of queries for retrieving these known-items are chosen randomly on the basis of popularity of terms in the documents. The query length is randomly selected between 3 to 6 terms.
3. These queries are then issued against the complete collection, and the performance of different buckets that how effectively their known-items are retrieved at top ranked positions are measured through Mean Reciprocal Rank (MRR). Thus, to qualify for the best retrievability scoring function, the low retrievability scores buckets should provide low effectiveness, since in principle the documents inside the buckets are difficult to retrieve by the retrieval models, and the high retrievability scores buckets should provide high effectiveness.

Tables 9, 10 11 and 12 are showing the correlation between retrievability scoring functions and MRR measure for different collections. The correlation is computed on the basis of Spearman's rank correlation coefficient. In an ideal scenario this correlation should be positive (close to 1). This would indicate that high retrievable documents are easy to retrieve than low retrievable documents. The negative

Table 10 Correlation between the MRR and the retrievability scoring functions for the *ChemAppPat* collection

Negative correlation indicates that the high retrievable documents have low effectiveness, thus difficult to find through queries

Retrieval model	$r(d)$	$\hat{r}(d)$
NormTFIDF	−0.10	0.43
BM25	0.06	0.49
DirS	0.11	0.66
JM	−0.08	0.82
AbsDis	−0.38	0.64
TwoStage	0.29	0.67
TFIDF	0.63	0.78
SMART	0.88	0.89

Table 11 Correlation between the MRR and the retrievability scoring functions for the *DentPat* collection

Negative correlation indicates that the high retrievable documents have low effectiveness, thus difficult to find through queries

Retrieval model	$r(d)$	$\hat{r}(d)$
NormTFIDF	−0.07	0.65
BM25	−0.23	0.56
DirS	0.17	0.61
JM	0.04	0.82
AbsDis	0.53	0.51
TwoStage	0.03	0.72
TFIDF	0.39	0.77
SMART	0.92	0.87

correlation close to −1 indicates that it is hard to retrieve high retrievable documents than low retrievable documents. From the results presented in tables, the following conclusions can be drawn.

When there is a high positive correlation between the $r(d)$ and the $\hat{r}(d)$, then there is not a very high difference between both functions with known-items search method. The high retrievable documents have high MRR effectiveness and the low retrievable documents have low MRR effectiveness. Such scenario is clearly visible by looking at the results of *TFIDF* and *SMART* models. Both functions have significant positive correlation with the MRR measure, thus verifying the above stated hypothesis.

On the other hand, when there is a moderate or low correlation between $r(d)$ and $\hat{r}(d)$, then $\hat{r}(d)$ appears to be more positively correlated with the MRR effectiveness than $r(d)$. On several places, when there is a negative correlation between the $r(d)$ and $\hat{r}(d)$, then $r(d)$ has negative correlation with the MRR effectiveness. The negative correlation of $r(d)$ indicates that the documents in the high retrievability scores buckets have low MRR effectiveness, thus for these documents users need more effort in order to retrieve them at top ranked positions. This happened due to their low percentage of retrievability out of total queries. On the other hand, a significant positive correlation between the $\hat{r}(d)$ and the MRR on similar retrieval models indicates that ranking retrievability of documents on the basis of relative retrievability scores provides huge benefits for correctly analyzing the relationship between the retrievability and the MRR effectiveness. With $r(d)$, the high retrievability scores buckets mostly have high MRR effectiveness and the low retrievability scores buckets have low MRR effectiveness. This satisfies our hypothesis. Thus again, retrievability as measured by $\hat{r}(d)$ produces better retrievability ranks of documents than $r(d)$.

Table 12 Correlation between the MRR and the retrievability scoring functions for the *ATNews* collection

Negative correlation indicates that the high retrievable documents have low effectiveness, thus difficult to find through queries

Retrieval model	$r(d)$	$\hat{r}(d)$
NormTFIDF	−0.59	0.95
BM25	−0.52	0.79
DirS	0.23	0.15
JM	−0.50	0.86
AbsDis	−0.43	0.76
TwoStage	0.27	0.14
TFIDF	0.90	0.75
SMART	0.47	0.95

8 Conclusion

In this paper we investigate the retrieval models effectiveness using retrievability measurement. Retrievability reflects the ease with which documents can be found through a retrieval model. The motivation for such a measure stems from the concern over bias within retrieval models, and the need to ensure that information is accessible through such retrieval models. This is because of the growing reliance of users to engage such retrieval models in order to find their desired information. We start this analysis by first performing comparison between different retrieval models on the basis of retrievability measures. Our experiments revealed that retrieval models are significantly and substantially differ in terms of retrieval bias that they imposed on the individual or collection of documents. We performed these experiments on four collections using eights standard retrieval models. We also addressed a limitation of the standard retrievability scoring function. This limitation revealed that in case of exhaustive query generation the standard retrievability scoring function exhibits high retrievability scores towards long documents than short documents. In order to handle this problem, we proposed a normalized retrievability scoring function that normalizes the retrievability scores of documents relative to the total number of queries of documents. This is helpful for removing any unnecessary bias from the retrieval bias that could arise due to ignoring the difference between the document lengths. Finally, in order to investigate that which retrievability scoring function has better effectiveness than the other for correctly producing the document retrievability ranks, we compared the effectiveness of both functions using known-items search method. Our results on this comparison showed that the normalized retrievability scoring function has better effectiveness than the standard retrievability scoring function.

In future work, we want to analyze the effect of query diversity on document retrievability. In our current experiments, we do not examine to what extent documents are retrievable from diverse queries. If we ignore query diversity then a document could still give a higher retrievability score if has high term weights for only a few number of terms. As a post processing analysis of retrievability, one important future direction is to cluster the queries on the basis of their terms similarity, and then try to examine to what extent the documents have higher retrievability for different query clusters. This research can be further extent for examining how many paragraphs of a document or its different text segments according to their length have low retrievability.

Another important research direction that has worth to investigate is to analyze the effect of query popularity on document retrievability. This would help in analyzing to what extent popular queries shape the accessibility of documents. Information about how much a query is popular can be obtained by examining query logs of users.

References

Arampatzis, A., Kamps, J., Kooken, M., Nussbaum, N. (2007). Access to legal documents: exact match, best match, and combinations. In *Proceedings of the 16th text retrieval conference (TREC'07)*.

Azzopardi, L., & Bache, R. (2010). On the relationship between effectiveness and accessibility. In *SIGIR '10: Proceeding of the 33rd annual international ACM SIGIR conference on research and development in information retrieval, Geneva, Switzerland* (pp. 889–890).

Azzopardi, L., de Rijke, M., Balog, K. (2007). Building simulated queries for known-item topics: an analysis using six European languages. In *SIGIR '07: Proceedings of the 30th annual international ACM SIGIR conference on research and development in information retrieval, Amsterdam, The Netherlands* (pp. 455–462).

Azzopardi, L., & Vinay, V. (2008). Retrievability: an evaluation measure for higher order information access tasks. In *CIKM '08: Proceeding of the 17th ACM conference on information and knowledge management, Napa Valley, CA, USA* (pp. 561–570).

Bache, R., & Azzopardi, L. (2010). Improving access to large patent corpora. In *Transactions on large-scale data- and knowledge-centered systems II* (Vol. 2, pp. 103–121). Springer.

Baeza-Yates, R., & Ribeiro-Neto, B. (1999). *Modern information retrieval*. ACM Press.

Bashir, S., & Rauber, A. (2009a). Analyzing document retrievability in patent retrieval settings. In *DEXA'09: Proceedings of the 20th international conference on database and expert systems applications* (pp. 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 2009* (pp. 1863–1866).

Bashir, S., & Rauber, A. (2010a). Improving retrievability and recall by automatic corpus partitioning. In *Transactions on large-scale data- and knowledge-centered systems II* (Vol. 2, pp. 122–140). Springer.

Bashir, S., & Rauber, A. (2010b). Improving retrievability of patents in prior-art search. In *Advances in information retrieval, 32nd European Conference on IR Research, ECIR 2010* (pp. 457–470).

Callan, J., & Connell, M. (2001). Query-based sampling of text databases. *ACM Transactions on Information Systems (TOIS) Journal, 19*(2), 97–130.

Chowdhury, G.G. (2004). *Introduction to modern information retrieval* (2nd ed.). London: Facet Publishing.

Gastwirth, J.L. (1972). The estimation of the Lorenz curve and Gini index. *The Review of Economics and Statistics, 54*(3), 306–316.

Harter, P.S. & Hert, A.C. (1997). Evaluation of information retrieval systems: approaches, issues, and methods. *Annual Review of Information Science and Technology (ARIST), 32*, 3–94.

Lauw, W.H., 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, Philadelphia, PA, USA* (pp. 625–630).

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, J.F.G. (2010). Pres: a score metric for evaluating recall-oriented information retrieval applications. In *SIGIR'10: ACM SIGIR conference on research and development in information retrieval* (pp. 611–618). ACM.

Manning, D., Raghavan, C.P., Schutze, H. (2008). *Introduction to information retrieval*. Cambridge: Cambridge University Press.

Mowshowitz, A., & Kawaguchi, A. (2002). Bias on the web. *Communications of the ACM, 45*(9), 56–60.

Ounis, I., De Rijke, M., Macdonald, C., Mishne, G., Soboroff, I. (2006). Overview of the trec 2006 blog track. In *Proc. of the text retrieval conference, TREC'06*.

Owens, C. (2009). *A study of the relative bias of web search engines toward news media providers*. Master Thesis, University of Glasgow.

Robertson, S.E., & Walker, S. (1994). Some simple effective approximations to the 2-poisson model for probabilistic weighted retrieval. In *SIGIR '94: Proceedings of the 17th annual international ACM SIGIR conference on research and development in information retrieval, Dublin, Ireland* (pp. 232–241).

Sanderson, M., & Zobel, J. (2005). Information retrieval system evaluation: effort, sensitivity, and reliability. In *SIGIR'05: ACM SIGIR conference on research and development in information retrieval* (pp. 162–169). ACM.

Singhal, A. (1997). At&t at trec-6. In *The 6th text retrieval conference (TREC6)* (pp. 227–232).

Singhal, A. (2001). Modern information retrieval: a brief overview. *IEEE Data Engineering Bulletin, 24*, 34–43.

Vaughan, L., & Thelwall, M. (2004). Search engine coverage bias: evidence and possible causes. *Information Processing and Management Journal, 40*(4), 693–707.
Voorhees, M.E. (2001). Overview of the trec 2001 question answering track. In *Proc. of the text retrieval conference, TREC'01* (pp. 42–51).
Voorhees, M.E. (2002). The philosophy of information retrieval evaluation. In *CLEF'01* (pp. 355–370). Springer.
Voorhees, M.E., & Harman, K.D. (2005). *Trec experiment and evaluation in information retrieval*. Cambridge, MA: MIT Press.
Zhai, C. (2002). *Risk minimization and language modeling in text retrieval*. PhD thesis, Carnegie Mellon University.