Neurocomputing 123 (2014) 216–232

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

Estimating retrievability ranks of documents using document features

Shariq Bashir *

Machine Intelligence Group, Department of Computer Science, National University of Computer and Emerging Sciences, Islamabad, Pakistan

ARTICLE INFO

Article history:
Received 2 May 2012
Received in revised form
28 April 2013
Accepted 14 July 2013
Communicated by Y. Chang
Available online 19 August 2013

Keywords:
Retrieval models evaluation
Retrieval systems bias analysis
Documents findability
Patent retrieval

ABSTRACT

Retrievability is a measure of access that quantifies how easily documents can be found using a retrieval system. Such a measure is of particular interest within the recall oriented retrieval domains such as patent or legal retrieval. This is because if a retrieval system for these retrieval domains makes some documents hard to find then professional searchers would have a difficult time when retrieving these documents. One main limitation of retrievability analysis is that it depends upon the processing of exhaustive number of queries. This requires large processing time and resources. In order to handle this problem, in this paper we use document features based approach in order to estimate the retrievability ranks of documents. In experiments, the strong correlation between features and retrievability scores on different collections confirms that it is possible to estimate the retrievability ranks of documents without processing queries. One major advantage of this approach is that it requires fewer resources, and can be computed more quickly as compared to query based approach. While, on the other hand, one major disadvantage of this approach is that it can only estimate the retrievability ranks of documents, but cannot calculate how much there is retrievability inequality (retrieval bias) between the documents of collection.

1. Introduction

Access to large amounts of information from the web and internet is playing an important part in the transformation of society. One important part of this overall process is information retrieval (IR) systems. IR systems deal with the storage (indexing), organization, management and retrieval of information. After indexing one important factor that shapes the access to information is the role of the retrieval strategy (model). 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 return the relevant results to the users according to descending order of their relevance, so that the users can view the most relevant information at top ranked positions. In last few years, a large number of retrieval models have been proposed for various kinds of retrieval tasks. One main problem that is always remained in the IR researchers' attention is how to choose the right model for a given retrieval task. This is a tedious task, and falls under the research domain of evaluation of retrieval models. Historically, research on the evaluation of retrieval models is always focused on either the effectiveness or the efficiency (speed/memory). These are the only two measures that are focused on the core research of IR community for determining the quality of retrieval models. The main limitation of these measures is that they focus almost exclusively on only few documents, i.e., the fact that the (most) relevant documents are returned at the top of ranked list, as this constitutes the primary criterion of interest in most of the standard retrieval tasks (web retrieval, question answering, opinion retrieval, etc.). With evaluation measures such as recall and F_β, 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 an indication of how easily the information within the collection can be reached or accessed with the given retrieval model [3]. This offers a higher level and abstract level of view for understanding 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 is given in the form of judged relevant documents by a group of few people. This is particularly important for recall-oriented retrieval domains like patent or legal retrieval, where focus of retrieval is more given toward 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. Furthermore, it specifically examines whether the lack of access to information actually impedes one's ability to access the required information within the collection.

The retrievability estimation framework proposed by Azzopardi and Vinay [3] is divided into two phases, namely (1) query list generation (if no query list is known in advance), and (2) the

* Tel.: +971 262 847 38.
E-mail address: shariq.bashir@nu.edu.pk

0925-2312/$ - see front matter © 2013 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.neucom.2013.07.011

processing of these queries with different retrieval models for analyzing their retrieval bias. Among both, query processing is difficult phase, since it requires large processing time and resources due to exhaustive number of queries. In this paper, rather than estimating retrievability by processing queries, we try to estimate the retrievability ranks of documents using query independent approach via document features. We categorize these features into three classes on the basis of their surface level, terms weighting methods, and density around nearest neighbors characteristics. The first class consists of features that are based on the document surface level characteristics. These are defined on the basis of the distributional characteristics of the terms frequencies either within a document or over the whole collection (document frequencies). The features of second class are based on the terms weighting methods of retrieval models. These features do not calculate absolute terms frequencies within the documents, but are defined over the terms weighting methods of different retrieval models. The features of third class consider to what extent the low or high density around the nearest neighbors of documents affects the relative retrievability ranks of document. Our results on different collections confirm that there exist several features that show a high correlation with the retrievability ranks of documents. This creates the possibility of estimating retrievability ranks of documents without processing exhaustive number of queries. One major advantage of this approach is that it requires fewer resources, and can be computed more quickly as compared to query based approach. While, on the other hand, one major disadvantage of this approach is that it can only estimate the retrievability ranks of documents, but cannot calculate how much there is retrievability inequality between the documents of collection (e.g. Gini-Coefficient measuring retrieval bias) with different retrieval models.

In related work on retrievability, estimating retrievability of documents automatically via documents features has proven an important factor in collection partition based retrieval approach [6,8,16]. Query-based retrievability estimation approach is not suitable in this case because it requires large processing time, and second it is not suitable if some new documents are added in the collection or existing documents in the collection are updated after some period. In collection partition a single collection is split into two equal sized low and high retrievable documents partitions. Partition is done by first estimating the retrievability scores of documents using correlated document features. Documents are then ordered on the basis of increasing retrievability scores, and afterwards split into two independent partitions. After splitting the collection into these two categories documents is then retrieved by treating these classes as independent partitions and queries are processed independently for each partition and subsequently results lists are combined afterwards. This ensures that the final result list will always include also the documents having a low retrievability score, i.e., that would rarely or never have been returned within a certain rank cut-off in a standard retrieval setting independent of collection partition. The performance of this approach highly depends upon the effectiveness of document features that are used for the partitioning of collection.

The remainder of this paper is structured as follows. Section 2 reviews related work on the bias assessment of retrieval models using retrievability measurement. Section 3 summarizes the concept of retrievability measurement. In Section 4 we describe settings for experiments, the collections, retrieval models and the queries that are used for retrieval bias analysis. Section 5 describes different classes of document features and their correlation with retrievability ranks of documents. Finally, Section 7 briefly summarizes the key lessons learned from this study.

2. Related work

In this section we provide a review of related work on retrievability. All these studies use exhaustive queries in order to estimate retrievability of documents.

Azzopardi and Vinay [3] introduced a measure for the quantification of retrieval bias 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 large proportion of documents that have very low retrievability scores (sometimes more than 80% of the documents in case of high retrieval bias) 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 [4] performed retrievability experiments on a patent collection using standard retrieval models. 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 [3] experiments, Bashir and Rauber [5] 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 [7,9]. 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 a small subset of documents, creating a high retrieval bias. To overcome this limitation, they proposed techniques to select relevant documents for pseudo-relevance feedback on the basis of documents clustering [7] and term-proximity based methods [9]. Their experiments with different collections of patent documents indicate 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 [8] 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 equal sized 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 approach 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 [1] 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.

Bashir and Rauber [10] also studied how to approximate retrievability without processing exhaustive number of queries. They grouped queries based on their different characteristics and ranges of these characteristics, and then individually analyzed their relationship with retrievability. The query characteristics that they considered are as follows: (a) the effect of high/low query term frequencies, (b) queries that retrieve few or large number of documents, and (c) the query quality based on different query quality predictors. In their experiments they found a significant correlation between the ranges of these characteristics and the different levels of retrieval bias. Their experiments on different collections indicated that the ranges of query characteristics do not dramatically alter the retrievability ranks of documents, but only affect the level (magnitude) of the (retrieval bias) approximation. This allows to approximate the retrieval bias without processing an exhaustive number of queries. The retrievability estimation method that is present in this paper is completely different from the methods proposed in [10]. The methods proposed in [10] still rely on query processing, while this work presents an approach that how to estimate retrievability without processing queries via document features.

3. Retrievability measurement

The following description of retrievability measurement as introduced by [3] (adopted from [8]) provides a quick introduction of how it can be measured.

Given a collection D, a retrieval model processes 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 [3] in their experiments used query set based sampling [11]. Q the query set could either be a historical sample of queries or an artificial simulated substitute similar to users queries. Then, each $q \in Q$ is issued to the retrieval model, and the retrieved documents along with their positions in the ranked list are recorded. Intuitively, retrievability of a document d is likely to be high in the following cases:

1. when 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 in order to obtain 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 $r(d)$ will be high when there are a large number of highly probable queries that can retrieve the document d at the rank position less than c, and the value of $r(d)$ will 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 the $r(d)$ is zero.

The cumulative measure of the retrievability score of a document on the basis of binary $f(k_{dq}, c)$ function ignores the ranking position of a document in the ranked list, i.e., how accessible the document is in the ranking. A 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

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

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 the *Lorenz Curve* [12]. In Economics and the Social Sciences, a Lorenz Curve is used to visualize the inequality of the 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 [3] employed similar idea in the context of a population of documents, where the wealth of documents are represented by $r(d)$ function. The more skewed the plot, the greater the amount of inequality, or high bias within the population. The *Gini-Coefficient* [12] 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:

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

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 some basic 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 dataset consists of 1.2 million patent documents from the TREC

Table 1
The properties of document collections 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.*

Chemical Retrieval Track (2009) (*TREC-CRT*)[1] [13]. 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* serves 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 $c=150$.

- *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.[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 the 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 as *ATNews Collection*. As compared to the above three collections, the documents in this collection are mostly short, however, the distributions of document length and vocabulary size are highly 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 the 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.

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

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 term 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 follows:

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

$tf_{t,d}$ 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 toward large absolute term frequencies. It is possible to address the length bias by using document length $|d|$, and defined 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 is 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. We used the following standard function of BM25 proposed by [14]:

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

$|\overline{d}|$ 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 [15] at AT&T Labs:

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

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

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

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:

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

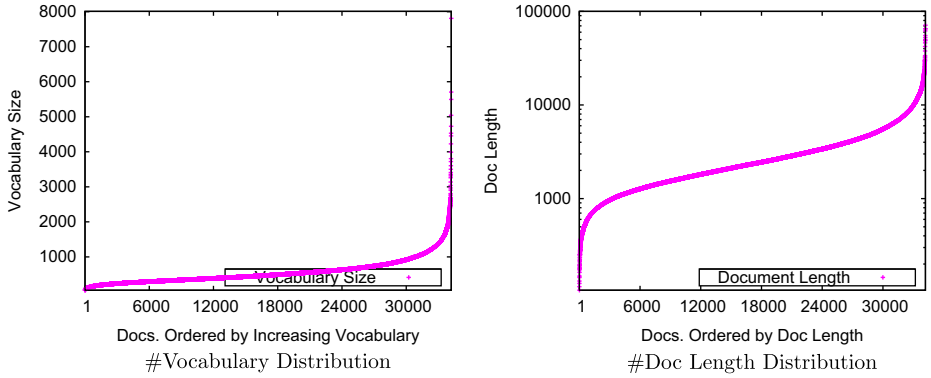

Fig. 1. Document vocabulary size and length distribution on the *TREC-CRT* collection.

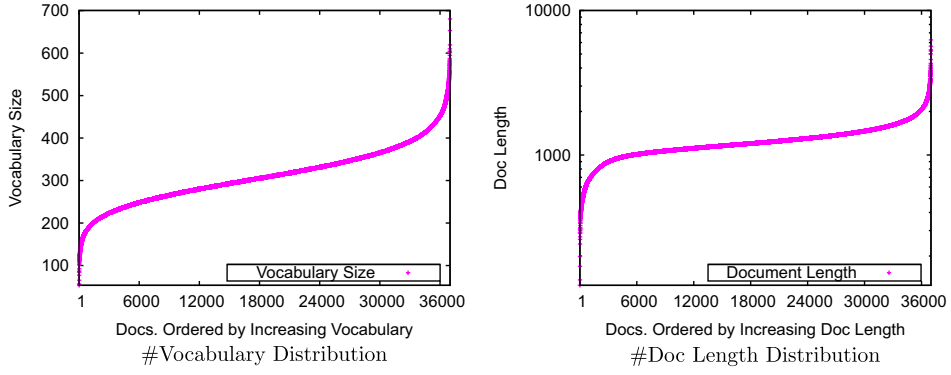

Fig. 2. Document vocabulary size and length distribution on the *ChemAppPat* collection.

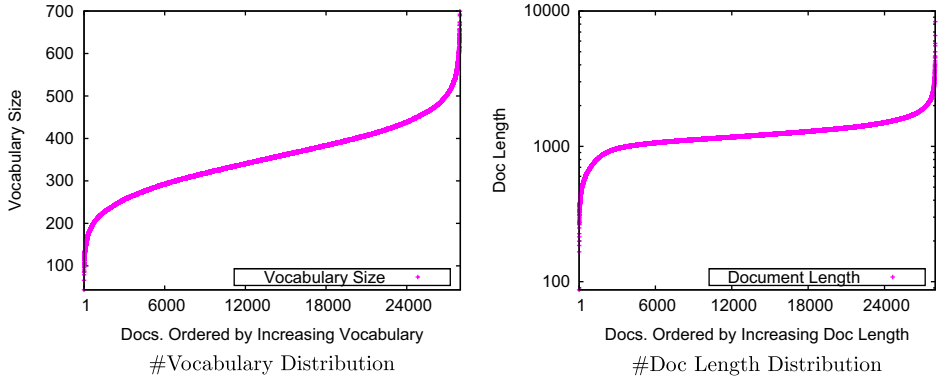

Fig. 3. Document vocabulary size and length distribution on the *DentPat* collection.

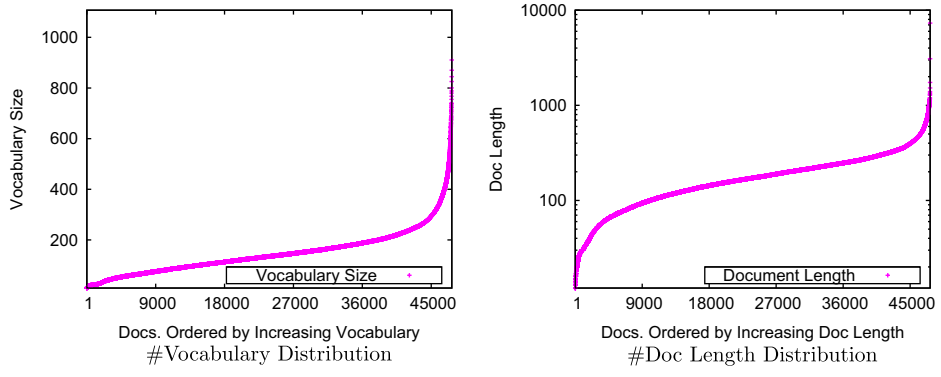

Fig. 4. Document vocabulary size and length distribution on the *ATNews* collection.

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

The overall similarity score for the query and the document could be zero if some of the 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*: Jelinek–Mercer smoothing [17] 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:

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

  $P(t|D)$ 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 [17], 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:

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

  According to Zhai's [17] suggestion, we use the μ with 2000.
- *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 parameters $\mu = 2000$ and $\lambda = 0.7$.
- *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

Although the focus of this paper is to estimate the retrievability ranks of documents using document features. However, for the validation of results it is important to test to what extent the features scores have strong correlation with retrievability scores that are estimated through processing exhaustive number of queries. In order to generate queries, we consider all sections (title, abstract, claims, description, background summary) of patent documents for both retrieval and query generation. Stop words are removed 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% of 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 document. 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 along with the user ability to formulate the query and the retrieval bias of retrieval model that affects the retrievability of documents 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 highly depend upon the retrieval bias of retrieval model. If a retrieval model has low retrieval bias then it would make a large number of documents highly retrievable at the 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 portion of documents and thus the bias of retrieval models would play less part on the retrievability of documents.

Therefore, in order to precisely analyze the effect of retrieval bias, the sizes of query result lists neither should be too close to the user's rank cutoff nor should be too large. Large result lists indicate that queries are generated through frequent terms of the collection, and the users would rarely use them for searching their information. Therefore, in order to reasonably approximate the retrieval bias we remove all those queries from Q that either retrieve only a few number of documents or retrieve a very large number of documents. Under this setting, 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 query result list sizes, and select only top 30 million queries (low frequent 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 the different collections. Fig. 5 shows the distributions of the total number of queries per document relative to the vocabulary size of documents. The *TREC-CRT* and *ATNews* collections have large differences between documents vocabulary size, thus for these collections this distribution is highly skewed. The *ChemAppPat* and *DentPat* collections have less differences between documents vocabulary size, thus for these collections the distribution of queries is less skewed.

4.4. Normalized retrievability scoring function

The retrievability measure that is defined above cumulates the retrievability scores of documents over all queries. Thus, in case of exhaustive query generation long documents potentially have large number of query combinations possible than short documents due to their large vocabulary sizes. Fig. 5 shows the distribution of the total number of queries for different

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

Table 2
Properties of Q that is 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	1636	156	161	87
Avg # of Queries/Document	531,442	127,032	173,516	54,551

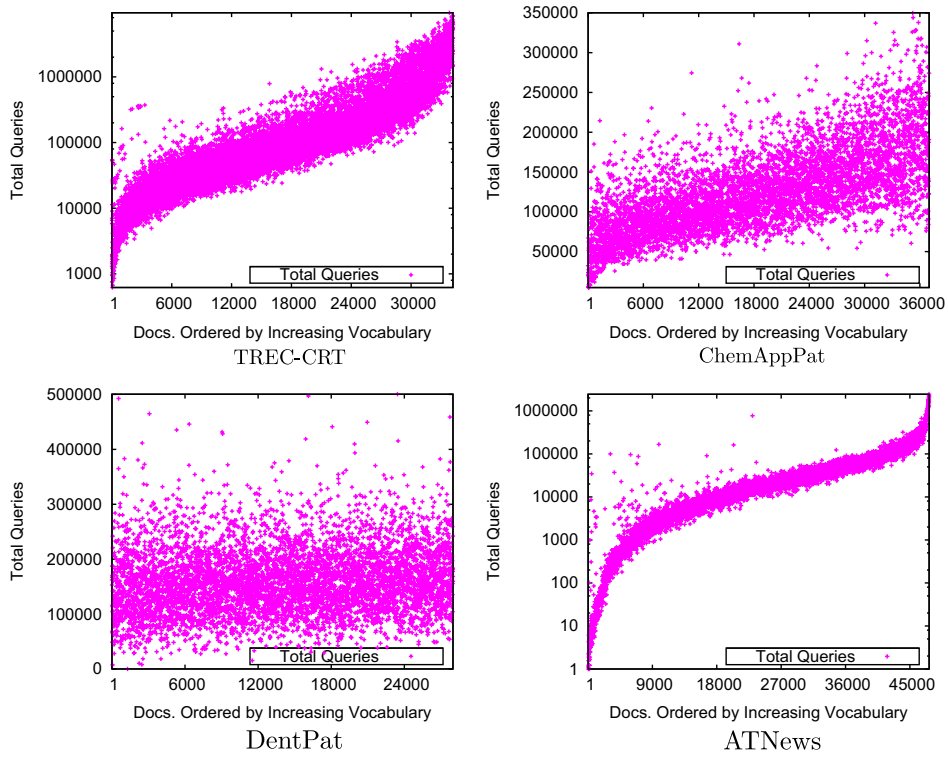

Fig. 5. The distribution of total number of queries per document for different collections. Documents are ordered by the increasing vocabulary size.

Table 3
Retrieval bias representation with $r(d)$ and $\hat{r}(d)$. G refers to Gini-Coefficient value.

Docs.	Unique terms	Total queries	IR model A	IR model B	IR model C
Retrieval Bias with $r(d)$					
Doc1	40	9880	791	5928	9880
Doc2	35	6545	851	3600	6545
Doc3	8	56	55	40	56
Doc4	28	3276	525	2130	3276
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	9880	0.08	0.60	1
Doc2	35	6545	0.13	0.55	1
Doc3	8	56	0.98	0.70	1
Doc4	28	3276	0.16	0.65	1
Doc5	10	120	0.98	0.75	1
Doc6	12	220	0.85	0.80	1
	Overall Bias		$G=0.48$	$G=0.08$	$G=0$

documents. Long documents have large query sets and short documents have small query sets. 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 a large query combinations. (For the context of this example, we are assuming only the 3-terms queries.) From 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 computing retrieval bias, the retrieval model C and the retrieval model B should show low retrieval bias than the 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 not considering the differences between the vocabulary richness of documents while computing the retrieval bias. We thus propose to normalize the 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 the $\hat{Q}_{(d)}$. 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. The 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 the few queries that can be created from their vocabulary. This is where a normalized retrievability score accounting 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 models' retrieval bias. 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.

4.5. Retrieval bias analysis

Tables 4–7 list the retrievability inequality providing Gini-Coefficients for a range of rank cutoff factors for the different collections. Note that the high bias is 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 user to search deeper down into the result list. If user examines only a few portion of the result list, then he/she would 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*.

Table 4
Gini-Coefficient scores representing the retrieval bias of different retrieval models on various rank cutoff factors for the *TREC-CRT* collection. As rank cutoff factor increases, bias steadily decreases indicating that the low retrieval bias is experienced when considering the long ranked lists.

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

Table 5
Gini-Coefficient scores representing the retrieval bias of different retrieval models on various rank cutoff factors for the *ChemAppPat* collection. As rank cutoff factor increases bias steadily decreases indicating that the low retrieval bias is experienced when considering the long ranked lists.

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

Table 6
Gini-Coefficient scores representing the retrieval bias of different retrieval models on various rank cutoff factors for the *DentPat* collection. As rank cutoff factor increases bias steadily decreases indicating that the low retrieval bias is experienced when considering the long ranked lists.

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

Table 7
Gini-Coefficient scores representing the retrieval bias of different retrieval models on various rank cutoff factors for the *ATNews* collection. As rank cutoff factor increases bias steadily decreases indicating that the low retrieval bias is experienced when considering the long ranked lists.

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

4.6. Comparing $r(d)$ and $\hat{r}(d)$ effectiveness using known-items search method

In the above section we analyze the retrieval biases of different retrieval models using two retrievability scoring functions. If we compare both functions only on the basis of Gini-Coefficient scores, 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 as proposed in [2].

Our hypothesis behind performing this test is to analyze that if a user tries to retrieve the documents of varying retrievability scores, then we can expect that it would be more difficult to formulate queries for retrieving the less retrievable documents than retrieving the high retrievable documents. In order to perform this test 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 using known-items search method [2].

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 a 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. [2] 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 (b) the discriminative terms (mixture of term frequency and inverse document frequency). By repeatedly performing this task it is possible to generate numerous queries for retrieving the known-items, and this would help in constructing a cheap test bed for checking the effectiveness of retrieval models. To perform this experiment 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. These buckets are created individually for both $r(d)$ and $\hat{r}(d)$. After partitioning, the first buckets contain the 3.33% documents of the collection that has high retrievability scores, while the last bucket contains the 3.33% documents of the collection that has low retrievability scores.
2. From each bucket we randomly pick 40 documents as known-items topics (total 30*40=1200 topics). Next, the terms of queries for retrieving these known-items are chosen randomly on the basis of probability of terms inside these documents. The query length is randomly selected between 3 and 6 terms.
3. These queries are then issued against the complete collection, and the effectiveness of different buckets that how efficiently their known-items are retrieved at the top ranked positions is measured through Mean Reciprocal Rank (MRR). Thus, if a retrievability scoring function produces correct retrievability ranks of document then its low retrievability scores buckets should have low effectiveness, since in principle the documents inside the buckets are difficult to retrieve by the retrieval models, and its high retrievability scores buckets should have high effectiveness.

Tables 8–11 are showing the correlation between the retrievability scoring functions and the MRR measure for different collections. The correlation is computed on the basis of Spearman's rank correlation coefficient. In a ideal scenario this correlation should be positive (close to 1). This indicates that the high retrievable documents are easy to retrieve than the low retrievable documents. The negative correlation close to −1 indicates that it is hard to retrieve the high retrievable documents than the 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 on 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 where both the 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 the $r(d)$ and the $\hat{r}(d)$, then $\hat{r}(d)$ is appeared to be more positively correlated with the MRR effectiveness than the $r(d)$. In several results, 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 same retrieval models indicates that estimating retrievability of documents on the basis of relative retrievability scores provides huge benefits in correctly analyzing the relationship between the retrievability and the MRR effectiveness. With $\hat{r}(d)$, the high retrievability scores buckets mostly have high MRR effectiveness and the low retrievability scores buckets have low

Table 8
Correlation between the MRR and the retrievability scoring functions for the *TREC-CRT* collection. Positive correlation indicates that high retrievable documents have high effectiveness. This indicates that retrievability scoring function produces correct retrievability ranks of documents.

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

Table 9
Correlation between the MRR and the retrievability scoring functions for the *ChemAppPat* collection. Positive correlation indicates that high retrievable documents have high effectiveness. This indicates that retrievability scoring function produces correct retrievability ranks of documents.

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 10
Correlation between the MRR and the retrievability scoring functions for the *DentPat* collection. Positive correlation indicates that high retrievable documents have high effectiveness. This indicates that retrievability scoring function produces correct retrievability ranks of documents.

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

Table 11
Correlation between the MRR and the retrievability scoring functions for the *ATNews* collection. Positive correlation indicates that high retrievable documents have high effectiveness. This indicates that retrievability scoring function produces correct retrievability ranks of documents.

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

MRR effectiveness. This indicates that $\hat{r}(d)$ produces better retrievability ranks of documents than $r(d)$.

5. Estimating retrievability ranks of documents using document features

We compute a number of statistical and information-theoretic features from the documents for analyzing their relationship with the document retrievability ranks.[6] We categorized these features into three classes: (a) Surface level features, (b) features based on the term weights, and (c) density around nearest neighbors of documents.

5.1. Surface level features

This features set captures the distributional characteristics of the document terms on the basis of term frequencies within the document and term document frequencies within the whole collection:

- *Normalized Average Term Frequencies* (*NATF*): NATF is defined as the average of the normalized term frequencies of all the terms of a document. Large or small term frequencies may create significantly effect on the retrievability scores of documents. Therefore this feature tries to capture what is the term frequencies distribution for the high and low retrievable documents with different retrieval models:

$$NATF(d) = \frac{\sum_{t \in T_d} \frac{tf_{t,d}}{|d|}}{|T_d|} \tag{17}$$

  T_d represents the set of all unique terms in a document d. $tf_{t,d}$ is the frequency of term t in d, and $|d|$ represents the length of document.
- *Number of Frequent Terms* (*freq*): NATF scores could be smaller for long documents as compared to short documents. This is because in long documents there could be a large number of small frequency terms, and these can decrease the NATF average score. As large term frequencies create strong impact on the document retrievability scores, however, for the long documents there could be a small ratio of such terms out of total terms. The main question is how to count them. For solving this problem, in this feature we use the term frequency threshold, and count how many terms ft_d in a document d have frequency above this threshold. We use $tf_{t,d}/|d| \geq 0.03$ for this purpose.
- *NATF of Frequent Terms* (*NATF_freq*): This feature calculates the *NATF* scores with only the frequent terms of documents, i.e., terms having normalized term frequencies $tf_{t,d}/|d| \geq 0.03$. This eliminates the impact of a potentially large number of terms having small frequencies:

$$NATF\_freq(d) = \frac{\sum_{t \in ft_d} \frac{tf_{t,d}}{|d|}}{|ft_d|} \tag{18}$$

  ft_d represents the set of frequent terms in document d.
- *Gini-Coefficient of Term Frequencies* (*GC_terms*): *freq* counts the frequent terms on the basis of a fixed threshold. Since the value of this threshold is heuristically set *unique* for all documents, therefore the main question is what should be the best value of this threshold that can accurately count the presence of all frequent terms for all kinds of short and long documents. One solution of this problem can be to count the freq feature with different threshold values. However, this increases the total number of features.
  In order to count the frequent terms with a single feature, *GC_terms* counts the frequent terms after computing the inequality between the term frequencies of a document using Gini-Coefficient. It tries to capture how balanced is the distribution of term frequencies within a document. *GC_terms* captures whether all terms have similar or rather different frequencies:

$$GC\_terms(d) = \frac{\sum_{t \in T_d}(2 \cdot i(t) - |T_d| - 1) \cdot tf_{t,d}}{(|T_d| - 1)\sum_{t \in T_d} tf_{t,d}} \tag{19}$$

  T_d is the set of unique terms of d, $i(t)$ is the rank of a term t in set T_d. The ranks are calculated after sorting all terms of d in ascending order of their frequencies.
- *Number of Frequent Terms based on Gini-Coefficient* (*freq_GC*): Rather than relying on a fixed threshold as we use for *NATF_freq*, terms having large frequencies within d are iteratively removed until the resulting Gini-Coefficient for the entire document does not drop below $GC\_terms = 0.25$, i.e., is rather homogeneous. The number of removed terms provides a different measure for the number of frequent terms contributing to the retrievability.
- *Average Document Frequency* (*ADF*): Low document frequency (df_t) of terms creates significant effect on the retrievability scores of documents. This feature captures the effect of average terms df_t of documents on the retrievability ranks distribution. It captures to what extent a document consists of rather common or rather specialized vocabulary by summing up the df_t values of its vocabulary:

$$ADF(T_d, d) = \frac{\sum_{t \in T_d} df_t}{|T_d|} \tag{20}$$

- *Frequent Terms with Low Document Frequency* (*freq_low_df*): This feature is similar to *freq*, but differ in this sense that it counts the frequent terms based on the term document frequencies rather than term frequencies within the document. Frequent terms are counted with the threshold $df_t/|D| = 5\%$.
- *Average Document Frequency of Frequent Terms* (*ADF_freq*): Similar to NATF_freq, this feature computes the ADF scores of the documents with only those terms of documents that are identified with the help of freq_low_df feature. This helps in capturing the exoticity of the frequent terms in the vocabulary of a document.
- *Document Length*: This feature may help to capture the relationship between the document length and the retrievability ranks.
- *Vocabulary Size*: This feature may help to capture the relationship between the document vocabulary size and the retrievability ranks.

5.2. Features based on term weights

The retrieval models that we use for retrievability analysis do not rely on the absolute term frequencies within documents for calculating the document relevance scores. In order to provide better relevance scores of documents, they modify the absolute term frequencies with the help of different features (i.e., length, vocabulary size, term document frequency, etc.) and parameters. These modified scores are called terms weights. The features of this feature set are defined on the basis of distributional characteristics of term weights. The distributional characteristics are based on the average of term weights within the documents, and the ranks of term weights relative to other documents of the collection. The ranks of terms are defined with the help of average of term rank positions in the inverted lists, the variance of term rank positions in the inverted lists, the term weights differences

[6] The complete feature scores for all collections are available at http://www.ifs.tuwien.ac.at/~bashir/Automatic_classification.htm.

relative to median weight, and the term low rank ratio relative to all terms of the document:

- *Average of Term Weights* (*ATW*): ATW is defined as the average of all term weights within the document. Terms weights are calculated on the basis of the different retrieval models term weighting schemes. High value of this feature indicates that the given document has many high weights terms, and thus it has high probability of high retrievability than low ATW scores documents:

$$ATW(d) = \sum_{t \in T_d} \frac{w_{t,d}}{|T_d|} \tag{21}$$

  $w_{t,d}$ is the weight of term t in d and it calculated on the basis of a given retrieval model term weighting scheme.
- *Average of Term Rank Positions* (*ATRP*): Inverted list is a popular indexing technique in IR. It maps the term occurrences to documents. In this feature, we first prepare the inverted list for each term of the collection together with their weights based on a given retrieval model's term weighting scheme. We then sort all the documents in the inverted lists according to descending order of their term weights and map them to the rank positions. We repeat this step for every term's inverted list. Next, for each document we compute the average rank positions of its terms in the sorted inverted list. This defines the value of this feature. Large value of this feature for any document indicates that this document has a large number of terms having high weights relative to other documents of the collection:

$$ATRP(d) = \sum_{t \in T_d} \frac{\frac{\hat{i}_{t,d,t)}}{df_t}}{|T_d|} \tag{22}$$

  $\hat{i}_{d,t}$ is the rank position of document d in the sorted inverted list of term t.
- *Variance of Term Rank Positions* (*VTRP*): ATRP is defined by the average of term rank positions but does not consider to what extent the term rank positions are spread out from each other. VTRP calculates the variance of term rank positions within the documents.
- *Term Weights Differences from the Median Weight* (*DiffMedianWeight*): In this feature, we first compute the median weight of each inverted list, and then for each document we calculate the average of its term weights differences from the median weights. This defines the value of this feature:

$$DiffMedianWeight(d) = \sum_{t \in T_d} \frac{w_{t,d} - \overline{w}_t}{|T_d|} \tag{23}$$

  $\overline{w}_t$ is the median weight of term t in its inverted list. $w_{t,d}$ is the weight of term t in d and it is based on the given retrieval model term weighting scheme.
- *Term Low Rank Ratio* (*LowRankRatio*): By using the ATRP sorted terms' inverted lists, the value of this feature is defined by considering how many terms of a document relative to all terms do not appear in the top 200 rank positions of the sorted inverted lists.

5.3. Document density based features

This feature set is based on the density around the nearest neighbor of documents:

- *Average Density of K-Nearest Neighbors* (*AvgDensity*): This feature is defined as follows. By taking each document of collection as a vector of terms, all other documents of the collection are sorted based on their distances from it, and then the average density of k nearest neighbors is used as a feature score. *AvgDensity* is calculated over 50, 100 and 150 nearest neighbors. In experiments we further test the *AvgDensity* with only top 40 (high frequency) terms of the documents.

6. Relationship between features and retrievability

6.1. Correlation measure

In the above section we propose different features, however, in order to verify that whether the proposed features are working as expected, it is important to analyze how strong the features scores are correlated with the retrievability ranks of documents that are estimated after processing exhaustive number of queries. Ideally for this type of task where measuring relationship between two vectors is important Spearman's rank correlation coefficient is mostly used. Given two vectors, where first vector contains rank positions of documents after sorting the values of this vector on the basis of ascending retrievability scores, and the second vector contains feature scores of documents. The Spearman's rank correlation coefficient of both vectors explains relationship (correlation) between them. The correlation score always lies between -1 and $+1$. The correlation score close to $+1$ or close to -1 indicates that there exists a high relationship between the feature and the retrievability ranks. This indicates that on the basis of feature it is possible to accurately estimate the retrievability ranks of documents. If the correlation score is close to 0 then this means that there exists no relationship between the feature and the retrievability ranks. Thus on the basis of given feature, it is difficult to estimate the retrievability ranks of documents.

6.2. Discussion on results

Tables from 12 to 23 show the correlation between the retrievability ranks of documents and the features for all collections. From the Table results the following conclusions can be drawn:

- Most surface level features have significant correlation with the retrievability ranks. One factor that we consider important to mention here is that the surface level features are defined over the combinations of the following four attributes: (a) normalized term frequency relative to document length, (b) term document frequency or the term collection frequency relative to the total size of collection, (c) document length, and (d) the vocabulary size. Due to these attributes, the correlation between the feature and the retrievability ranks of retrieval model depends upon, whether or not the combination of these attributes is presented in the feature and the retrieval model, and how they are controlled in the retrieval model (i.e., depends on any parameter, or does not depend on any parameter). Since most of the attributes are present in the *BM25*, *NormTFIDF* and the four language modeling approaches, therefore, these models have a moderate correlation with *NATF*, *freq*, *NATF_freq*, *ADF*, *freq_low_df*, *document length*, and *vocabulary size*. In case of *NormTFIDF* the first attribute (normalized term frequency relative to document length) and the second attribute (term document frequency or term collection frequency relative to total size of collection) are not controlled with any parameter, thus *NormTFIDF* has somewhat high correlation with *NATF*, *freq*, *NATF_freq*, *ADF*, and *freq_low_df* as compared to *BM25* and four language modeling approaches (parameter controlled retrieval models). Standard *TFIDF* does not use first, third and the fourth attributes, thus *TFIDF* has a very low

Table 12
Correlation between the retrievability ranks of documents and the surface level features for the *TREC-CRT* collection.

Feature	NormTFIDF	BM25	DirS	JM	AbsDis	TwoStage	TFIDF	SMART
NATF	0.77	0.80	0.81	0.86	0.86	0.66	−0.24	0.26
freq	0.55	0.41	0.48	0.50	0.43	0.50	0.16	0.44
NATF_freq	0.29	0.23	0.26	0.27	0.23	0.26	0.09	0.32
GC_terms	−0.22	−0.32	−0.28	−0.37	−0.46	−0.03	0.68	0.63
freq_GC	−0.70	−0.69	−0.72	−0.79	−0.79	−0.55	0.33	0.07
ADF	0.63	0.66	0.66	0.68	0.66	0.56	−0.08	0.30
freq_low_df	−0.67	−0.69	−0.69	−0.73	−0.72	−0.57	0.17	−0.11
ADF_freq	0.29	0.31	0.30	0.30	0.31	0.26	0.00	0.24
Document Length	−0.53	−0.56	−0.57	−0.64	−0.66	−0.38	0.42	0.22
Vocabulary Size	−0.70	−0.72	−0.73	−0.78	−0.78	−0.59	0.24	−0.08

Table 13
Correlation between the retrievability ranks of documents and the surface level features for the *ChemAppPat* collection.

Feature	NormTFIDF	BM25	DirS	JM	AbsDis	TwoStage	TFIDF	SMART
NATF	0.60	0.62	0.59	0.67	0.57	0.52	0.22	0.57
freq	0.48	0.18	0.39	0.35	0.19	0.44	0.47	0.57
NATF_freq	0.30	0.11	0.23	0.22	0.12	0.27	0.29	0.31
GC_terms	0.38	0.06	0.37	0.21	0.02	0.49	0.74	0.82
freq_GC	−0.45	−0.40	−0.38	−0.52	−0.44	−0.30	0.04	−0.15
ADF	0.39	0.36	0.37	0.40	0.32	0.33	0.18	0.47
freq_low_df	−0.46	−0.46	−0.45	−0.50	−0.42	−0.39	−0.18	−0.49
ADF_freq	0.14	0.13	0.13	0.14	0.12	0.10	0.06	0.22
Document Length	−0.06	−0.24	−0.03	−0.22	−0.29	0.10	0.49	0.38
Vocabulary Size	−0.57	−0.59	−0.57	−0.65	−0.55	−0.51	−0.21	−0.54

Table 14
Correlation between the retrievability ranks of documents and the surface level features for the *DentPat* collection.

Feature	NormTFIDF	BM25	DirS	JM	AbsDis	TwoStage	TFIDF	SMART
NATF	0.40	0.47	0.43	0.49	0.45	0.45	0.21	0.49
freq	0.33	0.13	0.29	0.25	0.16	0.39	0.35	0.46
NATF_freq	0.18	0.05	0.15	0.13	0.08	0.24	0.23	0.25
GC_terms	0.26	0.06	0.30	0.14	0.04	0.45	0.51	0.68
freq_GC	−0.43	−0.38	−0.32	−0.48	−0.39	−0.20	0.09	−0.12
ADF	0.25	0.27	0.27	0.29	0.25	0.30	0.20	0.37
freq_low_df	−0.49	−0.50	−0.45	−0.53	−0.43	−0.35	−0.10	−0.48
ADF_freq	0.02	0.05	0.04	0.04	−0.05	0.08	0.12	0.18
Document Length	−0.15	−0.28	−0.08	−0.28	−0.29	0.08	0.33	0.24
Vocabulary Size	−0.56	−0.59	−0.53	−0.62	−0.53	−0.43	−0.11	−0.52

Table 15
Correlation between the retrievability ranks of documents and the surface level features for the *ATNews* collection.

Feature	NormTFIDF	BM25	DirS	JM	AbsDis	TwoStage	TFIDF	SMART
NATF	0.91	0.91	0.07	0.92	0.88	−0.43	0.03	0.53
freq	0.60	0.59	0.19	0.59	0.59	−0.07	0.17	0.62
NATF_freq	0.51	0.49	0.18	0.50	0.49	−0.02	0.17	0.57
GC_terms	−0.39	−0.39	0.37	−0.42	−0.34	0.65	0.41	0.41
freq_GC	−0.66	−0.65	0.25	−0.69	−0.59	0.62	0.30	0.06
ADF	0.84	0.84	0.10	0.85	0.82	−0.38	0.06	0.55
freq_low_df	−0.87	−0.87	−0.08	−0.88	−0.84	0.40	−0.04	−0.53
ADF_freq	0.07	0.07	0.14	0.07	0.08	0.11	0.14	0.22
Document Length	−0.87	−0.87	0.00	−0.89	−0.83	0.49	0.05	−0.41
Vocabulary Size	−0.88	−0.88	−0.06	−0.89	−0.85	0.42	−0.02	−0.50

Table 16
Correlation between the retrievability ranks of documents and the term weights features for the *TREC-CRT* collection.

Feature	NormTFIDF	BM25	DirS	JM	AbsDis	TwoStage	TFIDF	SMART
ATW	0.76	0.23	0.78	0.84	0.85	0.61	0.52	0.07
ATRP	0.69	0.75	0.76	0.82	0.85	0.54	0.56	0.86
VTRP	−0.64	−0.67	−0.70	−0.77	−0.78	−0.46	−0.21	−0.64
DiffMedianWeight	0.78	0.74	0.79	0.87	0.86	0.72	0.60	0.73
LowRankRatio	−0.61	−0.30	−0.35	−0.65	−0.59	−0.45	−0.59	−0.26

Table 17
Correlation between the retrievability ranks of documents and the term weights features for the *ChemAppPat* collection.

Feature	NormTFIDF	BM25	DirS	JM	AbsDis	TwoStage	TFIDF	SMART
ATW	0.47	−0.03	0.55	0.60	0.54	0.40	0.62	−0.11
ATRP	0.18	0.49	0.20	0.40	0.42	−0.05	0.37	0.94
VTRP	−0.10	−0.31	−0.07	−0.29	−0.27	0.08	−0.12	−0.90
DiffMedianWeight	0.55	0.58	0.64	0.71	0.59	0.66	0.69	0.80
LowRankRatio	−0.35	−0.20	−0.33	−0.43	−0.34	−0.38	−0.47	−0.27

Table 18
Correlation between the retrievability ranks of documents and the term weights features for the *DentPat* collection.

Feature	NormTFIDF	BM25	DirS	JM	AbsDis	TwoStage	TFIDF	SMART
ATW	0.29	−0.02	0.44	0.46	0.43	0.40	0.40	−0.08
ATRP	0.08	0.34	0.11	0.28	0.31	0.07	−0.04	0.78
VTRP	0.01	−0.21	0.00	−0.19	−0.17	0.05	0.20	−0.72
DiffMedianWeight	0.35	0.45	0.49	0.52	0.46	0.56	0.47	0.76
LowRankRatio	−0.39	−0.33	−0.36	−0.45	−0.38	−0.32	−0.19	−0.31

Table 19
Correlation between the retrievability ranks of documents and the term weights features for the *ATNews* collection.

Feature	NormTFIDF	BM25	DirS	JM	AbsDis	TwoStage	TFIDF	SMART
ATW	0.90	0.30	0.15	0.92	0.86	0.07	0.42	−0.15
ATRP	0.93	0.90	0.06	0.94	0.88	0.01	0.11	0.92
VTRP	−0.91	−0.64	−0.02	−0.93	−0.85	0.04	0.01	−0.88
DiffMedianWeight	0.91	0.84	0.39	0.93	0.87	0.36	0.54	0.76
LowRankRatio	−0.68	−0.64	−0.11	−0.68	−0.62	−0.11	−0.28	−0.44

Table 20
Correlation between the retrievability ranks of documents and the density based features for the *TREC-CRT* collection.

Feature	NormTFIDF	BM25	DirS	JM	AbsDis	TwoStage	TFIDF	SMART
AvgDensity ($k=50$)	0.67	0.47	0.78	0.83	0.85	0.59	−0.06	0.05
AvgDensity ($k=100$)	0.67	0.47	0.78	0.83	0.85	0.59	−0.06	0.05
AvgDensity ($k=150$)	0.67	0.47	0.78	0.83	0.85	0.59	−0.06	0.05
AvgDensity-Top40Terms ($k=50$)	0.56	0.34	0.71	0.78	0.81	0.47	0.03	0.04
AvgDensity-Top40Terms ($k=100$)	0.56	0.34	0.71	0.78	0.80	0.47	0.04	0.04
AvgDensity-Top40Terms ($k=150$)	0.56	0.34	0.71	0.78	0.80	0.47	0.04	0.04

Table 21
Correlation between the retrievability ranks of documents and the density based features for the *ChemAppPat* collection.

Feature	NormTFIDF	BM25	DirS	JM	AbsDis	TwoStage	TFIDF	SMART
AvgDensity ($k=50$)	0.34	0.15	0.52	0.59	0.57	0.38	0.24	−0.17
AvgDensity ($k=100$)	0.34	0.14	0.52	0.59	0.57	0.39	0.24	−0.17
AvgDensity ($k=150$)	0.34	0.14	0.52	0.59	0.57	0.39	0.24	−0.17
AvgDensity-Top40Terms ($k=50$)	−0.08	−0.06	0.07	0.31	0.32	0.10	0.11	−0.12
AvgDensity-Top40Terms ($k=100$)	−0.09	−0.06	0.07	0.31	0.32	0.10	0.11	−0.12
AvgDensity-Top40Terms ($k=150$)	−0.09	−0.06	0.07	0.31	0.33	0.10	0.11	−0.12

Table 22
Correlation between the retrievability ranks of documents and the density based features for the *DentPat* collection.

Feature	NormTFIDF	BM25	DirS	JM	AbsDis	TwoStage	TFIDF	SMART
AvgDensity ($k=50$)	0.14	0.08	0.36	0.42	0.44	0.35	0.11	−0.14
AvgDensity ($k=100$)	0.14	0.08	0.36	0.42	0.44	0.35	0.11	−0.14
AvgDensity ($k=150$)	0.14	0.08	0.36	0.42	0.44	0.35	0.11	−0.14
AvgDensity-Top40Terms ($k=50$)	−0.13	−0.10	0.03	0.26	0.25	0.18	−0.07	−0.18
AvgDensity-Top40Terms ($k=100$)	−0.13	−0.11	0.03	0.25	0.25	0.18	−0.07	−0.18
AvgDensity-Top40Terms ($k=150$)	−0.13	−0.11	0.03	0.25	0.25	0.18	−0.07	−0.18

correlation with *NATF*, *freq*, and *NATF_freq* features. *SMART* also does not use the first three attributes, thus it has a low correlation with *NATF*, *freq*, *NATF_freq*, *ADF*, and *freq_low_df*. However, *SMART* uses vocabulary size attribute, therefore, as compared to *TFIDF* it has relatively a good correlation with the *vocabulary size* feature. The *GC-terms* is appeared as a best

Table 23
Correlation between the retrievability ranks of documents and the density based features for the *ATNews* collection.

Feature	NormTFIDF	BM25	DirS	JM	AbsDis	TwoStage	TFIDF	SMART
AvgDensity ($k=50$)	0.81	−0.18	0.18	0.89	0.63	0.15	0.43	−0.38
AvgDensity ($k=100$)	0.82	−0.18	0.17	0.89	0.63	0.15	0.43	−0.38
AvgDensity ($k=150$)	0.82	−0.18	0.17	0.89	0.63	0.14	0.44	−0.38
AvgDensity-Top40Terms ($k=50$)	0.90	0.79	0.08	0.93	0.87	0.14	−0.08	0.22
AvgDensity-Top40Terms ($k=100$)	0.90	0.79	0.08	0.93	0.87	0.14	−0.08	0.23
AvgDensity-Top40Terms ($k=150$)	0.90	0.79	0.08	0.93	0.87	0.14	−0.08	0.23

feature for *TFIDF* and *SMART*. The main reason is that *GC-terms* relies only on the absolute term frequencies within the documents, and similar to *TFIDF* and *SMART* it does not depend on any attribute. This attribute independent characteristics of *GC-terms* make it a good ranking predictor for *TFIDF* and *SMART* models.

- The features based on term weights also have significant correlation with the retrievability ranks. However, their correlation depends upon the amount of length and vocabulary skewness in the collection. On two less skewed collections (*DentPat* and *ChemAppPat*), the term frequencies (within the documents) differences are less extreme into different documents, and this makes the distribution of rank positions of terms in the sorted inverted lists less uniformed. Thus for these collections taking average or variance of rank positions of the terms in the form of *ATRP* or *VTRP* do not seem suitable idea. Due to this reason, the features *ATRP* and *VTRP* have low correlation, however, on the same collections, the features *ATW* and *DiffMedianWeight* have somewhat moderate correlation. The reason behind this perhaps could be that these features consider only the absolute term weights rather than term rank positions, thus seems suitable for capturing the s mall variation in the weights of terms. On two high skewed collections (*TREC-CRT* and *ATNews*), the term frequencies (within the documents) differences are somewhat high extreme, and due to this reason the correlations of *ATRP* and *VRTP* are moved from low to moderate level, and the correlations of *ATW* and *DiffMedianWeight* are moved from moderate to high level.
- Density based features have good correlation with high skewed collections (*TERC-CRT* and *ATNews*), but do not have good correlation with less skewed collections *ChemAppPat* and *DentPat*). Figs. 6 and 7 show the graphical relationship of $\hat{r}(d)$ with density based feature (*AvgDensity* ($k=50$)) for *TREC-CRT* and *ChemAppPat* collections respectively. Additionally, for most retrieval models, the density based features have positive correlation with the retrievability ranks (i.e., high retrievable documents have high average density, while low retrievable documents have low average density). This indicates that the low retrievable documents are mostly in the high density areas, and their neighbor documents have mostly similar term weights, and due to this reason their probability of retrievability is decreased. In case of *ChemAppPat* and *DentPat*) collections, the differences between the term weights of documents are less extreme, and due to this reason low and high retrievable documents have mostly similar average densities. This is the main reason why these collections do not have good correlation with density based features.
- Overall, there exists no feature that performs globally best for all retrieval models and for all collections. *NATF* has somewhat good correlation, however it breaks down in case of *TFIDF* and *SMART*. The main reason is that *NATF* is calculated on the basis of normalized term frequency, and *TFIDF* and *SMART* do not perform terms normalization relative to document length, thus give low correlation. The *Vocabulary Size* and *Document Length* features perform well in case of two high skewed collections (*TREC-CRT* and *ATNews*), however their performance break down in case of less skewed collections (*ChemAppPat* and *DentPat*). *GC-term* relies on the absolute term frequencies within the documents, thus achieves good performance for the *TFIDF* and *SMART*, but it has low correlation for the other retrieval models. *ATW* performs well for most of collections and retrieval models, however in case of *TFIDF* and *SMART* its correlation becomes low. Perhaps this could be because *TFIDF* and *SMART* rely on the absolute term frequencies, and for these models average scores of *ATW* are drifted away due to presence of some high frequency terms in the documents. Comparatively high correlation can be observed from the *DiffMedianWeight* and *LowRankRatio* features in most of the cases.

6.3. Combining multiple features

In the above section we analyze the correlation with the single features. We observe that there exists no feature that performs globally best for all retrieval models and for all collections. Thus it is worth to analyze to what extent combining multiple features improves the prediction performance. We combine multiple features and estimate the retrievability ranks using a regression tree. Regression tree is a powerful machine learning method that allows us to estimate the variance in a continuous dependent variable based on the combinations of multiple independent variables. The process of constructing a regression tree is similar to that of categorical classification tree. However, when building regression tree, there is no need to give prior class information. Regression tree builds the class information automatically by recursively partitioning the training samples into sub-groups (terminal nodes) that are internally more homogeneous than the parent nodes. This splitting is done with the help of either least squares (LS) or the least absolute deviation functions. At each terminal node, the mean value of the dependent variable is used as the predicted value.

We use WEKA machine learning toolkit[7] for performing the regression tree experiments. Additionally, bagging is used with the classifier to reduce the variance. Before building the tree, all features are normalized within the range [0–1] using min–max normalization. The regression tree is trained over training samples, and the correlations of predicted outcomes are verified over separate testing samples. We use all known retrievability samples (documents) of collections, and use training/testing samples with 50%/50% split combination.

Table 24 shows the correlation of the retrievability ranks of documents with multiple features together with the single best feature and the percentage of improvement gained by combining multiple features. Overall, a significant improvement is achieved by combining multiple features as compared to relying only on the single features. If we focus only on the retrieval models, then *BM25*

[7] http://www.cs.waikato.ac.nz/ml/weka/.

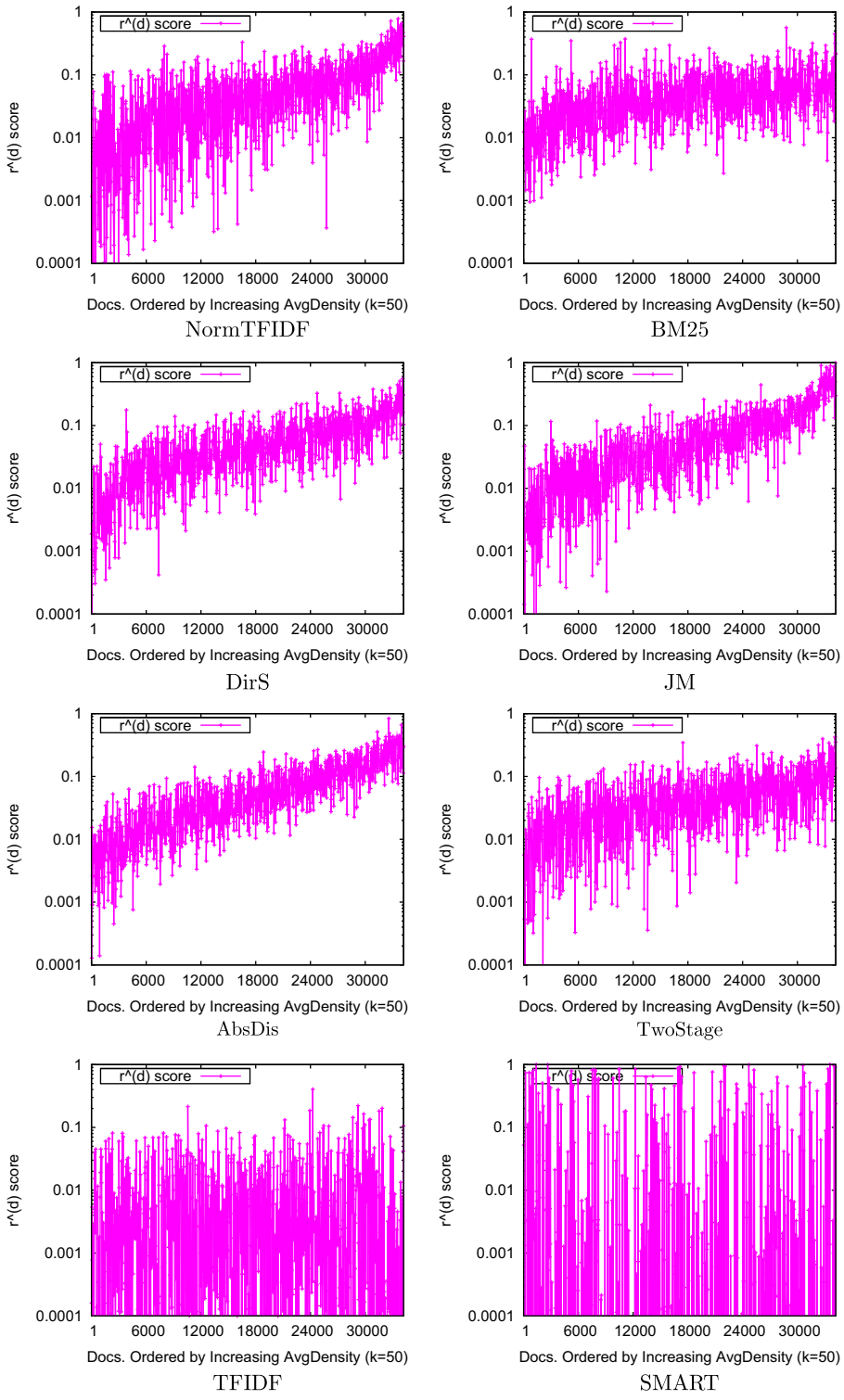

Fig. 6. Graphical relationship of $\hat{r}(d)$ with Density based Feature (*AvgDensity* ($k=50$) for *TREC-CRT* collection.

and *NormTFIDF* gained a large percentage of increased in the correlation on almost all collections. If we focus only on the collections, then large performance is achieved over *DentPat* collection. On *ATNews* collection, the highest correlation of *DirS* with the single best feature was 0.39. However, after combining features its correlation is increased from 0.39 to 0.58 with 49% of improvement. Similar performance improvement is observed with *TFIDF* on *ATNews* collection, when the correlation improves from 0.54 to 0.72 with 33% increment.

7. Conclusion

Documents retrievability is a novel measurement for the analysis of retrieval models effectiveness for recall-oriented retrieval domains. In recent years, several attempts are made for the bias quantification of retrieval models using this concept. 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

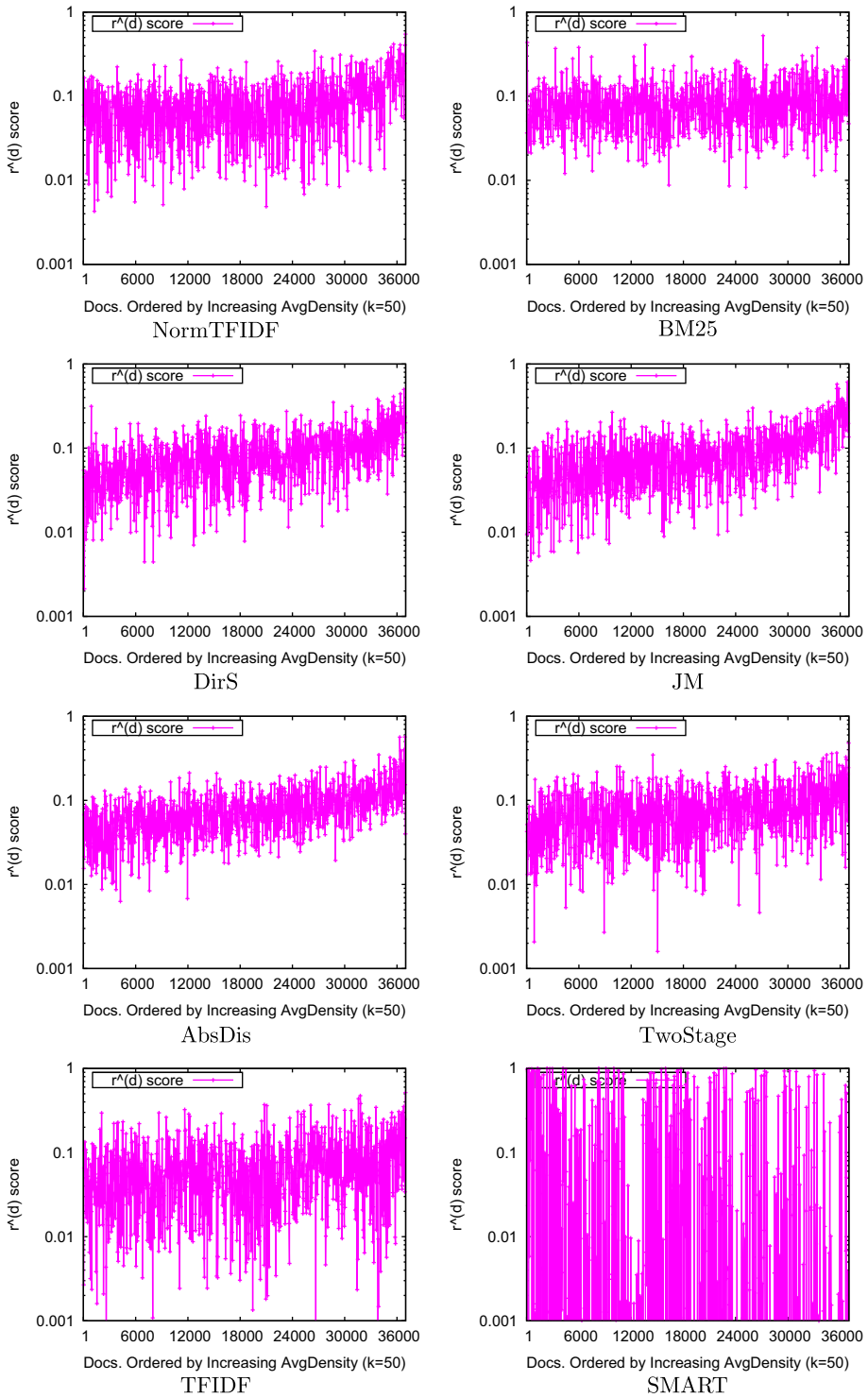

Fig. 7. Graphical relationship of $\hat{r}(d)$ with Density based Feature (*AvgDensity* ($k=50$)) for *ChemAppPat* collection.

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. One main limitation of retrievability analysis is that it requires the processing of exhaustive number of queries. This requires large processing time and resources. In order to handle this problem, in this paper we analyze the correlation between retrievability ranks of documents and document features. Our results confirm that there exist several features that show a high correlation with the retrievability ranks. This creates the possibility of estimating retrievability ranks of documents without processing queries. One major advantage of this approach is that it computes the retrievability ranks of documents more efficiently with less processing time and fewer resources than query based approach. However, on the other side, one major disadvantage of this approach is that it only estimates the retrievability ranks of documents, but cannot analyze how much there is a retrievability inequality between the documents of a collection with different retrieval models.

Table 24
Regression tree correlation with the document retrievability ranks for all collection. Table's cells represent three values. First value shows the correlation on the basis of combining multiple feature via regression tree, the second value in bracket shows the correlation with the single best feature, and the third value shows the percentage of improvement gained by combining multiple features.

Retrieval model	*TREC−CRT*	*ChemAppPat*	*DentPat*	*ATNews*
TFIDF	0.76 [0.68] [+12%]	0.77 [0.74] [+04%]	0.80 [+0.51] [+57%]	0.72 [0.54] [+33%]
NormTFIDF	0.82 [0.78] [+18%]	0.67 [0.60] [+12%]	0.74 [−0.56] [+32%]	0.93 [0.93] [+00%]
BM25	0.84 [0.80] [+05%]	0.72 [0.62] [+16%]	0.77 [−0.59] [+31%]	0.93 [0.90] [+02%]
SMART	0.89 [0.86] [+01%]	0.95 [0.94] [+01%]	0.94 [+0.78] [+21%]	0.95 [0.92] [+03%]
DirS	0.84 [0.81] [+04%]	0.69 [0.64] [+08%]	0.76 [−0.53] [+43%]	0.58 [0.39] [+49%]
JM	0.89 [0.87] [+02%]	0.74 [0.71] [+04%]	0.79 [−0.62] [+27%]	0.94 [0.94] [+00%]
AbsDis	0.89 [0.86] [+03%]	0.68 [0.59] [+15%]	0.74 [−0.53] [+40%]	0.90 [0.87] [+03%]
TwoStage	0.76 [0.72] [+06%]	0.69 [0.66] [+05%]	0.76 [+0.56] [+36%]	0.59 [0.65] [−09%]

References

[1] L. Azzopardi, R. Bache, 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, 2010, pp. 889–890.

[2] L. Azzopardi, M. de Rijke, K. Balog, 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, 2007, pp. 455–462.

[3] L. Azzopardi, V. Vinay, 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, California, USA, 2008, pp. 561–570.

[4] R. Bache, L. Azzopardi, Improving access to large patent corpora, in: Transactions on Large-Scale Data- and Knowledge-Centered Systems II, vol. 2, Springer, 2010, pp. 103–121.

[5] S. Bashir, A. Rauber, Analyzing document retrievability in patent retrieval settings, in: DEXA '09: Proceedings of the 20th International Conference on Database and Expert Systems Applications, Springer, Linz, Austria, 2009, pp. 753–760.

[6] S. Bashir, A. Rauber, Identification of low/high retrievable patents using content-based features, in: PaIR '09: Proceedings of the 2nd International Workshop on Patent Information Retrieval, 2009, pp. 9–16.

[7] S. Bashir, A. Rauber, Improving retrievability of patents with cluster-based pseudo-relevance feedback documents selection, in: CIKM '09: Proceedings of the 18th ACM Conference on Information and Knowledge Management, Hong Kong, China, 2–6 November, 2009, pp. 1863–1866.

[8] S. Bashir, A. Rauber, Improving retrievability and recall by automatic corpus partitioning, in: Transactions on Large-Scale Data- and Knowledge-Centered Systems II, vol. 2, Springer, 2010, pp. 122–140.

[9] S. Bashir, A. Rauber, Improving retrievability of patents in prior-art search, in: ECIR '10: 32nd European Conference on Information Retrieval Research, Springer, Milton Keynes, UK, 28–31 March, 2010, pp. 457–470.

[10] S. Bashir, A. Rauber, On the relationship between query characteristics and ir functions retrieval bias, Journal of the American Society for Information Science and Technology 62 (August (8)) (2011) 1512–1532.

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

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

[13] M. Lupu, J. Huang, J. Zhu, J. Tait, TREC-CHEM: large scale chemical information retrieval evaluation at trec, in: SIGIR Forum, vol. 43, no. 2, ACM, 2009, pp. 63–70.

[14] S.E. Robertson, S. Walker, 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.

[15] A. Singhal, 1997. At&t at trec-6, in: The 6th Text Retrieval Conference (TREC6), pp. 227–232.

[16] E. Weigl, Mitigating the Bias of Retrieval Systems by Corpus Splitting an Evaluation in the Patent Retrieval Domain, MS. Thesis, Vienna University of Technology, 2011.

[17] C. Zhai, Risk Minimization and Language Modeling in Text Retrieval, PhD Thesis, Carnegie Mellon University, 2002.

Shariq Bashir is currently working as a Research Scientist at Center of Science and Engineering, New York University, Abu Dhabi. He is also an Assistant Professor of computer science at NUCES-FAST, Islamabad. He received his Ph.D. in Computer Science from Vienna University of Technology, Austria. His research interests include information retrieval, unsupervised IR systems evaluation, retrieval bias analysis, documents retrievability analysis, query expansion, learning to rank and opinion-based entity ranking. He has published more than 30 research papers in leading international conferences and journals of information retrieval and data mining.