Improving Retrievability and Recall by Automatic Corpus Partitioning

Shariq Bashir and Andreas Rauber

Institute of Software Technology and Interactive Systems
Vienna University of Technology, Austria
http://www.ifs.tuwien.ac.at

Abstract. With increasing volumes of data, much effort has been devoted to finding the most suitable answer to an information need. However, in many domains, the question whether any specific information item can be found at all via a reasonable set of queries is essential. This concept of Retrievability of information has evolved into an important evaluation measure of IR systems in recall-oriented application domains. While several studies evaluated retrieval bias in systems, solid validation of the impact of retrieval bias and the development of methods to counter low retrievability of certain document types would be desirable.

This paper provides an in-depth study of retrievability characteristics over queries of different length in a large benchmark corpus, validating previous studies. It analyzes the possibility of automatically categorizing documents into low and high retrievable documents based on document properties rather than complex retrievability analysis. We furthermore show, that this classification can be used to improve overall retrievability of documents by treating these classes as separate document corpora, combining individual retrieval results. Experiments are validated on 1.2 million patents of the TREC Chemical Retrieval Track.

1 Introduction

The objective of Information Retrieval (IR) systems is to maximize effectiveness. In order to do so, IR systems attempt to discriminate between relevant and non-relevant documents. For measuring effectiveness, metrics such as Average Precision, Q-measure, Normalized Discounted, Cumulative Gain, Rank-Based Precision, Binary Preference (bref) are used [15]. The main limitation of these is, that they focus almost exclusively on precision, i.e. the fact that the (most) relevant documents are returned on top of a ranked list, as this constitutes the primary criterion of interest in most standard IR settings. With evaluation measures such as recall and F_β, aspects of the completeness of the result set are being brought into consideration.

Most information retrieval settings, such as web search, are typically precision-oriented, i.e. they focus on retrieving a small number of highly relevant documents. However, in specific domains, such as patent retrieval or law, recall becomes more relevant than precision: in these cases the goal is to find all relevant documents, requiring algorithms to be tuned more towards recall at the

A. Hameurlain et al. (Eds.): TLDKS II, LNCS 6380, pp. 122–140, 2010.

cost of precision. This raises important questions with respect to retrievability and search engine bias: depending on how the similarity between a query and documents is measured, certain documents may be more or less retrievable in certain systems, up to some documents not being retrievable at all within common threshold settings. Biases may be oriented towards popularity of documents (increasing weight of references), towards length of documents, favor the use of rare or common words; rely on structural information such as metadata or headings, etc.

This gave rise to a new evaluation measure for retrieval systems, namely *retrievability* [1]. Retrievability measures, in how far a system is able to retrieve at least in principle (via a set of reasonably queries) any document in a given corpus. A number of studies on document corpora of limited size have shown, that different retrieval systems perform differently on this task, i.e. that they exhibit a certain bias towards some type of documents (influenced e.g. by document length, vocabulary richness). It can be shown, that in some cases certain documents cannot be retrieved at all within the set of top-c documents returned for any query (within certain constraints, e.g. up to a certain number of query terms) [1,3]. This is due to the fact, that any retrieval system is inherently biased towards certain document characteristics. Bias to some document characteristics [16] is a concept used in Information Retrieval (IR) to describe the fact that a retrieval system gives preference to certain features of documents when ranking retrieval results. There are several techniques for calculating document relevance with respect to query terms. For example **PageRank** [12] calculates web page relevance by favoring large inlink over small inlink counts. In PageRank-style algorithms, if some documents have a higher number of inlinks, they will be ranked higher in the result list. Therefore, the PageRank algorithm is highly biased towards popular documents. Probabilistic retrieval systems such as **BM25** [14], **tf-idf** and **BM25F** [13] are biased towards documents which contain high term frequencies and many different terms, i.e. they favour long documents. Whatever the relevance criterium used in a retrieval system, the main purpose of introducing relevance in query terms is favouring certain types of documents over others, so that the users can retrieve the most relevant documents quickly from top rank results. There is a severe risk that a certain number of documents cannot be found in the top-n ranked results via any query terms that they would actually be relevant for, which ultimately decrease the usability of the retrieval system [2].

Using retrievability measurement, a document corpus can be analyzed, identifying, which documents are highly retrievable (i.e. they can be found by many queries), and which ones show low, down to no retrievability at all, i.e. they cannot be found in the top-c results by any query under a specific retrieval model. On top of this, research indicates (again on datasets of limited size) that it may be possible to identify for a given retrieval system, which documents are likely to show high or low retrievability based on document characteristics, i.e. without performing extensive retrievability measurement [4].

If the assumptions put forward by these results hold true for larger corpora as well, this raises the question, whether we may be able to improve overall retrievability by treating a document corpus as consisting of two different sub-corpora of documents, namely those with high and low retrievability in a given retrieval system. The experiments presented in this paper show, that this is in fact the case.

However, before validating this hypothesis, it is important to double-question the results reported so far in literature on retrievability analysis. While extensive experiments involving a massive amount of query processing were performed, all numbers published on retrievability results suffer from some limitations, by it either that only a comparatively small number of documents was used (e.g. 7,000 or 50,000 docs in [3,4], and/or that only rather short queries (only up to 2 terms combinations in [1]) were processed, and/or limiting the total number of queries issued per document rather than creating an exhaustive set of queries (e.g. max. 90 queries in [4]). Specifically the latter will usually penalize longer documents that potentially could be found via a larger number of (more exotic) query terms combinations, which would also reflect realistic settings in many search scenarios. This also seems to be reflected in the somewhat counter-intuitive figures published in said research, showing a higher retrievability bias (and thus lower overall retrievability) for longer queries, while intuitively higher retrievability should be expected for more specific (i.e. longer) queries.

In this paper we present a series of experiments on a large-scale document corpus in the same application domain as the previous studies on retrievability, i.e. patent retrieval. A representative benchmark corpus of 1.2 million patents used in the TREC Chemical Retrieval Track (TREC-CRT) is being used to validate the hypothesis of uneven retrievability in a large corpus [11]. Specifically, we verify whether retrievability is lower even when using longer queries. To this end we first perform standard exhaustive retrievability evaluation on short queries, followed by query generation using longer queries for documents exhibiting low retrievability on short queries.

We then replicate experiments to classify the entire document corpus into documents with high and low retrievability, yet using a significantly simpler set of features. Extensive experiments analyze the parameter settings required to obtain a suitable training corpus defining the classes of documents showing extremely high retrievability (i.e. dominating result lists on a huge number of queries), as well as documents showing extremely low retrievability, i.e. which are virtually impossible to retrieve.

Having classified the entire document corpus into these two categories, we then perform retrieval by treating these classes as independent partitions, processing queries independently for each and subsequently combining the result sets. We can show that this helps in increasing overall retrievability, reducing the dominance of certain documents in query processing and thus reducing the bias of a retrieval system. This approach thus provides a higher probability of being able to at least potentially find each document in a corpus.

Finally, an evaluation over the TREC-CRT prior art search task reveals, that this retrieval via two partitions of documents with high and low retrievability also helps in increasing overall recall for all baseline systems evaluated.

The remainder of this paper is structured as follows. Section 2 reviews related work on retrievability analysis, emphasizing the key messages learned from these as well as pointing to some limitations in the evaluations. Section 3 first introduces the concept of retrievability measurement and presents a modified score considering a potential bias introduced by the way queries are generated. It then describes the TREC-CRT benchmark corpus used for the experiments in this paper and retrievability results for several configurations of this corpus and different retrieval models. Section 4 then presents an approach to automatically classify documents into potentially high and low retrieval classes based on features capturing term distribution characteristics. Section 5 finally evaluates retrieval performance on the partitioned corpus, analyzing both retrievability as well as recall for the TREC-CRT prior art task. Section 6 briefly summarizes the key lessons learned and points to future work to evaluate the impact on real-life systems.

2 Related Work

2.1 Patent Retrieval

Patent Retrieval is a highly recall-oriented domain, aiming at identifying all documents relevant to a particular query. Several specific types of query processes may be identified in this domain, such as

Priot Art Search: This is a core step when planning to file a new patent application. Here, a survey is conducted in each national intellectual office for checking whether there exist any inventions similar to a given patent application. The mechanism that is generally used when collecting relevant patents applications for such a survey is keyword based query. Query terms are mostly extracted from the Claim sections. Query expansion is used to add related terms for improving the breath of the retrieval process [8].

Invalidity Search: In invalidity search the examiners have to find out the existing patent specifications that describe the same invention for collecting claims to make a particular patent invalid. In this search process, the examiners extract relevant query terms from patent applications particular from the Claim sections for creating query sets [9].

Right-to-Use: Right-to-use searches are conducted prior to marketing a new product for confirming whether a new patent application is infringing any existing patent application or not. In this application area, the method that is generally used is keyword based retrieval. However, query terms that are used for searching documents do not depend solely on a single patent application. Clustering is also widely used for identifying more target oriented queries which can cover all the related applications [6].

There are also many other patent processing applications such as patent map generation, current awareness search, legal status report, patenting activity report and trend mining. In all of these applications knowledge discovery methods such as data mining and machine learning are generally used for discovering competitive intelligence, which are not directly related to keyword based search.

2.2 Evaluating Retrievability

The evaluation of retrieval systems has always received much attention in the IR research community. Conventionally, retrieval systems are evaluated using a variety of precision and recall based measures [15]. However, these do not evaluate, what we can find and cannot find in a collection. Yet, for some specific retrieval applications like *patents* (or the legal domain in general), recall is considered more important than precision.

In addition to using traditional IR metrics for evaluation, Azzopardi et al. [1] introduce a measure for evaluating systems on the basis of retrievability scores of individual documents. It measures, how likely a document can be found at all by a specific system, with the analysis of the individual retrievability scores of documents performed using Lorenz curves and Gini coefficients. Their experiments with AQUAINT and .GOV datasets reveal that with a TREC-style evaluation a proportion of the documents with very low retrievability scores (sometimes more than 80% of the documents) can be removed without significantly degrading performance. This is because the retrieval systems are unlikely to ever retrieve these documents due to the bias they exhibit over the collection of documents.

In [3] we analyze 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. That evaluation was based on using a rather limited set of queries. Experiments revealed, that 90% of patents which are highly retrievable across all types of queries, are not highly retrievable on their relevant query sets.

One of the limitations of the approaches published so far is, that – due to the enormous amount of queries involved – only short (max 2-terms) queries are usually evaluated. Where longer queries are selected, they are limited to a drastically small subsets of all potential longer queries, virtually eliminating their effect on retrievability analysis for a document. We are thus trying to replicate a few of the experiment set-ups to study, in how far more specific, longer queries can help mitigate the problem of low retrievability. Jordan et al. [10] consider controlled query generation for evaluating the impact of retrieval systems performance. The main purpose of their study was to expose the performance of different algorithms, specifically how they react to queries of varying length and term quality (in case of noisy terms), which may also have an impact on retrievability.

Another caveat may lie in the retrievability measure used, which does not consider the (different) numbers of queries generated for each document. Both approaches rely on exhaustive query generation based on terms combinations of a

document's vocabulary. This leads to drastically different numbers of queries that can retrieve particularly vocabulary-rich, longer documents. We thus propose a slight adoption of the retrievability score considering the number of queries used to try to retrieve a particular document.

Techniques that address query expansions for the legal domain have been proposed by Custis et al. [5]. They evaluate query expansion methods for legal domain applications retrieval on the basis of query document term mismatch. For this purpose, they systematically introduce query document term mismatch into a corpus in a controlled manner and then measure the performance of IR systems as the degree of term mismatch changes.

For realistic evaluations of patent retrievability, the generic query generation approaches will need to be fine-tuned to specific domains. Fujii [7], for example, applies link analysis techniques to the citation structure for efficient patent retrieval. They first perform text based retrieval for obtaining the top-c patents. They then compute citation scores based on PageRank and topic-sensitive citation-based methods. Finally, both the text-based and citation-based scores are combined for better ranking. Also, more recently, full patents start being used as a query instead of selecting relevant keywords from them for prior-art search [18].

Applying these concepts may lead to more realistic retrievability analysis results, once the principles and limitations of the new measure are fully understood.

3 Retrievability Evaluation

3.1 Relative Retrievability Measurement

Given a retrieval system RS with a collection of documents D, the concept of retrievability is to measure how well each document $d \in D$ is retrievable within the top-c rank results of all queries, if RS is presented with a large set of queries $q \in Q$. Retrievability of a document is essentially a cumulative score that is proportional to the number of times the document can be retrieved within that cut-off c over the set Q [1]. A retrieval system is called best retrievable, if each document d has nearly the same retrievability score, i.e. is equally likely to be found. More formally, retrievability $r(d)$ of $d \in D$ can be defined as follows.

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

$f(k_{dq}, c)$ is a generalized utility/cost function, where k_{dq} is the rank of d in the result set of query q, c denotes the maximum rank that a user is willing to proceed down the ranked list. The function $f(k_{dq}, c)$ returns a value of 1 if $k_{dq} \leq c$, and 0 otherwise.

Retrievability inequality can further be analyzed using the Lorenz Curve. Documents are sorted according to their retrievability score in ascending order, plotting a cumulative score distribution. If the retrievability of documents is distributed equally, then the Lorenz Curve will be linear. The more skewed the

Table 1. Experiment set-up: number of queries generated for each subset of documents (without duplicates)

Experiment Method	queries	no docs	1-terms	2-terms	3-terms	4-terms
A = complete	all-queries	1.2 million	437,038	236,513,386	-	-
B = 20 doc top/bottom	all queries	20	8,568	1,239,449	52,668,550	-
C = random 5% (training)	20% cut	60,000	46,693	25,672,536	1,689,346,590	3,635,659,348
D = prior-art	20% cut	34,200	29,347	10,926,552	1,037,061,490	2,366,376,269

curve, the greater the amount of inequality or bias within the retrieval system. The Gini coefficient G is used to summarize the amount of bias in the Lorenz Curve, and is computed as follows.

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

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

However, the retrievability measure as defined above is a cumulative score over all queries. Thus, longer documents that contain a larger vocabulary, potentially have a higher retrievability score than shorter documents. While this is desirable as a general measure of retrievability, in settings where the actual set of queries is created directly from the documents to be found, this may have a negative impact. This is because a larger number of queries are generated for these longer documents. We thus propose for evaluations like these to normalize the cumulative retrievability score by the number of queries that were created from and thus potentially can retrieve a particular document.

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

where $\hat{Q}$ is the set of queries that can retrieve d when not considering any rank cut-off factor.

3.2 Experiment Set-Up

We use the 1.2 million patents from the TREC Chemical Retrieval Track (TREC-CRT)[1], allowing validation of retrievability measurements on a large-sale corpus within a recall-oriented domain [11].

Retrievability is evaluated for three different retrieval models, namely standard TFIDF based retrieval ranking by the sum of *tfidf* values for the query

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

Table 2. Number of queries generated and average number of documents retrieved per query (Experiment D)

Query Set	# Queries	avg. #docs retrieved
Single Term Queries	29,347	33,542.94
Two Terms Queries	10,926,552	21,675.46
Three Terms Queries	1,037,061,490	12,178.82
Four Terms Queries	2,366,377,269	6,472.35

terms; the OKAPI retrieval function (BM25) [14]; and a Language Modeling approach based on Dirichlet Smoothing (LM) [19].

For each document in the corpus a set of queries is generated using all terms that appear more than once in the document. In total, four subsets of queries are created, consisting of all single terms, 2-, 3-, and 4-terms combinations. These queries are then posed against the complete corpus of 1.2 million documents as boolean queries with subsequent ranking according to the chosen retrieval model to determine retrievability scores as defined in Equ. 3.

An overview of the various retrievability experiments performed is provided in Table 1. First, initial retrievability scores are determined using the complete set of queries for single term and 2-terms queries, resulting in sorting of documents according to their cumulative retrievability score (experiment A).

As terms distributions in a document tend to have an impact on retrievability, we then selected the 10 documents with the highest and lowest average terms frequencies. For these 20 documents, an exhaustive set of 3-terms and 4-terms queries is passed against the entire corpus to determine, whether low retrievability also persists over longer queries, or whether previously reported results along this line could be an artifact of processing only a rather small subset of the entire set of longer queries (experiment B).

After validating that, indeed, documents showing high/low retrievability over one and two terms queries also show corresponding retrievability over longer queries, and validating that this behavior is not impacted by selecting only a subset of queries, a random 5% subset of the entire corpus was selected (experiment C). Exhaustive sets of single term as well as 2-terms queries were combined with a set of 3- and 4-terms queries with a limit of 20% of the maximum number of potential queries (i.e. terms permutations)

A last group of retrievability experiments are performed using the set of 34,200 patents that are used as ground-truth in the evaluation of the 1,000 *prior art search task* of TREC-CRT, where they are referenced as relevant prior art patents [11]. This subset was chosen to evaluate, whether improving retrievability also improves accuracy for recall-oriented retrieval results, or whether promoting low-retrieval documents would harm accuracy. It allows us to verify whether predominantly documents with a low retrievability score are missed in the prior art retrieval process.

For this subset, an exhaustive set of queries for single term and 2-terms queries, as well as 20% of all 3- and 4-terms queries have been processed against

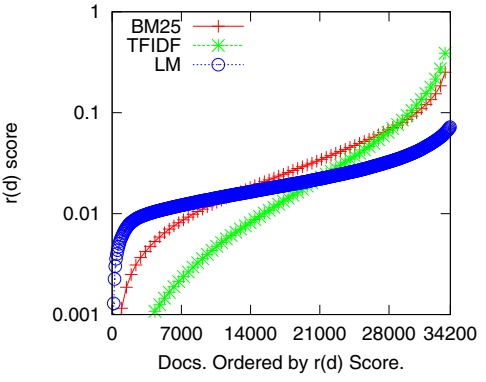


Fig. 1. Retrievability inequality for 3 retrieval systems, corpus (Experiment) A, rank cut-off c=150

the entire corpus of 1.2 million documents (experiment D). Each of these experiments was performed for the three retrieval models.

The exact configuration of the document corpora (document IDs) are available from our webpage[2]. Table 2 lists the number of queries generated, for example, for experiment corpus D and the average number of documents returned per query. This provides an indication of the impact for the rank cut-off factor. It also shows the significant decrease of the number of documents returned for longer and thus more specific queries – which, however, have to be selected from a drastically increasing number of potential query terms combinations.

Figure 1 compares the retrievability inequality of the 3 different retrieval models by sorting documents according to their $r(d)$ score. Ideally, all documents would be equally retrievable, i.e. they can be found by an equal fraction of all queries, resulting in a horizontal line. In reality, a significant number of documents (a few hundred for BM25, several 1,000 for TFIDF) cannot be found via any query, via a small number of documents are returned for a huge number of queries. TFIDF shows a stronger bias than BM25, whereas LM shows the lowest bias of all systems, not having any unretrievable documents.

Detailed evaluation of the retrievability values of the 20 documents in corpus (Experiment) B revealed, that the retrievability values determined on 2-terms and 3-terms queries lead to almost identical rankings by r(d) score. Thus, using r(d) scores based on the aggregated results of exhaustive single and 2-terms queries, combined with a sufficiently large number of 3- and 4-terms queries seems to be a solid basis for further analysis. However, better results may be obtainable by using more sophisticated query generation strategies which create the most probable subset of queries according to real-life scenario assumptions.

[2] `url.witheld.for.review/page/not/publicly/linked.xxxx`

4 Partitioning the Corpus

The experiments above reveal that, indeed, a corpus consists of documents that show highly different behavior in retrievability. Some documents are returned within the top-c results for a huge number of queries, possibly suppressing others that almost never show up within the top-c results for any query. This means that these documents are virtually inexistent for a searcher. One of the goals in recall-oriented application domains is to esure that all relevant documents are potentially found. We thus need to devise ways to ensure that documents exhibiting low retrievability can also be retrieved by queries that they are potentially relevant for. In order to do so, we propose to split a document corpus into two categories, consisting of documents with high and low retrievability, respectively. These two partitions can subsequently be accessed seperately, possibly via retrieval models that are optimized for the document characteristics, ensuring better overall retrievability.

The obvious way to divide a corpus into documents with high and low retrievability by performing extensive retrievability analysis unfortunately is prohibitive for any realistically-sized corpus. We thus pick up an idea proposed in [4] and try to classify documents into these two categories via a set of surface-level features. While the authors in that study propose a rather complex and extensive set of features to describe documents via co-location of word pairs within certain windows, we apply a simpler set of features that seems to compare favorably with the published results (although they cannot be compared directly as a different document corpus is used).

4.1 Features for Retrievability Classification

We compute a number of statistical and information-theoretic features from these documents, resulting in an only 10-dimensional feature vector, capturing the distributional characteristics of terms within a document and over the whole corpus.

- **Normalized Average Term Frequencies (NATF):** average of normalized term frequencies of all terms in a document, calculated as

$$NATF = \frac{\sum_{t \in T_d} \frac{f_{(t,d)}}{|d|}}{|T_d|} \tag{4}$$

  where T_d represents the set of all unique terms in a document d. $f_{(t,d)}$ is the frequency of term t in d (also referred to as tf), and $|d|$ represents the length of document.
- **Number of Frequent Terms (freq):** calculates, how many terms have a term frequency larger than a pre-defined threshold of, in our case, 6. This allows us to capture uneven distributions, identifying the absolute number of frequent terms, especially in collections with documents of drastically different length.

- **NATF of Frequent Terms (NATF_freq):** calculates NATF only for frequent terms, i.e. terms having a term frequency larger than e.g 6, to eliminate the impact of a potentially large number of rare terms having only low tf values.
- **Gini Coefficient of Term Frequencies (GC_terms):** measures how balanced the distribution of term frequencies is within a document. Similar to evaluating retrievability inequality, GC_terms captures, whether all terms have similar or rather different tf values.

$$GC\_terms = \frac{\sum_{t \in T_d}(2 \cdot i(t) - |T_d| - 1) \cdot f_{(t,d)}}{(|T_d| - 1) \sum_{t \in T_d} f_{(t,d)}} \tag{5}$$

  where $i(t)$ is the index of term t in set T_d after sorting terms in ascending order of their frequencies.
- **Number of Frequent Terms based on Gini-Coefficient (freq_GC):** rather than using a fixed threshold as for NATF_freq, terms with the highest tf values are iteratively removed until the resulting Gini Coefficient for the entire document drops below 0.25, i.e. is rather homogeneous. The number of terms removed provides a different measure for the number of frequent terms contributing to retrieval inequality.
- **Average Document Frequency (ADF):** measures in how far a document consists of rather common or rather specialized vocabulary by summing up the df values of its vocabulary.

$$ADF = \frac{\sum_{t \in T_d} f(d,t)}{|T_d|} \tag{6}$$

  where $f(d,t)$ represents the document frequency (df) of term t in D.
- **Frequent Terms with Low Document Frequency (freq_low_df):** measures, how many frequent terms in a document have a rather low document frequency, i.e. are exotic terms in the corpus. Thresholds are set to min tf of 6, and max df of 3,000.
- **Average Document Frequency of Frequent Terms (ADF_freq):** captures the exoticity of the frequent terms in the vocabulary of a document.
- **Relative Term Frequency (TF_rel):** Relative term frequency captures, how the term frequencies in the current document compare to the term frequencies of the subset of the documents where the respective terms have the highest term frequencies. It is calculated by determining the average top term frequency (ATTF) of each term in the top 10% of documents where this term has the highest term frequency.

$$ATTF(t) = \frac{\sum_{d \in \hat{D}_t} f(t,d)}{|\hat{D}_t|} \tag{7}$$

  where $\hat{D}_t$ is the 10% set of documents that have the highest tf value for term t. These values are subsequently aggregated for a document as the relation between the ATTF and the tf value in the given document.

$$TF\_rel = \frac{\sum_{t \in T_d} \frac{ATTF(t)}{f(t,d)}}{|T_d|} \tag{8}$$

Table 3. Classification accuracy (high/low retrievable), Naïve Bayes

Retr. Sys.	rank cut-off factors				
	50	**100**	**150**	**250**	**350**
TFIDF	85%	84%	83%	82%	79%
BM25	82%	81%	81%	80%	77%
LM	80%	79%	78%	77%	76%

Table 4. Percentage of bottom and top r(d) score documents used as basis for training set definition (tr. low and tr. high)and resulting classification of entire corpus into low and highly retrieval documents.

Retr. System	tr. low	tr. high	% clas. high	% class. low
TFIDF	35%	35%	56%	44%
BM25	25%	40%	45%	55%
LM	25%	40%	42%	58%

If TF_rel is high, then the document has rather low tf values for its terms compared to the top tf values for these terms in the corpus.

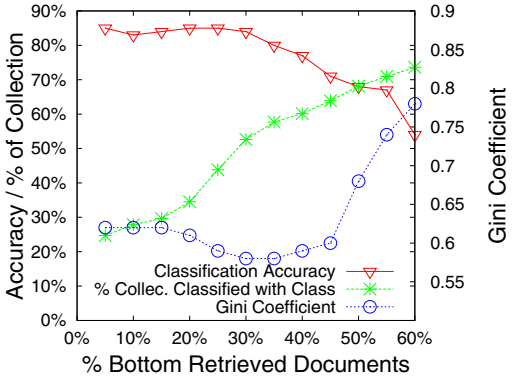
– **Patent Length (PL):** represents the length of document in words ($|d|$).

4.2 Model Training

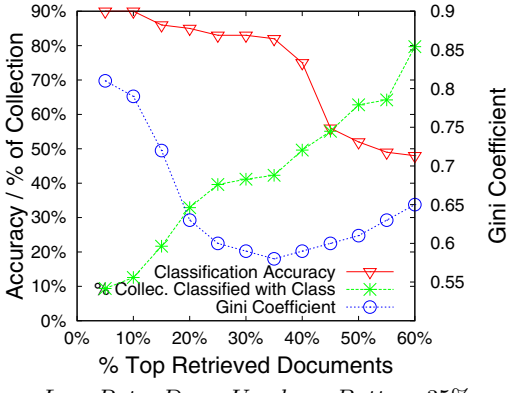
In order to train a model, we first need to identify a suitable training set configuration, picking certain subsets from the documents showing low and high retrievability. We perform retrievability analysis on corpus (Experiment) C, i.e. a random 5% selection of all documents of the TREC-CRT corpus for which retrievability scores are calculated with exhaustive 1- and 2-terms queries, as well as 20% of all 3- and 4-terms queries, with a rank cut-off factor of $c = 150$.

The set of documents is subsequently split into 3 subsets consisting of documents with high, medium and low retrievability scores. Training instances are picked only from the set of documents with high and low retrievability. A number of different configurations have been evaluated using 10-fold cross-validation training a Naïve Bayes classifier as implemented in the WEKA toolkit [17] to determine the optimal split of the training corpus. Experiments setting the split in 5% increments for documents with low and high $r(d)$ as the bottom and top 5% to 60% revealed that the optimal split was to use the bottom 25% and the top 40% documents in the case of the BM25, whereas the optimal split for the TFIDF retrieval model was both a top and bottom threshold of 35%.

Figures 2 and 3 show the resulting classification accuracies with either the top or bottom threshold fixed, varying the other. The percentage of documents classified into the respective classes when applying this classifier to the entire document collection is also depicted in these figures. We furthermore analyzed



High Retr. Docs. Used = Top 35%
(a) with TFIDF and Low Retr. Docs.

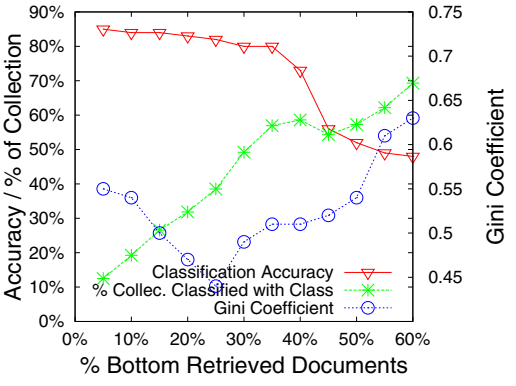


Low Retr. Docs. Used = Bottom 35%
(b) with TFIDF and High Retr. Docs.

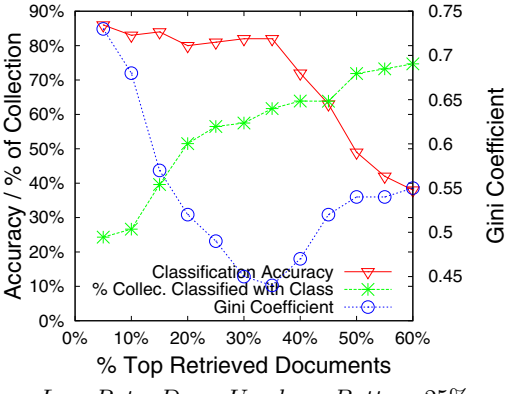
Fig. 2. Impact of selecting training documents from different subsets of the training corpus. rank cut-off c=150, TFIDF model, (a) fixing upper threshold 35%, (b) fixing lower threshold at 35%.

the classification accuracy on the training set for different rank cut-off factors as summarized in Table 3.

Details on the final classifier training split are provided in Table 4. For the TFIDF retrieval model, 56% of all documents were classified as potentially having low retrievability with a split of 35/35 for top/bottom categories for training set determination from which the actual training documents were sampled via 10-fold cross-validation. For BM25, the optimal split was 25/40, resulting in 45% of the entire corpus being classified as having potentially low $r(d)$ scores, whereas for the LM approach 42% were classified as low-retrievable with a 25/40 training set delimiter.



High Retr. Docs. Used = Top 40%
(a) with BM25 and Low Retr. Docs.



Low Retr. Docs. Used = Bottom 25%
(b) with BM25 and High Retr. Docs.

Fig. 3. Impact of selecting training documents from different subsets of the training corpus. rank cut-off c=150, BM25 model, (a) fixing upper threshold 40%, (b) fixing lower threshold at 25%.

5 Partition-Based Retrieval

Once a document corpus has been divided into two partitions containing documents with potentially high and low retrievability scores, queries can be passed to these corpora independently. On each partition queries are processed independently, and the final result set is merged to form a single result set. This ensures that the final result set will always include also 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 the query.

A number of merging principles can be envisaged. While relative similarity scores may be used, we used only rather simpler merge strategies, namely

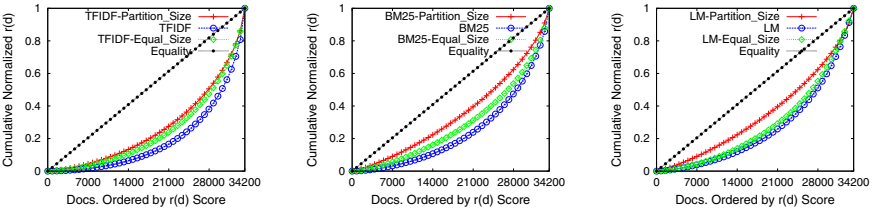


Fig. 4. Visualization of Documents Retrievability using Lorenz Curve, rank cut-off (c=150). Equality refers to an optimal system which has no bias.

Table 5. Bias of IR systems with different rank cut-off (c) factors and query sets

Retr. Sys.	Approach	Four Terms Queries					Three Terms Queries					Two Terms Queries				
		50	100	150	250	350	50	100	150	250	350	50	100	150	250	350
BM25	*no split*	0.75	0.67	0.56	0.54	0.51	0.63	0.56	0.52	0.51	0.49	0.50	0.49	0.44	0.41	0.37
	Part_Size	0.54	0.50	0.42	0.40	0.36	0.46	0.39	0.36	0.35	0.33	0.42	0.35	0.31	0.28	0.24
	Equal_Size	0.63	0.60	0.51	0.50	0.44	0.5 6	0.47	0.45	0.43	0.42	0.49	0.44	0.37	0.35	0.30
TFIDF	*no split*	0.88	0.83	0.72	0.65	0.64	0.77	0.70	0.61	0.57	0.56	0.69	0.62	0.53	0.49	0.44
	Part_Size	0.70	0.67	0.59	0.51	0.49	0.59	0.52	0.46	0.41	0.39	0.49	0.44	0.37	0.33	0.29
	Equal_Size	0.78	0.74	0.66	0.58	0.56	0.67	0.59	0.52	0.47	0.44	0.58	0.56	0.46	0.41	0.34
LM	*no split*	0.64	0.54	0.43	0.40	0.39	0.60	0.53	0.48	0.46	0.43	0.54	0.46	0.40	0.37	0.33
	Part_Size	0.44	0.38	0.29	0.25	0.22	0.40	0.34	0.30	0.28	0.26	0.38	0.34	0.27	0.25	0.23
	Equal_Size	0.52	0.47	0.38	0.34	0.32	0.49	0.43	0.41	0.39	0.36	0.45	0.39	0.34	0.30	0.28

- **equal_size**: an equal number of documents from the low and high retrievable subsets are returned, i.e. for a given cut-off factor c, $c/2$ documents were taken from each partition
- **partition_size**: in this case, the number of documents included in the final result set is relative to the size of the two partitions.

5.1 Retrievability Analysis

Figures 2 and 3 show the Gini coefficients using equal size based merging for the splits determined on the training set as an overlay to the parameter estimation process. These basically reveal, that optimal retrievability (i.e. lowest Gini coefficient) nicely co-insides with the configuration of training set thresholds that lead to the highest accuracy in the subsequent classification model.

Figure 4 shows the retrievability inequality of different IR systems using Lorenz Curves with a rank cut-off factor $c = 150$ with two different merging strategies in comparison to the default retrieval setting using a single corpus *(with all queries)*. It clearly shows that the retrieval inequality is lower when using the split corpus approach for all retrieval models. Merging the result set from the two partitions based on their relative size consistently leads to the lowest bias.

Table 6. Recall of IR systems with R150 for TREC-CRT prior-art search task on partitioned corpus and without partitioned corpus

Retr. Sys.	Approach	rank cut-off factor			
		R150	**R350**	**R550**	**R750**
TFIDF	Without Split	0.008	0.014	0.021	0.028
	Partition_Size	0.024	0.057	0.080	0.102
	Equal_Size	0.014	0.029	0.048	0.063
BM25	Without Split	0.022	0.042	0.055	0.076
	Partition_Size	0.077	0.138	0.177	0.216
	Equal_Size	0.039	0.084	0.115	0.168
LM	Without Split	0.021	0.042	0.061	0.080
	Partition_Size	0.074	0.116	0.184	0.242
	Equal_Size	0.043	0.083	0.147	0.177

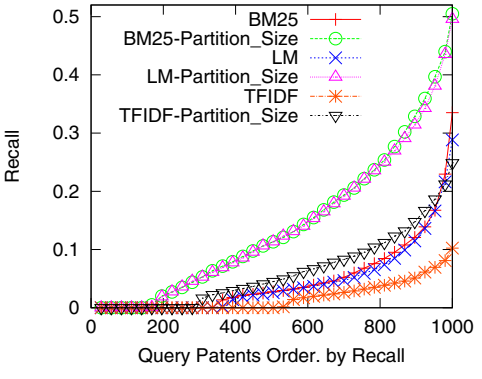


Fig. 5. Recall of IR systems for TREC-CRT prior-art search task on partitioned corpus and without partitioned corpus. Query Patents are ordered by increasing recall.

Table 5 lists the retrievability inequality for a range of other rank cut-off factors on different queries sets. As expected, the Gini coefficient tends to decrease slowly for all query sets and models as the rank cut-off factor increases. The retrievability inequality within the collection is mitigated by the willingness of the user to search deeper down into the ranking. If users examine only the top documents, they will face a greater degree of retrieval bias. However, it is also obvious, that processing queries separately for both partitions greatly reduces the retrievability bias. Again, merging based on relative partition size performs better consistently.

5.2 Recall Evaluation for Prior-Art Search Task

The experiments above only considered retrievability as an abstract measure for evaluating bias in certain retrieval models, as well as a way to reduce that bias

Table 7. Number of prior art documents (Experiment D) classified as high and low retrievable

Retr. Sys.	Low retrievable documents	%	High retrievable documents	%
TFIDF	18659	54%	15541	46%
BM25	14476	42%	19724	58%
LM	14745	43%	19455	57%

by applying it to a partitioned corpus. However, reducing retrievability bias does not automatically imply that this will also help us in improving accuracy within a certain rank cut-off for actual queries. In this section we thus analyze, in how far the approach of using a split document corpus will help increase the accuracy on actual prior-art search task patents.

To this end we select the 1,000 prior art query documents (Query Patents) of the TREC-CRT corpus. These documents have pointers to 34,200 prior art patents that have been identified as relevant prior art documents.

For each of the 1,000 query patents a set of 1- 2- 3- and 4-terms queries is created (exhaustive for the former two, a 20% limit for the later two). These are passed against the entire corpus of 1.2 million documents, using a rank cut-off of 150, 350, 550 and 750. For each document, the union of all result sets of queries created from these document is formed. The set is sorted according to the retrievability score, and the top-c documents are returned as the ranked query result.

Table 7 lists the number of documents classified as high or low retrievable for the various retrieval models evaluated. Roughly half of all target documents are classified into the low-retrievable category. These are highly unlikely to be found using the query generation process employed for the underlying study if posed against a single corpus.

Table 6 lists the resulting accuracy values using the different retrieval models and merge strategies for a range of rank cut-off factors. While results definitely offer room for further improvement by considering document similarity measures during the merging stages, as well as by using more sophisticated query generation algorithms. The results clearly show that overall retrievability is much higher using the retrieval via partitions size based merging, clearly outperforming the default strategy of performing retrieval on the entire corpus.

Overall, the default TFIDF retrieval model shows the worst accuracy. As we have seen in the initial retrievability experiments, it also exhibits the strongest retrievability inequality. This is a strong indicator that, at least for recall oriented applications, a strong retrieval bias (rendering many documents virtually unfindable) has a significant impact on retrieval performance. BM25 and LM perform almost identically as the recall on individual query patents depicted in Figure 5 shows.

The results above are not meant to be compared to accuracy rates of other retrieval engines operating on this corpus. They merely indicate that, by splitting

a corpus into partitions of documents with high and low retrievability, gains in the overall retrieval accuracy can be observed. The exact amount of improvement achievable will depend both on the retrieval model as well as on the query generation process used.

6 Conclusions

This paper has introduced an approach to improve recall in recall oriented application domains by improving retrievability of documents. We first presented a detailed analysis of the characteristics of retrieval inequality in document corpora using a range of configurations for query types and retrieval models. We then introduced the concept of classifying documents automatically into high and low retrievable partitions using a set of simple features capturing term distribution characteristics. The classifier achieved classification accuracies in the range of 85%. By partitioning a corpus into documents that have potentially high and low retrievability, we are now able to perform retrieval separately on these two partitions, merging the result set afterwards. This increases the likelihood that even documents with low inherent retrievability can be found, leading to an overall higher accuracy.

While the results achieved are promising, several questions require more detailed evaluation. These include a more detailed analysis of the behavior of both retrievability as well as the improvement possible with more targeted retrieval system, rather than relying on the generation of exhaustive query sets. Furthermore, retrieval models may be optimized to exhibit minimal bias on the respective partitions, further improving retrievability and thus recall.

References

1. Azzopardi, L., Vinay, V.: Retrievability: an evaluation measure for higher order information access tasks. In: CIKM '08: Proceeding of the 17th ACM Conference on Information and Knowledge Management, pp. 561–570. ACM, New York (2008)
2. Baeza-Yates, R.: Applications of web query mining. In: Losada, D.E., Fernández-Luna, J.M. (eds.) ECIR 2005. LNCS, vol. 3408, pp. 7–22. Springer, Heidelberg (2005)
3. Bashir, S., Rauber, A.: Analyzing document retrievability in patent retrieval settings. In: Bhowmick, S.S., Küng, J., Wagner, R. (eds.) DEXA 2009. LNCS, vol. 5690, pp. 753–760. Springer, Heidelberg (2009)
4. Bashir, S., Rauber, A.: Identification of low/high retrievable patents using content-based features. In: PaIR '09: Proceeding of the 2nd International Workshop on Patent Information Retrieval, pp. 9–16 (2009)
5. Custis, T., Al-Kofahi, K.: A new approach for evaluating query expansion: query-document term mismatch. In: SIGIR '07: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 575–582. ACM, New York (2007)
6. Doi, H., Seki, Y., Aono, M.: A patent retrieval method using a hierarchy of clusters at tut. In: NTCIR '05: In Proceedings of NTCIR-5 Workshop Meeting, Tokyo, Japan (December 6-9, 2005)

7. Fujii, A.: Enhancing patent retrieval by citation analysis. In: SIGIR '07: Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 793–794. ACM, New York (2007)
8. Graf, E., Azzopardi, L.: A methodology for building a patent test collection for prior art search. In: EVIA '08: The Second International Workshop on Evaluating Information Access, Tokyo, Japan, pp. 60–71 (2008)
9. Itoh, H., Mano, H., Ogawa, Y.: Term distillation in patent retrieval. In: Proceedings of the ACL-2003 Workshop on Patent Corpus Processing, pp. 41–45. Association for Computational Linguistics (2003)
10. Jordan, C., Watters, C., Gao, Q.: Using controlled query generation to evaluate blind relevance feedback algorithms. In: JCDL '06: Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries, pp. 286–295. ACM, New York (2006)
11. Lupu, M., Huang, J., Zhu, J., Tait, J.: Trec-chem: large scale chemical information retrieval evaluation at trec. SIGIR Forum 43(2), 63–70 (2009)
12. Page, L., Brin, S., Motwani, R., Winograd, T.: The pagerank citation ranking: Bringing order to the web (1999)
13. Robertson, S., Zaragoza, H., Taylor, M.: Simple bm25 extension to multiple weighted fields. In: CIKM '04: Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management, pp. 42–49. ACM, New York (2004)
14. Robertson, S.E., Walker, S.: 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, pp. 232–241. Springer, New York (1994)
15. Sakai, T.: Comparing metrics across trec and ntcir: the robustness to system bias. In: CIKM '08: Proceeding of the 17th ACM Conference on Information and Knowledge Management, pp. 581–590. ACM, New York (2008)
16. Vaughan, L., Thelwall, M.: Search engine coverage bias: evidence and possible causes. Inf. Process. Manage. 40(4), 693–707 (2004)
17. Witten, I.H., Frank, E.: Data mining: practical machine learning tools and techniques, 2nd edn. Morgan Kaufmann, USA (2005)
18. Xue, X., Croft, W.B.: Transforming patents into prior-art queries. In: SIGIR '09: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 808–809. ACM, New York (2009)
19. Zhai, C.: Risk minimization and language modeling in text retrieval dissertation abstract. SIGIR Forum 36(2), 100–101 (2002)