Improving Retrievability of Patents with Cluster-Based Pseudo-Relevance Feedback Documents Selection

Shariq Bashir
Department of Software Technology and Interactive Systems
Vienna University of Technology, Austria
bashir@ifs.tuwien.ac.at

Andreas Rauber
Department of Software Technology and Interactive Systems
Vienna University of Technology, Austria
rauber@ifs.tuwien.ac.at

ABSTRACT

High findability of documents within a certain cut-off rank is considered an important factor in recall-oriented application domains such as patent or legal document retrieval. Findability is hindered by two aspects, namely the inherent bias favoring some types of documents over others introduced by the retrieval model, and the failure to correctly capture and interpret the context of conventionally rather short queries. In this paper, we analyze the bias impact of different retrieval models and query expansion strategies. We furthermore propose a novel query expansion strategy based on document clustering to identify dominant relevant documents. This helps to overcome limitations of conventional query expansion strategies that suffer strongly from the noise introduced by imperfect initial query results for pseudo-relevance feedback documents selection. Experiments with different collections of patent documents suggest that clustering based document selection for pseudo-relevance feedback is an effective approach for increasing the findability of individual documents and decreasing the bias of a retrieval system.

Categories and Subject Descriptors

H.3.3 [**Information Search and Retrieval**]: Retrieval Models

General Terms

Measurement, Algorithms, Experimentation

1. INTRODUCTION

Conventionally, retrieval systems are evaluated based on `Average Precision`, `Q-measure`, `Normalized Discounted Cumulative Gain`, and other metrics [12]. However, these metrics do not evaluate, what we can find and cannot find globally throughout the whole collection. In spite of low retrievability, systems can still achieve good precision performance [1], which is a perfectly just target in precision-oriented application domains, where the focus is on providing a small set of perfectly matching results. In this paper, we argue that in recall oriented domains like *legal* or *patent* retrieval settings, it is very important to considered retrievability of each and every document, that is, whether each document can potentially be found for a query that it is relevant for.

Document retrievability [1] is a measurement used in Information Retrieval (IR) for determining, in how far all documents in a collection can actually be found given a specific retrieval system. In other words, "retrievability"is used to analyze the bias of retrieval systems. To measure retrievability, a large set of potential queries (e.g. all combinations of all important keywords up to a pre-specified query length) are passed to a retrieval system, and the number of documents that can and that cannot be retrieved is evaluated. The resulting figure provides an estimate on the amount of bias introduced by a certain retrieval system, indicating its suitability for recall-oriented applications.

In this paper, we use Query Expansion (QE) as an approach for increasing the retrievability of individual documents in recall oriented settings. The main motivation behind our work is to analyze the strength of different QE approaches, such as to what extent they can make individual documents more findable. However, first experiments revealed, that state of art QE approaches showed worse retrievability performance. This is because in QE it is assumed, that the set of top-ranked documents used for Pseudo-Relevance Feedback (PRF) is relevant, and learning from these Pseudo-Relevance documents can improve the ranking [10]. However, due to system bias, top-ranked documents may contain noise [10], which can ultimately result in the query representation "*drifting*" away from the original query.

Targeting this limitation and for improving the retrievability of individual documents, in this paper, we propose an improved clustering based resampling method for PRF selection. The main idea is to use document clustering based on local frequent terms to find dominant documents for PRF.

2. RETRIEVABILITY MEASUREMENT

"*Retrievability*" measures, how likely each and every document $d \in D$ can be retrieved within the top c ranked results for all queries in Q. More formally, retrievability $r(d)$ of $d \in D$ can be defined as follows.

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

CIKM'09, November 2–6, 2009, Hong Kong, China.
Copyright 2009 ACM 978-1-60558-512-3/09/11 ...$10.00.

Here, $f(k_{dq}, c)$ 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 be further 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 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.

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

where $n = |D|$ is the number 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 retrievability bias imposed by the underlying retrieval system on the given document collection.

3. PSEUDO-RELEVANCE FEEDBACK SELECTION WITH CLUSTERING

Existing Approach: Lee et al. [10] select documents for PRF based on cluster analysis. Under their assumption, a document is considered relevant for PRF, if it can cluster lot of high similarity documents. A document which has either no nearest neighbor or some neighbors with low similarity is considered irrelevant for PRF. The main limitation of their approach is that, it is not useful for those application domains, in where large variation exists between documents length and large diversity exists in a domain's vocabulary for expressing information, like patent documents. For example, if longer length documents contain many vague or general terms [3], then they can cluster a large number of irrelevant documents with good similarity to the centroid, but with poor intra-cluster similarity. On the other hand, if the documents are short and contain many technical or synonym terms, then the clusters may be small, and more importantly it can become very difficult to cluster related documents into a single cluster.

For overcoming the limitations with [10] approach, in this paper, we describe an improved approach for clustering, where clusters are rejected or accepted for PRF on the basis of their intra-cluster similarity, rather than only on the basis of their size. Our hypothesis is that a cluster should be better in terms of both factors; its size and intra-cluster similarity. Low intra-cluster similarity of large size clusters indicates that they clustered other documents due to their noisy terms. The main features of our clustering process are that, it checks intra-cluster similarity of clusters on the basis of `local frequent terms`, which is more efficient. Smaller size related clusters are merged efficiently using `local frequent terms` of clusters and merged cluster feature vectors. After ranking clusters for PRF, top-ranked documents in high rank clusters for relevance feedback are selected based on their similarity with the centroid `local frequent terms` of the cluster.

(a) Constructing Clusters: After constructing initial clusters using k-nearest neighbors with cosine similarity, we use intra-cluster similarity scores of clusters for selecting relevant clusters for PRF. For fast computation, we check intra-cluster similarity of individual clusters on the basis of their number of `local frequent terms`. If a cluster contains at least `(min_local_terms)` number of `local frequent terms` with minimum support `(min_term_supp)`, then it is considered relevant for PRF. For subsequent analysis we use these `min_local_terms` of clusters as centroids.

(b) Merging Similar Clusters: As each document basically creates its own cluster of similar documents, most documents will end up in numerous clusters. This overlap is used to merge clusters into larger consistent groups. Two clusters which contain a high number of overlapping documents are merged [5]. Intuitively, a cluster C_a is good to merge with another cluster C_b, if there are many terms in C_a that are also locally frequent in C_b. For calculating final inter-similarity of C_a to C_b, we use the scoring function of Equation 3. The score of Equation 3 is normalized using Equation 4 to remove the effect of varying document size.

$$Score(C_i, doc(C_j)) = [\sum_{x \in X} n(x) \cdot s(x)] - [\sum_{x' \in X'} n(x') \cdot s(x')] \qquad (3)$$

$$Sim(C_i, C_j) = \frac{Score(C_i, doc(C_j))}{\sum_{x \in X} n(x) + \sum_{x' \in X'} n(x')} + 1 \qquad (4)$$

Where C_i and C_j are two clusters; $doc(C_j)$ stands for combining all the documents of cluster C_j into a single document. $x \in X$ represents a term in $doc(C_j)$ that is local frequent in C_i; $x' \in X'$ represents a local frequent term of $doc(C_j)$ that is not locally frequent in C_i; $n(x)$ and $n(x')$ are the weighted frequencies of x and x' in the feature vector of $doc(C_j)$. $s(x)$ represents term x support in cluster C_i, and $s(x')$ represents term x' in cluster C_j.

The similarity function in Equation 4 only calculates the similarity from one side $Sim(C_a, C_b)$. However, for removing the effect of noisy terms, it is necessary that both values $Sim(C_a, C_b)$ and $Sim(C_b, C_a)$ should be high Equation 5. We thus use the geometric mean of the two normalized scores. After merging two similar clusters the `local frequent terms` sets of both clusters are also merged.

$$Inter\_Sim(C_a \leftrightarrow C_b) = [Sim(C_a, C_b) \cdot Sim(C_b, C_a)]^{\frac{1}{2}} \qquad (5)$$

(c) Ranking Clusters: In [10] a cluster-based query likehood language model is used for ranking clusters for relevance feedback. After ranking clusters a set of top documents from high rank clusters are used as an input for relevance feedback [9]. We use the same method for ranking clusters. However, for selecting the set of top documents from high rank clusters for relevance feedback, we further rank individual documents in the selected clusters based on their similarity with the centroid `local frequent terms` of the cluster.

After calculating individual similarity scores of all documents with their centroid frequent terms, documents from high-rank clusters are order by decreasing order of their scores, and placed into single set S. Finally, the top K documents from set S are selected for relevance feedback, and terms for expansion are selected using Language Modeling approach [9].

$$\sum_{d \in K} P(d)P(w|d)P(q|d) \tag{6}$$

Where K is the set of documents that are relevant to the query q. We assume that $P(d)$ is uniform over the set. After this estimation, the most e terms (words) from $P(w|K)$ are chosen as for expanding queries. The values $P(w|d)$ and $P(q|d)$ can be calculated following Equations 7 and 8.

$$P(q|d) = \prod_{i=1}^{m} P(\gamma_i|d) \tag{7}$$

where γ_i is the i^{th} query term, m is the number of terms in a query q, and d is a document model.

Dirichlet smoothing is used to estimate non-zero values for terms in the query which are not in a document. It is applied to the query likelihood language model as follows.

$$P(w|d) = \frac{|d|}{|d| + \lambda} P_{ML}(w|d) + \frac{\lambda}{|d| + \lambda} P_{ML}(w|D) \tag{8}$$

$$P_{ML}(w|d) = \frac{freq(w,d)}{|d|}, P_{ML}(w|D) = \frac{freq(w,D)}{|D|} \tag{9}$$

where $P_{ML}(w|d)$ is the maximum likehood estimate of a term w in document d, D is the entire collection, and λ is the smoothing parameter. $|d|$ and $|D|$ are the lengths of a document d and collection D, respectively, $freq(w,d)$ and $freq(w,D)$ denote the frequency of a term w in d and D, respectively.

4. EXPERIMENTAL SETUP

For our experiments, we use a collection of freely available patents from the US patent and trademark office, downloaded from (`http://www.uspto.gov/`). We collected all patents that are listed under `United State Patent Classification` (USPC) classes *422* and *423* . There are a total of 54, 353 documents in our collection, with an average document size of 3, 317.41 words (without stop words removing). Seven state-of-the art IR models and QE methods along with our proposed clustering technique are used for evaluating the retrievability inequality. These are TFIDF, OKAPI retrieval function (BM25), Exact match model, Language Modeling with term smoothing (LM) [13], Kullback-Leibler divergence (KLD) [2], Term Selection value (QE-TS) [11], PRF document selection using clustering (Lee et al.) [10], and our clustering approach using local frequent terms (Cluster-LFT). For all QE models, we select the `top 35` documents for PRF and `50 terms` for query expansion. In our clustering approach, for accepting or rejecting clusters under intra-cluster similarity hypothesis, we set the values of parameters `min_local_terms` and `min_term_supp` as `5` and `60%`, respectively.

Controlled Query Generation: For generating reproducible and theoretically consistent queries for retrievability measurement, we use the method of controlled query generation (CQG) [7]. We use two different variations, based on how patent experts generate queries for searching their relevant information in patent corpus.

Query Generation combining Frequent Terms ***(QG-FT)*:** In this CQG approach, we try to reflect the way how patent examiners generate queries sets in *patent invalidity search* problems [8]. In this search process, the examiners extract relevant query terms from a new patent application, particularly from the *Claim* sections for creating query sets [6]. QG-FT first extract all those frequent terms that are present in the Claim sections of each patent document and have a support greater than a certain threshold. Then, QG-FT combines the single frequent terms of each individual patent document into `two`, `three` and `four` terms combinations for constructing longer length queries.

Query Size	CQG Appr.	#Queries	ARS
2 terms	*QG-FT*	548,390	335.9
	QG-DR	436,273	549.6
3 terms	*QG-FT*	753,682	303.5
	QG-DR	590,820	480.3
4 terms	*QG-FT*	855,215	225.6
	QG-DR	587,782	360.7
Total Queries	*QG-FT*	2,157,287	
	QG-DR	1,614,875	

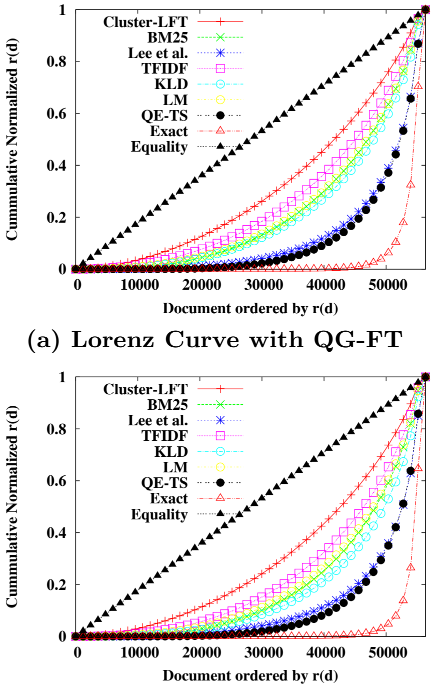
Table 1: Queries set sizes and average retrievability scores (ARS)

In QG-FT, we consider all the frequent single terms, which have a *minimum support*≥3 in the `Claim section`. For generating larger length queries for every document, we expand the single frequent terms into `two`, `three` and `four` term combinations. For documents which contain large number of single frequent terms, the different co-occurring term combinations of size `two`, `three` and `four` can become very large. Therefore, for generating similar number of queries for every document, we put an upper bound of 90 queries generated for every patent document.

Query Generation with Document Relatedness ***(QG-DR)*:** Patent applications may contain many vague or technical terms in order to hide the relation to other documents from patent examiners [3]. In such situations patent examiners extract relevant terms from other patent documents that are similar to the new patent application using the concept of document relatedness [4]. With QG-DR, we adopt this strategy. We first define a set of related documents for each document in the corpus based on *k*-nearest neighbor with cosine similarity. QG-DR then generates a set of queries based on each of these sets of related documents. Using the relative entropy of individual terms in language modeling [7], the most discriminating terms are identified for constructing `two`, `three` and `four` terms combinations queries.

In QG-DR mechanism, we construct the related document set for every document in the collection considering `35` neighbors. After applying language modeling on related documents sets, we select the top `70 terms` that contribute most to the relative entropy with the language model. `Two`, `three` and `four` term queries are constructed with the same approach and maximum number of queries per document boundary as above. Table 1 shows the distribution of different queries sets.

Discussion of Results: We show the retrievability inequality of different retrieval systems using Lorenz Curves with a `rank cut-off` factor of 30 (Figures 1). Tables 2 shows the retrievability inequality with other `rank cut-off` factors using Gini coefficient. The `exact match` method, which is widely used in professional patent retrieval system, consistently shows the worst performance. The clustering



(a) Lorenz Curve with QG-FT

(b) Lorenz Curve with QG-DR

Figure 1: Lorenz Curves for retrievability scores with *rank cut-off* c=30. *Equality* refers to a optimal system which has no bias.

approach of `Lee et al.` and `term selection` also do not perform too well with respect to providing potential access to all documents. The standard `TFIDF` approach performs surprisingly well, only surpassed by our clustering approach based on frequent terms with the enhanced cluster selection strategy. This indicates that our approach makes individual documents more findable than other systems, due to its improved selecting procedure for pseudo relevance feedback.

Table 2 depicts the *Gini coefficient* for various *rank cut-off* (c) parameter configurations. We can see that, as `c` increases, the Gini coefficient tends to slowly decrease on all different queries sets with both collections. This indicates that the retrievability inequality within the collection is mitigated by the willingness of the user to search deeper down into the ranking, as expected. If users are willing to examine only the top documents, then they will face a greater degree of retrieval bias.

5. CONCLUSIONS

In this paper, we evaluated the effectiveness of different retrieval systems using retrievability measurement with a focus on recall oriented application domains. Our results yield that state of the art QE methods provide a large inequality in the retrievability of documents, as compared to those systems which do not expand queries. This is due to their ineffective assumption that top rank documents for PRF are always relevant and learning from these relevance feedback documents for expanding queries can increase the effectiveness of retrieval systems. For overcoming this limitation, in this paper we utilize clustering approach for better document selection for PRF. On our extensive retrievability experiments with using large collection of queries, our clustering approach for PRF, provides less bias than all other systems on different rank factors.

Retrieval System	CQG Appr.	Rank cut-off factors				
		30	40	50	70	90
BM25	*QG-FT*	0.58	0.53	0.50	0.48	0.48
	QG-DR	0.62	0.57	0.53	0.50	0.49
TFIDF	*QG-FT*	0.50	0.43	0.39	0.41	0.43
	QG-DR	0.55	0.47	0.43	0.43	0.44
Exact	*QG-FT*	0.94	0.90	0.82	0.75	0.68
	QG-DR	0.95	0.92	0.86	0.80	0.74
LM	*QG-FT*	0.57	0.49	0.44	0.45	0.46
	QG-DR	0.61	0.53	0.47	0.46	0.46
KLD	*QG-FT*	0.60	0.54	0.49	0.48	0.48
	QG-DR	0.67	0.60	0.54	0.51	0.50
QE-TS	*QG-FT*	0.79	0.74	0.68	0.64	0.61
	QG-DR	0.81	0.75	0.69	0.65	0.62
Lee et al.	*QG-FT*	0.77	0.70	0.63	0.59	0.56
	QG-DR	0.78	0.72	0.65	0.61	0.58
Cluster-LFT	*QG-FT*	**0.39**	**0.31**	**0.33**	**0.37**	**0.40**
	QG-DR	**0.42**	**0.34**	**0.35**	**0.37**	**0.39**

Table 2: Gini coefficient values with different retrieval models with different *rank cut-off factors (c)*

6. REFERENCES

[1] L. Azzopardi, V. Vinay. Retrievability: an evaluation measure for higher order information access tasks. *In Proc. of CIKM '08*, 2008, USA.

[2] W. B. Croft. Advances in information retrieval. *Academic Publishers*, Norwell, MA, 2000.

[3] C. J. Fall, A. Torcsvari, K. Benzineb, G. Karetka. Automated categorization in the international patent classification. *ACM SIGIR Forum*, Volume 37, Issue 1 (Spring 2003), Pages 10–25.

[4] A. Fujii, M. Iwayama, N, Kando. Introduction to the special issue on patent processing. *Information Processing and Management*, Volume 43, Issue 5, 2007.

[5] B. C. M. Fung, K. Wang, M. Ester. Hierarchical document clustering using frequent itemsets. *In Proc. of SDM' 03*, 2003, USA.

[6] H. Itoh, H. Mano, Y, Ogawa. Term distillation in patent retrieval. *In Proc. of ACL '03*, 2003, Japan.

[7] C. Jordan, C. Wattters, Q. Gao. Using controlled query generation to evaluate blind relevance feedback algorithms. *In Proc. of JCDL '06*, 2006, USA.

[8] K. Konishi. Query terms extraction from patent document for invalidity search. *In Proc. of NTCIR '05*, 2005, Japan.

[9] V. Lavrenko, W. B. Croft. Relevance based language models. *In Proc. of SIGIR '01*, 2001, USA.

[10] K. S. Lee, W. B. Croft, J. Allan. A cluster-based resampling method for pseudo-relevance feedback. *In Proc. of SIGIR '08*, 2008, Singapore.

[11] S. E. Robertson, S. Walker. Okapi/Keenbow at TREC-8. *In Proc. TREC-8*, 1999, USA.

[12] T. Sakai. Comparing metrics across TREC and NTCIR: the robustness to system bias. *In Proc. of CIKM '08*, 2008, USA.

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