computers

Article

An Improved Retrievability-Based Cluster-Resampling Approach for Pseudo Relevance Feedback

Shariq Bashir

Information Management Department, College of Computer and Information Sciences,
Imam Muhammad Ibn Saud University, Riyadh 11564, Saudi Arabia; sbbashir@imamu.edu.sa;
Tel.: +966-9-9992-5011

Academic Editor: Kartik Gopalan
Received: 31 July 2016; Accepted: 10 November 2016; Published: 15 November 2016

Abstract: Cluster-based pseudo-relevance feedback (PRF) is an effective approach for searching relevant documents for relevance feedback. Standard approach constructs clusters for PRF only on the basis of high similarity between retrieved documents. The standard approach works quite well if the retrieval bias of the retrieval model does not create any effect on the retrievability of documents. In our experiments we observed when a collection contains retrieval bias, then high retrievable documents of clusters are frequently retrieved at top positions for most of the queries, and these drift the relevance feedback away from relevant documents. For reducing (retrieval bias) noise, we enhance the standard cluster construction approach by constructing clusters on the basis of high similarity and retrievability. We call this retrievability and cluster-based PRF. This enhanced approach keeps only those documents in the clusters that are not frequently retrieve due to retrieval bias. Although this approach improves the effectiveness, however, it penalizes high retrievable documents even if these documents are most relevant to the clusters. To handle this problem, in a second approach, we extend the basic retrievability concept by mining frequent neighbors of the clusters. The frequent neighbors approach keeps only those documents in the clusters that are frequently retrieved with other neighbors of clusters and infrequently retrieved with those documents that are not part of the clusters. Experimental results show that two proposed extensions are helpful for identifying relevant documents for relevance feedback and increasing the effectiveness of queries.

Keywords: document clustering; machine learning; information retrieval; pseudo-relevance feedback; query expansion; retrieval bias; retrievability measure

1. Introduction

Pseudo-relevance feedback (PRF)-based query expansion is an effective approach for increasing the effectiveness of queries. Most pseudo-relevance feedback methods assume that a set of top-retrieved documents is relevant and then learn from the pseudo-relevant documents to expand terms to increase the effectiveness of queries [1–5]. However, if the top retrieved documents are noisy (irrelevant to the search query), then this noise decreases the effectiveness. Recently, a deterministic sampling method based on overlapping clusters was proposed (cluster-based PRF) to select better documents for PRF [6,7]. By permitting overlapped clusters for the top-retrieved documents and repeatedly using the dominant documents that appear in multiple highly-ranked clusters, an expansion query can be represented to emphasize the core topics of a query.

Although it is well observed that cluster-based PRF improves effectiveness, however, further improvement is possible by removing noisy documents from the clusters. Related approaches construct clusters for PRF either using a query-dependent approach (after processing the initial query) or using a query independent approach (offline) [6–8]. Kalmanovich et al. [6] observed from their experiments that the query independent approach provides higher effectiveness than the query-dependent approach. This is because during cluster construction, the query independent approach utilizes the full context of a collection rather than the local context of a query [6]. Furthermore, the query independent approach is preferable over the query-dependent approach in terms of efficiency. In the query independent setting, the clusters are constructed offline and only once during the indexing of documents, and these offline clusters are used during query processing for selecting pseudo relevance feedback documents (PRF); whereas in the query-dependent setting, the clusters are needed to construct for every query, and this requires a long processing time and resources for selecting PRF documents.

The standard query independent approach constructs clusters for each document of a collection by assuming each document as the centroid of its cluster. Next, documents of clusters are retrieved through the *k*-nearest neighbor approach (kNN); the top *k* nearest neighbors of each centroid form the cluster, and this cluster is used for relevance feedback. kNN searches nearest neighbors through (well descriptive) long queries either using full text or the top most frequent terms of centroid document. Then, after constructing clusters, it is assumed that when the relevant documents of a query would be searched through the initial query, then a subset of their nearest neighbors that exist in the query's retrieved set would provide aid to them as a relevance feedback to retrieve these at top positions. This assumption works quite well if the retrieval bias of retrieval models does not create any effect on the retrievability of documents of clusters [9]. However, if the documents of the collection have large retrievability inequality between them (due to retrieval bias), then high retrievable documents of clusters have a large probability of retrievability at top positions for most of the queries. These biased documents drift the PRF selection towards noise [10]. This is because the documents that contain high retrievability have a large probability of retrievability at top positions for most of the queries compared to the documents that have low retrievability [9,11]. Although high retrievable documents of clusters are also a relevant part of the clusters, if retrieved frequently only due to retrieval bias, then these decrease the effectiveness of cluster-based relevance feedback [12].

Motivation and Contribution

The objective of this paper is to understand the effect of retrieval bias on PRF clusters and then improving the effectiveness of PRF by constructing clusters on the basis of high similarity and retrievability. We study this objective with the help of the following set of experiments. In the first experiment, we construct clusters using the standard *k*-nearest neighbor approach and examine which class of documents (if we partition the collection into several classes on the basis of low/high retrievability) contributes most to positive relevance feedback. In these experiments, we want to analyze whether the retrieval bias of the retrieval model affects PRF effectiveness. In the next set of experiments, we enhance the standard cluster construction approach proposed in [8] using retrievability and construct clusters by removing all of those documents from the clusters that add noise in the relevance feedback due to retrieval bias. In another extension, we enhance the basic retrievability mechanism by mining frequent neighbors of clusters. We then construct clusters by keeping only those documents in the clusters that not only have high similarity with the centroid of the clusters, but are also frequently retrieved with other documents of the cluster. We evaluate the effectiveness of all proposed approaches on the Text Retrieval Conference (TREC) chemical patent retrieval task collection (TREC-CRT) [13]. From the results, we observed that during constructing clusters, if we use retrieval bias and the retrievability of documents along with the similarity of documents to centroids, then this improves the overall effectiveness of cluster-based PRF. We compare the effectiveness of our approach with the cluster-based PRF approach presented in [8]. Our results show improvement over [8].

The remainder of this paper is structured as follows. Section 2 reviews related work on machine learning approaches that are used for the selection of PRF. Section 3 describes the TREC-CRT (TREC chemical patent retrieval task) benchmark collection, retrieval models and effectiveness measures that we used for experiments. Section 4 discusses the cluster-based PRF approach. In Section 5, we discuss the retrievability and cluster-based PRF. We explain this section by first describing the retrievability measure, and then, we show how retrieval bias effects the cluster-based PRF; finally, we proposed two extensions for cluster-based PRF using retrievability in order to improve the effectiveness. Section 6 briefly summarizes key lessons learned from this study.

2. Related Work

In information retrieval (IR), it is well studied that query expansion (QE) improves retrieval effectiveness. Research on query expansion using PRF falls into the following three classes. In the first class, different probabilistic approaches are investigated for selecting dominant (relevant) terms for expansion. These include Kullback–Leibler divergence [14], term selection by the Robertson and Walker method [15] and different variants of language modeling [3,16,17]. In the second class, different strategies are proposed for classifying the dominant expansion terms on the basis of data mining and machine learning techniques [18,19]. However, in both classes, it is assumed that the set of top-n documents is useful for relevance feedback. In the third class, different machine learning techniques are used for identifying relevant documents for PRF [6–8,20].

Huang et al. [19] observed that the effectiveness of PRF strongly relies on the quality of selected expansion terms from the top ranked documents. In their study, they used a number of machine learning classification techniques (naive Bayesian, decision tree, support vector machine) in the form of co-training for selecting the good terms for the expansion the from top-n documents. Cao et al. [18] also used a classifier to select the best terms from the top-n documents after analyzing the effects of each term on the effectiveness of training queries.

Collins-Thompson et al. [21] used bootstrap sampling on top-retrieved documents for identifying variants of the query by leaving a single term out. They assumed that if a set of query terms is noisy, then these decrease the robustness and effectiveness of queries. Sakai et al. [16] used clustering for skipping noisy relevance feedback documents that are retrieved at top positions.

Cluster-Based Pseudo-Relevance Feedback

There has also been work on term expansion using clustering in the vector space model [22–25]. At TREC 6 [23] used document clustering on the System for Manipulating and Retrieving Text (SMART), though the results of using clusters did not show improvements over the baseline feedback method. Huang et al. [20] found that the effectiveness of queries is highly sensitive to the selection of feedback documents. In their study, they proposed a number of techniques for selecting query-specific feedback documents by applying query the clarity score, discount cumulative gain and mixture models.

Lee et al. [8] used document clustering, which helps in selecting documents for PRF on the basis of their cluster size and similarity to other documents of the query. Under their assumption, a document is considered relevant for the PRF if it contains high similarity with other documents of the query and is irrelevant if it has either no nearest neighbor or some neighbors with low similarity. Their technique generates clusters for PRF after processing the initial query and for only the N number of top retrieved documents. This approach also forms the basis of the approaches presented in this paper. Although, this approach works quite well if the retrieval bias of the retrieval model does not create any effect on retrievability of documents [10], in our experiments, we observed when a collection contains retrieval bias, then high retrievable documents of clusters are frequently retrieved at top positions for most of the queries, and these drift the relevance feedback away from relevant documents. For reducing (retrieval bias) noise, we enhance this standard cluster construction approach by constructing clusters on the basis of high similarity and retrievability. We call this retrievability and cluster-based PRF. This enhanced approach keeps only those documents in the clusters that are not frequently retrieved due to retrieval bias.

In [26], Bashir et al. evaluated the effectiveness of different retrieval systems using retrievability measures with a focus on recall-oriented application domains. Their results indicate that state of the art query expansion methods provide a large inequality in the retrievability of documents as compared to those systems that 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. To overcome the limitations of the standard clustering approach in their paper, they proposed an improved approach. Their approach selects clusters for PRF on the basis of their intra-cluster similarity rather than only on the basis of clusters size. This is helpful for pruning irrelevant clusters when long documents cluster a large number of documents on the basis of their noisy terms. Moreover, they used local frequent terms of clusters for fast inter-cluster similarity checking. Smaller-sized related clusters are also merged using a fast method on the basis of local frequent terms. Finally, top documents in high rank clusters are selected for relevance feedback by ranking their similarity with centroid frequent terms of clusters. Although [26] used cluster-based PRF for query expansion, however, we could not compare the system of [26] with our proposed approach. This is because in [26], the authors used local frequent terms for merging small clusters and pruning irrelevant clusters that have low inter-cluster similarity, and these features provide additional benefit for increasing the effectiveness.

3. Experimental Setup

3.1. Collection

We select the prior-art (PA) task of the TREC-CRT collection for analyzing the effectiveness of retrieval models [13]. The total size of the collection is 1.2 million documents. Figure 1 shows the document length statistics of the collection. The PA task consisted of 1000 topic queries that are the full-text patent documents (i.e., consisting of at least claims and abstract or description) taken from both the European Patent Office (EPO) and the U.S. Patent Office (USPTO). The goal of searching a patent database for the prior-art search task is to find all previously-published related patents on a given topic [13,27,28]. It is a common task for patent examiners and attorneys to decide whether a new patent application is novel or contains technical conflicts with some already patented invention. They collect all related patents and report them in a search report. We use these reports as relevance judgments. Next, we apply a standard approach for query generation in the patent retrieval domain. From each topic, we select only the claim section, because it is regarded as being the most representative piece of text, characterizing the scope of invention well due to the rules of the patent system worldwide, as done also in [27–30].

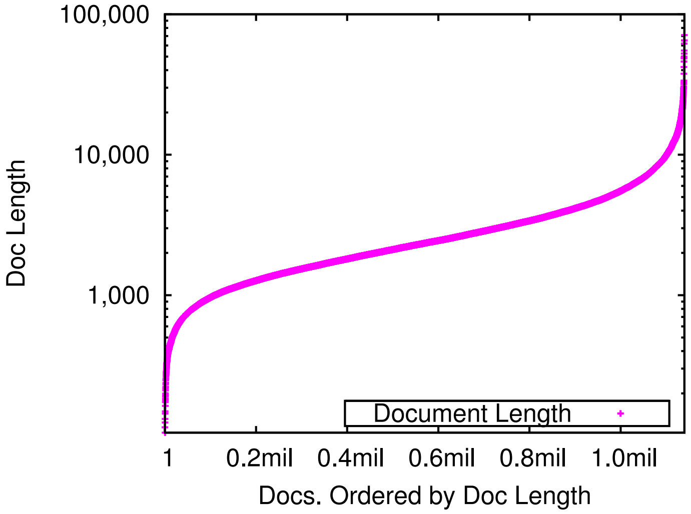

Figure 1. The document length statistics of the TREC chemical patent retrieval task collection.

3.2. Retrieval Models

Eight standard retrieval models and three query expansion methods along with cluster-based PRF (resampling [8]) are used for experiments. These are as follows.

3.2.1. Standard Retrieval Models

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

$$TFIDF(d,q) = \sum_{t \in q} tf_{t,d} * log\frac{|D|}{df_t} \tag{1}$$

  $tf_{t,d}$ 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 to and biased toward large absolute term frequencies. It is possible to address the length bias by using document length $|d|$ and defied normalized TFIDF (NormTFIDF) as:

$$NormTFIDF(d,q) = \sum_{t \in q} \frac{tf_{t,d}}{|d|} * log\frac{|D|}{df_t} \tag{2}$$

- BM25: Okapi (Best Match Retrieval Model BM25) is arguably one of the most important and widely-used information retrieval models. 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 [4]:

$$BM25(d,q) = \sum_{t \in q} log\frac{|D| - df_t + 0.5}{df_t + 0.5} \frac{tf_{t,d}(k+1)}{tf_{t,d} + k(1 - b + b\frac{|d|}{|\overline{d}|})} \tag{3}$$

  $|\overline{d}|$ 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 [31] at AT&T Labs.

$$SMART(d,q) = \sum_{t \in q} (W_d * W_q) \tag{4}$$

$$W_d = \frac{1 + log(tf_{t,d})}{1 + log(avtf)} * \frac{1}{0.8 + 0.2\frac{utf}{pivot}} \tag{5}$$

$$W_q = (1 + log(tf_{t,d})) * log\frac{|D| + 1}{df_t} \tag{6}$$

  $avtf$ 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.

3.2.2. Language Models with Term Smoothing

The language model tries to estimate the relevance of the 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:

$$P(q|M_d) = \prod_{t \in q} P(t|M_d) \tag{7}$$

$$P(t|M_d) = \frac{tf_{t,d}}{|d|} \tag{8}$$

In Equation (7), 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 a 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 term smoothing in our experiments.

- Jelinek–Mercer smoothing (JM): 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. Small smoothing values of λ close to 0 add only the contribution of term frequencies, while large λ values reduce the effect of relative term frequencies within the documents, and more importance is given toward the relative frequencies of terms in the collection.

$$P(t|M_d) = (1-\lambda)\frac{tf_{t,d}}{|d|} + \lambda P(t|D) \tag{9}$$

  $P(t|D)$ 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 [17]. Since long documents allow us to estimate the language model more accurately, therefore, this technique smooths them less, and this is done with the help of a parameter μ. Since the value of μ is added in the document length, thus small values of μ retrieve less long documents. If the μ is used with large values, then the distinction for the difference between document lengths becomes less extreme, and long documents are more favored over short documents. Again, this favoritism mostly occurs in the case of long Boolean OR queries.

$$P(t|M_d) = \frac{tf_{t,d} + \mu P(t|D)}{|d| + \mu} \tag{10}$$

  According to [17] suggestion, we use the μ with 2000.
- Two-stage smoothing (TwoStage): This smoothing technique first smooths 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 [17]. The query background model is based on the term frequency in the collection. The smoothing function is therefore:

$$P(t|M_d) = (1-\lambda)\frac{tf_{t,d} + \mu P(t|D)}{|d| + \mu} + \lambda P(t|D) \tag{11}$$

  where μ is the Dirichlet prior probability and λ is the Jelinek–Mercer parameter. In our experiments, we use the parameters $\mu = 2000$ and $\lambda = 0.7$, respectively.

- Absolute discount smoothing (AbsDis): This technique makes smoothing by subtracting a constant $\delta \in [0,1]$ from the counts of each seen term [17]. 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 $(1-\lambda)$.

$$P(t|M_d) = \frac{max(tf_{t,d}-\delta,0)}{|d|} + \frac{\delta|T_d|}{|d|}P(t|D) \tag{12}$$

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

3.2.3. Query Expansion Models

1. Query expansion using language modeling (TS-LM): [32]: This method uses the top-n documents for PRF selection. The candidate terms for the expansion in the PRF are ranked according to the sum of divergences between the documents in which they occurred and the importance of the terms in the whole collection (Equation (14)),
2. Query expansion using Kullback–Leibler divergence (TS-KLD): [14]: This method also uses the top-n for PRF selection. However, terms for the expansion in the PRF set are ranked according to the relative rareness of terms in the PRF set as opposed to the whole collection (Equation (13)),

$$KLD_t = \frac{P(t|P)}{|P|} * log(\frac{P(t|P)}{|P|} * \frac{C+0.01*|V|}{cf_t+0.01}) \tag{13}$$

t is the expansion term; P is the PRF set; cf_t represents the total count of term t in the collection; C represents the total count of all terms in the collection; and $P(t|P) = (\sum_{p\in P} tf_{t,p} / \sum_{p\in P}|p|)$ is the probability of term t occurrence in the PRF set P.
3. PRF selection using clustering (resampling): [8]: This technique selects PRF using document clustering (see Section 4 for more details). Resampling selects candidate terms for the expansion using TS-LM as described above.

For effectiveness analysis, we used BM25 for providing initial ordering of documents for the PRF selection. For all query expansion strategies, the top 10 documents are used for the PRF and the top 30 terms are used for query expansion.

4. Cluster-Based Pseudo-Relevance Feedback

In this section, we present the effectiveness results of retrieval models that we obtained after processing queries. We start this section by first describing the cluster-based PRF technique, and then, we compare the effectiveness of cluster-based PRF with standard query expansion approaches.

4.1. Constructing Clusters for PRF

We construct clusters using the k-nearest neighbors approach [7,8,33]. In this approach, each document of the collection plays a central role for forming its own cluster. Each document is assumed as a centroid of its cluster and k nearest neighbors are retrieved from the collection on the basis of their similarity with the centroid document. To check the similarity between two documents, we represent vectors of documents using the BM25 weighting scheme, and we used the top 30 most frequent terms of each document for calculating similarity. We kept the parameter k constant for all clusters; thus, all documents of different lengths have similar cluster sizes. The parameter k also helps in controlling the number of topics and their quality in the clusters. Ideally, the value of k should be not too small or too large. A large value for k increases the total number of available clusters for PRF, but decreases the effectiveness of PRF due to retrieving neighbors with low similarity. A very small value for k retrieves a small number of clusters for PRF and, thus, decreases the effectiveness of PRF. Next, during query processing, clusters are generated for only the top-retrieved 500 documents of the query to find dominant documents for PRF.

To select top documents of a query for relevance feedback, the formed clusters are ranked according to their cluster sizes. This approach can be easily understand from the following example.

Example 1. *Suppose we have a collection of 1.2 million documents, and we want to construct clusters for these 1.2 million documents using the k nearest neighbor approach. With this approach, each document of the collection plays a central role for forming its own cluster. Each document is assumed as a centroid of its cluster, and k nearest neighbors are retrieved from the collection on the basis of their similarity with the centroid document. Suppose the value of k is 2000, and thus, each document in the collection has an independent cluster of a size of 2000 documents. Now, suppose we want to process a query "Networking Cables", and we need pseudo feedback documents for query expansion. Suppose query "Networking Cables" retrieves only 80 documents from 1.2 million documents. The retrieved documents have ids from* $\{d_1, d_2 ... d_{80}\}$*. Now, suppose document* (d_1) *has its 20 cluster documents (out of 2000) in the 80 retrieved documents, and document* (d_2) *has its 10 cluster documents (out of 2000) in the 80 retrieved documents. Thus, for pseudo relevance feedback,* d_1 *has a larger cluster size in the retrieved set than* d_2.

4.2. Ranking Clusters and Selecting Documents for Relevance Feedback

To select top documents of a query for relevance feedback, the formed clusters are ranked according to their cluster sizes.

Once the clusters are ranked, the top S clusters (representing documents) are selected for the PRF. Next, terms for the expansion are ranked according to the sum of divergences between the PRF documents where they occurred and the importance of the terms in the whole collection [32] (Equation (14)). This penalizes those terms of PRF documents that occur more frequently in the collection.

$$score_t = \sum_{s \in S}((\lambda \cdot \frac{tf_{t,s)}}{|s|})) + (1-\lambda)\frac{cf_t}{C})log(\lambda \frac{tf_{t,s}}{|s|}\frac{cf_t}{C} + 1 - \lambda)) \tag{14}$$

where t is the expansion term, S is the PRF set, λ represents the smoothing parameter, cf_t represents the total count of term t in the collection, C is the total count of all terms in the collection and $tf_{t,s}$ is the term frequency of t in document s.

After selecting top terms for the expansion, the relevance scores of documents using expansion terms are again calculated using BM25. Figure 2 shows the architecture of cluster-based PRF. In the architecture, clusters for PRF are constructed using the query-independent approach, and these clusters are used to select relevant documents for PRF.

4.3. Parameter Setting for Constructing Clusters

The clustering process explained above controls the size of clusters with the help of the k parameter. Figure 3 shows the sensitivity of the ranges of k over effectiveness. In order to select the best values of k, we tune the values of the k over different ranges and examine its sensitivity with $P@30$. The k is varied within a range of 10, 50 and then from 100 to 3000 with each step with an increase of 100. The results reveal that the large values of k (between 1000 and 3000) increase the total number of topics in the clusters, but decrease the quality of the clusters since a large number of noisy neighbors with low similarity are also appearing in the clusters. This increases the probability of the appearance of noisy clusters in the query processing and decreases the effectiveness. The very small values of k are also not suitable. Small values of k decrease the probability of PRF documents' selection via clustering, and thus, most of the PRF documents are selected through standard query relevance scores. In our clustering process, we use k with 300.

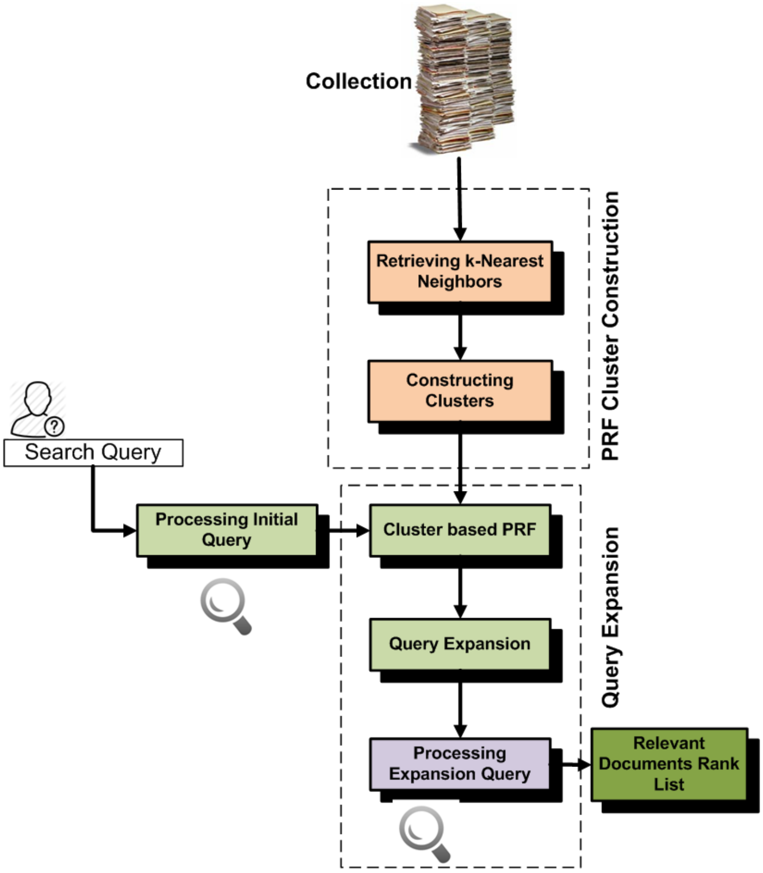

Figure 2. Architecture for cluster-based pseudo-relevance feedback. We construct the clusters using the query independent approach, and then during processing queries, we use these clusters for searching relevant documents for relevance feedback.

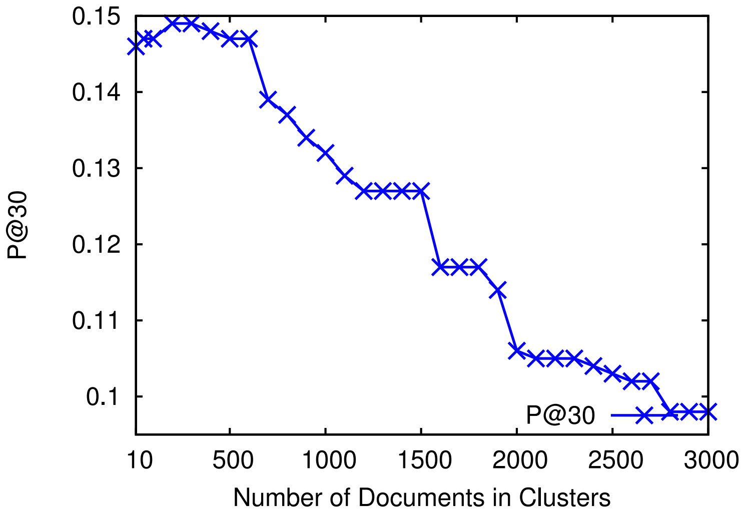

Figure 3. The maximum number of neighbors used for constructing clusters and their effect on effectiveness. Large values of *k* decrease the quality of topics in clusters and decrease the effectiveness.

4.4. Effectiveness Analysis

We select prior-art (PA) task of the TREC-CRT collection for analyzing the effectiveness of retrieval models. From each topic, we select only the claim section, because it is regarded as being the most representative piece of text, characterizing the scope of invention well due to the rules of the patent systems worldwide. In order to build prior-art queries from the claim section, we first sort all of the terms found in the claim section on the basis of their increasing term frequencies. We then select only the top 10 terms that have high frequencies and use these terms in the form of a long query for searching relevant documents. Using 10 terms' queries instead of very long 30 terms' queries allows us to examine more precisely the effectiveness of query expansion approaches over non-query expansion approaches.

Information retrieval experimentation has a strong underlying experimental methodology as used for example in the ongoing series of text retrieval conferences (TREC): a set of queries is run on a static collection of documents, with each query returning a list of answer resources. Humans assess the relevance of each document-query combination, and from this, a variety of system effectiveness metrics can be calculated. We use the following effectiveness metrics for analyzing the effectiveness of retrieval models.

- Recall: Recall cares about all relevant (judged) documents. It is the ratio of the number of retrieved relevant documents relative to the total number of documents in the collection that are desired to be retrieved.

  $$Recall = \frac{tp}{tp + fn} \tag{15}$$

  tp represents the total number of relevant documents retrieved. fn represents the false negative, the documents that are relevant, but could not be retrieved.
- Precision: Precision is the ratio of the number of retrieved relevant documents relative to the total number of retrieved documents. Precision measures the quality of the rank lists. However, since it does not consider the total number of relevant documents, therefore, a result list consisting of just a few retrieved and relevant documents might provide higher precision than a large result list with many relevant documents.

  $$Precision = \frac{tp}{tp + fp} \tag{16}$$

  fp represents the false positive, the documents that are retrieved, but are not relevant. Recall and precision are always used with rank cutoff levels. In our experiments we measured the recall with $R@100$ and precision with $P@30$ rank cutoff levels.
- Mean average precision (MAP): Precision and recall are not sensitive to the ranking order of documents (i.e., they do not consider how efficiently different retrieval models retrieve the relevant documents at the top ranked positions). Average precision cares for this factor by averaging the precision values obtained after each relevant document found. Thus, a retrieval model that ranks a large number of relevant documents at the top ranked positions would provide good average precision. It is calculated using the following equation.

  $$AveP(q) = \frac{\sum_{d \in D_q} (P@k_{dg}(q)) \cdot rel(d)}{tp + fn} \tag{17}$$

  D_q represents the set of retrieved documents of a query q, and k_{dq} is the rank of a document d in D_q. $rel(d)$ returns one, if d is a relevant judged document of q, otherwise zero. The mean average precision (MAP) is used for the average precision figures over a number of different queries.

  $$MAP = \frac{\sum_{q \in Q} AveP(q)}{|Q|} \tag{18}$$

- ***b-pref***: The *b-pref* measure is designed for those situations where the relevance judgments are known to be far from complete. It was introduced in the TREC 2005 terabyte track. *b-pref* computes a preference relation of whether the judged relevant documents are retrieved ahead of irrelevant documents. Thus, it is based only on the relative ranks of judged documents [34]. The *b-pref* measure is defined as:

$$b\text{-}pref = \frac{1}{J_q} \sum_{k_{jq} \in J_q} (1 - \frac{Number\ of\ e\ above\ k_{dq}}{(|E|)}) \quad (19)$$

where J_q is the set of judged relevant documents of a query q, E is the set of all judged irrelevant documents retrieved before the last judged document rank position in q, k_{jq} is the rank of judged document in q and e represents the count of irrelevant documents in E retrieved before the rank position k_{jq}. *b-pref* can be thought of as the inverse of the fraction of judged irrelevant documents that are retrieved before the relevant ones.

Table 1 lists the effectiveness of different retrieval strategies with *R*@100, *P*@30, MAP and *b-pref* on the TREC-CRT collection. If we compare only non-query expansion retrieval models, then the effectiveness of language modeling approaches (JM, AbsDis, DirS, TwoStage) is better than other retrieval models. If we compare only query expansion models, then TS-LM shows better effectiveness than TS-KLD. Overall, both query expansion models shows better effectiveness than non-query expansion models. As we can see from the results, the overall effectiveness of resampling (cluster-based PRF selection) is better. The resampling brings an improvement of (35%, 47%, 80%, 28%) on *R*@100, *P*@30, MAP and *b-pref* as compared to the language modeling (JM) approach and (11%, 23%, 29%, 05%) as compared to the standard PRF selection-based TS-LM approach. Table 2 shows the robustness of resampling, TS-LM, TS-KLD and JM to each other. The robustness is defined as the number of queries whose effectiveness is improved or hurt as the result of applying these methods over others. Resampling shows stronger robustness than JM (non-query expansion approach). It improves 540 queries and hurts 281, whereas the standard PRF selection approach TS-LM improves 493 queries and hurts 294. Although the resampling improves the effectiveness of only 47 more queries than TS-LM, however, the improvement obtained by the resampling is significantly large.

Table 1. Effectiveness of the retrieval models on the TREC-chemical patent retrieval task (CRT) collection. ⋆ indicates improvement on the effectiveness by applying cluster-based pseudo-relevance feedback (PRF). TFIDF, term frequency inverse document frequency; SMART, System for Manipulating and Retrieving Text; DirS, Dirichlet smoothing; JM, Jelinek–Mercer; AbsDis, absolute discount; LM, language modeling.

Retrieval Model	**Effectiveness**			
	R@100	*P*@30	**MAP**	***b-Pref***
TFIDF	0.015	0.006	0.005	0.133
NormTFIDF	0.057	0.032	0.016	0.243
BM25	0.121	0.079	0.037	0.351
SMART	0.036	0.016	0.010	0.210
DirS	0.144	0.093	0.045	0.406
JM	0.148	0.094	0.045	0.410
AbsDis	0.145	0.094	0.044	0.392
TwoStage	0.141	0.089	0.044	0.403
TS-LM	0.180	0.112	0.063	0.500
TS-KLD	0.157	0.097	0.050	0.429
Resampling	⋆0.199	⋆0.138	⋆0.081	⋆0.523

Table 2. Robustness of retrieval strategies relative to others using mean average precision (MAP). The values of the cell show that the numbers of queries improve or hurt by applying this method over others. The superscript α with $\star$ indicates statistically-significant improvements over others. We use the paired t-test with significance at $p < 0.05$.

	TS-LM	***Resampling***	***JM***	***TS-KLD***
TS-LM	-	292/448	$\star$493/294 $^{\alpha}$	$\star$449/313 $^{\alpha}$
Resampling	$\star$448/292 $^{\alpha}$	-	$\star$540/281 $^{\alpha}$	$\star$523/279 $^{\alpha}$
JM	294/493	281/540	-	305/371
TS-KLD	313/449	279/523	371/305	-

5. Retrieval Bias and Cluster-Based PRF

In the above section, we analyzed the effectiveness of the cluster-based PRF approach (resampling). We achieved higher effectiveness than standard PRF approaches. In standard cluster-based PRF, we construct clusters with only those neighbors of clusters that have high similarity with centroids. These nearest neighbors are retrieved using the top 30 frequent terms of centroid documents by assuming that during query processing, a subset of these cluster documents would be retrieved at top rank positions, thus increasing the effectiveness of PRF by retrieving relevant documents. This assumption works quite well when the retrieval bias of the retrieval model does not cause any effect on the retrievability of documents of clusters. However, if there exists a large retrievability inequality between documents of a collection (due to retrieval bias), then high retrievable documents of clusters are frequently retrieved at top rank positions for most of the queries due only to retrieval bias. These documents decrease the effectiveness of PRF. The objective of this section is to analyze to what extent retrieval bias effects the PRF effectiveness and how to improve the effectiveness by constructing clusters with the help of high similarity and retrievability. We perform the following set of experiments in order to understand this objective.

- In the first experiments, we construct clusters by retrieving k-nearest neighbors that have high similarity with centroid documents. We then partition all documents of the collection into different subsets according to their retrievability scores, and then, we analyze which subset contributes most to PRF effectiveness. These experiments help us with understanding which documents add noise in the PRF selection.
- In the second experiments, we construct clusters by retrieving k-nearest neighbors that have high similarity with the cluster centroids and that do not add noise due to retrieval bias. Basically, for these experiments, we remove high retrievable documents from the clusters. We then compare the effectiveness with the standard k-nearest neighbor approach and found high improvement.
- In the third experiments, we mine frequent neighbors of clusters and construct clusters by retrieving k-nearest neighbors that have high similarity with cluster centroids and are also frequently retrieved with documents of cluster. The standard retrievability approach always penalizes high retrievable documents even if these are most relevant to the clusters and frequently retrieved with documents of the cluster. Constructing clusters by mining frequent neighbors is helpful for increasing the effectiveness. We also compare the effectiveness of this approach with the retrievability-based cluster construction approach and found high improvement.

As all of the above experiments are based on retrievability, therefore, in the following sections, we first introduce the definition of the retrievability measure in IR; then, we describe the process of creating queries for calculating retrievability; and finally, we perform all experiments as described above.

5.1. Retrievability Measure

The following description of retrievability measurement as introduced by [9] provides a quick introduction of how it is measured.

Given a collection D, a retrieval model accepts a user query q and returns a ranked list of documents, which are deemed to be relevant to q. We can thus consider the retrievability of a document as influenced by two 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, [9] used query set-based sampling [35]. The query set Q could either be a historical sample of queries or an artificial simulated substitute similar to users' queries. Then, each user's $q \in Q$ is issued to the retrieval model, and the retrieved documents along with their positions in the ranked list are recorded. Intuitively, the retrievability of a document d is high when:

1. there are many probable queries in Q that can be expressed in order to retrieve d and
2. when retrieved, the rank r of the document d is lower 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 the 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.

$$r(d) = \sum_{q \in Q} p(q) \cdot \hat{f}(k_{dq}, c) \tag{20}$$

$\hat{f}(k_{dq}, c)$ 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 one if $k_{dq} \leq c$ and zero otherwise. $p(q)$ denotes the likeliness that a user actually issues query q. This probability may be hard to determine explicitly and is thus frequently set to one, i.e., to give all queries equal probabilities. More complex heuristics considering the length of the query, the specificity of the vocabulary, etc., may be considered. 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 is a large number of (highly probable) queries that can retrieve the document d at the rank less than c, and the value of $r(d)$ will be low when only a few 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 inequality between the retrievability score of documents can be further analyzed using the Lorenz curve [36]. 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. The work in [9] employed a similar idea in the context of a population of documents, where the wealth of documents is represented by the $r(d)$ function. The more skewed the plot, the greater the amount of inequality or bias within the population. The Gini coefficient [36] G is used to summarize the amount of retrieval bias in the Lorenz curve and provides a bird's eye view. It is computed as follows.

$$G = \frac{\sum_{i=1}^{|D|} (2 \cdot i - |D| - 1) \cdot r(d_i)}{(|D| - 1) \sum_{j=1}^{|D|} r(d_j)} \tag{21}$$

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.

5.2. Retrievability Analysis

We consider all sections (title, abstract, claims, description, background summary) of 1.2 million documents for both retrieval and query generation. Stop words are removed prior to indexing, and words stemming is performed with the Porter stemming algorithm. Additionally, we do not use those terms of the collection that have document frequency greater than 25% of the total collection size to remove high frequency stop words. For retrievability analysis, we generate the queries with the combinations of those terms that appear more than one time in the document. For these terms, all three-term and four-term combinations are used in the form of Boolean AND queries for creating the exhaustive set of queries Q, with duplicate queries being removed. Additionally, we consider only those queries that have a query result list size of more than the rank cutoff $c = 100$.

For calculating retrievability, we require the processing of all queries in Q on the full 1.2 million collection. This requires large processing time and resources. Thus, in order to complete the experiments in a reasonable time, we select a subset of two million queries from Q. However, rather than selecting this subset randomly, we select it on the basis of query quality prediction [37,38]. This is further motivated by earlier analysis of the relationship between query quality and retrieval bias [11]. In this method, we first order all queries in Q using the simplified query clarity score (SCS) [39]. Then, we select the two million queries that have the highest SCS scores. These queries are then used for document retrieval against the complete collection of 1.2 million documents as Boolean AND queries with subsequent ranking according to the chosen retrieval model to determine the retrievability scores of documents as defined in Equation (20).

Table 3 lists the retrieval bias of retrieval strategies using Gini coefficient for a range of rank cutoff factors. As expected, the Gini coefficient tends to decrease slowly for all query sets and models as the rank cutoff factor increases. This indicates that retrievability inequality within the collection is mitigated by the willingness of the user to search deeper down into the ranking. If users examine only the top documents, they will face a greater degree of retrieval bias. If we compare the retrieval bias of retrieval models, then we can observe that SMART has the greatest inequality between documents, while BM25 appears to provide the least inequality.

Table 3. Gini coefficients representing the retrieval bias of the retrieval models. High values indicate that retrieval models have a larger retrieval bias than others.

Retrieval Model	**High Quality Queries**		
	$c = 50$	$c = 100$	$c = 250$
NormTFIDF	0.68	0.60	0.49
BM25	0.55	0.50	0.43
DirS	0.59	0.53	0.46
JM	0.67	0.60	0.50
AbsDis	0.64	0.57	0.48
TwoStage	0.60	0.53	0.43
TFIDF	0.89	0.84	0.72
SMART	0.95	0.92	0.85

5.3. Retrieval Bias and PRF Effectiveness

In the above experiments, we observed that retrieval models add retrieval bias in the collection making a subset of documents more highly retrievable than others. For the first experiment, we want to analyze if a collection has large retrievability inequality between documents; then, do high retrievable documents of clusters decrease the effectiveness of cluster-based PRF? We want to analyze this hypothesis with the help of various partitions of the documents of clusters grouped according to retrievability scores. We want to analyze which partitions contribute most to PRF effectiveness than others. In order to perform this, we first sort the documents of the collection in ascending order according to their retrievability scores, and then, we divide the collection into p partitions (subsets). This means that the first subset contains (Partition 1) all those documents that accounted from the bottom $(100/p)\%$ of the cumulative retrievability scores, and the last subset (Partition 5) would contain the most retrievable documents containing the $(100 - (100/p))\%$ percentile. Since we generate clusters using BM25 weighting, therefore, for the experiments, we consider the retrievability scores that we calculated using BM25. For each subset, we then individually analyze the PRF effectiveness on 1000 known prior-art topics queries as explained above by keeping only those documents of clusters that are part of the analyzed subset and ignoring all those that are not part of the subset. The idea is to issue each topic query to the whole collection and then to analyze which subsets add noise and which subsets contribute most to PRF effectiveness than others.

Table 4 shows the PRF effectiveness of subsets with $p = 5$. This table also shows which subset is retrieved more frequently than others within the top 500 documents. We examine this by the retrieval probability of subsets within the top 500 documents (i.e., how many documents of a subset are retrieved within the top 500 documents). As we can observe from the results, high retrievable subsets have low PRF effectiveness than low retrievable subsets, although documents of high retrievable subsets are retrieved more frequently than the documents of low retrievable subsets (see Table 4). Although, high retrievable documents are also a relevant part of the clusters, however, when these are frequently retrieved at top positions for many irrelevant queries just due to retrieval bias, these decrease the effectiveness by generating noisy clusters. Low retrievable documents on the other side are not frequently retrieved due to retrieval bias; thus, when we construct clusters by taking only low retrievable documents, then the noise of retrieval bias does not decrease effectiveness. We found this the main reason why low retrievable partitions have high effectiveness for PRF than high retrievable partitions. These results confirm our hypothesis that the retrieval bias of retrieval models seriously degrades the retrieval effectiveness of cluster-based PRF.

Table 4. Effectiveness of different subsets with $p = 5$. We first sort the documents of the collection in ascending order according to their retrievability scores, and then, we divide the collection into p partitions. Partition 5 contains high retrievable documents containing the $(100 - (100/p))\%$ percentile, and Partition 1 contains all those documents that accounted for the bottom $(100/p)\%$ of the cumulative retrievability scores (low retrievability documents). High retrievable subsets show low PRF effectiveness due to retrieval bias. Subset % within the top 500 docs shows that the subset is retrieved more frequently than others within the top 500 documents. We examine this by the retrieval probability of subsets within the top 500 documents (i.e., how many documents of a subset are retrieved within the top 500 documents). ⋆ indicates results are better than others.

Retrieval Model	Subset % within Top 500 Documents	Effectiveness			
		R@100	*P@30*	**MAP**	***b-pref***
Partition 1	11%	⋆0.103	⋆0.221	⋆0.150	⋆0.551
Partition 2	14%	⋆0.098	⋆0.213	⋆0.143	⋆0.546
Partition 3	20%	0.095	0.212	0.136	0.543
Partition 4	⋆25%	0.090	0.199	0.130	0.540
Partition 5	⋆30%	0.091	0.191	0.129	0.540

5.4. *Retrievability and Cluster-Based PRF* (RetrClusPRF)

From t he above experiments, we observe that the standard approach constructs clusters for PRF without carrying retrievability inequality between documents. Therefore, during query processing, high retrievable documents of clusters are frequently retrieved at top positions for most of the queries, and these drift the PRF away from relevant documents. This decreases the effectiveness of cluster-based PRF. In order to improve the effectiveness, we construct clusters by retrieving *k*-nearest neighbors on the basis of high similarity and then using retrievability. For each cluster, first we rank the 2 × k-nearest neighbors of clusters on the basis of their high similarity with centroid documents, and then, we again re-rank *k*-nearest neighbors from 2 × k list on the basis of low retrievability scores. The rationale behind keeping low retrievable neighbors in the clusters is that we want to minimize the effect of retrieval bias. We perform this process for all clusters, and then, we use these clusters for PRF. Tables 5 and 6 show the effectiveness of this approach (RetrClusPRF) and the standard *k*-nearest neighbor approach (resampling as a baseline), where we construct clusters by retrieving only similar neighbors. Figure 4 shows the architecture of RetrClusPRF. As we can seen from the architecture, retrievability provides help in removing noisy (high retrievable) documents from the clusters. If we analyze the results, then RetrClusPRF brings an improvement of (12%, 15%, 32%, 6%) on *R*@100, *P*@30, MAP and *b-pref* as compared to resampling. Table 6 shows the robustness of RetrClusPRF with resampling and also with other retrieval approaches. RetrClusPRF shows stronger robustness than resampling. It improves 428 queries and hurts 321. As we can observe from the results, retrievability-based PRF (RetrClusPRF) achieves significantly better effectiveness than standard cluster-based PRF technique.

Table 5. Effectiveness of *RetrClusPRF* and *FreqNeigPRF* on the TREC-CRT collection. ⋆ indicates improvement on the effectiveness that is gained by applying enhanced cluster construction techniques over the baseline (resampling) approach.

Retrieval Model	**Effectiveness**			
	R@100	*P*@30	**MAP**	*b-pref*
TFIDF	0.015	0.006	0.005	0.133
NormTFIDF	0.057	0.032	0.016	0.243
BM25	0.121	0.079	0.037	0.351
SMART	0.036	0.016	0.010	0.210
DirS	0.144	0.093	0.045	0.406
JM	0.148	0.094	0.045	0.410
AbsDis	0.145	0.094	0.044	0.392
TwoStage	0.141	0.089	0.044	0.403
TS-LM	0.180	0.112	0.063	0.500
TS-KLD	0.157	0.097	0.050	0.429
Resampling	0.199	0.138	0.081	0.523
RetrClusPRF	⋆0.223 (+12%)	⋆0.159 (+15%)	⋆0.107 (+32%)	⋆0.552 (+06%)
FreqNeigPRF	⋆0.235 (+18%)	⋆0.168 (+22%)	⋆0.121 (+49%)	⋆0.572 (+09%)

Table 6. Robustness of RetrClusPRF and FreqNeigPRF relative to others using mean average precision (MAP). The values of the cell show that the number of queries improves or hurts by applying this method over others. The superscript α with ⋆ indicates statistically-significant improvements over others. We use the paired *t*-test with significance at $p < 0.05$.

	TS-LM	***Resampling***	***JM***	***TS-KLD***	***RetrClusPRF***	***FreqNeigPRF***
TS-LM	-	292/448	493/294 $^{\alpha}$	449/313 $^{\alpha}$	217/518	205/542
Resampling	⋆448/292 $^{\alpha}$	-	⋆540/281 $^{\alpha}$	⋆523/279 $^{\alpha}$	⋆321/428	⋆304/441
JM	294/493	281/540	-	305/371	231/589	201/608
TS-KLD	313/449	279/523	371/305	-	236/567	210/599
RetrClusPRF	⋆518/217 $^{\alpha}$	⋆428/321 $^{\alpha}$	⋆589/231 $^{\alpha}$	⋆567/236 $^{\alpha}$	-	⋆327/401
FreqNeigPRF	⋆542/205 $^{\alpha}$	⋆441/304 $^{\alpha}$	⋆608/201 $^{\alpha}$	⋆599/210 $^{\alpha}$	⋆401/327 $^{\alpha}$	-

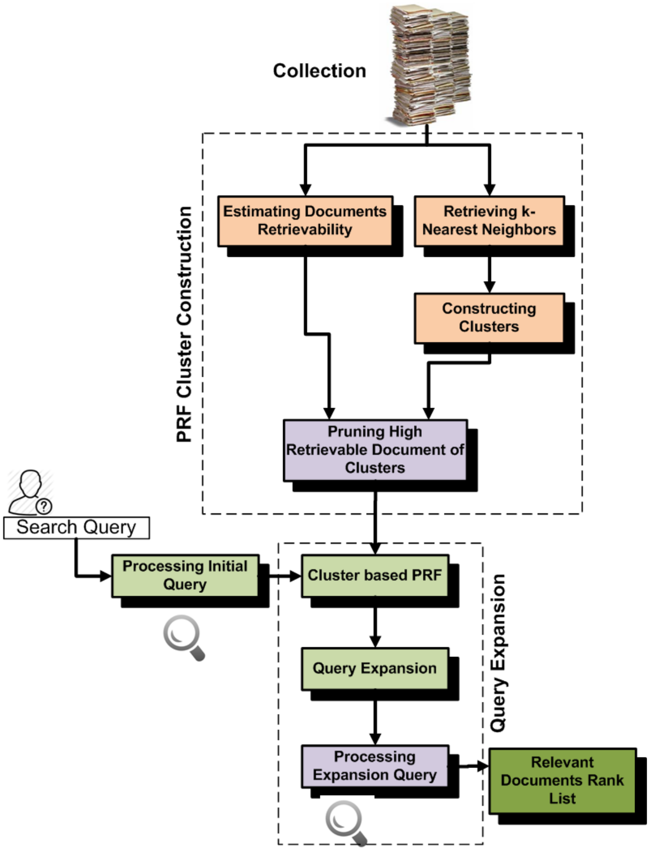
Collection
PRF Cluster Construction
Estimating Documents Retrievability
Retrieving k-Nearest Neighbors
Constructing Clusters
Pruning High Retrievable Document of Clusters
Search Query
Processing Initial Query
Cluster based PRF
Query Expansion
Processing Expansion Query
Query Expansion
Relevant Documents Rank List

Figure 4. Architecture of retrievability and cluster-based pseudo-relevance feedback (RetrClusPRF).

5.5. *Frequent Neighbor-Based PRF* (FreqNeig)

The above experiments indicate that retrievability-based clustering improves PRF effectiveness. The standard retrievability approach only analyzes individual retrievability scores of documents, but does not discover which documents are frequently retrieved together. As a result, this approach always penalizes high retrievable documents and can ignore many (high retrievable) relevant documents of clusters that are frequently retrieved with documents of the cluster. Further improvement is possible by ranking k-nearest neighbors that have a high confidence of retrieval with other documents of the cluster.

In order to mine frequent neighbors (FreqNeig), we represent the result list of queries as transactions and mine FreqNeig using the frequent itemset mining algorithm [40]. We mine all those FreqNeig that have a length larger than one and support greater than 0.05%. After mining FreqNeig, we first rank 2 × k-nearest neighbors on the basis of their high similarity with centroids, and then, we again re-rank k-nearest neighbors from the 2 × k list that have a high confidence of retrieval with the nearest neighbors of clusters. For each nearest neighbor, we assign a confidence score using the following approach. For each nearest neighbor k_i, we first obtain the set s_i of all those frequent itemsets that contain k_i. From the set s_i, we then remove all of those itemsets that have an item that is not present in the 2 × k set. After pruning itemsets, we generate association rules for the remaining itemsets using confidence support greater than 20%. For a rule $D_r \rightarrow d_i$, the higher confidence for document d_i implies that d_i is more likely to retrieve in the result lists of queries that also contain documents of set D_r. We generate association rules for all itemsets of s_i, and then, we rank neighbors in the 2 × k list on the basis of their average confidence score. Figure 5 shows the architecture of this approach.

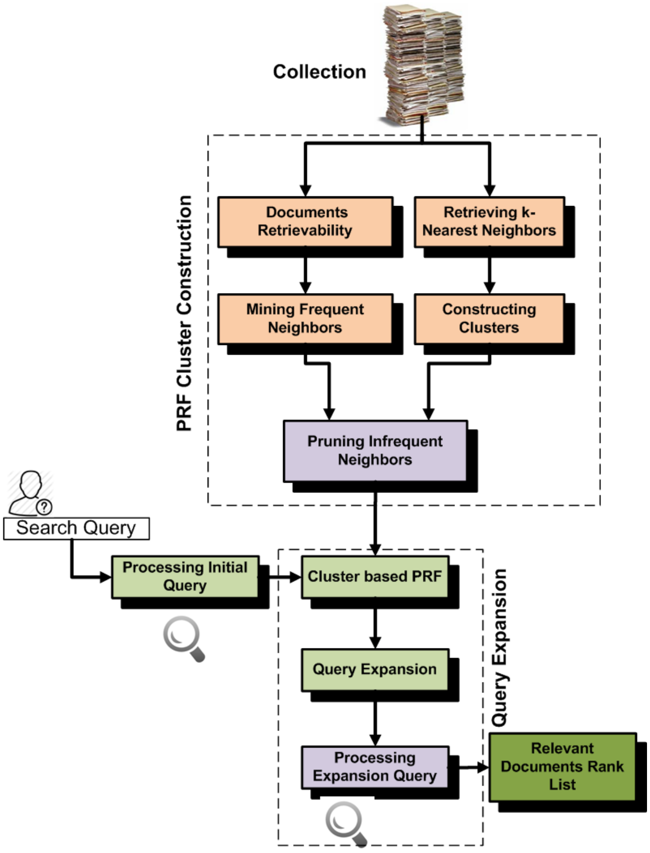

Figure 5. Architecture of frequent neighbor-based pseudo-relevance feedback (FreqNeigPRF).

Tables 5 and 6 show the comparison of effectiveness of this approach (FreqNeigPRF) with the retrievability-based PRF (RetrClusPRF) and standard k-nearest neighbor approach (resampling) (as a baseline). FreqNeigPRF brings an improvement of (5%, 6%, 13%, 4%) on *R*@100, *P*@30, MAP and *b-pref* as compared to RetrClusPRF and (18%, 22%, 49%, 9%) as compared to resampling. Table 6 shows the robustness of resampling, RetrClusPRF and FreqNeigPRF to each other. As expected, FreqNeigPRF shows stronger robustness than resampling. It improves 441 queries and hurts 304, whereas RetrClusPRF improves 428 queries and hurts 321. Although FreqNeigPRF improves the effectiveness of only 13 more queries than RetrClusPRF, however, the improvement obtained by FreqNeigPRF is large. As we can observe from the results, frequent retrieved document-based PRF achieves significantly higher effectiveness than other two approaches.

6. Conclusions

In this paper, we studied the effect of retrieval bias on cluster-based pseudo relevance feedback (PRF). We first showed that if we construct clusters for PRF only by retrieving highly similar neighbors and ignore retrievability, then high retrievable documents of clusters add noise in PRF due to retrieval bias. We then extend the standard approach by constructing clusters on the basis of high similarity and retrievability. In this approach, we first retrieve the documents of clusters on the basis of their similarity, and then, we re-ranked documents and keep only those documents in the clusters that do not decrease the effectiveness due to retrieval bias. We further improve this approach by mining frequent neighbors of clusters and keep only those documents in the clusters that are frequently retrieved with documents of the clusters and are infrequently retrieved with those documents that are not part of clusters. Experiments show that retrievability-based PRF is helpful for identifying better documents for pseudo relevance feedback.

Conflicts of Interest: The author declares no conflict of interest.

References

1. Attar, R.; Fraenkel, A.S. Local feedback in full-text retrieval systems. *J. ACM* **1977**, *24*, 397–417.
2. Buckley, C.; Salton, G.; Allan, J.; Singhal, A. *Automatic Query Expansion Using Smart: Trec 3*; DIANE Publishing: Collingdale, PA, USA, 1994.
3. Lavrenko, V.; Croft, W.B. Relevance based language models. In Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, New Orleans, LA, USA, 9–12 September 2001; ACM: New York, NY, USA, 2001; pp. 120–127.
4. Robertson, S.; Walker, S.; Beaulieu, M.; Gatford, M.; Payne, A. Okapi at trec-4. In Proceedings of the Fourth Text REtrieval Conference (TREC–4), Gaithersburg, MD, USA, 1–3 November 1995.
5. Salton, G.; Buckley, C. Improving retrieval performance by relevance feedback. *J. Am. Soc. Inf. Sci.* **1990**, *41*, 288–297.
6. Gelfer Kalmanovich, I.; Kurland, O. Cluster-based query expansion. In Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '09, Boston, MA, USA, 19–23 July 2009; pp. 646–647.
7. Lee, K.S.; Croft, W.B.; Allan, J. A cluster-based resampling method for pseudo-relevance feedback. In Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '08, Singapore, 20–24 July 2008; pp. 235–242.
8. Lee, K.S.; Croft, W.B. A deterministic resampling method using overlapping document clusters for pseudo-relevance feedback. *Inf. Process. Manag.* **2013**, *49*, 792–806.
9. Azzopardi, L.; Vinay, V. Retrievability: An evaluation measure for higher order information access tasks. In Proceedings of the 17th ACM Conference on Information and Knowledge Management, CIKM '08, Napa Valley, CA, USA, 26–30 October 2008; pp. 561–570.
10. Bashir, S.; Rauber, A. Improving retrievability of patents in prior-art search. In Proceedings of the 32nd European Conference on Advances in Information Retrieval, ECIR'2010, Milton Keynes, UK, 28–31 March 2010; Springer: Berlin/Heidelberg, Germany, 2010; pp. 457–470.
11. Bashir, S.; Rauber, A. On the relationship between query characteristics and ir functions retrieval bias. *J. Am. Soc. Inf. Sci. Technol.* **2011**, *62*, 1512–1532.
12. Azzopardi, L.; Bache, R. On the relationship between effectiveness and accessibility. In Proceedings of the 33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '10, Geneva, Switzerland, 19–23 July 2010; pp. 889–890.
13. Lupu, M.; Huang, J.; Zhu, J.; Tait, J. TREC-CHEM: Large scale chemical information retrieval evaluation at trec. In *SIGIR Forum*; ACM: New York, NY, USA, 2009; Volume 43, Number 2, pp. 63–70.
14. Croft, W.B. *Advances in Information Retrieval, Recent Research from the Center for Intelligent Information Retrieval*; Springer: Berlin/Heidelberg, Germany, 2000.
15. Robertson, S.E.; Walker, S. Okapi/keenbow at trec-8. In Proceedings of the Eighth Text REtrieval Conference (TREC-8), Gaithersburg, MD, USA, 19–21 November 1999.
16. Sakai, T.; Manabe, T.; Koyama, M. Flexible pseudo-relevance feedback via selective sampling. *Trans. Asian Lang. Inf. Process.* **2005**, *4*, 111–135.
17. Zhai, C.; Lafferty, J. A study of smoothing methods for language models applied to information retrieval. *ACM Trans. Inf. Syst.* **2004**, *22*, 179–214.
18. Cao, G.; Nie, J.-Y.; Gao, J.; Robertson, S. Selecting good expansion terms for pseudo-relevance feedback. In Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '08, Singapore, 20–24 July 2008; ACM: New York, NY, USA, 2008; pp. 243–250.
19. Huang, X.; Huang, Y.R.; Wen, M.; An, A.; Liu, Y.; Poon, J. Applying data mining to pseudo-relevance feedback for high performance text retrieval. In Proceedings of the Sixth International IEEE Computer Society Conference on Data Mining, ICDM '06, Washington, DC, USA, 8–22 December 2006; pp. 295–306.
20. Huang, Q.; Song, D.; Ruger, S. Robust query-specific pseudo feedback document selection for query expansion. In Proceedings of the IR Research, 30th European Conference on Advances in Information Retrieval, ECIR'08, Milton Keynes, Glasgow, UK, 30 March–3 April 2008; pp. 547–554.

21. Collins-Thompson, K.; Callen, J. Estimation and use of uncertainty in pseudo-relevance feedback. In Proceedings of the 30th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'07, Amsterdam, The Netherlands, 23–27 July 2007; pp. 295–306.
22. Boteanu, B.; Mironica, I.; Ionescu, B. Hierarchical clustering pseudo-relevance feedback for social image search result diversification. In Proceedings of the 13th International Workshop on Content-Based Multimedia Indexing, CBMI 2015, Prague, Czech Republic, 10–12 June 2015; pp. 1–6.
23. Buckley, C.; Mitra, M.; Walz, J.; Cardie, C. Using Clustering and Superconcepts within Smart: Trec 6. *Inf. Process. Manag.* **1998**, *36*, 109–131.
24. Lynam, T.R.; Buckley, C.; Clarke, C.L.A.; Cormack, G.V. A multi-system analysis of document and term selection for blind feedback. In Proceedings of the Thirteenth ACM International Conference on Information and Knowledge Management, CIKM '04, Washington, DC, USA , 8–13 November 2004; ACM: New York, NY, USA, 2004; pp. 261–269.
25. Yeung, D.L.; Clarke, C.L.A.; Cormack, G.V.; Lynam, T.R.; Terra, E.L. Task-specific query expansion (multitext experiments for trec 2003). In Proceedings of the 2002 Text REtrieval Conference, Gaithersburg, MD, USA, 19–22 November 2002.
26. Bashir, S.; Rauber, A. Improving retrievability of patents with cluster-based pseudo-relevance feedback documents selection. In Proceedings of the 18th ACM Conference on Information and Knowledge Management, CIKM '09, Hong Kong, China, 2–6 November 2009; pp. 1863–1866.
27. Fujita, S. Technology survey and invalidity search: A comparative study of different tasks for Japanese patent document retrieval. *Inf. Process. Manag.* **2007**, *43*, 1154–1172.
28. Mase, H.; Matsubayashi, T.; Ogawa, Y.; Iwayama, M.; Oshio, T. Proposal of two-stage patent retrieval method considering the claim structure. *Trans. Asian Lang. Inf. Process.* **2005**, *4*, 190–206.
29. Itoh, H. NTCIR-4 Patent Retrieval Experiments at RICOH. In Proceedings of the NTCIR-4 Workshop Meeting, NTCIR '04, Tokyo, Japan, 2–4 June 2004.
30. Shinmori, A.; Okumura, M.; Marukawa, Y.; Iwayama, M. Patent claim processing for readability: Structure analysis and term explanation. In Proceedings of the ACL-2003 Workshop on Patent Corpus Processing, Sapporo, Japan, 12 July 2003; Volume 20, pp. 56–65.
31. Singhal, A. At&t at trec-6. Available online: http://singhal.info/trec6.pdf (accessed on 11 November 2016).
32. Larkey, L.S.; Allan, J.; Connell, M.E.; Bolivar, A.; Wade, C. Umass at TREC 2002: Cross language and novelty tracks. In Proceedings of the Text Retrieval Conference (TREC), Gaithersburg, MD, USA, 19–22 November 2002; pp. 721–732.
33. Liu, X.; Croft, W.B. Cluster-based retrieval using language models. In Proceedings of the 27th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR'04, Sheffield, UK, 25–29 July 2004; pp. 186–193.
34. Different Evaluation Measures. Available online: http://trec.nist.gov/pubs/trec16/appendices/measures.pdf (accessed on 14 November 2016).
35. Callan, J.; Connell, M. Query-based sampling of text databases. *ACM Trans. Inf. Syst. (TOIS) J.* **2001**, *19*, 97–130.
36. Gastwirth, J.L. The estimation of the LORENZ curve and GINI index. *Rev. Econ. Stat.* **1972**, *54*, 306–316.
37. He, B.; Ounis, I. Query performance prediction. *Inf. Syst.* **2006**, *31*, 585–594.
38. Zhao, Y.; Scholer, F.; Tsegay, Y. Effective pre-retrieval query performance prediction using similarity and variability evidence. In Proceedings of the 30th European Conference on Advances in Information Retrieval, ECIR'08, Glasgow, UK, 30 March–3 April 2008; pp. 52–64.
39. Cronen-Townsend, S.; Zhou, Y.; Croft, W.B. Predicting query performance. In Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '02, Tampere, Finland, 11–15 August 2002; pp. 299–306.
40. Han, J.; Pei, J.; Yin, Y. Mining frequent patterns without candidate generation. In Proceedings of the 2000 ACM SIGMOD International Conference on Management of Data, SIGMOD '00, Dallas, TX, USA, 15–18 May 2000; pp. 1–12.