Information Processing and Management 49 (2013) 792–806

Contents lists available at SciVerse ScienceDirect

Information Processing and Management

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

A deterministic resampling method using overlapping document clusters for pseudo-relevance feedback

Kyung Soon Lee [a,*], W. Bruce Croft [b]

[a] Division of Computer Science and Engineering, CAIIT, Chonbuk National University, 567 Baekje-daero, Deokjin-gu, Jeonju, Jeollabuk-do 561-756, Republic of Korea
[b] Center for Intelligent Information Retrieval, Department of Computer Science, University of Massachusetts Amherst, 140 Governors Drive, Amherst, MA 01003-9264, USA

ARTICLE INFO

Article history:
Received 17 February 2010
Received in revised form 30 December 2012
Accepted 10 January 2013
Available online 28 February 2013

Keywords:
Information retrieval
Pseudo-relevance feedback
Relevance model
Deterministic resampling
Dominant documents
Query expansion

ABSTRACT

Typical pseudo-relevance feedback methods assume the top-retrieved documents are relevant and use these pseudo-relevant documents to expand terms. The initial retrieval set can, however, contain a great deal of noise. In this paper, we present a cluster-based resampling method to select novel pseudo-relevant documents based on Lavrenko's relevance model approach. The main idea is to use overlapping clusters to find dominant documents for the initial retrieval set, and to repeatedly use these documents to emphasize the core topics of a query.

The proposed resampling method can skip some documents in the initial high-ranked documents and deterministically construct overlapping clusters as sampling units. The hypothesis behind using overlapping clusters is that a good representative document for a query may have several nearest neighbors with high similarities, participating in several different clusters. Experimental results on large-scale web TREC collections show significant improvements over the baseline relevance model.

To justify the proposed approach, we examine the relevance density and redundancy ratio of feedback documents. A higher relevance density will result in greater retrieval accuracy, ultimately approaching true relevance feedback. The resampling approach shows higher relevance density than the baseline relevance model on all collections, resulting in better retrieval accuracy in pseudo-relevance feedback.

1. Introduction

Most pseudo-relevance feedback methods (e.g., Attar & Fraenkel, 1977; Buckley, Salton, Allan, & Singhal, 1995; Croft & Harper, 1979; Lavrenko & Croft, 2001; Robertson, Walker, Beaulieu, Gatford, & Payne, 1996) assume that a set of top-retrieved documents is relevant and then learn from the pseudo-relevant documents to expand terms or to assign better weights to the original query. This is similar to the process used in relevance feedback, when actual relevant documents are used (Salton & Buckley, 1990). In general, however, the top retrieved documents contain noise: when the precision of the top 10 documents (P@10) is 0.5, this means that five of them are non-relevant. This is common and even expected in all retrieval models. When combined with pseudo-relevance feedback, this noise, however, can cause the query representation to "drift" away from the original query.

This paper describes *a deterministic sampling method based on overlapping clusters* to select better documents for pseudo-relevance feedback. The sampling unit is a document cluster from the initial retrieval set which can represent an aspect of a

* Corresponding author. Tel.: +82 63 270 4138; fax: +82 63 270 2394.
E-mail addresses: selfsolee@jbnu.ac.kr (K.S. Lee), croft@cs.umass.edu (W.B. Croft).

0306-4573/$ - see front matter © 2013 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.ipm.2013.01.001

query, especially with large-scale web test collections, since the initial retrieval results may involve diverse subtopics for such collections. Since it is difficult to find one optimal cluster, we use several relevant groups for feedback. By permitting overlapped clusters for the top-retrieved documents and repeatedly using the *dominant documents* that appear in multiple highly-ranked clusters, we expect that an expansion query can be represented to emphasize the core topics of a query.

This is not the first time that clustering has been suggested as an improvement for relevance feedback. In fact, clustering was mentioned in some of the first work related to pseudo-relevance feedback (Attar & Fraenkel, 1977). Previous attempts to use clusters have not improved effectiveness. The work presented here is based on a new approach to using the clusters that produces significantly better results.

Our motivation for using overlapping clusters and resampling is as follows: the top-retrieved documents are a query-oriented ordering that does not consider the relationship between documents. We view the pseudo-relevance feedback problem of learning expansion terms closely related to a query to be similar to the classification problem of learning an accurate decision boundary, depending on training examples. There are two stages to expand query terms based on the pseudo-relevance feedback framework: (1) Classifying pseudo-relevant documents for the initial retrieval set. (2) Extracting an expansion query for the classified pseudo-relevant documents. Here, we focus on unsupervised learning with pseudo-relevant documents. We approach this problem by repeatedly selecting dominant documents biased toward *good* representative documents with high agreement in the initial retrieval set. In contrast, the boosting method for a weak learner repeatedly selects hard examples to change the decision boundary toward hard examples. For the query expansion problem, it is important to use good representative documents with high agreement in the initial set, resulting in better representative expansion terms for the topic of a query.

The hypothesis behind using overlapped document clusters is that a good representative document for a query may have several nearest neighbors with high similarities, participating in several different clusters. Since it plays a central role in forming clusters, this document may be dominant for this topic. Repeatedly sampling dominant documents can emphasize the topics of a query, rather than randomly resampling documents for feedback.

We show that resampling feedback documents based on clusters contributes to higher relevance density for feedback documents on a variety of TREC collections. The results on large-scale web collections such as the TREC WT10g and GOV2 collections show significant improvements over the baseline relevance model.

The rest of the paper is organized as follows: Section 2 presents related work. Section 3 describes a cluster-based resampling framework. Section 4 shows experimental results on TREC test collections, and results analyses. In Section 5, we justify the experimental results by studying relevance density. We conclude in Section 6.

2. Related work

Our approach is related to previous work on pseudo-relevance feedback, resampling approaches, and the cluster hypothesis in information retrieval.

Relevance feedback (RF) and pseudo-relevance feedback (PRF) have been shown to be effective ways of improving retrieval accuracy by reformulating an original query using relevant or pseudo-relevance documents from the initial retrieval result. New interest in relevance feedback has resulted in the establishment of a relevance feedback track at TREC (2008, 2009). This track provides a framework for exploring the effects of different factors on relevance feedback, such as initial retrieval, judgment procedure, core reformulation algorithm, and multiple iterations on the terabyte document collection. In TREC (2008), the relevance feedback task was to compare RF algorithms with exactly the same relevance judgments, and RF results for multiple amounts of relevance information – i.e. one relevant document or 10 judged documents with at least three relevant documents (Kaptein, Kamps, & Hiemstra, 2008; Lv & Zhai, 2008; Tsegay, Scholer, & Puglisi, 2008; Zhao, Liang, & Callan, 2008). In TREC (2009), the focus shifted to methods for finding good documents for each topic and the impact of different documents on algorithms. Specifically, sites compared results using the documents they found with those found by other sites' algorithms. Preliminary results indicate that the relationship between which documents are used and success of an algorithm is unclear (Buckley, 2009). The motivation of the track shows the current state of research: that relevance feedback is one of the successes of information retrieval over the past 30 years, in that it is applied in a wide variety of settings as both explicit and implicit feedback; however there is surprisingly little new basic research (Buckley & Robertson, 2008). At the RIA workshop (Buckley & Harman, 2004), there were comparative experiments on the effects of several factors for pseudo-relevance feedback. The report provides the effects of the number of documents, the number and source of terms used, the initial set of documents, and the effects of swapping documents and clusters by document clustering and passage-level clustering. The experimental setup is too complex to see the individual effects of clusters, since an outside source factor is mixed up with the clustering factor (Yeung, Clarke, Cormack, Lynam, & Terra, 2004): using outside sources for feedback itself affects the performance. Thus the analysis for the comparative experiments is inconclusive.

Traditional pseudo-relevance feedback algorithms (Robertson et al., 1996) are based on the assumption of relevancy for the top-retrieved documents. Research has been conducted to improve traditional PRF by using passages (Allan, 1995; Yeung et al., 2004) instead of documents, by using a local context analysis method (Xu & Croft, 1996), by using a query-regularized estimation method (Tao & Zhai, 2006), by drawing statistics from external corpora (Diaz & Metzler, 2006), and by using latent concepts (Metzler & Croft, 2007). These methods follow the basic assumption that the top-retrieved documents are relevant to a query.

Recently there has been some work on sampling and resampling techniques for the initial retrieval set. A selective sampling method by Sakai, Manabe, and Koyama (2005) skips some top-retrieved documents based on a clustering criterion. The

cluster is generated not by document similarity but all members of a cluster containing the same subset of query terms. The sampling purpose is to select a more varied and novel set of documents for feedback. Their assumption is that the top-ranked documents may be too similar or redundant. However, their results did not show significant improvements on NTCIR collections. Our approach of repeatedly and deterministically using dominant documents is based on a different assumption. A deterministic sampling is a sampling which behaves predictably (Vishkin, 1991). Given a particular input, it will always produce the same output. In applications such as motion planning and verification problems in robotics and graphics, deterministic sampling has shown benefits compared to random sampling (Yershova, Jain, & LaValle, 2010; Yershova & LaValle, 2004). A resampling method suggested by Collins-Thompson and Callan (2007) uses bootstrap sampling on the top-retrieved documents for the query and variants of the query obtained by leaving a single term out. The assumption behind query variants is that one of the query terms is a noise term. From their experimental analysis, the main gain is from the use of query variants, not document resampling. Their results on robustness and precision at 10 documents (P@10) show improvements, but the performance in terms of mean average precision (MAP) is lower than the baseline relevance model on TREC collections. Our approach primarily focuses on the effects of resampling the top-retrieved documents. Recently, a boosting approach proposed by Lv, Zhai, and Chen (2011) combined different document weighting strategies using a loss function defined to directly measure both robustness and effectiveness to improve the overall effectiveness of pseudo-relevance feedback without sacrificing the performance of individual queries too much.

On the other hand, many information retrieval techniques have adopted the cluster hypothesis to improve effectiveness. The cluster hypothesis states that *closely related documents tend to be relevant to the same request* (Jardine & Rijsbergen, 1971). Re-ranking using clusters (Lee, Kageura, & Choi, 2004; Lee, Park, & Choi, 2001) based on the vector space model has shown successful results. A cluster-based retrieval model (Liu & Croft, 2004) based on language modeling ranks clusters by the likelihood of generating the query. The results show improvements over the query-likelihood retrieval model on TREC collections. A local score regularization method (Diaz, 2005) uses a document affinity matrix to adjust initial retrieval scores so that topically related documents receive similar scores. The results on TREC collections show that regularized scores are significantly better than the initial scores. Our work is closely related to document re-ranking using cluster validation and label propagation (Yang, Ji, Zhou, Nie, & Xiao, 2006), document-based language models by the incorporation of cluster information (Kurland & Lee, 2004), re-ranking method using cluster-based language models within a graph-based framework (Kurland & Lee, 2006), re-ranking using an affinity graph (Zhang et al., 2005) and iterative pseudo-query processing using cluster-based language models (Kurland & Lee, 2005), and cluster-based fusion of retrieved lists (Kozorovitzky & Kurland, 2011).

There has also been work on term expansion using clustering in the vector space model (Buckley, Mitra, Walz, & Cardie, 1998; Lynam, Buckley, Clarke, & Cormack, 2004; Yeung et al., 2004). At TREC 6, Buckley et al. (1998) used document clustering on SMART though the results of using clusters did not show improvements over the baseline feedback method. Recently, a cluster-based query expansion method (Kalmanovich & Kurland, 2009) combined document clusters that are created offline and the top-retrieved documents for pseudo-relevance feedback in the relevance model.

This paper extends previous work by the authors (Lee, Croft, & Allan, 2008) by adding analyses of experimental results for all collections, justifying the hypothesis of *dominant documents* by redundancy ratio and the distribution of relevance density and average precision, and analyzing results for each query.

3. A cluster-based deterministic resampling framework for feedback

This section describes the rationale for the method for a deterministic sampling and our sampling procedure.

3.1. A deterministic sampling approach

The main issues in pseudo-relevance feedback are how to select (likely) relevant documents from the top-retrieved documents, and how to select expansion terms. Here we deal with the problem of selecting better feedback documents.

The problem in traditional pseudo-relevance feedback is obtaining a set of expansion terms from the top-retrieved documents that may have low precision. If a method can select better documents from the given sample, it can almost certainly contribute better expansion terms. For pseudo-relevance feedback, the initial retrieval set can be seen as the sample space from which we estimate the sampling distribution of documents.

In statistics, resampling (bootstrapping) (Efron, 1979) is a method for estimating the precision of sample statistics by sampling *randomly* with replacement from the original sample, leading to robust estimates. If a method is available for selecting better examples from the original sample space, ***deterministic resampling*** will perform better than ***random*** sampling. In some cases, deterministic sampling has shown benefits compared to random sampling (Vishkin, 1991; Yershova & LaValle, 2004). Boosting (Freund, 1990; Schapire, 1990) is a selective resampling method in machine learning. It is an iterative procedure used to *adaptively change the distribution of training examples* so that the weak learners focus on examples that previous weak learners misclassified. In contrast, in the query expansion problem, it is important to change the distribution toward good representative documents for a query. We assume that ***a dominant document for a query*** is one with *good representation* of the topics of a query—i.e. one with several nearest neighbors with high similarity. In overlapped clusters, a dominant document will appear in multiple highly-ranked clusters. Since a topic can contain several subtopics, the

retrieved set can be divided into several subtopic groups. A document that deals with all subtopics will likely be in all subtopic clusters, so we call that document *dominant*. From such a dominant document, terms that retrieve documents related to all subtopics can be selected as an expanded query.

Based on the above assumption, we *deterministically sample* documents for feedback using k-nearest neighbors (k-NN) clustering to generate overlapped clusters from the given top-retrieved documents space.

3.2. Deterministically resampling feedback documents using overlapping clusters

A cluster-based resampling method to get novel pseudo-relevant documents is based on the language model (Ponte & Croft, 1998), the cluster-based language model (Liu & Croft, 2004), and the relevance model (Lavrenko & Croft, 2001) frameworks. The essential point of our approach is that a document that appears in multiple highly-ranked clusters will contribute more to the query terms than other documents. The resampling process proceeds as follows:

(1) Deterministically constructing a sample space by selecting top-ranked N documents for each query based on language model from the collection of documents.
(2) Deterministically constructing sampling units by k-NN clustering based on the similarities of documents in the sample space.
(3) Deterministically sampling clusters by selecting top-ranked M clusters based on the cluster-based language model. All the documents in the top M clusters are selected as feedback documents with redundancy which means one document can be selected more than twice.
(4) Deterministically sampling expansion terms by selecting top-ranked E terms based on the relevance model.

The following are the details of each step.

3.2.1. Constructing a sample space

First, documents are retrieved for a given query by the query-likelihood language model (Ponte & Croft, 1998) with Dirichlet smoothing (Zhai & Lafferty, 2004). The sample space consists of the top-retrieved N documents from the collection of documents. (In our experiments, the size of sample space N is set to 100.)

A statistical language model is a probabilistic distribution over all the possible word sequences for generating a piece of text. In information retrieval, the language model treats documents themselves as models and a query as strings of text generated from these document models. The popular query-likelihood retrieval model estimates document language models using the maximum likelihood estimator. The documents can be ranked by their likelihood of generating or sampling the query from document language models: $P(Q|D)$.

$$P(Q|D) = \prod_{i=1}^{m} P(q_i|D) \tag{1}$$

where q_i is the ith query term, m is the number of words 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.

$$P(w|D) = \frac{|D|}{|D|+\mu} P_{ML}(w|D) + \frac{\mu}{|D|+\mu} P_{ML}(w|Coll) \tag{2}$$

$$P_{ML}(w|D) = \frac{freq(w,D)}{|D|}, \quad P_{ML}(w|Coll) = \frac{freq(w,Coll)}{|Coll|} \tag{3}$$

where $P_{ML}(w|D)$ is the maximum likelihood estimate of word w in the document D, $Coll$ is the entire collection, and μ is the smoothing parameter. $|D|$ and $|Coll|$ are the lengths of a document D and collection C, respectively. $freq(w,D)$ and $freq(w,Coll)$ denote the frequency of a word w in D and $Coll$, respectively. The smoothing parameter is learned using training topics on each collection in experiments.

3.2.2. Constructing sampling units

Next, clusters are generated by the k nearest neighbors (k-NN) clustering method (Fix & Hodges, 1951) for documents in the sample space to find dominant documents. Here, a sampling unit is that cluster considered for selection in the next stage of sampling. Note that one document can belong to several clusters.

In k-NN clustering, each document plays a central role in making its own cluster with its k closest neighbors by similarity. We represent a document using *tf.idf* weighting and cosine normalization. The cosine similarity is used to calculate similarities among the top-retrieved documents.

Our hypothesis is that a dominant document may have several nearest neighbors with high similarities, participating in several clusters. On the other hand, a non-relevant document ideally makes a singleton cluster with no nearest neighbors with high similarity, though in practice it will have neighbors due to noise such as polysemous or general terms. Document clusters can also reflect the association of terms and documents from similarity calculation. In this work, if a document is a

member of several clusters and the clusters are highly related to the query, we assume it to be a dominant document. The cluster-based resampling method repeatedly uses such dominant documents based on document clusters.

3.2.3. Deterministically sampling clusters

After forming clusters as sampling units, the clusters are ranked by the cluster-based language model (Liu & Croft, 2004) described below. The top-ranked M clusters are selected. All the documents in the sampled clusters are used for feedback with redundancy. Note that clusters are used only for selecting feedback documents.

A cluster can be treated as a large document so that we can use the successful query-likelihood retrieval model. Intuitively, each cluster can be represented by just concatenating documents which belong to the cluster. If *Clu* represents such a cluster, then:

$$P(Q|Clu) = \prod_{i=1}^{m} P(q_i|Clu) \tag{4}$$

$$P(w|Clu) = \frac{|Clu|}{|Clu| + \lambda} P_{ML}(w|Clu) + \frac{\lambda}{|Clu| + \lambda} P_{ML}(w|Coll) \tag{5}$$

$$P_{ML}(w|Clu) = \frac{freq(w, Clu)}{|Clu|}, \quad P_{ML}(w|Coll) = \frac{freq(w, Coll)}{|Coll|} \tag{6}$$

where $freq(w, Clu)$ is sum of $freq(w, D)$ for the document D which belongs to the cluster *Clu*.

3.2.4. Sampling expansion terms

Finally, expansion terms are selected using the relevance model for each document in the sampled clusters. Note that the set of feedback documents chosen from the selected clusters are used to estimate the relevance model with their initial query-likelihood probabilities.

A relevance model is a query expansion approach based on the language modeling framework. Relevance models have been shown to be a powerful way to construct a query model from the top-retrieved documents (Diaz & Metzler, 2006; Lavrenko & Croft, 2001). The relevance model is a multinomial distribution which estimates the likelihood of a word w given a query Q. In the model, the query words $q_1 \ldots q_m$ and any word w in relevant documents are sampled identically and independently from a distribution R. Following that work, we estimate the probability of a word in the distribution R using

$$P(w|R) = \sum_{D \in R} P(D)P(w|D)P(Q|D) \tag{7}$$

where R is the set of documents that are pseudo-relevant to the query Q. We assume that $P(D)$ is uniform over the set.

After this estimation, the most likely e terms from $P(w|R)$ are deterministically selected as the expansion terms for an original query. The final expanded query is combined with the original query using linear interpolation, weighted by a parameter λ. The combining parameter is learned using training topics on each collection in the experiments.

The original relevance model and traditional pseudo-relevance feedback methods use the initial retrieval set to get expansion terms directly after the first step. The problem is that the top-retrieved documents contain non-relevant documents, which add noise to expansion terms. Our effort uses overlapping clusters to reuse dominant documents and to skip non-dominant documents for the query to emphasize good representative terms in dominant documents and to deemphasize terms in non-dominant documents. It may still find non-relevant documents, but we will show it finds fewer of them.

4. Experiments

To validate the proposed method, we performed experiments on five TREC collections and compared the results with a baseline retrieval model, a baseline feedback model, and an upper-bound model.

4.1. Experimental set-up

4.1.1. Test collections

We tested the proposed method on homogeneous and heterogeneous test collections: the ROBUST, AP and WSJ collections are smaller and contain newswire articles, whereas GOV2 and WT10G are larger web collections. For all collections, the topic title field is used as the query. A summary of the test collections is shown in Table 1.

Version 2.3 of the Indri system (Strohman, Metzler, Turtle, & Croft, 2005) is used for indexing and retrieval. All collections are stemmed using a Porter stemmer. A standard list of 418 common words is removed at retrieval time.

4.1.2. Training and evaluation

For each test collection, we divide topics into training topics and test topics, where the training topics are used for parameter estimation and the test topics are used for evaluation. The sample space, which is the initial retrieval set (the size is set to 100), is deterministically constructed by the language model for each query from the document collection.

In order to find the best parameter settings, we sweep over values of the smoothing parameter for the language model ($\mu \in \{500, 750, 1000, 1500, 2000, \ldots, 5000\}$), the number of feedback documents which can be deterministically selected from

Table 1
Training and test collections.

Collection	Description	# of documents	Topics	
			Train	Test
GOV2	2004 crawl of .gov domain	25,205,179	50 (701–750)	50 (751–800)
WT10g	TREC web collection	1,692,096	50 (451–500)	50 (501–550)
ROBUST	Robust 2004 collection: Financial Times, the Federal Register 94, the LA Times, and FBIS	528,155	150 (301–450)	100 (601–700)
AP	Associated Press 88–90	242,918	100 (51–150)	50 (151–200)
WSJ	Wall Street Journal 87–92	173,252	100 (51–150)	50 (151–200)

the sample space for the relevance model ($|R| \in \{5,10,25,50,75,100\}$), the number of expansion terms, which is the final result of drawing samples by the relevance model ($e \in \{10,25,50,75,100\}$), and the weight of the original query ($\lambda \in \{0.1,0.2,\ldots,0.9\}$). To train the proposed model, we sweep over the number of feedback clusters that can be deterministically selected by the cluster-based language model ($|C| \in \{1,2,5,10,15,20\}$), which corresponds to the number of feedback documents since one cluster can have at most five documents as a member ($k = 5$; the maximum size of a sampling unit is 5) in our k-nearest neighbors clustering. The threshold for clustering is set to 0.25. Expansion terms are represented using the following Indri query form:

$$\#weight(\lambda\ \#combine(q_1 \ldots q_m) \quad (1-\lambda)\#weight(p_1 t_1\ p_2 t_2 \ldots p_e t_e))$$

where $q_1 \ldots q_m$ are the original query terms, $t_1 \ldots t_e$ are the e terms with expansion probabilities $p_1 \ldots p_e$, and λ is a parameter combining the original query and the expanded query.

All comparison methods are optimized on the training set using mean average precision (MAP) defined as,

$$MAP = \frac{1}{|Q|}\sum_{q \in Q} AP(q) \tag{9}$$

where $AP(q)$ is (uninterpolated) average precision for a query q in the topic set Q.

The best parameters on training for each test collection shown in Table 2 are used for experimental results with the test topics.

4.1.3. Comparisons

4.1.3.1. Baselines. We provide two baselines: the language model and the relevance model.

- Language model (*LM*): The performance of the baseline retrieval model. The relevance model and the resampling method use this baseline result as the starting point for selecting documents for feedback and clustering.
- Relevance model (*RM*): The performance of the baseline pseudo-relevance feedback model. The expanded query is combined with the original query; this formulation of relevance modeling is commonly referred to as RM3 (Diaz & Metzler,

Table 2
Trained parameters for each collection.

Collection	Model	Smoothing parameter, μ	Original query weight, λ	Feedback documents, $\|R\|$	Feedback clusters, $\|C\|$	Expansion terms, e
GOV2	LM	1500	–	–	–	–
	RM	1500	0.6	10	–	50
	Resampling	1500	0.6	avg 24.14	5	100
WT10G	LM	1000	–	–	–	–
	RM	1000	0.7	50	–	10
	Resampling	1000	0.5	avg 8.95	2	25
ROBUST	LM	1000	–	–	–	–
	RM	1000	0.2	25	–	50
	Resampling	1000	0.3	23.01	5	50
AP	LM	2000	–	–	–	–
	RM	2000	0.2	50	–	75
	Resampling	2000	0.2	23.68	5	100
WSJ	LM	2000	–	–	–	–
	RM	2000	0.2	25	–	100
	Resampling	2000	0.3	45.98	10	100

2006). The resampling method is based on the relevance model framework. The difference is the pseudo-relevant documents used.

4.1.3.2. Upper-bound: true relevance feedback. To investigate the performance of the upper-bound of the proposed method, we compare with true relevance feedback.

- True relevance feedback (*TrueRF*): The performance using actual relevant documents in the top-retrieved 100 documents, where relevance is determined by the provided TREC judgments. This performance presents the upper-bound when using the relevance model.

4.1.3.3. A cluster-based reranking method. To provide the effectiveness of clusters for the initial retrieval set, we also include a cluster-based reranking method.

- Reranking using clusters (*Rerank*): The performance of reranking by combining query likelihoods for documents and clusters based on *k*-NN clusters for the top-retrieved *N* documents. *N* and *k* are set to 1000 and 5, respectively.

$$P'(Q|D) = P(Q|D) \cdot MAX_{D \in Clu_i} P(Q|Clu_i) \tag{10}$$

Since a *dominant document* can be a member of several clusters, we choose the maximum query likelihood for the clusters *Clu* which the document *D* belongs to. The cluster-based reranking method (Lee et al., 2004) based on the vector-space model has shown good results. Here we applied the reranking method to the language model.

We also compared the cluster-based language model (CBLM) by linear combination of Liu and Croft (2004).

$$P(w|D) = \lambda P_{ML}(w|D) + (1-\lambda)P_{ML}(w|Clu) = \lambda P_{ML}(w|D) + (1-\lambda)[\beta P_{ML}(w|Clu) + (1-\beta)P_{ML}(w|Coll)] \tag{11}$$

$$P_{ML}(w|Clu) = \frac{freq(w, Clu)}{|Clu|} \tag{12}$$

where $P_{ML}(w|D)$ is the maximum likelihood estimate of word *w* in the cluster *Clu*. $|Clu|$ is the length of a cluster *Clu*. Eq. (11) of the cluster-based language model corresponds to Eq. (2) of the language model.

The reranking method shows the effects of dominant documents without the feedback procedure.

4.2. Experimental results

The results for the comparison methods on five test collections are presented in Table 3. The *Resampling* method significantly outperforms *LM* on all test collections, whereas *RM* does not significantly outperform *LM* on WT10g collection. For GOV2 and WT10g heterogeneous web test collections, *Resampling* significantly outperforms *RM*. The relative improvements over *RM* are 6.28% and 19.63% on GOV2 and WT10g, respectively. For the ROBUST newswire collection, *Resampling* shows slightly lower performance than *RM*. For AP and WSJ newswire collections, *Resampling* shows small, but not significant improvements over *RM*. In the precision at 5 (P@5) evaluation metric as shown in Table 6, *Resampling* shows 14.8%, 24.7%, 3.9%, 20.0%, and 11.9% improvements over *LM*, whereas *RM* shows −7.1%, 7.4%, 1.6%, 18.8% and 7.4% improvement on GOV2, WT10g, ROBUST04, AP and WSJ collections, respectively.

The *Rerank* method using clusters shows significant improvements over *LM* on all test collections. In fact, *Rerank* outperforms *RM* on WT10g collection. The results indicate that document clustering can help find relevant document groups for the initial retrieval set and provide implicit document context to the query. The dominant documents play a central role in building relevant groups. As shown in Table 4, we conducted an experiment to compare the cluster-based language model by linear combination of Liu and Croft (2004) in Eq. (11) to show a strong baseline for cluster-based reranking. The proposed reranking method outperforms the reranking by linear combination.

TrueRF shows significant improvements over all methods on test collections. The results provide upper-bound performance on each collection, showing what might happen if we are able to choose better pseudo-relevant documents, approaching the set of true relevant documents.

Table 3
Performance comparisons using mean average precision for the test topics on test collections. The superscripts α, β, γ and δ indicate statistically significant improvements over LM, Rerank, RM and Resampling, respectively. We use the paired *t*-test with significance at $p < 0.05$.

Collection	LM	Rerank	RM	Resampling	TrueRF
GOV2	0.3258	0.3406^{α}	$0.3581^{\alpha\beta}$	$\mathbf{0.3806}^{\alpha\beta\gamma}$	$0.4315^{\alpha\beta\gamma\delta}$
WT10g	0.1861	0.2044^{α}	0.1966	$\mathbf{0.2352}^{\alpha\beta\gamma}$	$0.4030^{\alpha\beta\gamma\delta}$
ROBUST	0.2920	0.3206^{α}	$\mathbf{0.3591}^{\alpha\beta}$	$0.3515^{\alpha\beta}$	$0.5351^{\alpha\beta\gamma\delta}$
AP	0.2077	0.2361^{α}	$0.2803^{\alpha\beta}$	$\mathbf{0.2906}^{\alpha\beta}$	$0.4253^{\alpha\beta\gamma\delta}$
WSJ	0.3258	0.3611^{α}	$0.3967^{\alpha\beta}$	$\mathbf{0.4033}^{\alpha\beta}$	$0.5306^{\alpha\beta\gamma\delta}$

Table 4
Performance comparisons on combination methods in Rerank.

	LM	Rerank	
		Linear combination (Eq. (11))	Multiplication (Eq. (10))
GOV2	0.3258	0.3363	**0.3406**
WT10g	0.1861	0.1948	**0.2044**
ROBUST	0.2920	0.3181	**0.3206**
AP	0.2077	0.2303	**0.2361**
WSJ	0.3258	0.3555	**0.3611**

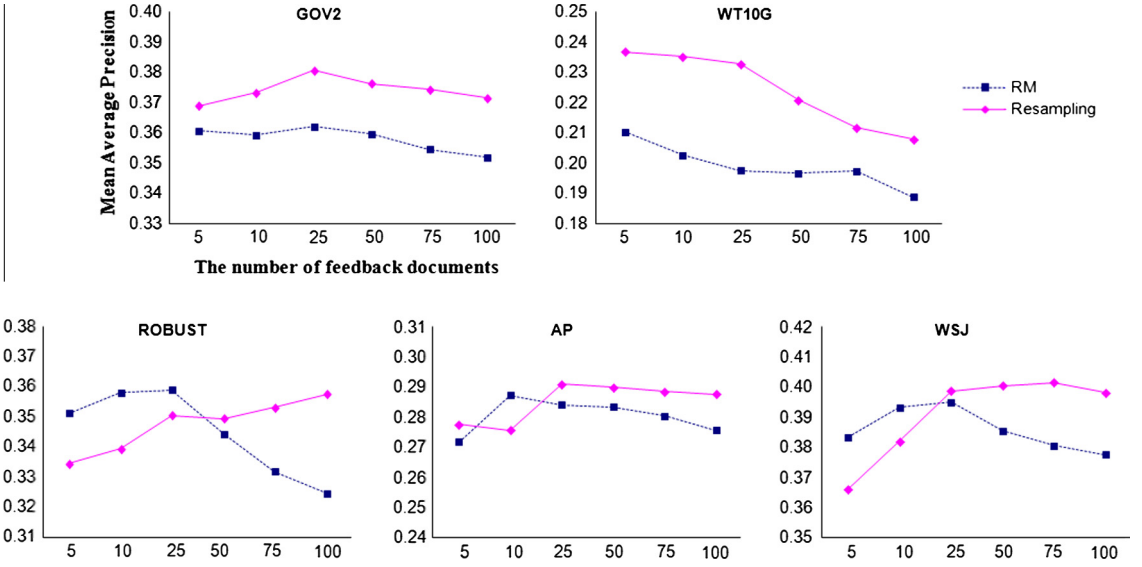

Fig. 1. Performance (mean average precision) on the test set for RM and Resampling according to the number of feedback documents.

We have also examined the effectiveness as the number of feedback documents varies. The best parameters (μ, e, and λ) learned on each collection are set for test queries since our efforts is on selecting pseudo-relevant documents. As shown in Fig. 1, *Resampling* achieves better performance over *RM* regardless of the number of feedback documents on GOV2 and WT10g collection. On ROBUST, AP and WSJ collection, *RM* shows better performance in 25 feedback documents, but *Resampling* shows better performance as the number of feedback documents increases.

Fig. 2 shows the performance of each query for *LM*, *RM*, *Resampling*, and *TrueRF*. TrueRF is an upper-bound performance of the relevance model with the true relevant documents contained in the top 100 retrieved documents. On GOV2 collection, *Resampling* shows relatively stable improvements over *RM* overall, although some queries perform worse using *Resampling* compared to *RM*. On WT10g collection, *Resampling* shows some variations in performance depending on queries. Basically, the performance of *Resampling* depends on the accuracy in clustering stage.

4.3. Retrieval robustness

We analyze the robustness of the relevance model and the resampling method over the baseline language model. Here, retrieval robustness is defined as the number of queries whose performance is improved (i.e. # of improved queries/# of total queries) and average effectiveness by the amount of improvement as the result of applying these methods (Metzler & Croft, 2007).

Overall, our resampling method improves the effectiveness over the language model for 82%, 61%, 63%, 66%, and 70% of the queries for GOV2, WT10g, ROBUST, AP and WSJ, respectively.

Fig. 3 presents an analysis of the robustness of the relevance model, the resampling method and true relevance feedback model. The resampling method shows strong robustness for GOV2. For the GOV2 collection, *Resampling* improves 41 queries and hurts 9, whereas *RM* improves 37 and hurts 13, and *TrueRF* improves 45 and hurts 5. For the WT10g and WSJ collection, although *RM* improves the performance of 2 more queries than *Resampling*, the amount of improvement in average precision for each query exhibited by *Resampling* is significantly larger for WT10g. For the ROBUST collection, *Resampling* improves 63 and hurts 36, whereas the relevance model improves 64 and hurts 35.

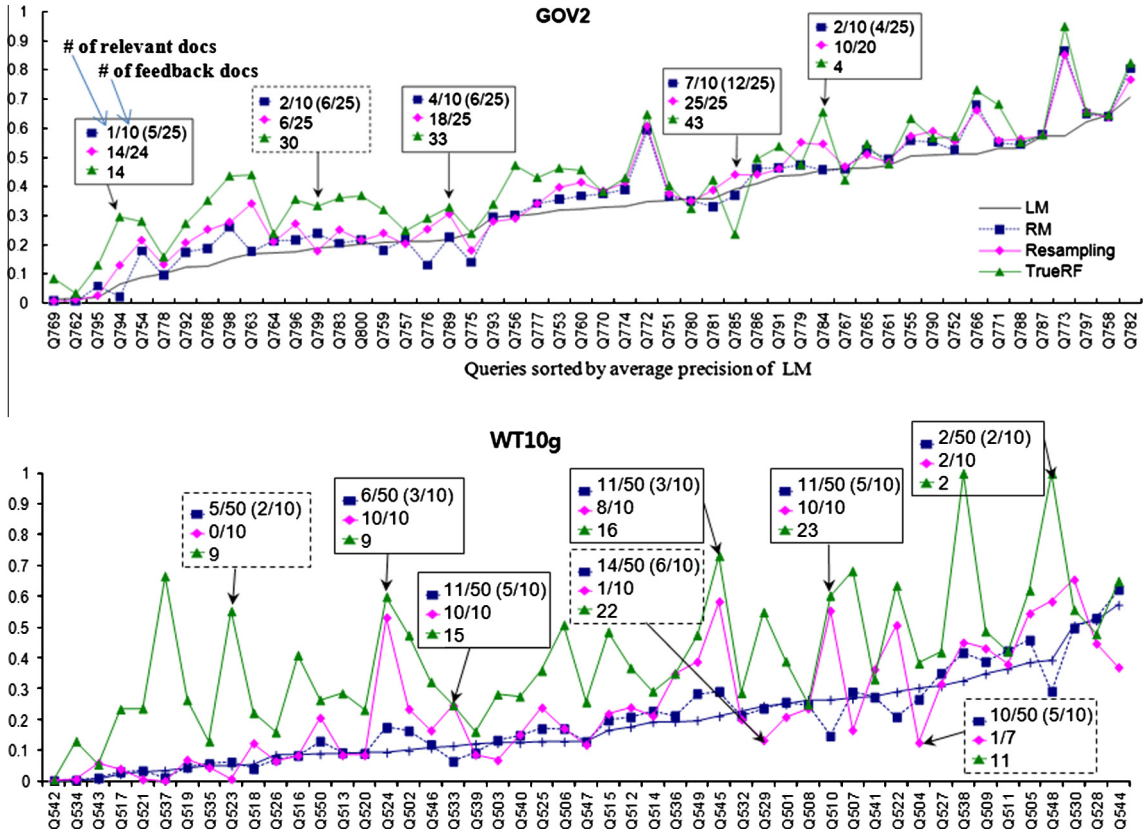

Fig. 2. Performance comparisons on each query sorted by average precision of LM. RM uses 10 feedback documents and Resampling uses at most 25 documents (in 5 clusters). On WT10g collection, *RM* uses 50 feedback documents and Resampling uses at most 10 documents (in 2 clusters).

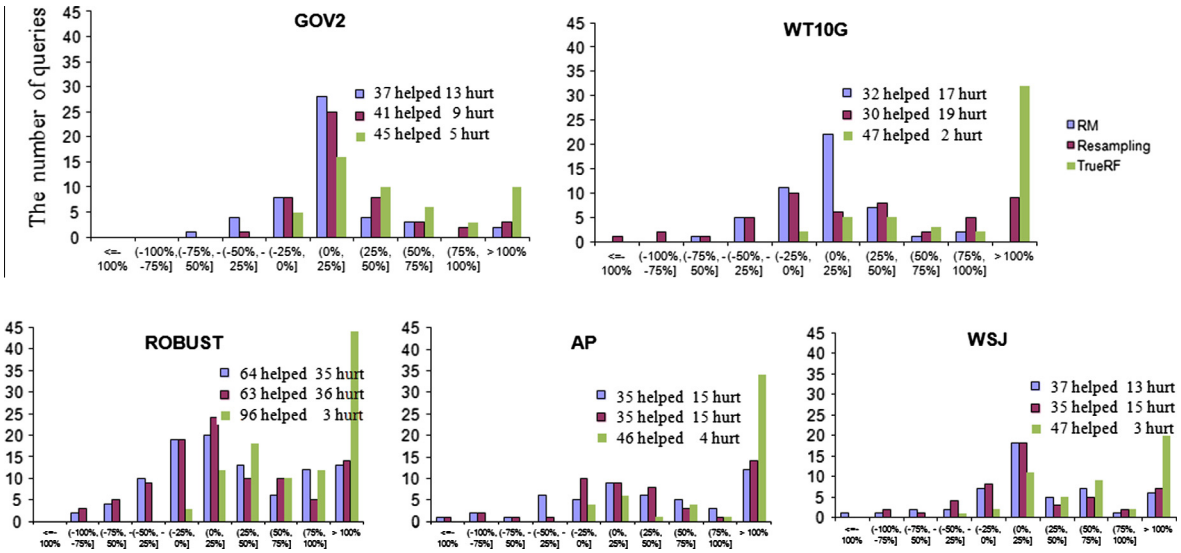

Fig. 3. Robustness of the relevance model and the resampling method over the language model. Each graph shows the number of queries that were hurt or helped by the *RM* or *Resampling* approaches.

The average effectiveness (i.e. performance improvement in average precision for each query) of overall queries of *RM* and *Resampling* over the language model are 12.45% and 23.09% on GOV2, 9.07% and 55.04% on WT10g, 39.28% and 30.96% on ROBUST, 35.65% and 36.23% on WSJ, and 48.79% and 81.09% on AP collection.

On GOV2, WT10g, and AP collections in the amount of improvement, our resampling method shows strong effectiveness, although it is less robust for the ROBUST collection. However, *RM* and *Resampling* do hurt some queries since the pseudo-relevance feedback model tends to sensitive to the initial retrieval result for a query. *Resampling* method showed higher variance in performance improvement than *RM* on WT10g since non-relevant documents can be repeatedly used for feedback in the cluster-based resampling method. It can be overcome by finding optimal clusters for a query.

5. Justification by relevance density

In this section, we aim to develop a deeper understanding of why expansion by the cluster-based resampling method helps. To justify the cluster-based resampling approach using overlapping clusters, we have analyzed relevance density with dominant documents and the performance of feedback without redundant documents.

The rationale for the proposed method is that resampling documents using clusters is an effective way to find dominant documents for a query from the initial retrieval set. We measure relevance density to justify our assumption that dominant documents are relevant to the query and redundantly appear over the top-ranked clusters.

Relevance density is defined to be the proportion of the feedback documents that contain relevant documents.

$$\text{Relevance density} = \frac{\text{the number of relevant feedback documents}}{\text{the number of feedback documents}} \tag{8}$$

A higher relevance density implies greater retrieval accuracy, ultimately approaching true relevance feedback.

If a resampling method is effective, it will produce higher relevance densities for pseudo-relevant documents than a set of top-retrieved documents. To justify the cluster-based resampling method, we will examine the relevance density of feedback documents through experimental analysis.

5.1. Relevance density of feedback documents

We can expect that higher relevance density produces higher performance since more relevant documents are used for feedback.

As shown in Fig. 4, the resampling method shows higher density compared to the relevance model on 100 feedback documents for all test collections. Relevance density for *TrueRF* is measured with all true relevant documents in the relevance judgment set.

When the number of feedback documents is set to 100, we can expect that the resampling method outperforms the relevance model since the resampling method uses more relevant documents for feedback.

To verify our expectation for density, we compared performance with the number of feedback documents and terms set to 100. The performance of feedback on fixed documents is shown in Table 5. The resampling method outperforms the

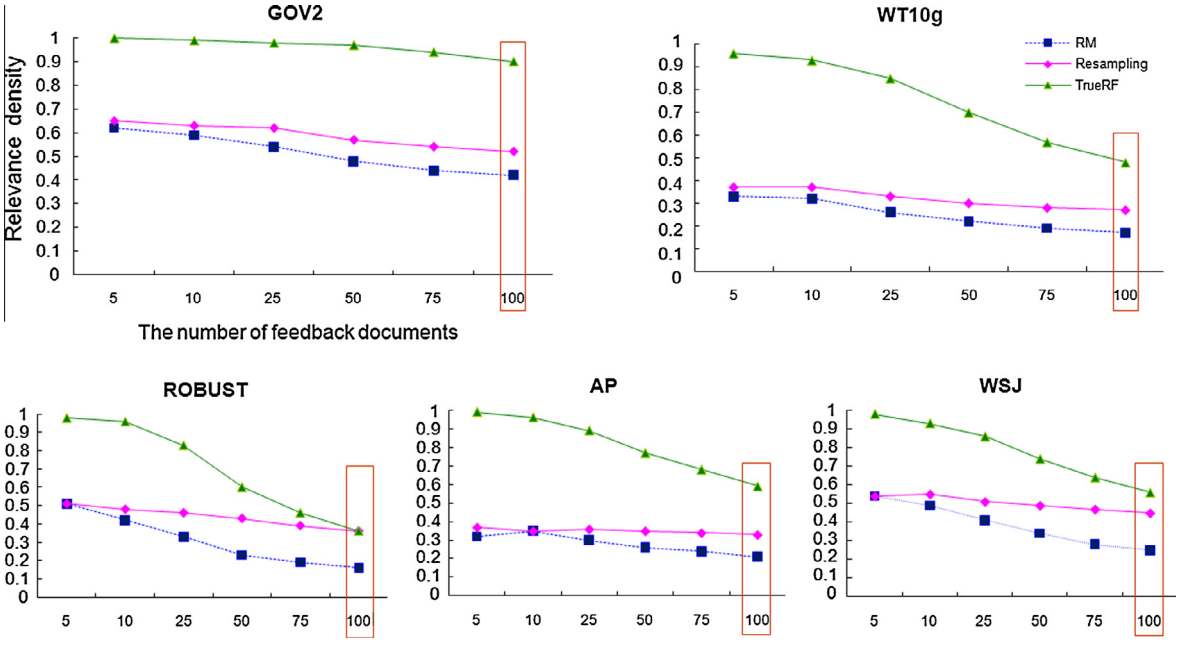

Fig. 4. Relevance density for *RM*, *Resampling*, and *TrueRF* according to the number of feedback documents. Relevance density for *TrueRF* is less than 100% because there are not always 100 documents judged relevant for a query.

Table 5
Performance on fixed feedback documents and terms. The number of feedback documents and terms are both set to 100. The superscripts α and β indicate statistically significant improvements over LM and RM, respectively. We use the paired t-test using significance at $p < 0.05$.

Collection	LM	RM	*chg%*	Resampling	*chg%*
GOV2	0.3258	0.3519^{α}	*8.01*	$\mathbf{0.3764}^{\alpha\beta}$	*15.53*
WT10g	0.1861	0.1886	*1.34*	$\mathbf{0.2072}^{\alpha}$	*11.34*
ROBUST	0.2920	0.3262^{α}	*11.71*	$\mathbf{0.3549}^{\alpha\beta}$	*21.54*
AP	0.2077	0.2758^{α}	*32.79*	$\mathbf{0.2853}^{\alpha}$	*37.36*
WSJ	0.3258	0.3785^{α}	*16.18*	$\mathbf{0.4009}^{\alpha\beta}$	*23.05*

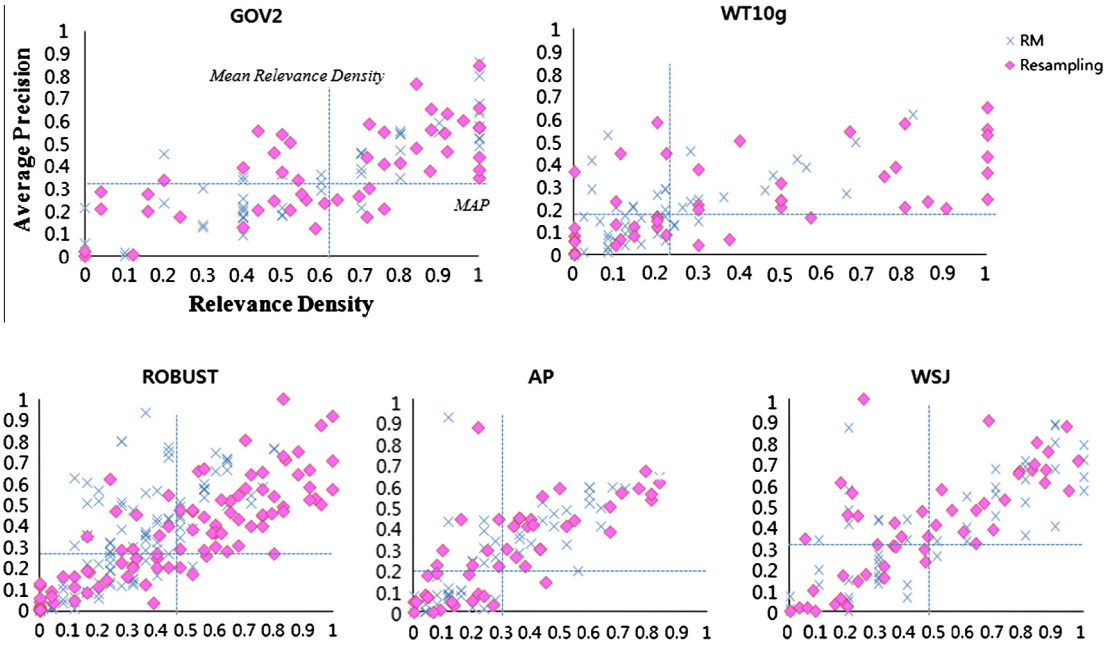

Fig. 5. The distribution of relevance density vs. average precision for each query.

relevance model for all collections. The results show that the density of relevant documents supports the improvements from the resampling approach which extracts better feedback documents from the top-ranked 100 documents.

From the results of density as related to the number of feedback documents and effectiveness on all collections, we can see that the redundant dominant documents help the density of the relevant documents.

Table 5 shows performance on fixed feedback documents and terms. *Resampling* outperforms *RM* for all collections. Resampling takes better feedback documents (with higher relevance density) from the top-ranked 100 documents.

Fig. 5 shows the distribution of relevance density vs. average precision on test queries for all test collections. The line labeled MAP (mean average precision) is the score of the language model on each collection. When relevance density is higher than *Mean Relevance Density*, it tends to have higher average precision than MAP for all test collections. The distribution shows the general pattern that higher relevance density produces higher average precision though some queries with low relevance density show high average precision for all test collections.

5.2. Redundancy in feedback documents

To support the observation of relevance density and performance, we have examined performance by removing redundant documents in feedback. That is, a document is not repeated in the feedback even if it occurs in multiple clusters.

We assumed that dominant documents for the initial retrieval set are relevant and redundant documents that play a central role in making overlapping clusters. Table 6 shows the performance of *Sampling without Replacement*. It outperforms *RM*, but is worse than *Resampling* for GOV2, WT10g, AP, and WSJ collection except for ROBUST collection. The results show that dominant documents give positive effects for feedback.

We examined how many documents are redundantly used for feedback for each query.

$$\text{Redundancy ratio} = 1 - \frac{\text{the number of unique documents}}{\text{the number of feedback documents}} \tag{9}$$

Table 6
The effect of dominant documents for feedback.

Collection	Model	MAP	chg%	P@5	chg%	P@10	chg%
GOV2	LM	0.3258	–	0.6200	–	0.5860	–
	Rerank	0.3406	4.54	0.6720	8.39	0.6340	8.19
	RM	0.3581	9.91	0.5760	−7.1	0.5960	1.71
	Sampling without replacement	0.3745	14.95	0.6760	9.03	0.6280	7.17
	Resampling	**0.3806**	16.82	0.7120	14.84	0.6600	12.63
WT10G	LM	0.1861	–	0.3306	–	0.3204	–
	Rerank	0.2044	9.83	0.4041	22.23	0.3429	7.02
	RM	0.1966	5.64	0.3551	7.41	0.3286	2.56
	Sampling without replacement	0.2193	17.84	0.3957	19.69	0.3468	8.24
	Resampling	**0.2352**	26.38	0.4122	24.68	0.3531	10.21
ROBUST	LM	0.2920	–	0.5152	–	0.4293	–
	Rerank	0.3206	9.79	0.5434	5.47	0.4707	9.64
	RM	**0.3591**	22.98	0.5232	1.55	0.4657	8.48
	Sampling without replacement	*0.3560*	21.92	0.5273	2.35	0.4707	9.64
	Resampling	*0.3515*	20.38	0.5354	3.92	0.4657	8.48
AP	LM	0.2077	–	0.3200	–	0.3460	–
	Rerank	0.2361	13.67	0.3920	22.50	0.3660	5.78
	RM	0.2803	34.95	0.3800	18.75	0.3640	5.20
	Sampling without replacement	0.2834	36.45	0.3520	10.00	0.3620	4.62
	Resampling	**0.2906**	39.91	0.3840	20.00	0.3940	13.87
WSJ	LM	0.3258	–	0.5400	–	0.4860	–
	Rerank	0.3611	10.83	0.5760	6.67	0.5280	8.64
	RM	0.3967	21.76	0.5800	7.41	0.5400	11.11
	Sampling without replacement	0.3920	20.32	0.5800	7.41	0.5280	8.64
	Resampling	**0.4033**	23.79	0.6040	11.85	0.5340	9.88

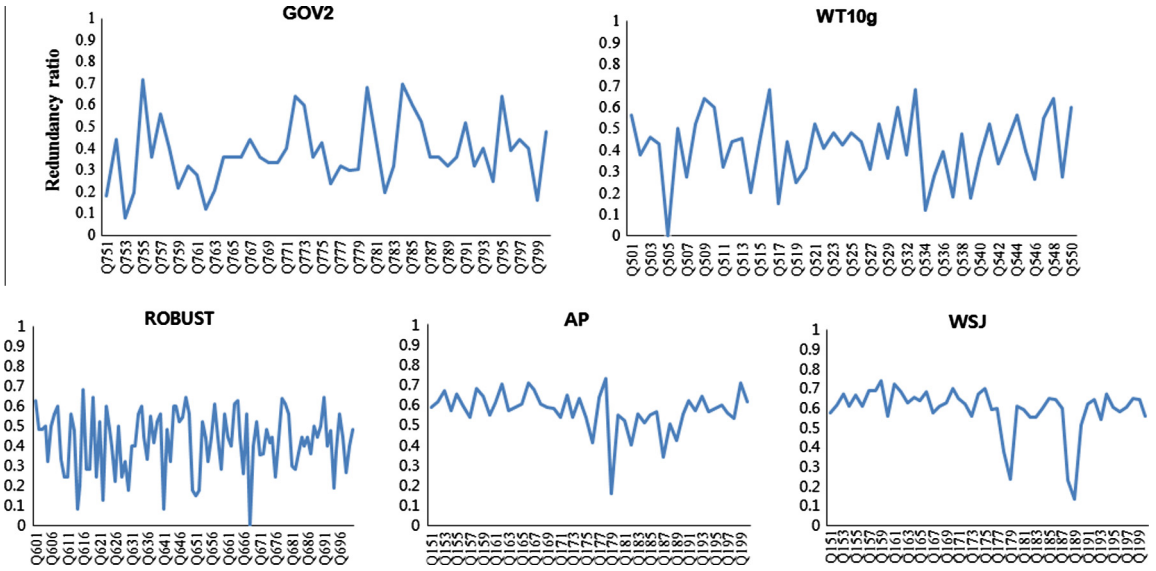

Fig. 6. Redundancy ratio of feedback documents for each query.

For example, the unique documents are six among ten feedback documents which means four documents redundantly appear in more than one clusters. Then redundancy ratio is 0.4.

Fig. 6 shows that redundancy ratio for overall queries are high for all test collections. The average redundancy ratio is 0.38, 0.41, 0.42, 0.57, and 0.59 on GOV2, WT10g, ROBUST, AP and WSJ, respectively.

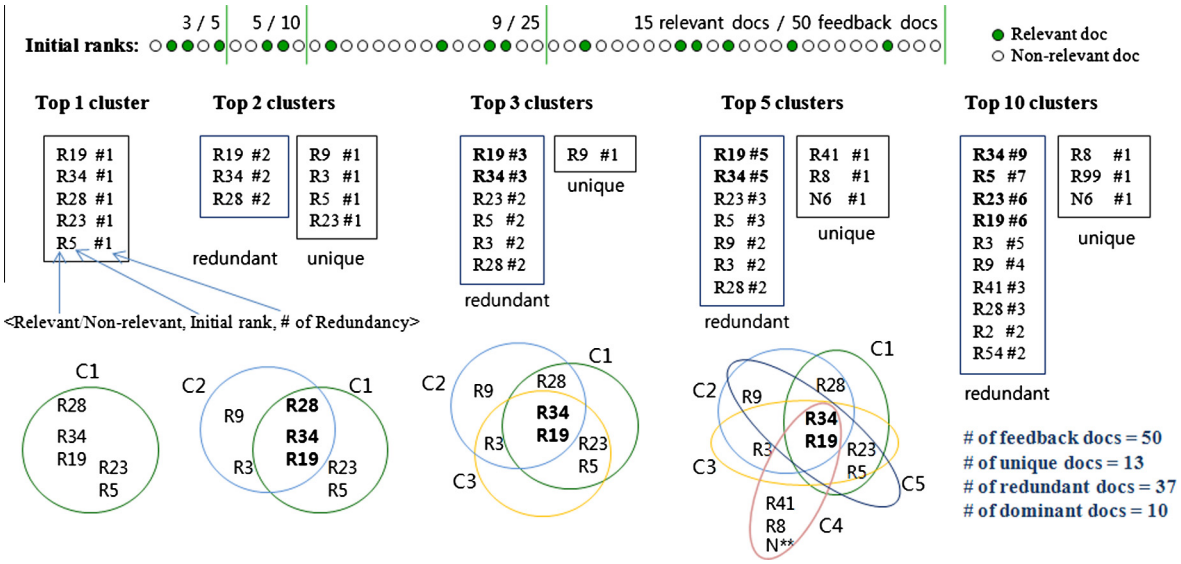

Fig. 7. The illustration of the level of redundancy for a query on WT10g.

5.3. Result analysis: Inside the clusters sampled for a example query

We have also examined inside the clusters how redundancy affects the number of relevant documents in the feedback sample for a query in WT10g shown in Fig. 7. The format of a document in the cluster is as follows: <Relevant/Non-relevant, initial rank, # of redundancy>. For example, 'R34 #9' describes 'rank by LM is 34, relevant, and appeared 9 times for the top 10 clusters, and 'N' in 'N6 #1' means non-relevant. If we look at the top 5, 10, 25, 50, 75, and 100 documents, we find the following. For the *RM* approach, the relevance density was 0.6, 0.5, 0.36, 0.3, and 0.23, respectively. This means that initial retrieved results include 23 relevant documents for the top 100 documents. For *Resampling*, however, the relevance densities were almost perfect: 1.0, 1.0, 0.96, 0.98, 0.97, and 0.89, respectively. To illustrate the level of redundancy, consider the query where the top 10 clusters contained 50 documents, 49 of which were relevant: 37 of those relevant documents appeared in other clusters. One relevant document appeared in nine of the top 10 clusters and another was in seven. The ten dominant documents are sampled 47 times repeatedly for feedback. The three non-dominant documents are sampled 3 times.

Such dominant documents that appear in multiple highly-ranked clusters and their redundancy contribute to query expansion terms.

6. Conclusions

Resampling the top-ranked documents using clusters is effective for pseudo-relevance feedback. All the procedures of the proposed method for constructing the sample space, building sampling units, sampling clusters, and sampling expansion terms are deterministic. The improvements obtained were consistent across nearly all test collections, and for large web collections, such as GOV2 and WT10g, the approach showed substantial gains. The relative improvements on GOV2 collection are 16.82% and 6.28% over LM and RM, respectively. The improvements on the WT10g collection are 19.63% and 26.38% over LM and RM, respectively. We showed that the relevance density was higher than the baseline feedback model for all test collections as a justification of why expansion by the cluster-based resampling method helps. Experimental results also show that deterministically resampling clusters are helpful for identifying pseudo-relevant documents for a query.

For future work, we will study how the resampling approach can adopt query variants by considering query characteristics. Additionally, in our experiments we simply represent a cluster by concatenating documents. Using a better representation of a cluster to improve cluster ranking and combining weak learners based on several cluster representations for boosting should improve the performance of pseudo-relevance feedback without sacrificing the performance of individual queries.

Acknowledgements

This research was supported in part by Basic Science Research Program through the National Research Foundation of Korea (NRF) grant funded by the Ministry of Education, Science and Technology (MEST) (611-2006-1-D00025), and by the

Center for Intelligent Information Retrieval. Any opinions, findings and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the sponsor.

References

Allan, J. (1995). Relevance feedback with too much data. In *Proc. 18th ACM SIGIR conference on research and development in information retrieval* (pp. 337–343).

Attar, R., & Fraenkel, A. S. (1977). Local feedback in full-text retrieval systems. *Journal of the ACM, 24*(3), 397–417.

Buckley, C. (2009). *Relevance feedback 2009: Overview. Presentation at TREC 2009 workshop, unpublished*.

Buckley, C., & Harman, D. (2004). *Reliable information access final workshop report*. <http://nrrc.mitre.org/NRRC/publications.htm>.

Buckley, C., Mitra, M., Walz, J., & Cardie, C. (1998). Using clustering and superconcepts within SMART: TREC 6. In *Proc. 6th text retrieval conference (TREC-6)*.

Buckley, C., & Robertson, S. (2008). *Proposal for relevance feedback 2008 track*. <http://groups.google.com/group/trec-relfeed>.

Buckley, C., Salton, G., Allan, J., & Singhal, A. (1995). Automatic query expansion using SMART: TREC 3. In *Proc. 3rd text retrieval conference (TREC-3)* (pp. 69–80).

Collins-Thompson, K., & Callan, J. (2007). Estimation and use of uncertainty in pseudo-relevance feedback. In *Proc. 30th ACM SIGIR conference on research and development in information retrieval* (pp. 303–310).

Croft, W. B., & Harper, D. J. (1979). Using probabilistic models of document retrieval without relevance information. *Journal of Documentation, 35*(4), 285–295.

Diaz, F. (2005). Regularizing ad hoc retrieval scores. In *Proc. 14th ACM international conference on information and knowledge management (CIKM)* (pp. 672–679).

Diaz, F., & Metzler, D. (2006). Improving the estimation of relevance models using large external corpora. In *Proc. 29th ACM SIGIR conference on research and development in information retrieval* (pp. 154–161).

Efron, B. (1979). Bootstrap methods: Another look at the jackknife. *The Annals of Statistics, 7*, 1–26.

Fix, E., & Hodges, L. (1951). *Discriminatory analysis: Nonparametric discrimination: Consistency properties. Technical report*. USAF School of Aviation Medicine, Randolph Field, Texas, Project 21-49-004.

Freund, Y. (1990). Boosting a weak learning algorithm by majority. In *Proc. 3rd annual workshop on computational learning theory*.

Jardine, N., & Rijsbergen, C. J. V. (1971). The use of hierarchic clustering in information retrieval. *Information Storage and Retrieval, 7*, 217–240.

Kaptein, R., Kamps, J., & Hiemstra, D. (2008). The impact of positive, negative and topical relevance feedback. In *Proc. 17th text retrieval conference (TREC 2008)*.

Kozorovitzky, A. K., Kurland, O. (2011). Cluster-based fusion of retrieved lists. In *Proc. ACM SIGIR conference on research and development in information retrieval* (pp. 893–902).

Kurland, O., & Lee, L. (2004). Corpus structure, language models, and ad hoc information retrieval. In *Proc. 27th ACM SIGIR conference on research and development in information retrieval* (pp. 194–201).

Kurland, O., & Lee, L. (2005). Better than the real thing? Iterative pseudo-query processing using cluster-based language models. In *Proc. 28th ACM SIGIR conference on research and development in information retrieval* (pp. 19–26).

Kurland, O., & Lee, L. (2006). Respect my authority! HITS without hyperlinks, utilizing cluster-based language models. In *Proc. 29th ACM SIGIR conference on research and development in information retrieval* (pp. 83–90).

Kalmanovich, I. G., & Kurland, O. (2009). Cluster-based query expansion. In *Proc. 32nd ACM SIGIR conference on research and development in information retrieval* (pp. 646–647).

Lavrenko, V., & Croft, W. B. (2001). Relevance-based language models. In *Proc. 24th ACM SIGIR conference on research and development in information retrieval* (pp. 120–127).

Lee, K. S., Croft, W. B., & Allan, J. (2008). A cluster-based resampling method for pseudo-relevance feedback. In *Proc. 31st ACM SIGIR conference on research and development in information retrieval* (pp. 235–242).

Lee, K. S., Park, Y. C., & Choi, K. S. (2001). Re-ranking model based on document clusters. *Information Processing and Management, 37*, 1–14.

Lee, K. S., Kageura, K., & Choi, K. S. (2004). Implicit ambiguity resolution based on cluster analysis in cross-language information retrieval. *Information Processing and Management, 40*, 145–159.

Liu, X., & Croft, W. B. (2004). Cluster-based retrieval using language models. In *Proc. 27th ACM SIGIR conference on research and development in information retrieval* (pp. 186–193).

Lv, Y., & Zhai, C. (2008). A study of adaptive relevance feedback – UIUC TREC2008 relevance feedback experiments. In *Proc. 17th text retrieval conference (TREC'08)*.

Lv, Y., Zhai, C., & Chen, W. (2011). A boosting approach to improving pseudo-relevance feedback. In *Proc. 34st ACM SIGIR conference on research and development in information retrieval* (pp. 165–174).

Lynam, T., Buckley, C., Clarke, C., & Cormack, G. (2004). A multi-system analysis of document and term selection for blind feedback. In *Proc. 13th ACM international conference on information and knowledge management (CIKM'04)* (pp. 261–269).

Metzler, D., & Croft, W. B. (2007). Latent concept expansion using Markov random fields. In *Proc. 30th ACM SIGIR conference on research and development in information retrieval* (pp. 311–318).

Ponte, J. M., & Croft, W. B. (1998). A language modeling approach to information retrieval. In *Proc. 21st ACM SIGIR conference on research and development in information retrieval* (pp. 275–281).

Robertson, S. E., Walker, S., Beaulieu, M., Gatford, M., & Payne, A. (1996). Okapi at TREC-4. In *Proc. 4th text retrieval conference (TREC)*.

Sakai, T., Manabe, T., & Koyama, M. (2005). Flexible pseudo-relevance feedback via selective sampling. *ACM Transactions on Asian Language Information Processing (TALIP), 4*(2), 111–135.

Salton, G., & Buckley, C. (1990). Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science, 41*(4), 288–297.

Schapire, R. (1990). Strength of weak learnability. *Journal of Machine Learning, 5*, 197–227.

Strohman, T., Metzler, D., Turtle, H., & Croft, W. B. (2005). Indri: A language model-based search engine for complex queries. In *Proc. international conference on intelligence analysis*.

Tao, T., & Zhai, C. (2006). Regularized estimation of mixture models for robust pseudo-relevance feedback. In *Proc. 29th ACM SIGIR conference on research and development in information retrieval* (pp. 162–169).

TREC (2008). In *Proc. 17th text retrieval conference (TREC 2008)*.

TREC (2009). In *Proc. 18th text retrieval conference (TREC 2009)*.

Tsegay, Y., Scholer, F., & Puglisi, S. (2008). RMIT University at TREC 2008: Relevance feedback track. In *Proc. 17th text retrieval conference (TREC 2008)*.

Vishkin, U. (1991). Deterministic sampling – A new technique for fast pattern matching. *SIAM Journal of Computing, 20*(1), 22–40.

Xu, J., & Croft, W. B. (1996). Query expansion using local and global document analysis. In *Proc. 19th ACM SIGIR conference on research and development in information retrieval* (pp. 4–11).

Yang, L., Ji, D., Zhou, G., Nie, Y., & Xiao, G. (2006). Document re-ranking using cluster validation and label propagation. In *Proc. 15th ACM international conference on information and knowledge management (CIKM)* (pp. 690–697).

Yershova, A., & LaValle, S. M. (2004). Deterministic sampling methods for spheres and SO(3). In *Proc. IEEE international conference on robotics and automation* (pp. 3974–3980).

Yershova, A., Jain, S., & LaValle, S. M. (2010). Generating uniform incremental grids on SO (3) using the Hopf fibration. *The International Journal of Robotics Research, 29*(7), 801–812.
Yeung, D. L., Clarke, C. L. A., Cormack, G. V., Lynam, T. R., & Terra, E. L. (2004). Task-specific query expansion. In *Proc. 12th text retrieval conference (TREC)* (pp. 810–819).
Zhang, B., Li, H., Liu, Y., Ji, L., Xi, W., Fan, W., et al. (2005). Improving web search results using affinity graph. In *Proc. 28th ACM SIGIR conference on research and development in information retrieval* (pp. 504–511).
Zhai, C., & Lafferty, J. (2004). A study of smoothing methods for language models applied to information retrieval. *ACM Transactions on Information Systems, 22*(2), 179–214.
Zhao, L., Liang, C., & Callan, J. (2008). Extending relevance model for relevance feedback. In *Proc. 17th text retrieval conference (TREC 2008).*