Two are better than one! Re-ranking search results using an additional retrieved list

Lior Meister, Oren Kurland, and Inna Gelfer Kalmanovich
Faculty of Industrial Engineering and Management
Technion — Israel Institute of Technology
meister@tx.technion.ac.il,kurland@ie.technion.ac.il,innagel@tx.technion.ac.il

ABSTRACT

We present a novel approach to re-ranking a list that was retrieved in response to a query so as to improve precision at the very top ranks. The approach is based on utilizing a second list that was retrieved by using, for example, a different retrieval method and/or query representation. In contrast to commonly-used methods for *fusion* of retrieved lists, our approach also exploits *inter-document-similarities* between the lists — a potentially rich source of additional information. Empirical evaluation shows that our methods are effective in re-ranking a high quality TREC *run* using a second high-quality run; the resultant performance also favorably compares with that of a state-of-the-art fusion method. Furthermore, we show that our methods can help to effectively tackle two long-standing challenges; namely, integration of document-based and cluster-based retrieved results; and, improvement of the *performance robustness*, and overall effectiveness, of pseudo-feedback-based retrieval.

Categories and Subject Descriptors: H.3.3 [Information Search and Retrieval]: Retrieval models

General Terms: Algorithms, Experimentation

Keywords: ad hoc retrieval, re-ranking, inter-document-similarities, cluster-based retrieval, robust pseudo-feedback-based retrieval

1. INTRODUCTION

Attaining high precision at the very top ranks of results returned in response to a query is an important challenge that search engines have to address. To that end, researchers have proposed, among others, a *re-ranking* paradigm: automatically re-ordering the documents in an initially retrieved list so as to improve precision at top ranks [47, 25, 34, 19, 28]. The motivation is based on the fact that the ratio of relevant to non-relevant documents in the initial list is often much larger than that in the entire corpus.

We present a novel approach to re-ranking an initially retrieved list. Our approach is based on utilizing information induced from a second list that is retrieved in response to the query by using, for example, a different retrieval method and/or query representation. Specifically, we exploit information induced from similarities between documents in the two lists so as to re-rank the initial list.

Indeed, it has long been acknowledged that *fusing* retrieved lists — i.e., conceptually combining "experts' recommendations" — is quite effective for retrieval [22, 32, 16, 3, 38, 6]. Many fusion methods "reward" documents that are highly ranked in many of the lists. The effectiveness of this principle is often attributed to the fact that there is high overlap between relevant documents in the lists and low overlap between non-relevant documents [32]. However, it turns out that on many occasions, the lists to be fused contain different relevant documents, and, hence, fusion methods tend to fall short in such cases [18, 23, 42, 5].

Our models address this potential relevant-set "mismatch" by letting similar — but not necessarily identical — documents to provide *relevance-status* support to each other. Case in point, similar documents can potentially be viewed as discussing the same topics. Specifically, if relevant documents are assumed to be similar following the *cluster hypothesis* [44], then they can "support" each other via inter-document similarities.

Thus, the basic principle underlying our methods — inspired by work on re-ranking a list using inter-document similarities within that list [19, 28] — is as follows. We reward documents in the initial list that are highly ranked, and that are similar to documents that are highly ranked in the second list.

Our models are shown to be effective in re-ranking a high quality TREC *run*, using a second high-quality run; the performance also favorably compares with that of a state-of-the-art fusion method used to integrate the two runs.

As it turns out, our models are also effective in addressing two long-standing challenges. The first is the integration of results produced by standard document-based retrieval with those produced by *cluster-based* retrieval [24, 14]. It was observed [23] that using the two retrieval paradigms yields result-lists with different sets of relevant documents — a scenario motivating the development of our models. Indeed, using our methods to re-rank the result-list of document-based retrieval using that of cluster-based retrieval yields performance that is superior to that of using each alone. Furthermore, the performance is also superior to that of a state-of-the-art fusion method that integrates the two lists; and, to that of a state-of-the-art re-ranking method that operates only on the document-based retrieved list.

The second challenge that we address is improving the *performance robustness* (and overall effectiveness) of pseudo-

feedback-based query-expansion [37, 2, 17, 12, 33]. While pseudo-feedback-based retrieval methods improve retrieval effectiveness on average, the performance for many queries is substantially lower than that of using the original query rather than the expanded one. A major cause of this performance robustness problem is *query drift* [37] — the change in underlying intention between the original query and its expanded form. We re-rank the retrieval results of a state-of-the-art pseudo-feedback-based method using those retrieved in response to the original query. The idea is to reward documents that are highly ranked when using pseudo-feedback, and, which "preserve" the original-query "intent" by the virtue of being similar to documents highly ranked in response to the original query. Our methods yield performance that is better, and more robust, than that of using pseudo-feedback-based retrieval alone; and, than that of using a state-of-the-art fusion method for this task.

2. RETRIEVAL FRAMEWORK

We use q and d to denote a query and a document, respectively. Our goal is to re-rank an initial list of documents, $\mathcal{L}_{init}$, which was retrieved in response to q by some search algorithm that was run over a given corpus, so as to improve precision at top ranks. To that end, we assume that a second list, $\mathcal{L}_{help}$, was also retrieved in response to q using, for example, a different search algorithm or/and query representation. Since $\mathcal{L}_{init}$ and $\mathcal{L}_{help}$ could be considered as the outputs of two classifiers [16], information induced from $\mathcal{L}_{help}$ can potentially help in re-ranking $\mathcal{L}_{init}$.

To indicate that d is a member of list $\mathcal{L}$, we write $d \in \mathcal{L}$; we use $Score_{\mathcal{L}}(d)$ to denote d's (non-negative) retrieval score in $\mathcal{L}$. (For $d \notin \mathcal{L}$, $Score_{\mathcal{L}}(d) \stackrel{def}{=} 0$.) The methods we present use a measure $sim(x, y)$ of the similarity between texts x and y; we describe the measure in Section 5. We will also make use of Kronecker's delta function: for argument s, $\delta[s] = 1$ if s holds, and 0 otherwise.

2.1 Similarity-based re-ranking

One potential way to re-rank $\mathcal{L}_{init}$ using $\mathcal{L}_{help}$ is by rewarding documents that are highly ranked in $\mathcal{L}_{init}$, and that are also positioned at high ranks of $\mathcal{L}_{help}$. Indeed, such ranking principle underlies most standard approaches for *fusion* of retrieved lists [22, 32, 16]. However, there is evidence that different retrieved lists might contain different relevant documents [42]. This observation holds whether the lists are retrieved in response to different query representations [18], or produced by different retrieval algorithms using the same query representation [23]. Hence, standard fusion methods can fall short in these cases.

To address this challenge, we can exploit a potentially rich source of information not utilized by current fusion methods, namely, *inter-document similarities*. Case in point, if the top-ranked document in $\mathcal{L}_{init}$ is not the same document as the top-ranked one in $\mathcal{L}_{help}$, but their content overlaps to a major extent, then the latter can potentially provide "relevance-status" support to the former. More generally, documents that are highly ranked in $\mathcal{L}_{help}$, and hence, are presumed to be relevant by the ranking method that created $\mathcal{L}_{help}$, can provide relevance-status support to those documents in $\mathcal{L}_{init}$ that they are similar to.

Thus, the following (re-)ranking principle for documents in $\mathcal{L}_{init}$ emerges. A document in $\mathcal{L}_{init}$ should be ranked high if it is (i) initially highly ranked in $\mathcal{L}_{init}$, and (ii) (very) similar to (many) documents that are highly ranked in $\mathcal{L}_{help}$.

The (qualitative) relevance-scoring principle just described conceptually generalizes the one underlying standard fusion approaches. That is, if we deem two documents "similar" if and only if they are the same document, then a document is presumed to be relevant if it is highly ranked in both retrieved lists. Furthermore, this relevance-scoring principle is conceptually a generalization of recently-proposed approaches for re-ranking a list based on inter-document similarities within the list [4, 19, 28]. Such methods reward documents in the list that are both highly ranked and highly similar to many other (highly ranked) documents. Thus, if $\mathcal{L}_{help}$ is set to be $\mathcal{L}_{init}$, our relevance-scoring principle echoes these approaches.

2.2 Algorithms

Following some previous work on re-ranking search results [19, 28], we let document d_h in $\mathcal{L}_{help}$ to provide relevance-status support only to documents in $\mathcal{L}_{init}$ that are most similar to it — i.e., its nearest neighbors in the similarity space. Formally, we use $Neighbors(d_h; \alpha)$ to denote the set of α documents d in $\mathcal{L}_{init}$ that yield the highest $sim(d_h, d)$; α is a free parameter. (Ties are broken by document ID.)

Thus, document d in $\mathcal{L}_{init}$ gets relevance-status support from its set of *supporters*: the documents in $\mathcal{L}_{help}$ that it is among the nearest-neighbors of; formally, $supporters(d) \stackrel{def}{=} \{d_h \in \mathcal{L}_{help} : d \in Neighbors(d_h; \alpha)\}$.

We can now quantify the above-stated relevance-scoring principle. That is, we reward document d in $\mathcal{L}_{init}$ if it is initially highly ranked, and if its supporters are highly ranked in $\mathcal{L}_{help}$ and provide d with a large extent of support; we use the inter-document similarity estimate to measure the level of support. The resultant algorithm, **SimRank**, then scores d by[1]:

$$Score_{SimRank}(d) \stackrel{def}{=} Score_{\mathcal{L}_{init}}(d) \sum_{d_h \in supporters(d)} Score_{\mathcal{L}_{help}}(d_h) sim(d_h, d). \quad (1)$$

Note that documents in $\mathcal{L}_{init}$ that appear in $\mathcal{L}_{help}$ receive self support by the virtue of self similarity. We might want, however, to further reward these documents as work on fusion of retrieved lists has demonstrated the merits in doing so [22, 32]. Therefore, inspired by the CombMNZ fusion method [22, 32], we also consider the **SimMNZRank** algorithm that doubles the score of d if it appears in $\mathcal{L}_{help}$:[2]

$$Score_{SimMNZRank}(d) \stackrel{def}{=} (\delta[d \in \mathcal{L}_{init}] + \delta[d \in \mathcal{L}_{help}]) Score_{SimRank}(d). \quad (2)$$

[1]Note that normalizing retrieval scores using the *sum* of scores so as to yield probability distributions has no effect on the induced ranking. Furthermore, experiments show that normalizing the inter-document-similarities values does not result in performance improvements.

[2]Note that $\delta[d \in \mathcal{L}_{init}] = 1$ since $d \in \mathcal{L}_{init}$; "MNZ" stands for "multiply non zero similarities" — i.e., non-zero retrieval scores, which in our case is either 1 or 2.

3. APPLICATIONS

The proposed algorithms can be used with any two lists that are retrieved in response to q. Indeed, in Section 5.1 we demonstrate their effectiveness when employed with high-quality TREC *runs* [46].

As it turns out, our methods can also be used to tackle two long-standing problems; namely, integrating the results of cluster-based retrieval with those of document-based retrieval [23]; and, improving the *robustness* (and overall effectiveness) of pseudo-feedback-based query expansion methods [2, 17, 12, 33].

3.1 Integrating document-based and cluster-based retrieval

One of the most common cluster-based retrieval paradigms follows van Rijsbergen's *cluster hypothesis* [44], which states that "closely-associated documents tend to be relevant to the same requests". Specifically, the idea is to cluster the corpus into clusters of similar documents; then, at retrieval time, the constituent documents of the clusters most similar to the query are presented as results. The hypothesis is that these clusters contain a high percentage of relevant documents [24, 14, 45, 23]. Griffiths et al. [23] observed that the list of documents retrieved in this fashion yields the same retrieval effectiveness as that resulting from comparison of the query with documents (i.e., document-based retrieval). However, the overlap between the relevant document sets in the two lists was somewhat small. While noting that integration of the two lists could potentially yield significant merits over using each alone, they left the question of how to automatically perform the integration open.

This mismatch between sets of relevant documents was an important motivating factor in deriving our approach. Hence, we will test the effectiveness of our methods in re-ranking the results of document-based retrieved ($\mathcal{L}_{init}$) using those of cluster-based retrieval ($\mathcal{L}_{help}$) in Section 5.2.

3.2 Robust pseudo-feedback-based retrieval

Pseudo-feedback-based query expansion is a highly effective retrieval paradigm [10]. The approach is based on performing standard document-based retrieval in response to the query, and, then, using terms from the top-retrieved documents for expanding the query; the expanded query form is then used for ranking. While such methods improve retrieval effectiveness on average, there are many queries for which the performance is substantially lower than that of using the original query with no expansion [2, 17, 12, 33]. One of the major reasons for this *performance robustness* problem is *query drift* — the shift in intention between the information need represented by the original query and that represented by its expanded form [37].

Our methods can be used to potentially ameliorate the query-drift problem. Specifically, we can re-rank a list produced by pseudo-feedback-based retrieval ($\mathcal{L}_{init}$) using a list retrieved in response to the original query ($\mathcal{L}_{help}$). The potential merit in doing so is as follows. Consider a document d, which is highly ranked in $\mathcal{L}_{init}$, but which is not in $\mathcal{L}_{help}$ — e.g., d does not exhibit high surface-level query similarity. (Note that if d is relevant, then this is an example for a relevant-documents-sets mismatch that our methods try to address.) Now, d is rewarded by our methods if it is similar to highly-ranked documents in $\mathcal{L}_{help}$; those could be considered as reflecting the corpus-context of the query by the virtue of the way $\mathcal{L}_{help}$ was created. In such a case, d can potentially be considered as "faithful" to the original query, and, hence, less probable to manifest query drift.

4. RELATED WORK

If we consider documents to be similar if and only if they are the same document, then our methods reduce to using the principle underlying most fusion methods; that is, rewarding documents that are highly ranked in many of the fused lists [16, 21, 6, 38]. Yet, we note that our methods re-rank $\mathcal{L}_{init}$, while fusion methods produce result-lists that can contain documents that are in $\mathcal{L}_{help}$ but not in $\mathcal{L}_{init}$. In Section 5 we demonstrate the merits of our approach with respect to a state-of-the-art fusion method.

Traditional fusion methods use either the ranks of documents, or their relevance-scores, but not the documents' content [22, 16, 21, 6, 38]. One reason for doing so is lack of (quick) access to the document content. We hasten to point that our re-ranking methods need not necessarily compute inter-document similarities based on the entire document content. Case in point, *snippets* (i.e., document summaries) can potentially be used for computing inter-document similarities, as is the case, for example, in some work on clustering results of Web search engines [49]. Snippets (and other document features) were also utilized in some fusion models [13, 7, 41]; in contrast to our methods, inter-document similarities were not exploited.

There is a large body of work on re-ranking a retrieved list using inter-document similarities within the list (e.g., [47, 34, 4, 19, 28, 51, 26, 29, 35, 48, 20, 36]). As described in Section 2, our models could conceptually be viewed as a generalization of some of these approaches (e.g., [4, 19, 28]) to the two-lists case. Furthermore, our SimRank algorithm is reminiscent of a cluster-based re-ranking model wherein similarities between documents in the list and *clusters* of documents in the list are utilized [29]. We demonstrate the merits of our methods with respect to a state-of-the-art cluster-based (one-list) re-ranking method [26] in Section 5.2.

We posed our re-ranking methods as a potential means for integrating document-based and cluster-based retrieval results. Some previous work [34, 27] has shown the effectiveness of cluster-based smoothing of *document language models* as a way for integrating document-based and cluster-based information. We demonstrate in Section 5.2 the merits in using our methods with respect to one such state-of-the-art approach [27].

Another proposed use of our methods is for improving the robustness of pseudo-feedback-based (PF) retrieval, specifically, by re-ranking the PF results ($\mathcal{L}_{init}$) using those retrieved in response to the original query ($\mathcal{L}_{help}$). A recently-proposed approach resembles ours in that it fuses $\mathcal{L}_{init}$ and $\mathcal{L}_{help}$ using traditional fusion methods [52]. One potential benefit of our approach is the consideration of inter-document similarities. Case in point, a document not highly ranked in response to the original query (e.g., that does not exhibit high surface-level query similarity) can still be positioned at high ranks of the final result list — the underlying idea of using PF — if it is similar to documents highly ranked in $\mathcal{L}_{help}$. We further demonstrate the merits of our approach in Section 5.3.

Re-ranking PF-based retrieved results using those produced in response to the original query is conceptually rem-

iniscent of work on fusing the results retrieved in response to various (manually created) query representations [40, 9, 8]; however, inter-document-similarities — the importance of which for this task is demonstrated in Section 5.3 — were not used. In addition, we note that previously-proposed approaches for improving PF-based retrieval robustness (e.g., [1, 37, 39, 43, 12, 11, 33]), which improve the query-model used for ranking, can potentially benefit from employing our methods upon the retrieved results, as we show (for one such method [1]) in Section 5.3.

5. EVALUATION

We next explore the effectiveness (or lack thereof) of our re-ranking methods when (i) employed over TREC runs (Section 5.1), (ii) used to integrate document-based and cluster-based retrieval (Section 5.2), and (iii) used to improve the robustness of pseudo-feedback-based retrieval (Section 5.3).

Following some previous work on re-ranking [28, 29], we use a language-model-based approach to estimate similarities between texts. Specifically, let $p_z^{Dir[\mu]}(\cdot)$ denote the unigram, Dirichlet-smoothed, language model induced from text z, where μ is the smoothing parameter [50]. (Unless otherwise specified, we set $\mu = 1000$ following previous recommendations [50].) We then define the inter-text-similarity estimate for texts x and y:

$$sim(x, y) \stackrel{def}{=} \exp\left(-D\left(p_x^{Dir[0]}(\cdot) \,\middle\|\, p_y^{Dir[\mu]}(\cdot)\right)\right);$$

D is the KL divergence. This estimate was shown to be effective in previous work on utilizing inter-document similarities for re-ranking [28, 29].

We apply tokenization, Porter stemming, and stopword removal (using the INQUERY list) to all data used in our experiments via the Lemur toolkit (www.lemurproject.org), which is also used for language-model induction.

We posed our re-ranking methods as a means for improving precision at top ranks. Therefore, we use the precision of the top 5 and 10 documents (p@5, p@10), and the mean reciprocal rank of the first relevant document (MRR) as evaluation measures. We use the Wilcoxon two-tailed test at a 95% confidence level to determine statistically-significant differences in performance.

We set α, the ancestry parameter used by our methods, to a value in $\{5, 10, 20, 30, 40, 50\}$ so as to optimize p@5. (In most cases, $\alpha = 20$ yields effective performance.) If two values yield the same p@5, we choose the one *minimizing* p@10 so as to provide conservative estimates of performance; in case of a tie for both p@5 and p@10, we choose the value minimizing MRR.

5.1 Re-ranking TREC runs

Our first order of business is to study the general effectiveness of our re-ranking approach. To that end, we re-rank TREC *runs* [46] (i.e., lists submitted by TREC participants in response to a query). We set $\mathcal{L}_{init}$, the list upon which re-ranking is performed, to the top-k documents in the run that yields the **best** p@5 — the evaluation measure for which we optimize performance — among all submitted runs. The list $\mathcal{L}_{help}$, which is used for re-ranking $\mathcal{L}_{init}$, is set to the k highest ranked documents in the run that yields the **second-best** p@5. We set $k = 50$ following findings that using inter-document similarities for re-ranking is most effective when employed over relatively short lists [19, 28].

We normalize retrieval scores for inter-compatibility. (While this is not required by our methods, except for cases of negative retrieval scores, it is crucial for the CombMNZ method that we use as a reference comparison.) Specifically, let $s_{\mathcal{L}}(d)$ be d's original retrieval score in the run that was used to create $\mathcal{L}$; we set $Score_{\mathcal{L}}(d)$, the retrieval score used by our methods, to $\frac{s_{\mathcal{L}}(d) - min_{d' \in \mathcal{L}} s_{\mathcal{L}}(d')}{\max_{d' \in \mathcal{L}} s_{\mathcal{L}}(d') - min_{d' \in \mathcal{L}} s_{\mathcal{L}}(d')}$.

To study the importance of utilizing inter-document similarities between documents in $\mathcal{L}_{init}$ and $\mathcal{L}_{help}$, we use two fusion approaches as reference comparisons to our re-ranking methods. The first is **CombMNZ** [22, 32] — a state of the art fusion method that scores document d in $\mathcal{L}_{init} \cup \mathcal{L}_{help}$ by $(\delta[d \in \mathcal{L}_{init}] + \delta[d \in \mathcal{L}_{help}])(Score_{\mathcal{L}_{init}}(d) + Score_{\mathcal{L}_{help}}(d))$. The second reference comparison that we consider, **CombMult**, could be regarded as a special case of SimRank wherein two documents are deemed similar if and only if they are the same document. Specifically, CombMult scores d in $\mathcal{L}_{init} \cup \mathcal{L}_{help}$ by $Score_{\mathcal{L}_{init}}(d) \cdot Score_{\mathcal{L}_{help}}(d)$; to avoid zero-score multiplication, we define (only for CombMult) $Score_{\mathcal{L}}(d) \stackrel{def}{=} \min_{d' \in \mathcal{L}} Score_{\mathcal{L}}(d')$ if $d \notin \mathcal{L}$.

For experiments we use the ad-hoc track of trec3, the Web tracks of trec9 and trec10, and the robust track of trec12; these were used in previous work on fusion (e.g., [3, 38]).

The performance numbers of all methods are presented in Table 1. Our first observation is that both SimRank and SimMNZRank are in general quite effective in re-ranking $\mathcal{L}_{init}$; specifically, in all reference comparisons (track $\times$ evaluation measure), except for those of trec9, their performance is better — in many occasions to a substantial extent and to a statistically significant degree — than that of the best run used to create $\mathcal{L}_{init}$. For trec9, the performance of both methods is lower than that of the best run, although not to a statistically significant degree. (Note that both fusion methods, CombMNZ and CombMult, also post performance that is inferior to that of the best run for trec9. The substantial, and statistically-significant, performance differences between the best run and the second-best run for trec9 imply to the challenge of improving on the performance of the former.)

Another observation that we make based on Table 1 is that the performance of SimRank and SimMNZRank is superior in most relevant comparisons, and, often, to a statically significant degree, to that of the second-best run that was used to create $\mathcal{L}_{help}$.

In comparing SimRank with SimMNZRank, we see that the latter posts performance that is superior in most relevant comparisons to that of the former. Thus, rewarding documents in $\mathcal{L}_{init}$ that also appear in $\mathcal{L}_{help}$, as is done by SimMNZRank as opposed to SimRank, has positive impact on re-ranking performance.

We also see in Table 1 that SimRank and SimMNZRank post performance that is in most relevant comparisons superior to that of the CombMult fusion method. Furthermore, their performance also transcends that of CombMNZ in most comparisons for trec3, trec9 and trec10; for trec12, CombMNZ posts better performance. (The performance differences between our methods and CombMNZ, however, are not statistically significant.) We also note that both fusion methods (CombMult and CombMNZ) rarely post statistically significant performance improvements over the best

	trec3			trec9			trec10			trec12		
	p@5	p@10	MRR	p@5	p@10	MRR	p@5	p@10	MRR	p@5	p@10	MRR
best($\mathcal{L}_{init}$)	76.0	71.2	84.9	**59.2**	**51.8**	**76.1**	63.2	58.8	80.4	54.8	45.2	75.4
second-best($\mathcal{L}_{help}$)	74.4	72.2	84.2	45.6^{i}	38.4^{i}	61.2^{i}	55.6	46.8^{i}	70.6	52.8	48.8	74.7
CombMNZ	80.4^{h}	73.8	88.2	56.8^{h}	49.4^{h}	70.1	69.6^{h}	59.6^{h}	90.3^{ih}	$\mathbf{58.2}^{h}$	$\mathbf{49.9}^{i}$	**78.9**
CombMult	80.8^{h}	75.4	86.4	52.4^{h}	44.0^{ih}_{c}	70.6^{h}	67.6^{h}	56.8^{h}	$\mathbf{90.9}^{ih}$	57.2^{h}	49.0^{i}	78.6
SimRank	82.0^{ih}	75.4^{i}	87.8	56.8^{h}	49.6^{h}_{m}	71.8	$\mathbf{72.4}^{ih}$	$\mathbf{62.8}^{h}_{m}$	89.7^{h}	57.2^{h}	48.4^{i}	77.4
SimMNZRank	$\mathbf{82.8}^{ih}$	$\mathbf{76.0}^{i}$	**89.2**	57.2^{h}	49.0^{h}	71.8	71.2^{ih}	60.4^{h}	90.1^{ih}	57.6^{ih}	48.6^{i}	78.0

Table 1: Performance numbers of re-ranking (using SimRank and SimMNZRank) the best-p@5 run in a TREC track using the second-best-p@5 run. The performance of the CombMNZ and CombMult fusion methods is presented for comparison. The best result in a column is boldfaced. Statistically significant differences with best, second-best, CombMNZ, and CombMult, are marked with 'i', 'h', 'c', and 'm', respectively.

run, while our re-ranking methods do so in about a third (SimRank) and a half (SimMNZRank) of the relevant comparisons (track × evaluation measure). Thus, we see that there is merit in utilizing inter-document similarities between $\mathcal{L}_{init}$ and $\mathcal{L}_{help}$ as is done by our methods as opposed to the fusion methods.

5.2 Integrating document-based and cluster-based retrieval

We next study the effectiveness of our methods in re-ranking a list retrieved by a standard document-based approach ($\mathcal{L}_{init}$) using a list retrieved by a cluster-based method ($\mathcal{L}_{help}$).

To create $\mathcal{L}_{init}$, we use the standard KL-retrieval approach [30], denoted **DocRet**. Specifically, we set $\mathcal{L}_{init}$ to the 50 documents d in the corpus that yield the highest $sim(q, d)$. Following previous work on re-ranking [28, 29], we set μ, the document language model smoothing parameter, to a value optimizing MAP@1000. Although MAP is not one of the evaluation metrics that we consider, it is a general-purpose evaluation criterion that yields ranking of a reasonable quality. Naturally, then, we use the same retrieval method with μ optimized for p@5 — the measure for which we optimize the performance of our re-ranking methods — as a reference comparison, denoted **DocRetOpt**.

To create $\mathcal{L}_{help}$, we first cluster the corpus into *static* (offline) nearest-neighbor overlapping clusters [23, 27]. Specifically, for each document d we define a cluster that contains d and its $\alpha - 1$ nearest-neighbors d' ($d' \neq d$) according to $sim(d, d')$; we set $\alpha = 10$, since such small nearest-neighbor clusters are known to yield effective retrieval performance [23, 27]. As is common in work on cluster-based retrieval in the language modeling framework [34, 27, 29], we represent cluster c by the big document that results from concatenating its constituent documents. The order of concatenation has no effect since we only use unigram language models that assume term independence.

To exploit the overlap of clusters, we use the effective Bag-Select retrieval algorithm [28], referred to here as **ClustRet**. Specifically, let $TopRetClust(q)$ be the set of 50 clusters c that yield the highest $sim(q, c)$. The score of document d is $Score_{ClustRet}(d) \stackrel{def}{=} sim(q, d) \cdot \#\{c \in TopRetClust(q) : d \in c\}$. Thus, d is ranked high if it is a member of many top-retrieved clusters and if it exhibits high query-similarity. Finally, we set $\mathcal{L}_{help}$ to the list of documents that are members of the clusters in $TopRetClust(q)$ ordered by their ClustRet-assigned scores.

As at the above, we use the CombMNZ fusion method — employed over the lists $\mathcal{L}_{init}$ and $\mathcal{L}_{help}$ using normalized retrieval scores — as a reference comparison.

We also consider two additional reference comparisons that utilize a state-of-the-art cluster-based retrieval approach, namely, the *interpolation* algorithm [27]. Specifically, given a set of clusters S, document d that belongs to at least one cluster in S is assigned with the score $\lambda sim(q, d) + (1 - \lambda) \sum_{c \in S} sim(q, c) sim(c, d)$; λ is a free parameter. The interpolation algorithm was shown to be highly effective for re-ranking the cluster-based retrieved list $\mathcal{L}_{help}$ (i.e., setting $S \stackrel{def}{=} TopRetClust(q)$) [27]; we use **Interp(stat)** to denote this implementation. The interpolation algorithm was also shown to yield state-of-the-art performance in re-ranking a list retrieved by document-based retrieval ($\mathcal{L}_{init}$) [26]; specifically, such implementation, denoted **Interp(dyn)**, uses 50 *dynamic*, query-specific nearest-neighbor clusters (of 10 documents) that are created from documents in $\mathcal{L}_{init}$ [26]. For both Interp(stat) and Interp(dyn), we set λ to a value in $\{0, 0.1, \ldots, 1\}$ so as to optimize p@5.

To further study the importance of using the cluster-based retrieved list for re-ranking the document-based retrieved list, we also experiment with an implementation of SimRank, denoted **SimRankNoHelp**[3], wherein $\mathcal{L}_{help} \stackrel{def}{=} \mathcal{L}_{init}$ — i.e., inter-document similarities within $\mathcal{L}_{init}$ are used for its re-ranking. (The value of α is chosen, as at the above, to optimize p@5.)

Since clustering the web corpora from trec9 and trec10 is computationally demanding, we used for the experiments here the following four TREC corpora, which were used in some previous work on re-ranking [34, 28, 29]:

corpus	queries	disk(s)
ROBUST	301-450, 601-700	4,5 (-CR)
AP	51-150	1-3
WSJ	151-200	1-2
SJMN	51-150	3

The performance numbers of our re-ranking methods and those of the reference comparisons are presented in Table 2. Our first observation is that the performance of SimRank and SimMNZRank is better in most relevant comparisons — often to a statistically significant degree — than that of the document-based retrieval (DocRet) that was used to create the list $\mathcal{L}_{init}$ upon which re-ranking is performed. Furthermore, the performance of both methods is also consistently

[3]This is reminiscent of a previously-proposed re-ranking algorithm [28].

	ROBUST			AP			WSJ			SJMN		
	p@5	p@10	MRR	p@5	p@10	MRR	p@5	p@10	MRR	p@5	p@10	MRR
DocRet($\mathcal{L}_{init}$)	48.2	43.2	**68.8**	44.6	43.5	59.5	55.6	49.8	71.5	33.2	28.7	48.5
ClustRet($\mathcal{L}_{help}$)	48.8	44.9	65.5^{i}	49.7^{i}	48.2^{i}	61.6	58.8	**54.6**^{i}	75.9	35.4	31.8^{i}	52.3
DocRetOpt	48.3	44.0^{i}	67.2	45.5	43.5^{h}	58.8	57.6	50.0	75.2	33.6	28.7	50.7
CombMNZ	49.4	44.8^{i}	67.3^{h}	49.7^{i}_{o}	47.0^{i}_{o}	63.6_{o}	58.4	53.4^{i}	**79.5**^{i}	35.6	33.2^{i}_{o}	51.9
Interp(stat)	48.9	43.3_{c}	68.6	**52.3**$^{i}_{o}$	47.5	61.4	60.4^{i}	53.2^{i}	77.7	37.8^{i}	33.8^{i}_{o}	50.9
Interp(dyn)	50.2^{i}	45.1^{i}_{s}	68.1	50.1^{i}	47.8^{i}_{o}	59.3	60.8	53.4^{i}	77.6	36.0	32.9^{i}_{o}	47.9
SimRankNoHelp	49.4	45.3^{i}_{s}	67.5	50.7^{i}_{o}	48.4^{i}_{o}	60.9	59.2^{i}	53.2^{i}_{o}	77.4	**38.2**^{i}	**34.3**$^{i}_{o}$	**53.1**$^{i}_{d}$
SimRank	50.8^{ih}_{os}	**45.8**$^{ih}_{s}$	67.2^{h}	52.1^{i}_{o}	49.1^{i}_{o}	63.3	61.6^{i}	53.6^{i}	77.1	37.6^{i}_{c}	33.1^{i}_{o}	50.8
SimMNZRank	**50.9**$^{ih}_{ocs}$	**45.8**$^{ih}_{os}$	67.2^{h}	**52.3**$^{i}_{o}$	**49.3**$^{i}_{oc}$	**65.1**$^{i}_{on}$	**62.0**$^{i}_{c}$	54.0^{i}	76.8	37.4^{i}	33.9^{i}_{o}	51.4

Table 2: Performance numbers of SimRank and SimMNZRank when re-ranking a list ($\mathcal{L}_{init}$) retrieved by a document-based approach (DocRet) using a list ($\mathcal{L}_{help}$) retrieved by a cluster-based approach (ClustRet). The performance of optimized document-based retrieval (DocRetOpt), the CombMNZ method, a state-of-the-art cluster-based approach (interpolation [27]) — implemented either with static clusters (Interp(stat)) to re-rank $\mathcal{L}_{help}$, or with dynamic, query-specific clusters (Interp(dyn)) to re-rank $\mathcal{L}_{init}$ — and SimRank implemented over $\mathcal{L}_{init}$ without using $\mathcal{L}_{help}$ (SimRankNoHelp) is presented for reference. The best result in a column is boldfaced. Statistically significant differences with DocRet, ClustRet, DocRetOpt, CombMNZ, Interp(stat), Interp(dyn), and SimRankNoHelp, are marked with 'i', 'h', 'o', 'c', 's', 'd', and 'n', respectively.

better than that of the optimized document-based retrieval method (DocRetOpt). In addition, both re-ranking methods post performance that is better in almost all relevant comparisons than that of the cluster-based retrieval method (ClustRet) that was used to create $\mathcal{L}_{help}$.

We can also see in Table 2 that the re-ranking methods (especially SimMNZRank) post performance that is superior in a majority of the relevant comparisons (sometimes to a statistically significant degree) to that of the CombMNZ fusion method. Hence, inter-document similarities induced between the lists seem to be a helpful source of information for re-ranking as was the case for the TREC tracks.

Another observation that we make based on Table 2 is that SimMNZRank is superior in most relevant comparisons to the interpolation algorithms. While only a few of these performance differences are statistically significant (see the ROBUST case), SimMNZRank posts more statistically significant performance improvements over DocRet and DocRetOpt than the interpolation algorithms do. The performance improvements posted by SimMNZRank with respect to Interp(dyn) attest to the benefit in using the cluster-based retrieved list ($\mathcal{L}_{help}$) to re-rank $\mathcal{L}_{init}$. (Recall that Interp(dyn) is a state-of-the-art re-ranking method that uses information only within $\mathcal{L}_{init}$.) This finding is further supported by the fact that in a majority of the relevant comparisons, both SimRank and SimMNZRank post better performance than that of SimRankNoHelp that uses inter-document-similarities only within $\mathcal{L}_{init}$.

5.3 Robust pseudo-feedback-based query expansion

The next challenge that we address is that of robust pseudo-feedback-based query expansion. Our goal is to re-rank the results of pseudo-feedback-based retrieval ($\mathcal{L}_{init}$) using those retrieved in response to the original query ($\mathcal{L}_{help}$) so as to potentially ameliorate query drift.

We use the *relevance model* approach [31, 1], which is a state-of-the-art pseudo-feedback-based query expansion technique, to create $\mathcal{L}_{init}$. Specifically, let w denote a term in the vocabulary, $\{q_i\}$ be the bag of query terms, $\mathcal{D}_q^{[m]}$ be the set of m documents d in the corpus that yield the highest $sim(q,d)$, and $p_x^{JM[\lambda]}(\cdot)$ denote x's Jelinek-Mercer smoothed language model with smoothing parameter λ [31]. The basic relevance model, a.k.a. RM1, is defined as

$$p_{RM1}(w;\lambda) \stackrel{def}{=} \sum_{d \in \mathcal{D}_q^{[m]}} p_d^{JM[\lambda]}(w) \frac{\prod_i p_d^{JM[\lambda]}(q_i)}{\sum_{d_j \in \mathcal{D}_q^{[m]}} \prod_i p_{d_j}^{JM[\lambda]}(q_i)}.$$

Following previous work [1], we *clip* RM1 by setting $p_{RM1}(w;\lambda)$ to 0 for all but the γ terms with the highest $p_{RM1}(w;\lambda)$; further normalization is performed to yield a probability distribution $\tilde{p}_{RM1}(\cdot;\lambda,\gamma)$. To potentially prevent query drift at the language model level, RM1 is anchored to the original query [1]; that is, the resultant language model, a.k.a. **RM3**, which we use in the experiments to follow, is defined as follows:

$$p_{RM3}(w;\lambda,\gamma,\psi) \stackrel{def}{=} \psi p_q^{JM[1]}(w) + (1-\psi)\tilde{p}_{RM1}(w;\lambda,\gamma).$$

Then, $\mathcal{L}_{init}$, the list to be re-ranked, is set to the 50 documents d in the corpus that yield the lowest KL divergence $D\left(p_{RM3}(\cdot;\lambda,\gamma,\psi) \,\middle|\middle|\, p_d^{Dir[\mu]}(\cdot)\right)$.

In what follows, we set $\lambda = 0.9$ following previous recommendations [31]. The values of the other free parameters are chosen from the following sets to optimize MAP@1000, so as to ensure that the list $\mathcal{L}_{init}$ is of a reasonable quality: $m \in \{10, 50, 100, 500\}$, $\gamma \in \{25, 50, 100, 500, ALL\}$, where "*ALL*" stands for using all terms in the vocabulary (i.e., no clipping), and $\psi \in \{0, 0.1, 0.2, \ldots, 0.9\}$; μ is set to 1000 as in all other algorithms. Consequently, a natural choice for a reference comparison to our methods is **RM3opt** — the RM3 model with the values of the free parameters optimized for p@5 — the metric for which we optimize the performance of our re-ranking methods.

The list $\mathcal{L}_{help}$ using which our methods re-rank the pseudo-feedback-based results ($\mathcal{L}_{init}$), is set to the 50 documents d in the corpus that yield the highest $sim(q,d)$ — i.e., the standard KL-retrieval method with $\mu = 1000$; we denote this approach here as **DocRetOrigQuery** so as to emphasize that this retrieval is performed in response to the original query. As in all other experiments, we use the CombMNZ

	ROBUST			AP			WSJ			SJMN		
	p@5	p@10	MRR	p@5	p@10	MRR	p@5	p@10	MRR	p@5	p@10	MRR
RM3($\mathcal{L}_{init}$)	49.0	45.6	68.2	49.3	46.7	59.4	58.8	55.0	75.6	38.4	34.5	50.5
DocRetOrigQuery($\mathcal{L}_{help}$)	48.2	43.2^{i}	**68.8**	44.6^{i}	43.5	59.5	55.6	49.8^{i}	71.5	33.2^{i}	28.7^{i}	48.5
RM3opt	$\mathbf{51.2}^{ih}$	$\mathbf{46.1}^{h}$	67.6	50.7^{h}	48.5^{h}	58.4	$\mathbf{62.4}^{h}$	56.0^{h}	77.0	$\mathbf{39.6}^{h}$	34.1^{h}	49.5^{i}
CombMNZ	48.9_{o}	45.2^{h}	$\mathbf{68.8}^{i}$	47.1	47.5^{h}	59.3	57.6_{o}	53.4^{h}	77.1^{h}	36.8^{h}	32.4^{ih}	$\mathbf{52.4}^{i}_{o}$
SimRank	49.7	44.8_{o}	65.8^{h}_{oc}	50.9^{h}_{c}	48.1^{h}	61.3	59.6	$\mathbf{56.4}^{h}_{c}$	**77.7**	39.2^{h}	$\mathbf{35.4}^{h}_{c}$	**52.4**
SimMNZRank	50.0	44.7_{o}	66.0^{h}_{c}	$\mathbf{51.5}^{h}_{c}$	$\mathbf{48.8}^{h}$	**61.5**	59.2	$\mathbf{56.4}^{h}_{c}$	**77.7**	39.2^{h}	35.0^{h}_{c}	51.9

Table 3: Performance numbers of re-ranking the results of pseudo-feedback-based retrieval (RM3) with those of standard document-based retrieval performed in response to the original query (DocRetOrigQuery). The performance of optimized pseudo-feedback-based retrieval (RM3opt) and that of the CombMNZ fusion method is presented for reference. The best result in a column is boldfaced. Statistically significant differences with RM3, DocRetOrigQuery, RM3opt, and CombMNZ, are marked with 'i', 'h', 'o', and 'c', respectively.

fusion method upon $\mathcal{L}_{init}$ and $\mathcal{L}_{help}$ (with normalized retrieval scores[4]) as a reference comparison to our methods.

Recall that our motivation for re-ranking pseudo-feedback-based retrieval results, in addition to improving their overall effectiveness, is to improve performance robustness. To measure the success, or lack thereof, of the different methods in accomplishing this goal, we use the "reliability of improvement index" (RI) [39], a.k.a. the "robustness index" [12]. Specifically, for retrieval method $\mathcal{M}$, we define $RI \stackrel{def}{=} \%N_+ - \%N_-$, where $\%N_+$ ($\%N_-$) is the percentage of queries for which the p@5-performance of $\mathcal{M}$ is better (worse) than that of the retrieval based on the original query (DocRetOrigQuery); note that RI takes 100 and -100 as maximum and minimum values, respectively.

The performance numbers of our methods, along with those of the reference comparisons, are presented in Table 3. Table 4 presents the performance-robustness numbers of the methods. Our first observation based on Table 3 is that both re-ranking methods post performance that is in most relevant comparisons superior (although not to a statistically significant degree) to that of RM3, which was used to create the list $\mathcal{L}_{init}$. Moreover, the robustness of both re-ranking methods is almost always better than that of RM3 as can be seen in Table 4. As RM3 attempts to prevent query-drift at the language-model-level (see the above), we view these results as gratifying.

When considering the CombMNZ fusion method, we see that it posts performance that is in most cases inferior to, and less robust than, that of RM3 and our re-ranking methods. This finding supports the benefit in using inter-document-similarities between the retrieved lists: we want to both ameliorate query drift and to let documents not retrieved in response to the original query to be retrieved by the pseudo-feedback-based approach.

We can also see in Table 3 that SimMNZRank is superior to RM3opt, the optimized relevance model, in 7 out of 12 relevant comparisons (corpus × evaluation measure); however, these improvements are not statistically significant. RM3opt does manage to yield statistically-significant improvements over our methods for the ROBUST corpus, albeit not with respect to p@5 — the metric for which performance was optimized. In addition, we see in Table 4 that for the AP and SJMN corpora, the performance of SimMNZRank is more robust than that of RM3opt; for the ROBUST and WSJ corpora, the reverse holds.

[4]We use $\exp(-D\left(p_{RM3}(\cdot;\lambda,\gamma,\psi) \,\middle|\middle|\, p_d^{Dir[\mu]}(\cdot)\right))$ as the pseudo-feedback-based retrieval score of d, and normalize it as at the above.

	ROBUST	AP	WSJ	SJMN
RM3($\mathcal{L}_{init}$)	1.6	14.1	10.0	11.0
RM3opt	**12.4**	13.1	**18.0**	13.0
CombMNZ	3.2	9.1	8.0	9.0
SimRank	6.4	19.2	**18.0**	11.0
SimMNZRank	8.0	**20.2**	16.0	**14.0**

Table 4: Performance robustness of the methods from Table 3; bold marks the best result in a column. Robustness is computed by $RI \stackrel{def}{=} \%N_+ - \%N_-$, where $\%N_+$ ($\%N_-$) is the percentage of queries for which the p@5-performance of the approach is better (worse) than that of retrieval performed in response to the original query (DocRetOrigQuery).

In comparing SimRank with SimMNZRank we see that none dominates the other with respect to overall effectiveness. (Refer to Table 3.) However, SimMNZRank does yield performance that is for most corpora more robust than that of SimRank (see Table 4). The latter finding resonates with the fact that SimMNZRank performs "stronger" query-anchoring than that of SimRank by the virtue of rewarding documents that are both in $\mathcal{L}_{init}$ and in $\mathcal{L}_{help}$; recall that the latter was retrieved in response to the original query.

All in all, the findings at the above attest that our re-ranking methods are a potential means to improving the robustness and overall effectiveness of the RM3 model.

6. CONCLUSION

We presented a novel approach to re-ranking an initially retrieved list so as to improve precision at the very top ranks. Our approach is based on utilizing similarities between documents in the list and documents in a second list that was produced using, for example, a different retrieval method and/or query representation. We demonstrated the effectiveness of our methods in re-ranking high quality TREC *runs*. In addition, we showed that our methods can be used to integrate document-based and cluster-based retrieved results; and, to improve the overall effectiveness, and performance robustness, of a state-of-the-art pseudo-feedback-based query expansion method.

7. REFERENCES

[1] N. Abdul-Jaleel, J. Allan, W. B. Croft, F. Diaz, L. Larkey, X. Li, M. D. Smucker, and C. Wade. UMASS at TREC 2004 —

novelty and hard. In *Proceedings of TREC-13*, pages 715–725, 2004.

[2] G. Amati, C. Carpineto, and G. Romano. Query difficulty, robustness, and selective application of query expansion. In *Proceedings of ECIR*, pages 127–137, 2004.

[3] J. A. Aslam and M. Montague. Models for metasearch. In *Proceedings of SIGIR*, pages 276–284, 2001.

[4] J. Baliński and C. Daniłowicz. Re-ranking method based on inter-document distances. *Information Processing and Management*, 41(4):759–775, 2005.

[5] S. M. Beitzel, E. C. Jensen, A. Chowdhury, O. Frieder, D. A. Grossman, and N. Goharian. Disproving the fusion hypothesis: An analysis of data fusion via effective information retrieval strategies. In *Proceedings of SAC*, pages 823–827, 2003.

[6] S. M. Beitzel, E. C. Jensen, A. Chowdhury, D. A. Grossman, O. Frieder, and N. Goharian. Fusion of effective retrieval strategies in the same information retrieval system. *JASIST*, 55(10):859–868, 2004.

[7] S. M. Beitzel, E. C. Jensen, O. Frieder, A. Chowdhury, and G. Pass. Surrogate scoring for improved metasearch precision. In *Proceedings of SIGIR*, pages 583–584, 2005.

[8] N. Belkin, P. Kantor, E. Fox, and J. Shaw. Combining evidence of multiple query representation for information retrieval. *Information Processing and Management*, 31(3):431–448, 1995.

[9] N. J. Belkin, C. Cool, W. B. Croft, and J. P. Callan. The effect of multiple query representations on information retrieval system performance. In *Proceedings of SIGIR*, pages 339–346, 1993.

[10] C. Buckley, G. Salton, J. Allan, and A. Singhal. Automatic query expansion using SMART: TREC3. In *Proceedings of TREC-3*, pages 69–80, 1994.

[11] G. Cao, J.-Y. Nie, J. Gao, and S. Robertson. Selecting good expansion terms for pseudo-relevance feedback. In *Proceedings of SIGIR*, pages 243–250, 2008.

[12] K. Collins-Thompson and J. Callan. Estimation and use of uncertainty in pseudo-relevance feedback. In *Proceedings of SIGIR*, pages 303–310, 2007.

[13] N. Craswell, D. Hawking, and P. B. Thistlewaite. Merging results from isolated search engines. In *Proceedings of the Australian Database Conference*, pages 189–200, 1999.

[14] W. B. Croft. A model of cluster searching based on classification. *Information Systems*, 5:189–195, 1980.

[15] W. B. Croft, editor. *Advances in Information Retrieval: Recent Research from the Center for Intelligent Information Retrieval*. Number 7 in The Kluwer International Series on Information Retrieval. Kluwer, 2000.

[16] W. B. Croft. Combining approaches to information retrieval. In Croft [15], chapter 1, pages 1–36.

[17] S. Cronen-Townsend, Y. Zhou, and W. B. Croft. A language modeling framework for selective query expansion. Technical Report IR-338, Center for Intelligent Information Retrieval, University of Massachusetts, 2004.

[18] P. Das-Gupta and J. Katzer. A study of the overlap among document representations. In *SIGIR*, pages 106–114, 1983.

[19] F. Diaz. Regularizing ad hoc retrieval scores. In *Proceedings of CIKM*, pages 672–679, 2005.

[20] F. Diaz. A method for transferring retrieval scores between collections with non overlapping vocabularies. In *Proceedings of SIGIR*, pages 805–806, 2008. poster.

[21] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar. Rank aggregation methods for the Web. In *Proceedings of the WWW Conference*, pages 613–622, Hong Kong, 2001.

[22] E. A. Fox and J. A. Shaw. Combination of multiple searches. In *Proceedings of TREC-2*, 1994.

[23] A. Griffiths, H. C. Luckhurst, and P. Willett. Using interdocument similarity information in document retrieval systems. *Journal of the American Society for Information Science (JASIS)*, 37(1):3–11, 1986.

[24] N. Jardine and C. J. van Rijsbergen. The use of hierarchic clustering in information retrieval. *Information Storage and Retrieval*, 7(5):217–240, 1971.

[25] J. Kleinberg. Authoritative sources in a hyperlinked environment. Technical Report Research Report RJ 10076, IBM, May 1997.

[26] O. Kurland. *Inter-document similarities, language models, and ad hoc retrieval*. PhD thesis, Cornell University, 2006.

[27] O. Kurland and L. Lee. Corpus structure, language models, and ad hoc information retrieval. In *Proceedings of SIGIR*, pages 194–201, 2004.

[28] O. Kurland and L. Lee. PageRank without hyperlinks: Structural re-ranking using links induced by language models. In *Proceedings of SIGIR*, pages 306–313, 2005.

[29] O. Kurland and L. Lee. Respect my authority! HITS without hyperlinks utilizing cluster-based language models. In *Proceedings of SIGIR*, pages 83–90, 2006.

[30] J. D. Lafferty and C. Zhai. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of SIGIR*, pages 111–119, 2001.

[31] V. Lavrenko and W. B. Croft. Relevance-based language models. In *Proceedings of SIGIR*, pages 120–127, 2001.

[32] J. H. Lee. Analyses of multiple evidence combination. In *Proceedings of SIGIR*, pages 267–276, 1997.

[33] K.-S. Lee, W. B. Croft, and J. Allan. A cluster-based resampling method for pseudo-relevance feedback. In *SIGIR*, pages 235–242, 2008.

[34] X. Liu and W. B. Croft. Cluster-based retrieval using language models. In *Proceedings of SIGIR*, pages 186–193, 2004.

[35] X. Liu and W. B. Croft. Experiments on retrieval of optimal clusters. Technical Report IR-478, Center for Intelligent Information Retrieval (CIIR), University of Massachusetts, 2006.

[36] X. Liu and W. B. Croft. Evaluating text representations for retrieval of the best group of documents. In *Proceedings of ECIR*, pages 454–462, 2008.

[37] M. Mitra, A. Singhal, and C. Buckley. Improving automatic query expansion. In *Proceedings of SIGIR*, pages 206–214, 1998.

[38] M. Montague and J. A. Aslam. Condorcet fusion for improved retrieval. In *Proceedings of CIKM*, pages 538–548, 2002.

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

[40] T. Saracevic and P. Kantor. A study of information seeking and retrieving. iii. searchers, searches, and overlap. *Journal of the American Society for Information Science*, 39(3):197–216, 1988.

[41] S. B. Selvadurai. Implementing a metsearch framework with content-directed result merging. Master's thesis, North Carolina State University, 2007.

[42] I. Soboroff, C. K. Nicholas, and P. Cahan. Ranking retrieval systems without relevance judgments. In *Proceedings of SIGIR*, pages 66–73, 2001.

[43] T. Tao and C. Zhai. Regularized esitmation of mixture models for robust pseudo-relevance feedback. In *Proceedings of SIGIR*, pages 162–169, 2006.

[44] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, second edition, 1979.

[45] E. M. Voorhees. The cluster hypothesis revisited. In *Proceedings of SIGIR*, pages 188–196, 1985.

[46] E. M. Voorhees and D. K. Harman. *TREC: Experiments and evlautaion in information retrieval*. The MIT Press, 2005.

[47] P. Willett. Query specific automatic document classification. *International Forum on Information and Documentation*, 10(2):28–32, 1985.

[48] L. Yang, D. Ji, G. Zhou, Y. Nie, and G. Xiao. Document re-ranking using cluster validation and label propagation. In *Proceedings of CIKM*, pages 690–697, 2006.

[49] O. Zamir and O. Etzioni. Web document clustering: a feasibility demonstration. In *Proceedings of SIGIR*, pages 46–54, 1998.

[50] C. Zhai and J. D. Lafferty. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of SIGIR*, pages 334–342, 2001.

[51] B. Zhang, H. Li, Y. Liu, L. Ji, W. Xi, W. Fan, Z. Chen, and W.-Y. Ma. Improving web search results using affinity graph. In *Proceedings of SIGIR*, pages 504–511, 2005.

[52] L. Zighelnic and O. Kurland. Query-drift prevention for robust query expansion. In *Proceedings of SIGIR*, pages 825–826, 2008.