Utilizing inter-passage and inter-document similarities for re-ranking search results

EYAL KRIKON and OREN KURLAND — Faculty of Industrial Engineering and Management, Technion — Israel Institute of Technology

and

MICHAEL BENDERSKY — Center for Intelligent Information Retrieval, Department of Computer Science, University of Massachusetts

We present a novel language-model-based approach to *re-ranking* search results; that is, re-ordering the documents in an initially retrieved list so as to improve precision at top ranks. Our model integrates whole-document information with that induced from *passages*. Specifically, inter-passage, inter-document, and query-based similarities, which constitute a rich source of information, are combined in our model. Empirical evaluation shows that the precision-at-top-ranks performance of our model is substantially better than that of the initial ranking upon which re-ranking is performed. Furthermore, the performance is substantially better than that of a commonly-used passage-based document ranking method that does not exploit inter-item similarities. Our model also generalizes and outperforms a recently-proposed re-ranking method that utilizes inter-document similarities, but which does not exploit passage-based information. Finally, the model's performance is superior to that of a state-of-the-art pseudo-feedback-based retrieval approach.

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

General Terms: Algorithms, Experimentation

Additional Key Words and Phrases: ad hoc retrieval, re-ranking, passage-based retrieval, inter-passage similarities, inter-document similarities, document centrality, passage centrality

1. INTRODUCTION

Users of search engines expect to see the documents most pertaining to their queries at the very high ranks of the returned results. One paradigm for addressing this challenge, which has attracted quite a lot of research attention lately, is based on automatically *re-ranking* the documents in an initially retrieved list so as to improve precision at top ranks (e.g., Willett [1985], Kleinberg [1997], Liu and Croft [2004],

Portions of this work appeared in Krikon et al. [2009].

Authors' addresses: Eyal Krikon and Oren Kurland, Faculty of Industrial Engineering and Management, Technion — Israel Institute of Technology, Haifa 32000, Israel; email: krikon@tx.technion.ac.il, kurland@ie.technion.ac.il. Michael Bendersky, Center for Intelligent Information Retrieval, Department of Computer Science, University of Massachusetts, Amherst MA 01003; email: bemike@cs.umass.edu. Email: krikon@tx.technion.ac.il; kurland@ie.technion.ac.il; bemike@cs.umass.edu

Diaz [2005], Kurland and Lee [2005], Kurland and Lee [2006], Liu and Croft [2006], and Yang et al. [2006]).

An information source often utilized by these re-ranking methods is *inter-document* similarities. For example, documents that are similar to many other documents in the list, and hence, are considered as *central*, have higher chances of being relevant by the virtue of the way the list was created [Kurland and Lee 2005]; that is, in response to the query at hand.

However, an issue often not accounted for in re-ranking methods that utilize inter-document similarities is that long and/or heterogeneous relevant documents could contain many parts (*passages*) that have no query-related information. This issue is addressed by approaches that use passage-based information for ranking *all* documents in the corpus [Salton et al. 1993; Callan 1994; Kaszkiel and Zobel 2001; Liu and Croft 2002]. Nevertheless, these passage-based methods do not exploit similarity relationships between documents, nor between their passages — a potentially rich source of information for re-ranking as noted above.

We present a novel language-model-based approach to re-ranking an initially retrieved list. Our model leverages the strengths of the two research directions just described: utilizing inter-item similarities and exploiting passage-based information.

We derive our model by using passages as proxies for documents. Passage-based information can help, for example, in estimating the document-query similarity, because passages could be considered as more coherent units than documents. More specifically, our model integrates query-similarity information induced from documents and passages with passage and document centrality information; the latter is induced from inter-passage and inter-document similarities, respectively. Using an array of experiments, we study the relative contribution of these various types of information to the overall effectiveness of our model. Among the key findings that we present is that inter-passage-similarities can often be a much more effective source of information for document re-ranking than passage-query similarities. The latter have been the focus of most previous work on passage-based document retrieval.

Empirical evaluation shows that the precision-at-top-ranks performance obtained by our model is substantially better than that of the initial ranking upon which re-ranking is performed. Furthermore, the attained performance is often substantially better than that of a commonly used passage-based ranking method that does not utilize inter-document and inter-passage similarities. In addition, we show that our model generalizes and outperforms a previously-proposed re-ranking method that utilizes inter-document similarities, but which does not exploit passage-based information. Finally, we show that our model is also superior to a state-of-the-art pseudo-feedback-based query expansion method, namely, the relevance model [Lavrenko and Croft 2001].

2. RETRIEVAL FRAMEWORK

Let q, d and $\mathcal{D}$ denote a query, a document, and a corpus of documents, respectively. We use g to denote a passage, and write $g \in d$ if g is part of d. The model we present is not committed to any specific type of passages. To measure similarity

between texts x and y, we use a language-model-based estimate $p_x(y)$; we describe our language-model induction method in Section 4.1.

Our goal is to re-rank an initial list $\mathcal{D}_{\text{init}}$ ($\mathcal{D}_{\text{init}} \subset \mathcal{D}$) of documents, which was retrieved in response to q by some search algorithm, so as to improve precision at top ranks.

2.1 Passage-based document re-ranking

We take a probabilistic approach and rank document d in $\mathcal{D}_{\text{init}}$ by the probability $p(d|q)$ of its being relevant to the information need expressed by q. This probability can be written as

$$p(d|q) = \frac{p(q|d)p(d)}{p(q)}. \quad (1)$$

We interpret $p(q|d)$ as the probability that q can serve as a summary of d's content, or in other words, the probability of "generating" the terms in q from a model induced from d (cf., the language modeling approach to retrieval [Ponte and Croft 1998; Croft and Lafferty 2003])[1]; $p(d)$ is the prior probability of d being relevant; $p(q)$ is q's prior probability.

Since passages are shorter than documents, and hence, are often considered as more focused (coherent) units, they can potentially aid in generating summaries that are more "informative" than those generated from whole documents. Indeed, it has long been acknowledged that passages can serve as effective proxies for estimating the document-query match ($p(q|d)$ in our case), especially for long and/or heterogeneous documents [Salton et al. 1993; Callan 1994; Wilkinson 1994; Kaszkiel and Zobel 1997; 2001]. Following this observation we develop our passage-based document re-ranking model in Section 2.1.1.

The derivation of our model is inspired, among others, by a recently proposed method of ranking *document clusters* by the presumed percentage of relevant documents that they contain [Kurland 2008]. This cluster-ranking model uses documents as proxies for clusters. Given that the cluster-document relationship is conceptually reminiscent of the document-passage relationship (i.e., a unit vs. smaller units it is composed of), it turns out that some of the principles for ranking clusters can be adapted to *re-ranking* documents. We hasten to point out, however, that the assumptions that guide our model derivation are often different than those used in the cluster ranking setting [Kurland 2008], as the document-passage scenario is different than that of the cluster-document. A case in point, document clusters are sets of documents that are *similar* to each other by the virtue of the way the clusters are defined. In contrast, passages in documents need not necessarily be similar to each other, as is the case for heterogeneous/long documents that we opt to tackle here.

In addition, we note that one of the main novel aspects of the re-ranking model that we present is the use of inter-passage-similarities computed within and across documents. As we show in Section 4.3, these similarities can often be a much

[1]While it is convenient to use the term "generate" in reference to work on utilizing language models for IR, we do not assume an underlying generative theory as opposed to Lavrenko and Croft [2001] and Lavrenko [2004], inter alia.

more effective source of information for document re-ranking than passage-query-similarities, although the integration of the two yields performance superior to that of using each alone. Passage-query similarity has traditionally been the focus of work on passage-based document retrieval.

2.1.1 *Model derivation.* We use $p(g_i|d)$ to denote the probability that some passage g_i in the corpus is chosen as d's proxy for "query generation" ($\sum_{g_i} p(g_i|d) = 1$); $p(g_i)$ will be used to denote the prior probability of choosing g_i as a query generator — i.e., the prior probability of g_i being relevant to *any* query. Using probability algebra, Equation 1 then becomes

$$p(d|q) = \frac{p(d)}{p(q)} \sum_{g_i} p(q|d, g_i)p(g_i|d). \tag{2}$$

To estimate $p(q|d, g_i)$, the probability of generating q from d and g_i, we use a simple mixture of the probabilities of generating q from each [Bendersky and Kurland 2008b]:

$$\lambda p(q|d) + (1 - \lambda)p(q|g_i); \tag{3}$$

λ is a free parameter. Using this estimate in Equation 2 we get

$$\frac{p(d)}{p(q)} \sum_{g_i} [\lambda p(q|d) + (1 - \lambda)p(q|g_i)]p(g_i|d).$$

Since $\sum_{g_i} p(g_i|d) = 1$, we can apply some probability algebra so as to get the following ranking principle for documents:[2]

$$Score(d) \stackrel{def}{=} \lambda \frac{p(d)p(q|d)}{p(q)} + \frac{(1 - \lambda)}{p(q)} \sum_{g_i} p(q|g_i)p(d|g_i)p(g_i). \tag{4}$$

Equation 4 scores d using a two component mixture model. The first is based on the probability of generating q directly from d and on the prior probability of d being relevant. The second component is based on the probability of generating q from passages. The relative impact of passage g_i depends on its (i) prior probability of being a query generator ($p(g_i)$), (ii) "association" with d ($p(d|g_i)$), and (iii) probability of generating q ($p(q|g_i)$). Indeed, if g_i is strongly associated with (e.g., textually similar to) d, and it is a-priori a good candidate for query generation, then it can serve as d's "faithful" proxy; yet, g_i can potentially be more effective than d for query generation by the virtue of being more focused.

To alleviate the computational cost of estimating Equation 4, we make the assumption that d's most effective proxies are its passages — i.e., that $p(d|g_i)$ is much larger for passages in d than for passages not in d. Consequently, we truncate the summation in Equation 4 to yield:

$$Score(d) \stackrel{def}{=} \lambda \frac{p(d)p(q|d)}{p(q)} + \frac{(1 - \lambda)}{p(q)} \sum_{g_i \in d} p(q|g_i)p(d|g_i)p(g_i). \tag{5}$$

[2]The shift in notation and terminology from "$p(d|q)$" to "score of d" echoes the transition from using (model) probabilities to estimates of such probabilities.

Equation 5 considers each of d's passages, g_i, to the extent it represents d as measured by $p(d|g_i)$. However, in some relevance judgment regimes, such as that of TREC's [Voorhees and Harman 2005], little evidence for relevance in d is enough for deeming d relevant. Thus, the ranking principle in Equation 5 can fall short in such cases; specifically, if d is a long (heterogeneous) document with only a single passage containing query-pertaining information. To address this issue, we truncate the summation in Equation 5 by using only d's passage for which the "evidence" for relevance, as manifested in $p(q|g_i)p(g_i)$, is the strongest. Since this passage could be a relatively weak representative of d (as measured by $p(d|g_i)$), and d could still be deemed relevant, we exploit the fact that $p(d|g_i) \leq 1$ so as to use an upper bound of the truncated sum:

$$Score(d) \stackrel{def}{=} \lambda \frac{p(d)p(q|d)}{p(q)} + \frac{(1-\lambda)}{p(q)} \max_{g_i \in d} p(q|g_i)p(g_i). \quad (6)$$

We now turn to complete the derivation of our ranking model by addressing the query prior, $p(q)$. It is a fact that $p(q) = \sum_{d'} p(q|d')p(d')$. If we assume, as in work on pseudo-feedback-based retrieval [Lavrenko and Croft 2003], that $p(q|d')$ is much higher for documents in $\mathcal{D}_{\text{init}}$ (the initially retrieved list of documents upon which re-ranking is performed) than for documents not in $\mathcal{D}_{\text{init}}$ — which holds, for example, if $p(q|d')$ represents the surface-level match between d' and q — then we can use the approximation:

$$p(q) \approx \sum_{d' \in \mathcal{D}_{\text{init}}} p(q|d')p(d'). \quad (7)$$

Now, the scoring function in Equation 6 provides an estimate for $p(q|d')p(d')$, that is, $\lambda p(d')p(q|d') + (1-\lambda)\max_{g' \in d'} p(q|g')p(g')$, by the virtue of the way Equation 6 was derived (from Equation 1). Thus, we can set $\lambda = 1$ so as to get a document-based estimate for $p(q)$ using Equation 7; or, set $\lambda = 0$ so as to get a passage-based estimate for $p(q)$ using Equation 7. Note that the former case corresponds to the assumption that a query is independent of a passage given a document, and the latter corresponds to the assumption that the query is independent of a document given a passage (see Equation 3). Using these two query-prior estimates in Equation 6, for the document and passage components, respectively, we get our primary ranking principle:

$$Score(d) \stackrel{def}{=} \lambda \frac{p(d)p(q|d)}{\sum_{d' \in \mathcal{D}_{\text{init}}} p(d')p(q|d')} + (1-\lambda)\frac{\max_{g_i \in d} p(q|g_i)p(g_i)}{\sum_{d' \in \mathcal{D}_{\text{init}}} \max_{g' \in d'} p(q|g')p(g')}. \quad (8)$$

Note that both the document component and the passage component in Equation 8 constitute probability distributions over $\mathcal{D}_{\text{init}}$. Consequently, $Score(d)$ is a probability distribution over $\mathcal{D}_{\text{init}}$ as well.

Furthermore, we note that previously proposed passage-based document ranking approaches of the form $Score(d) \stackrel{def}{=} \lambda sim(d,q) + (1-\lambda)\max_{g \in d} sim(g,q)$, where $sim(\cdot,\cdot)$ is an inter-text similarity measure [Buckley et al. 1994; Callan 1994; Wilkinson 1994; Cai et al. 2004; Bendersky and Kurland 2008b], can be viewed as specific instantiations of Equation 8. Specifically, if $p(q|x)$, where x is either a document or a passage, is estimated using some similarity measure, and the document and passage priors are assumed to be uniform, then these previous approaches can

be derived from Equation 8. In the next section we discuss the potential use of non-uniform priors, and demonstrate their merits in Section 4.3.

2.1.2 *Algorithm.* The scoring function in Equation 8 is not committed to any specific paradigm of estimating probabilities. Following common practice in work on language models for IR [Liu and Croft 2002; Croft and Lafferty 2003] we use the language-model-based estimate $p_x(y)$ for $p(y|x)$.

The remaining task for deriving a specific algorithm from Equation 8 is estimating the document and passage priors, $p(d)$ and $p(g)$, respectively. The standard choice would be a uniform distribution as is common in the language modeling framework [Croft and Lafferty 2003].

However, previous work on re-ranking search results has demonstrated the merits of using a measure of the *centrality* of a document with respect to $\mathcal{D}_{\text{init}}$ as a *document bias* [Kurland and Lee 2005]. Specifically, a document is considered central if it is similar to many other (central) documents in $\mathcal{D}_{\text{init}}$. The idea is that central documents reflect the context of $\mathcal{D}_{\text{init}}$, which was retrieved in response to the query, and, hence, have high chances of being relevant. Indeed, various centrality measures were shown to be connected with relevance [Kurland and Lee 2005; 2006]. Following the same line of reasoning we hypothesize that *passage centrality* with respect to $\mathcal{D}_{\text{init}}$ is an indicator for passage relevance, that is, for the passage "ability" to serve as an effective query generator. Specifically, we consider a passage of a document in $\mathcal{D}_{\text{init}}$ to be central to the extent it is similar to many other passages of documents in $\mathcal{D}_{\text{init}}$.

To derive specific document and passage centrality measures, we adopt a previously-proposed method for document-centrality induction [Kurland and Lee 2005]. We construct two weighted graphs — a document-only graph that is composed of documents in $\mathcal{D}_{\text{init}}$, and a passage-only graph that is composed of all passages of documents in $\mathcal{D}_{\text{init}}$. Edge weights in these graphs represent inter-item similarities. We use the PageRank [Brin and Page 1998] score of item x with respect to its ambient graph as its centrality value $Cent(x)$. For documents not in $\mathcal{D}_{\text{init}}$ and for passages that are not parts of documents in $\mathcal{D}_{\text{init}}$ we set $Cent(x) \stackrel{def}{=} 0$. Consequently, $Cent(d)$, which will serve as the document bias $p(d)$, and $Cent(g)$, which will serve as the passage bias $p(g)$, are probability distributions over all documents in the corpus, and over all passages of documents in the corpus, respectively[3]. Details of the graph-construction and centrality induction methods [Kurland and Lee 2005] are provided in Appendix A.

[3]We use the term "bias" and not "prior" as these are not "true" prior-distributions, because of the virtue by which $\mathcal{D}_{\text{init}}$ was created, that is, in response to the query. Nevertheless, these biases form valid probability distributions by construction.

Using the estimates and measures described above we can now fully instantiate Equation 8 so as to produce our primary (re-)ranking algorithm, **PsgAidRank**:

$$Score_{PsgAidRank}(d) \stackrel{def}{=} \lambda \frac{Cent(d)p_d(q)}{\sum_{d' \in \mathcal{D}_{\text{init}}} Cent(d')p_{d'}(q)} + (1-\lambda)\frac{\max_{g_i \in d} p_{g_i}(q) Cent(g_i)}{\sum_{d' \in \mathcal{D}_{\text{init}}} \max_{g' \in d'} p_{g'}(q) Cent(g')}. \quad (9)$$

Setting $\lambda = 1$ results in a recently-proposed re-ranking method [Kurland and Lee 2005], which utilizes document centrality and document-query generation information, but which does not utilize passage-based information. Hence, this method is a specific instance of PsgAidRank, which utilizes centrality information and query-generation information induced from both the document as a whole and its passages. We present an in-depth study of the relative effect of these information types on the overall effectiveness of PsgAidRank in Section 4.3.

3. RELATED WORK

The most commonly used methods for passage-based document retrieval are ranking a document by the highest query-similarity score assigned to any of its passages [Callan 1994; Wilkinson 1994; Kaszkiel and Zobel 2001; Liu and Croft 2002; Bendersky and Kurland 2008b; Na et al. 2008]; and, by interpolating this similarity score with a document-query similarity score [Buckley et al. 1994; Callan 1994; Wilkinson 1994; Cai et al. 2004; Bendersky and Kurland 2008b]. We showed in Section 2.1.1 that these approaches are specific instances of our ranking model if uniform document and passage priors are utilized. We demonstrate the relative merits of PsgAidRank, which uses centrality measures for priors, in Section 4.3.

There is a large body of work on using graph-based approaches for (re-)ranking documents (e.g., Diaz [2005], Baliński and Daniłowicz [2005], Kurland and Lee [2005], Zhang et al. [2005], Kurland and Lee [2006], Yang et al. [2006], Bendersky and Kurland [2008a], and Mei et al. [2008]). Most of these methods utilize document-solely graphs, while some use cluster-document [Kurland and Lee 2006] and passage-document [Bendersky and Kurland 2008a] graphs. We compare PsgAidRank's performance with that of using these graphs in Section 4.3.2.

Information induced from clusters of similar documents in the initial list was also utilized for re-ranking (e.g., Willett [1985], Liu and Croft [2004], Liu and Croft [2006], Kurland and Lee [2006], Kurland [2008], and Liu and Croft [2008]). As mentioned at the above, our method of using passages as proxies for documents is inspired by a method that uses documents as proxies for clusters for the task of *cluster ranking* [Kurland 2008]. We discuss and compare this approach with PsgAidRank in Section 4.3.2.

Some work on automatic summarization, question answering, and clustering utilizes passage-only graphs for inducing passage centrality in ways somewhat similar to ours [Erkan and Radev 2004; Mihalcea 2004; Mihalcea and Tarau 2004; Otterbacher et al. 2005; Erkan 2006]. Inter-passage similarities, along with passage-document similarities, were also utilized in work on *query-by-example* [Wan et al. 2008]. Since most of these tasks are quite different than ours, inter-document

and document-query similarities were not used, in contrast to the case with PsgAidRank. We demonstrate the importance of document-based similarities for the re-ranking task in Section 4.3.1.

Inter-passage similarities *within* a document were also used for inducing passage language models [Bendersky and Kurland 2008b], and for devising discriminative passage-based document retrieval models [Wang and Si 2008]; the latter uses an initial standard passage-based document ranking, and therefore can potentially benefit from using PsgAidRank instead. Furthermore, we note that PsgAidRank uses inter-passage similarities both *across* and *within* documents.

Recent work on passage-based document retrieval [Bendersky and Kurland 2008b] resembles ours in that it uses passages as proxies for documents. In contrast to PsgAidRank, the proposed method does not utilize document and passage centrality, the importance of which we demonstrate in Section 4.3. The major performance gains in this work stem from using a novel passage language model that can be used by PsgAidRank so as to potentially improve its performance.

There is a large body of work on identifying (and using) different passage types [Hearst and Plaunt 1993; Callan 1994; Mittendorf and Schäuble 1994; Wilkinson 1994; Kaszkiel and Zobel 1997; Ponte and Croft 1997; Denoyer et al. 2001; Cai et al. 2004; Jiang and Zhai 2004], and on inducing passage language models [Liu and Croft 2002; Abdul-Jaleel et al. 2004; Hussain 2004; Murdock and Croft 2005; Wade and Allan 2005; Bendersky and Kurland 2008b] for various tasks. These could be used by PsgAidRank, which is not committed to a specific type of passage and passage model, to potentially improve its performance.

4. EVALUATION

The following evaluation is designed to explore the effectiveness of PsgAidRank in re-ranking, and to study the impact of various factors on its performance.

4.1 Inter-item-similarity estimate

Let $p_x^{Dir[\mu]}(\cdot)$ denote the unigram, Dirichlet-smoothed, language model induced from text x, where μ is the smoothing parameter [Zhai and Lafferty 2001]. To compensate for the length-bias, and consequently underflow issues, incurred by the use of unigram language models when assigning probabilities to long texts [Lavrenko et al. 2002; Kurland and Lee 2005], we adopt a previously-proposed estimate [Kurland and Lee 2005; 2006]

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

D is the KL divergence. This estimate was mathematically shown to compensate for length-bias [Lafferty and Zhai 2001; Kurland and Lee 2005], and empirically demonstrated to be effective in various re-ranking models [Kurland and Lee 2005; 2006].

We note that the estimate just described does not constitute a probability distribution, as is the case for probabilities assigned by unigram language models to term sequences. However, the resultant estimates that it is used for in Equation 8 (the document and passage score components) are valid probability distributions

as noted in Section 2.1.1. Hence, we use the estimate as is, and as was done in some other re-ranking models [Kurland and Lee 2005; 2006].

4.2 Experimental setup

We conducted experiments using the following TREC corpora. Several of these were used in work on re-ranking methods [Kurland and Lee 2005; 2006; Bendersky and Kurland 2008a; Kurland 2008] with which we compare our model.

corpus	# of docs	avg. doc length	queries	disks
AP	242,918	464	51-150	1-3
FR	45,820	1498	51-150	1-2
TREC8	528,155	481	401-450	4-5
WSJ	173,252	452	151-200	1-2
WT10G	1,692,096	611	451-550	WT10G

AP and WSJ are news corpora. TREC8, which is considered a hard benchmark [Voorhees 2005], is mainly composed of news documents, but also contains federal register records. FR is composed of only federal register records. Due to the high average document length in FR, passage-based document retrieval methods are often more effective than whole-document-based retrieval approaches for this corpus [Callan 1994; Liu and Croft 2002; Bendersky and Kurland 2008b]. WT10G is a Web corpus.

We used titles of TREC topics for queries[4]. We applied tokenization and Porter stemming via the Lemur toolkit[5], which was also used for language-model induction.

We set $\mathcal{D}_{\text{init}}$, the list upon which re-ranking is performed, to be the 50 highest-ranked documents by an *initial ranking* induced over the corpus using $p_d(q)$ — i.e., a standard language model approach; the document language model smoothing parameter (μ) is set to a value optimizing MAP (at 1000) so as to yield an initial ranking of a reasonable quality. Such initial-list construction also facilitates the comparison with previous re-ranking models that use the same approach [Kurland and Lee 2005; 2006]. We note that exploiting information induced from inter-document-similarities among top-retrieved documents — specifically, by using centrality measures as we do here — was shown to be most effective when the initially retrieved list is quite short [Diaz 2005; Kurland 2006] (i.e., when the documents in the list exhibit relatively high query similarity, and hence, could be thought of as providing a "good" corpus context for the query).

The goal of re-ranking methods is to improve precision at top ranks. Therefore, we use the precision of the top 5 and 10 documents (p@5, p@10) for evaluation metrics. Statistically significant differences of performance are determined using the two-tailed Wilcoxon test at a 95% confidence level.

Our first goal is to perform an in-depth exploration of the performance of PsgAidRank, and the relative impact of the different information types that it leverages, regardless of the question of whether effective values of the free parameters that it incorporates generalize across queries. To that end, we start by neutralizing free-parameter effects. That is, following some previous work on graph-based

[4]Queries with no relevant documents were not considered.
[5]www.lemurproject.org

re-ranking [Kurland and Lee 2005; 2006; Bendersky and Kurland 2008a] we set the values of the free parameters of PsgAidRank, and those of all reference comparisons, so as to optimize the *average* p@5 performance over the *entire* set of given queries per corpus[6]. Then, in Section 4.3.1 we present the effect of varying the values of these parameters on PsgAidRank's performance. Finally, in Section 4.3.3 we present the performance of PsgAidRank, and that of the reference comparisons, when the values of all free parameters are set using a leave-one-out cross validation procedure performed over the query set for each corpus.

We set λ, the interpolation parameter of PsgAidRank, to a value in $\{0, 0.1, \ldots, 1\}$. The centrality induction method that we use incorporates two free parameters (see Appendix A): the graph out-degree, m, is set in *both* the document and passage graphs to α percent of the number of nodes in the graph, where $\alpha \in \{4, 8, 18, 38, 58, 78, 98\}$; PageRank's smoothing factor, δ, is set for both graphs to a value in {0.05,0.1,0.2,. . . ,0.9,0.95}. The ranges for the graph parameters' values were chosen to comply with previous work on graph-based re-ranking [Kurland and Lee 2005; 2006]. Thus, PsgAidRank incorporates three free parameters (λ, α, and δ). The document and passage language models smoothing parameter, μ, is set to 2000 in all the methods we consider following previous recommendations [Zhai and Lafferty 2001], except for the estimate for $p_d(q)$ where we use the value chosen to create $\mathcal{D}_{\text{init}}$ so as to maintain consistency with the initial ranking.

While there are several types of passages we can implement our model with [Kaszkiel and Zobel 2001], our focus is on the effectiveness of the underlying principles of our ranking approach. Hence, we use half-overlapping fixed window passages of 150 terms that are marked prior to retrieval time. These passages were shown to be effective for document retrieval [Callan 1994; Wang and Si 2008], specifically, in the language model framework [Liu and Croft 2002; Bendersky and Kurland 2008b]. In Section 4.3.1 we study the effect of passage length on the performance of PsgAidRank.

Finally, it is important to note that the computational overhead incurred by our re-ranking method on top of the initial retrieval is not significant. Specifically, the document graph is composed of 50 nodes, and the passage graph contains a few hundred nodes at most. Thus, the Power method [Golub and Van Loan 1996] used for computing PageRank scores (i.e., centrality values) converges in a few iterations.

4.3 Experimental results

Our first order of business is evaluating the effectiveness of the PsgAidRank method in re-ranking the initial list $\mathcal{D}_{\text{init}}$ so as to improve precision at top ranks. Recall that the initial ranking used to create $\mathcal{D}_{\text{init}}$ is based on a language model approach ($p_d(q)$) wherein the smoothing parameter (μ) is set to optimize MAP. We therefore also compare PsgAidRank with *optimized baselines*, which use $p_d(q)$ to rank *all* documents in the corpus, and in which μ is set to optimize p@5 and p@10, independently. As can be seen in Table I, PsgAidRank consistently and substantially outperforms both the initial ranking and the optimized baselines; note that the performance of the optimized baselines is not statistically distinguishable from

[6]If two parameter settings yield the same p@5, we choose the one *minimizing* p@10 so as to provide conservative estimates of performance.

Table I. Comparison with the initial ranking and optimized baselines; 'i' and 'o' mark statistically significant differences with the former and the latter, respectively. The best result in a column is boldfaced.

	AP		FR		TREC8		WSJ		WT10G	
	p@5	p@10	p@5	p@10	p@5	p@10	p@5	p@10	p@5	p@10
init. rank.	45.7	43.2	24.8	18.5	50.0	45.6	53.6	48.4	33.9	28.0
opt. base.	46.5	43.9	24.8	18.9	51.2	46.4	56.0	49.4	34.1	28.2
PsgAidRank	$\mathbf{53.7}^{i}_{o}$	$\mathbf{49.5}^{i}_{o}$	**26.7**	**19.3**	$\mathbf{56.8}^{i}$	**48.0**	$\mathbf{58.0}^{i}$	**51.0**	$\mathbf{37.1}^{i}_{o}$	**30.0**

Table II. The MAP(@50) performance of a MAP-optimized version of PsgAidRank in comparison to that of the initial ranking. The best result in a column is boldfaced. Statistically significant differences with the initial ranking are marked with 'i'.

	AP	FR	TREC8	WSJ	WT10G
init. rank.	9.3	24.0	17.5	22.2	13.3
PsgAidRank	$\mathbf{10.0}^{i}$	**25.6**	$\mathbf{18.3}^{i}$	**23.7**	$\mathbf{14.2}^{i}$

that of the initial ranking. Moreover, the p@5 performance differences between PsgAidRank and the initial ranking are statistically significant for four out of the five corpora. These results attest to the effectiveness of PsgAidRank as a re-ranking method[7].

We posed PsgAidRank as a method for improving precision at top ranks of a given retrieved list. As such, it could also be expected to post MAP performance that is better than that of the initial ranking. Indeed, PsgAidRank's MAP-optimized version consistently improves (and often, to a statistically significant degree) on the initial ranking in terms of MAP@50 as can be seen in Table II.

Comparison with pseudo-feedback-based retrieval. The PsgAidRank method utilizes information from the initial list $\mathcal{D}_{\text{init}}$ for re-ranking it. Pseudo-feedback-based query expansion techniques [Buckley et al. 1994; Xu and Croft 1996], on the other hand, utilize information from $\mathcal{D}_{\text{init}}$ to construct a query model using which the corpus is (re-)ranked. To contrast the two paradigms, we compare the performance of PsgAidRank with that of *relevance model number 3* (RM3) [Lavrenko and Croft 2001; Abdul-Jaleel et al. 2004; Diaz and Metzler 2006], which is a state-of-the-art pseudo-feedback-based query expansion method. We also examine the performance of RM3(re-rank) that uses RM3 to re-rank $\mathcal{D}_{\text{init}}$ rather than the entire corpus.

We use Lemur's implementation of the relevance model. The values of the free parameters of RM3 and RM3(re-rank) are set to optimize p@5 (as is the case for PsgAidRank). Specifically, the value of the (Jelinek-Mercer) smoothing parameter used for relevance-model construction is chosen from $\{0, 0.1, 0.3, \ldots, 0.9\}$; the number of terms used by the models is chosen from $\{25, 50, 75, 100, 500, 1000, 5000, ALL\}$, where "*ALL*" stands for using all terms in the corpus; and, the interpolation

[7]For TREC8, for example, PsgAidRank's performance is top-tier with respect to that of systems that participated in TREC8 [Voorhees and Harman 2000]. The performance of the initial ranking on the other hand, which was re-ranked using PsgAidRank, is somewhat mediocre.

Table III. Comparison with a relevance model used either to rank all documents in the corpus (RM3) or to re-rank the initial list (RM3(re-rank)). Best result in a column is boldfaced; 'i' marks statistically significant differences with the initial ranking.

	AP		FR		TREC8		WSJ		WT10G	
	p@5	p@10	p@5	p@10	p@5	p@10	p@5	p@10	p@5	p@10
init. rank.	45.7	43.2	24.8	18.5	50.0	45.6	53.6	48.4	33.9	28.0
RM3	50.3^i	48.6^i	25.2	18.1	54.4	**49.4**	58.4^i	51.0	35.7	28.7
RM3(re-rank)	51.1^i	48.2^i	25.2	18.1	54.4	48.6	**58.8**^{i}	**52.0**	36.3	29.4
PsgAidRank	**53.7**^{i}	**49.5**^{i}	**26.7**	**19.3**	**56.8**^{i}	48.0	58.0^i	51.0	**37.1**^{i}	**30.0**

parameter that controls the reliance on the original query is set to a value in $\{0, 0.1, 0.2, \ldots, 0.9\}$. The (Dirichlet) document language model smoothing parameter (μ) used for ranking with a relevance model is set to 2000 as in all other methods.

Table III shows that the performance of PsgAidRank is superior in most cases to that of the relevance models. Although the performance of PsgAidRank and of the relevance models is not statistically distinguishable, PsgAidRank posts statistically significant p@5 improvements over the initial ranking for four out of the five corpora, while the relevance models posts statistically significant improvements for only two corpora. Thus, we see that PsgAidRank is a highly effective approach for obtaining high precision at top ranks.

4.3.1 *Deeper inside PsgAidRank .* We now turn to analyze the relative impact of different factors that affect the performance of PsgAidRank.

The PsgAidRank method utilizes different information types for ranking document d: (i) **DocQueryGen** — the possibility that d "generates" q ($p_d(q)$); this is exactly the estimate using which the initial ranking was created, (ii) **DocCent** — d's centrality ($Cent(d)$), (iii) **PsgQueryGen** — the possibility that d's passages "generate" q ($p_{g_i}(q)$), and (iv) **PsgCent** — the centrality of d's passages ($Cent(g_i)$).

Hence, our next goal is to explore the relative contribution of each of these information types, and their combinations, to the overall effectiveness of PsgAidRank. To that end, we apply the following manipulations to PsgAidRank: (i) setting λ to 1 (0) to have only the document (passages) generate q, (ii) using uniform distribution for $Cent(d)$ over $\mathcal{D}_{\text{init}}$ and/or for $Cent(g_i)$ over $G(\mathcal{D}_{\text{init}})$, where $G(\mathcal{D}_{\text{init}})$ is the set of all passages of documents in $\mathcal{D}_{\text{init}}$; this manipulation amounts to assuming that all documents in $\mathcal{D}_{\text{init}}$ and/or passages in $G(\mathcal{D}_{\text{init}})$ are central to the same extent, and (iii) fixing $p_d(q)$ ($p_{g_i}(q)$) to the same value, thereby assuming that all documents in $\mathcal{D}_{\text{init}}$ (passages in $G(\mathcal{D}_{\text{init}})$) have the same probability of generating q. For example, setting λ to 0 and $p_{g_i}(q)$ to some constant, we rank d by PsgCent — the maximal centrality value assigned to any of d's passages: $max_{g_i \in d} Cent(g_i)$. Table IV presents the resultant ranking methods that we explore ("$\wedge$" indicates that a method utilizes two types of information), and their performance.

Our first observation based on Table IV is that for AP, TREC8 and WSJ, centrality information is more effective for re-ranking than query-generation information; specifically, DocCent is as effective as DocQueryGen, PsgCent is superior to Ps-

Table IV. Performance analysis of the different information types utilized by PsgAidRank. Statistically significant differences with DocQueryGen (the method using which the initial ranking was induced) and with PsgAidRank are marked with 'i' and 'a', respectively. The best result in each column is boldfaced.

Method	$Score(d)$
DocQueryGen	$p_d(q)$
DocCent	$Cent(d)$
DocQueryGen∧DocCent	$Cent(d)p_d(q)$
PsgQueryGen	$\max_{g_i \in d} p_{g_i}(q)$
PsgCent	$\max_{g_i \in d} Cent(g_i)$
PsgQueryGen∧PsgCent	$\max_{g_i \in d} p_{g_i}(q) Cent(g_i)$
DocQueryGen∧PsgQueryGen	$\lambda \frac{p_d(q)}{\sum_{d' \in \mathcal{D}_{\text{init}}} p_{d'}(q)} + (1-\lambda) \frac{\max_{g_i \in d} p_{g_i}(q)}{\sum_{d' \in \mathcal{D}_{\text{init}}} \max_{g' \in d'} p_{g'}(q)}$
DocCent∧PsgCent	$\lambda \frac{Cent(d)}{\sum_{d' \in \mathcal{D}_{\text{init}}} Cent(d')} + (1-\lambda) \frac{\max_{g_i \in d} Cent(g_i)}{\sum_{d' \in \mathcal{D}_{\text{init}}} \max_{g' \in d'} p_{g'}(q) Cent(g')}$
PsgAidRank	$\lambda \frac{Cent(d) p_d(q)}{\sum_{d' \in \mathcal{D}_{\text{init}}} Cent(d') p_{d'}(q)} + (1-\lambda) \frac{\max_{g_i \in d} p_{g_i}(q) Cent(g_i)}{\sum_{d' \in \mathcal{D}_{\text{init}}} \max_{g' \in d'} p_{g'}(q) Cent(g')}$

Method	AP		FR		TREC8		WSJ		WT10G	
	p@5	p@10	p@5	p@10	p@5	p@10	p@5	p@10	p@5	p@10
DocQueryGen	45.7	43.2	24.8	18.5	50.0	45.6	53.6	48.4	33.9	28.0
DocCent	51.9	48.1	11.5^{i}_{a}	9.3^{i}_{a}	50.0	45.6	53.6	48.6	31.4	26.3
DocQueryGen∧DocCent	53.3^{i}	49.2^{i}	25.6	18.0	54.0^{i}	**48.0**^{i}	57.2^{i}	49.6	35.9	29.4
PsgQueryGen	46.1_{a}	41.7_{a}	24.8	19.4	44.8^{i}_{a}	43.0_{a}	48.8^{i}_{a}	44.6_{a}	32.9_{a}	29.3
PsgCent	50.1	46.6_{a}	19.3_{a}	15.7	52.4	46.2	56.0	50.8	23.1^{i}_{a}	23.1^{i}_{a}
PsgQueryGen∧PsgCent	50.9^{i}	45.5_{a}	25.6	**20.0**	52.4	46.8	54.8	47.8_{a}	34.3	29.8
DocQueryGen∧PsgQueryGen	46.3_{a}	42.1_{a}	**26.7**	**20.0**	50.4_{a}	46.2	54.8	48.8	36.1^{i}	29.7
DocCent∧PsgCent	52.3	48.0	20.4	14.1_{a}	55.2	46.2	56.0	50.8	31.4	26.3
PsgAidRank	**53.7**^{i}	**49.5**^{i}	**26.7**	19.3	**56.8**^{i}	**48.0**	**58.0**^{i}	**51.0**	**37.1**^{i}	**30.0**

gQueryGen, and DocCent∧PsgCent is superior to DocQueryGen∧PsgQueryGen. These results echo previous findings in reports on re-ranking using document centrality [Kurland and Lee 2005] and cluster centrality [Kurland and Lee 2006]. For FR and WT10G, however, the opposite holds — query-generation information is clearly more effective than centrality information. This finding could be attributed to the fact that documents in these two corpora are much longer than those in the other corpora. Hence, they might contain many passages that are non-query related. As document centrality and passage centrality are computed in a query-independent fashion — although the documents themselves are those retrieved in response to the query — non-query related aspects might have a significant impact on centrality induction. Nevertheless, integrating centrality and query-generation information yields in most reference comparisons for all corpora performance that is superior to that of using each alone; specifically, DocQueryGen∧DocCent is superior to DocQueryGen and DocCent, and PsgQueryGen∧PsgCent is often superior to PsgQueryGen and PsgCent.

In comparing the effectiveness of information induced from the document as a whole with that induced from its passages we can see that the results are corpus dependent. This finding echoes those in previous work on passage-based document retrieval [Callan 1994; Liu and Croft 2002; Bendersky and Kurland 2008b; Wang and Si 2008] — e.g., that passage-based document ranking methods are more effective for scoring heterogeneous documents, while whole-document-based methods are more effective for scoring homogeneous documents. Here, we find, for example, that for AP and WT10G, document-based information is more effective than passage-

based information. (Compare DocQueryGen with PsgQueryGen; DocCent with PsgCent; and, DocQueryGen∧DocCent with PsgQueryGen∧PsgCent.) For FR, on the other hand, information induced from passages is in general more effective than whole-document-based information. This finding, which is in accordance with those in previous reports [Callan 1994; Liu and Croft 2002; Bendersky and Kurland 2008b; Wang and Si 2008], is not surprising given the extremely large length of FR documents. For TREC8 and WSJ, information induced from the whole document is more effective than that induced from passages only when query-generation information is used. When query-generation information is not used the opposite holds. Perhaps the more important message rising from Table IV in that respect is that integrating information induced both from the document as a whole and from its passages is superior to using each alone in a vast majority of the relevant comparisons (corpus × evaluation measure); specifically, DocCent∧PsgCent is often superior to DocCent and PsgCent, and DocQueryGen∧PsgQueryGen is superior to both DocQueryGen and PsgQueryGen. This finding supports the merit of integrating passage-based and whole-document-based information — the underlying principle of our approach.

It is not a surprise, then, that PsgAidRank that integrates centrality and query-generation information that are induced both from the document as a whole and from its passages is the most effective method in most relevant comparisons among those in Table IV. Furthermore, the performance of PsgAidRank is in many cases statistically significantly better (and is never statistically significantly worse) than that of the other methods in Table IV.

Specifically, PsgAidRank is superior in most reference comparisons to DocQueryGen∧PsgQueryGen— a commonly used method for passage-based document ranking [Buckley et al. 1994; Callan 1994; Wilkinson 1994; Cai et al. 2004; Bendersky and Kurland 2008b], which we use here for re-ranking. Some of these performance differences are quite substantial (e.g., for AP, TREC8 and WSJ) and statistically significant (e.g., for AP and TREC8). Furthermore, PsgAidRank posts many more statistically significant improvements over the initial ranking than DocQueryGen∧PsgQueryGen does. These findings attest to the benefits of utilizing centrality information induced from inter-item similarities, as is done by PsgAidRank in contrast to DocQueryGen∧PsgQueryGen.

Balancing document-based and passage-based information. The reliance of PsgAidRank on whole-document-based versus passage-based information is controlled by the parameter λ. (Refer back to Equation 9 in Section 2.) Note that $\lambda = 0$ (i.e., using only passage-based information) amounts to PsgQueryGen∧PsgCent, and $\lambda = 1$ (i.e., using only document-based information) amounts to DocQueryGen∧DocCent. (See Table IV for details.) Figure 1 depicts the p@5 performance curve of PsgAidRank when varying λ; the initial ranking performance is drawn with an horizontal line for reference.

We can see in Figure 1 that integrating passage-based and document-based information is often superior to using each alone, as noted above. (Note that the best performance in each graph is obtained for $\lambda \neq 0, 1$.) Furthermore, for most values of λ, PsgAidRank posts performance that is substantially better than that of the initial ranking.

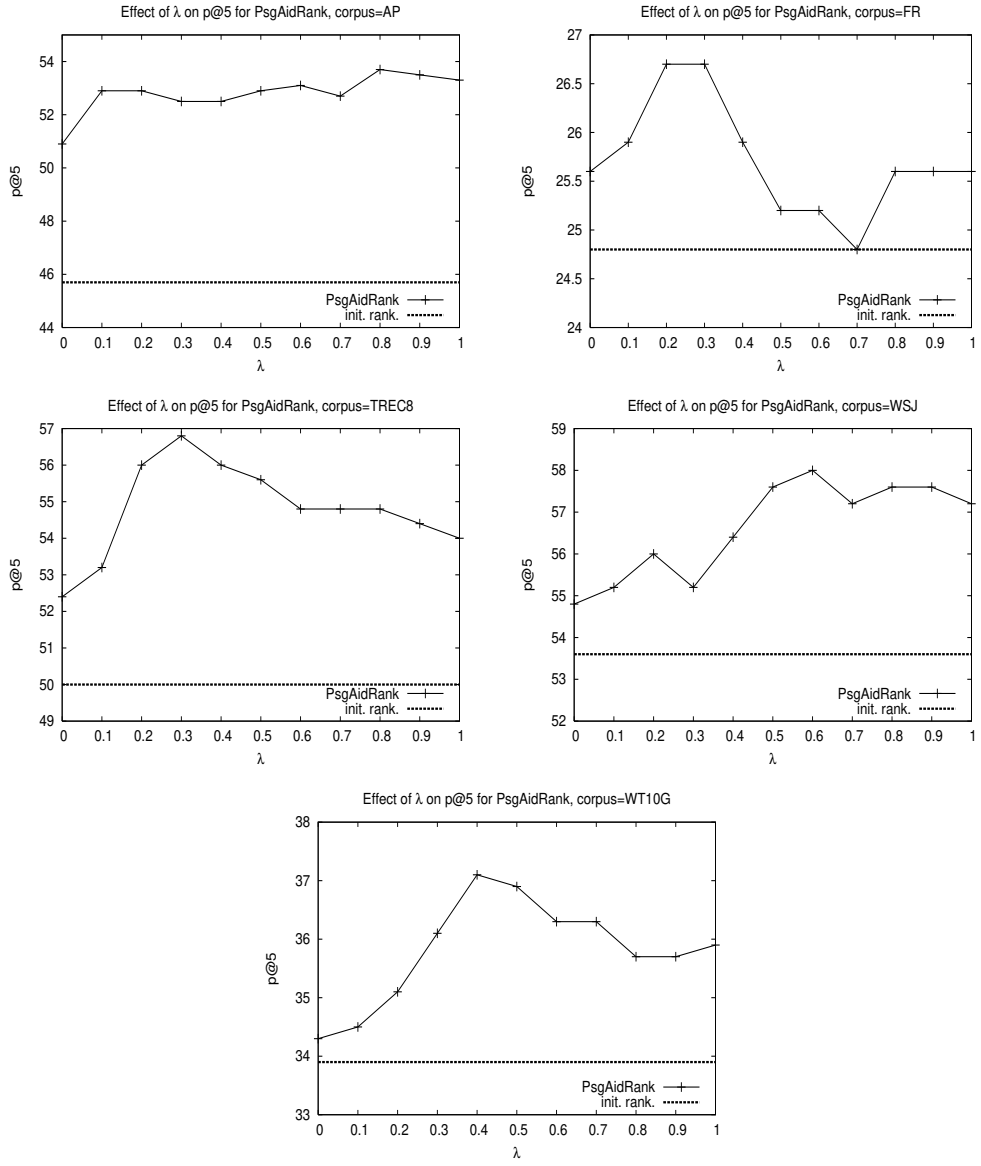

Fig. 1. Effect of varying λ on the p@5 performance of PsgAidRank; 0 and 1 correspond to PsgQueryGen∧PsgCent and DocQueryGen∧DocCent, respectively. The performance of the initial ranking is depicted with an horizontal line. Note: figures are not to the same scale.

Another observation that we make based on Figure 1 is that values of λ that yield optimal p@5 performance are not necessarily consistent *across corpora*. For FR and TREC8, optimal performance is attained for relatively low values of λ ($\{0.2, 0.3\}$) — i.e., putting much emphasis on passage-based information. For the

other corpora (especially AP and WSJ), higher values of λ, which reflect increased emphasis on whole-document-based information, yield optimal (or near optimal) performance. These findings echo those from above, and those from previous work [Callan 1994; Liu and Croft 2002; Bendersky and Kurland 2008b; Wang and Si 2008], with respect to the relative effectiveness of document-based and passage-based information being corpus dependent. We hasten to point out, however, that within a corpus, effective values of λ tend to generalize well *across queries* as the performance results that we present in Section 4.3.3 attest.

Parameters affecting centrality induction. We now turn to examine the effect of the two graph parameters, the out-degree percentage (α) and PageRank's smoothing factor (δ), which are used for centrality induction, on the performance of PsgAidRank.

We can see in Figures 2 and 3 that except for very high values of δ for the FR corpus, all values of α and δ yield performance that is better (often to a substantial extent) than that of the initial ranking. This finding attests to the relative performance robustness of PsgAidRank with respect to free-parameter values that affect centrality induction.

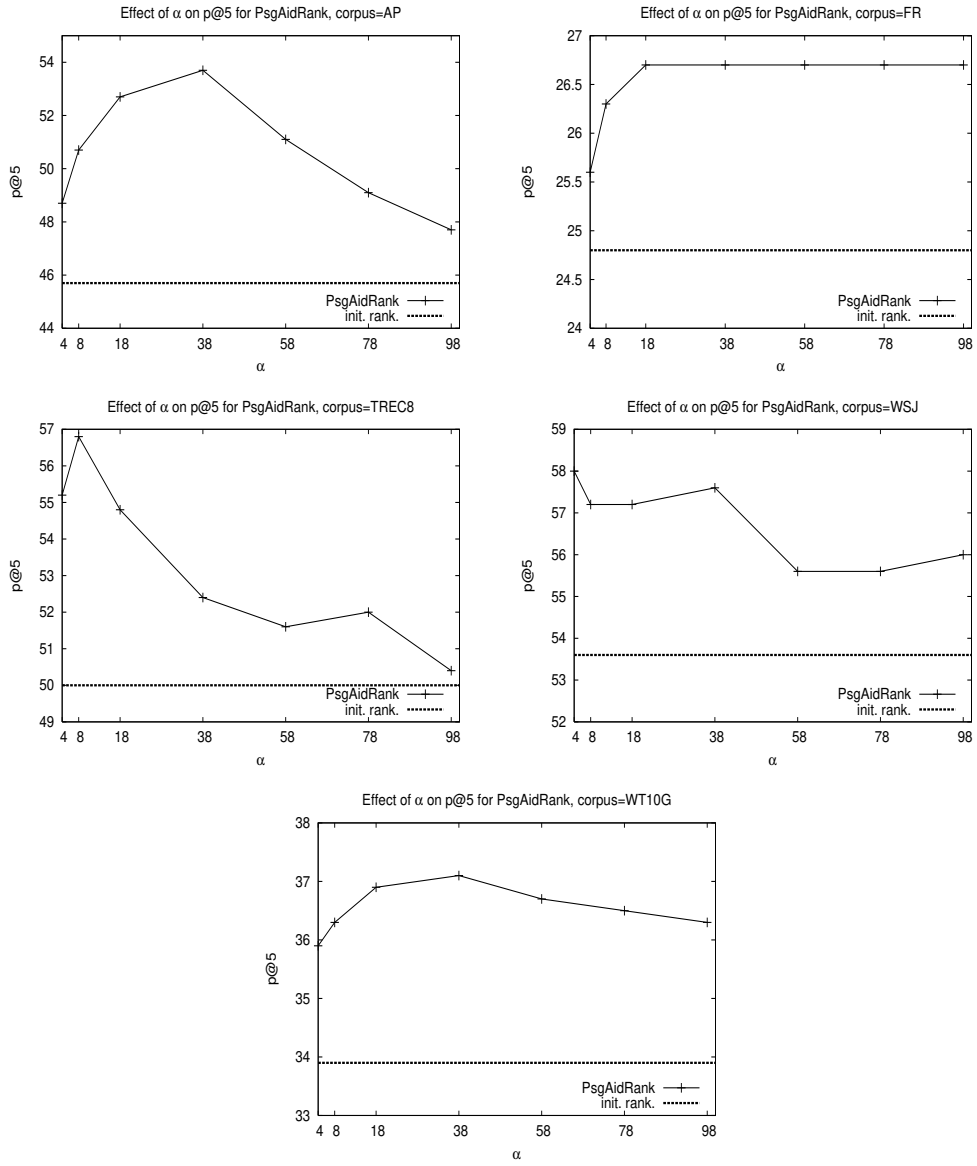

Fig. 2. Effect of varying α, the graph out-degree (percentage) parameter, on the p@5 performance of PsgAidRank. The performance of the initial ranking is depicted with an horizontal line. Note: figures are not to the same scale.

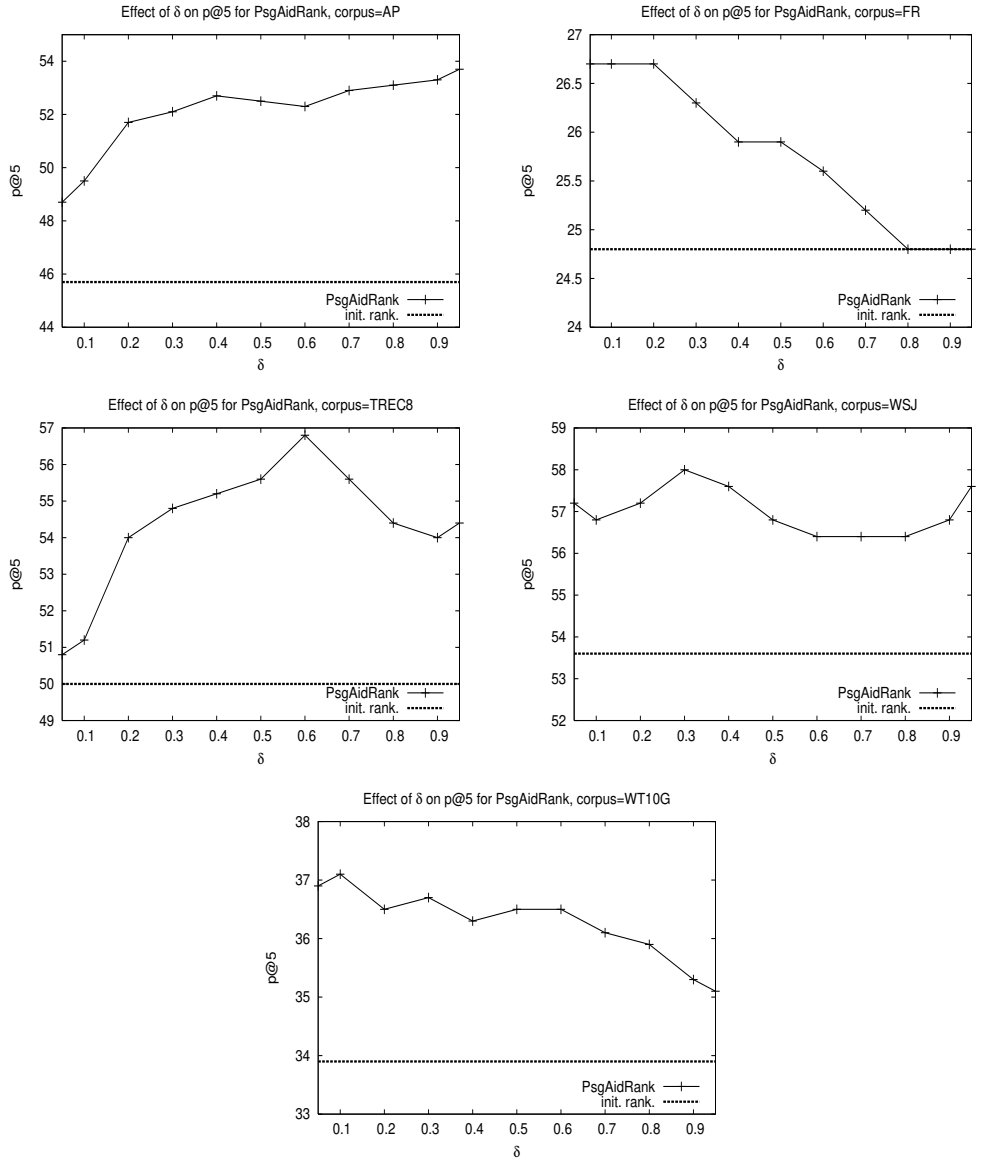

Fig. 3. Effect of varying PageRank's smoothing factor, δ, on the p@5 performance of PsgAidRank. The performance of the initial ranking is depicted with an horizontal line. Note: figures are not to the same scale.

Table V. Comparison of PsgAidRank with a method, PsgAidRank[AllPsg], which uses all passages in a document and their association with the document for scoring it. Best result in a column is boldfaced; 'i' marks statistically significant differences with the initial ranking.

	AP		FR		TREC8		WSJ		WT10G	
	p@5	p@10	p@5	p@10	p@5	p@10	p@5	p@10	p@5	p@10
init. rank.	45.7	43.2	24.8	18.5	50.0	45.6	53.6	48.4	33.9	28.0
PsgAidRank[AllPsg]	53.5^i	49.1^i	25.6	18.0	55.6	47.8	$\mathbf{61.6}^i$	$\mathbf{53.0}^i$	35.9	29.4
PsgAidRank	$\mathbf{53.7}^i$	$\mathbf{49.5}^i$	**26.7**	**19.3**	$\mathbf{56.8}^i$	**48.0**	58.0^i	51.0	$\mathbf{37.1}^i$	**30.0**

Using all passages in a document vs. using a single passage. We derived the PsgAidRank method in Section 2.1.1 by using a single passage from each document in the document scoring function. We did so by truncating the summation in Equation 5 so as to consider only the passage in the document for which the evidence for relevance is the highest. We now turn to examine the alternative of using *all* the passages in a document for scoring it. The resultant ranking method, henceforth referred to as **PsgAidRank[AllPsg]**, which is derived from Equation 5, scores document d by:

$$Score_{PsgAidRank[AllPsg]}(d) \stackrel{def}{=} \lambda \frac{Cent(d)p_d(q)}{\sum_{d' \in \mathcal{D}_{\text{init}}} Cent(d')p_{d'}(q)} + (1-\lambda)\frac{\sum_{g_i \in d} p_{g_i}(q)p_{g_i}(d)Cent(g_i)}{\sum_{d' \in \mathcal{D}_{\text{init}}} \sum_{g' \in d'} p_{g'}(q)p_{g'}(d')Cent(g')}. \quad (10)$$

Note that in PsgAidRank[AllPsg], in contrast to PsgAidRank, the passage-document association, $p_{g_i}(d)$, is also considered. The performance comparison of PsgAidRank with PsgAidRank[AllPsg] is presented in Table V.[8]

As we can see in table V, both PsgAidRank and PsgAidRank[AllPsg] consistently and substantially outperform the initial ranking. The performance improvements are often statistically significant. This finding further supports the merits of integrating document-based and passage-based information, and of using inter-item-similarities information, whether utilizing a single (most "query-pertaining") document passage or all document's passages in the scoring function.

We can also see in Table V that PsgAidRank is superior to PsgAidRank[AllPsg] in all corpora except for WSJ. However, the performance differences between the two methods are not statistically significant. Nevertheless, PsgAidRank posts p@5 — the metric for which performance was optimized — performance that is statistically significantly better than that of the initial ranking for four out of the five corpora, while PsgAidRank[AllPsg] does so for only two corpora. These findings demonstrate the merits of using in the document-scoring function a single passage from each document, for which query-relevance evidence is the "strongest", with respect to using all passages in the document.

[8]In the conference version of this paper [Krikon et al. 2009], a variant of PsgAidRank[AllPsg] was termed "PsgAidRank", and was the focus of the paper.

Table VI. The effect of passage length (number of terms in a window) on the performance of PsgAidRank. Best result in a column is boldfaced. Statistically significant differences with the initial ranking are marked with 'i'.

	AP		FR		TREC8		WSJ		WT10G	
	p@5	p@10	p@5	p@10	p@5	p@10	p@5	p@10	p@5	p@10
init. rank.	45.7	43.2	24.8	18.5	50.0	45.6	53.6	48.4	33.9	28.0
PsgAidRank (PsgLength = 50)	53.5^i	49.2^i	**28.5**	19.1	56.0^i	**48.6**^{i}	57.6^i	**51.0**	36.7	29.6
PsgAidRank (PsgLength = 150)	**53.7**^{i}	**49.5**^{i}	26.7	**19.3**	**56.8**^{i}	48.0	**58.0**^{i}	**51.0**	**37.1**^{i}	**30.0**

The effect of passage length. We now turn to study the effect of the passage length — the number of terms in a window which constitutes a passage — on the performance of PsgAidRank. Recall that insofar we have used passages of 150 terms. The effectiveness of PsgAidRank with passages of 50 terms in comparison to that of using 150 terms is presented in Table VI.

As we can see in Table VI, the performance of PsgAidRank is consistently better — often to a statistically significant degree — than that of the initial ranking with passages of 150 and 50 terms.

We can also see in Table VI that for most corpora, using passages of 150 terms in PsgAidRank yields somewhat better performance than that of using passages of 50 terms. The notable exception is the FR corpus for which the average document length is much higher than that for the other corpora. This finding could potentially be explained by the fact that the extremely long FR documents are often quite heterogeneous. Hence, using shorter passages might better help to capture "coherent" units in the document. Indeed, previous work on using passage-based document ranking methods for the FR corpus has also shown the merits of using passages of 50 terms with respect to using passages of 150 terms [Liu and Croft 2002; Bendersky and Kurland 2008b].

4.3.2 *Further comparisons.* The performance analysis of PsgAidRank demonstrated the important role of centrality information. We therefore compare PsgAidRank's performance with that of some previous centrality-based approaches for re-ranking. These methods utilize graph-based approaches wherein edge-weights represent inter-item similarities as in PsgAidRank.

The first method that we consider, **DocGraph** [Kurland and Lee 2005], scores d by $p_d(q)Cent(d)$ — i.e., no passage-based information is used. This is the DocQueryGen∧DocCent method from Table IV, which is a specific case of PsgAidRank with $\lambda = 1$, as mentioned in Section 2. We note that the graph out-degree parameter and PageRank's smoothing factor, which affect centrality induction, are set to values that optimize p@5, and which are selected from the same value-ranges used for the parameters of PsgAidRank, as was the case in Table IV.

Inducing document centrality using information induced from *query-specific* clusters (i.e., clusters of documents from the initial list $\mathcal{D}_{\text{init}}$) has also shown merit for re-ranking [Kurland and Lee 2006]. Specifically, d's *authority* value as computed over a bipartite cluster-document graph by the HITS (hubs and authorities) algorithm [Kleinberg 1997] serves as d's centrality value; to rank documents, the centrality value is scaled by the document-query similarity score ($p_d(q)$). We use **ClustDocGraph** to denote this method, and set the graph out-degree parameter

Table VII. Comparison of PsgAidRank with graph-based re-ranking methods that utilize document-only graphs (DocGraph [Kurland and Lee 2005]), cluster-document graphs (ClustDocGraph [Kurland and Lee 2006]), and passage-document graphs (PsgDocGraph [Bendersky and Kurland 2008b]). Statistically significant differences between a method and the initial ranking are marked with 'i'; statistically significant differences between PsgAidRank and ClusterDocGraph and PsgDocGraph, are marked with 'c' and 'p', respectively. There are no statistically significant differences between PsgAidRank and DocGraph. The best result in a column is boldfaced.

	AP		FR		TREC8		WSJ		WT10G	
	p@5	p@10	p@5	p@10	p@5	p@10	p@5	p@10	p@5	p@10
init. rank.	45.7	43.2	24.8	18.5	50.0	45.6	53.6	48.4	33.9	28.0
DocGraph	53.3^{i}	49.2^{i}	25.6	18.0	54.0^{i}	48.0^{i}	57.2^{i}	49.6	35.9	29.4
ClustDocGraph	$\mathbf{53.7}^{i}$	49.4^{i}	20.7	15.9	**57.2**	**48.4**	57.2	**51.0**	36.1	$\mathbf{30.5}^{i}$
PsgDocGraph	50.3	47.3	16.7^{i}	14.6	55.6	47.8	53.2	49.2	29.4	26.5
PsgAidRank	$\mathbf{53.7}^{i}$	$\mathbf{49.5}^{i}$	$\mathbf{26.7}^{c}_{p}$	$\mathbf{19.3}^{c}_{p}$	56.8^{i}	48.0	$\mathbf{58.0}^{i}$	**51.0**	$\mathbf{37.1}^{i}_{p}$	30.0

to a value in {2,4,9,19,29,39,49} [Kurland and Lee 2006] to optimize p@5; these out-degree values correspond (in terms of percent of the total number of nodes) to those used by PsgAidRank in the document-only and passage-only graphs.

Some recent work [Bendersky and Kurland 2008a] has shown the merits of utilizing passage-centrality information induced over a passage-document graph for re-ranking. Specifically, a document from the initial list $\mathcal{D}_{\text{init}}$ is ranked by $p_d(q) \max_{g_i \in d} auth(g_i)$, where $auth(g_i)$ is induced by the HITS algorithm over a bipartite passage-document graph. We follow this work [Bendersky and Kurland 2008a] and set the graph out-degree parameter to a value in $\{9, 19, \ldots, 99\}$ to optimize p@5 performance. Note that in contrast to this re-ranking method, denoted **PsgDocGraph**, PsgAidRank induces passage centrality using a passage-only graph; in addition, PsgAidRank utilizes the centrality of d as a whole ($Cent(d)$).

The performance comparison between PsgAidRank and the graph-based reference comparisons is presented in Table VII. Our first observation is that PsgAidRank outperforms DocGraph on all corpora, though never to a statistically significant degree. Furthermore, PsgAidRank posts statistically significant p@5 improvements over the initial ranking for four corpora, while DocGraph does so for three corpora; recall that p@5 is the evaluation metric for which the performance of all methods was optimized. These findings attest to the potential merit of integrating document and passage information for re-ranking — the basic idea that motivated the development of PsgAidRank.

We can also see in Table VII that PsgAidRank is consistently superior to PsgDocGraph; sometimes, to a statistically significant degree. Thus, the information sources utilized by PsgAidRank and which are not utilized by PsgDocGraph (see the above) have an important contribution to the resultant effectiveness.

Finally, we can see in Table VII that PsgAidRank outperforms ClustDocGraph on FR, WSJ, and WT10G; the improvements for FR are substantial and statistically significant. For the AP corpus, PsgAidRank and ClustDocGraph yield comparable performance, while for TREC8 the performance of ClustDocGraph is slightly better than that of PsgAidRank, albeit not to a statistically significant degree. Thus, while in general PsgAidRank seems to be more effective than ClustDocGraph, integrating

Table VIII. Comparison with cluster ranking (ClustRanker) [Kurland 2008], for which p@5 denotes the percentage of relevant documents in the highest ranked cluster that contains 5 documents. Boldface marks the best result in a column; 'i' marks statistically significant differences with the initial ranking.

	AP	FR	TREC8	WSJ	WT10G
	p@5	p@5	p@5	p@5	p@5
init. rank.	45.7	24.8	50.0	53.6	33.9
ClustRanker	52.7^i	22.6	**57.6**	56.0	**39.8**^{i}
PsgAidRank	**53.7**^{i}	**26.7**	56.8^i	**58.0**^{i}	37.1^i

cluster-based information in PsgAidRank seems to be an interesting venue for future work.

Comparison with cluster ranking. As noted in Section 2, the derivation of PsgAidRank was inspired by a model, **ClustRanker**, for ranking *query specific* clusters by the presumed percentage of relevant documents that they contain [Kurland 2008]. Thus, we now turn to compare the effectiveness of PsgAidRank with that of ClustRanker when used to produce document ranking.

To that end, we follow the implementation details in Kurland [2008] and measure the percentage of documents in the highest ranked cluster of size 5. This percentage is exactly the p@5 performance obtained by positioning the constituent documents of the highest-ranked cluster at the top of the returned results. The values of the free parameters of ClustRanker are set to optimize p@5, as is the case for PsgAidRank.

As can be seen in Table VIII, PsgAidRank outperforms ClustRanker on AP, FR and WSJ, while the reverse holds for TREC8 and WT10G; however, these performance differences are not statistically significant. Nevertheless, PsgAidRank posts statistically significant improvements over the initial ranking for four corpora, while ClustRanker posts such statistically significant improvements only for two corpora. As stated above, integrating cluster-based and passage-based information for document re-ranking is a challenge we leave for future work.

4.3.3 *Learning parameter values.* Our goal in the study presented above was to evaluate the *potential* performance of PsgAidRank — specifically, in comparison to that of various reference comparisons — and the factors that affect its performance. To that end, we neutralized issues that rise from free-parameter values by setting the free parameters of *all* methods to values optimizing average p@5 performance with respect to the entire set of queries.

We now turn to examine whether effective values of the free parameters incorporated by PsgAidRank generalize across queries; that is, whether these values can be learned using a held-out query set. Such a study is different than that presented in Figures 1, 2 and 3, which addressed the robustness of the average (over queries) performance of PsgAidRank with respect to free-parameter values.

We showed in Section 4.3.1 that effective balance between using whole-document-based and passage-based information can vary from one corpus to another, as was the case in previous work on passage-based document ranking [Liu and Croft 2002; Bendersky and Kurland 2008b; Wang and Si 2008]. Hence, learning free parameter values across queries is performed per each corpus. Specifically, we employ a leave-

Table IX. Performance numbers when learning free parameter values using a leave-one-out cross validation procedure. Statistically significant differences of a method with the initial ranking, RM3, RM3(re-rank), DocQueryGen∧PsgQueryGen, and DocQueryGen∧DocCent are marked with 'i', 'r', 'R', 'p' and 'd', respectively. The best result in a column is boldfaced.

	AP		FR		TREC8		WSJ		WT10G	
	p@5	p@10	p@5	p@10	p@5	p@10	p@5	p@10	p@5	p@10
init. rank.	45.7	43.2	24.8	18.5	50.0	45.6	53.6	48.4	33.9	28.0
RM3	49.9	48.5^{i}	23.7	18.1	50.8	**49.4**	**54.8**	**51.0**	30.4^{i}	28.1
RM3(re-rank)	51.1^{i}	48.2^{i}	23.7	18.1	50.4	47.6	52.0_{r}	50.6	$\mathbf{36.3}_{r}$	29.4_{r}
DocQueryGen∧PsgQueryGen	44.6^{R}_{r}	42.8^{R}_{r}	**26.3**	**20.6**	50.0	46.2	**54.8**	48.8	36.1^{i}_{r}	**29.7**
DocQueryGen∧DocCent	53.1^{i}_{p}	49.2^{i}_{p}	24.1	18.7	48.8	48.8	52.0	48.0	34.3_{r}	28.8
PsgAidRank	$\mathbf{53.7}^{i}_{p}$	$\mathbf{49.5}^{i}_{p}$	**26.3**	**20.6**	$\mathbf{56.8}^{iR}_{pd}$	48.0	52.4	49.4	35.3_{r}	**29.7**

one-out cross validation approach: the free-parameter values of a method for a query are set to those optimizing the average p@5 performance[9] over all other queries for the same corpus.

We note that we have also tested the performance of PsgAidRank with 2-fold, 3-fold, 5-fold, and 10-fold cross validation procedures (using multiple sampled random folds) for setting free parameter values. As could be expected, the more queries the training is based on, the better is the resultant performance of PsgAidRank. Specifically, leave-one-out was the train/test regime using which PsgAidRank attained the best performance in most relevant comparisons (corpus × evaluation measure). This finding also holds for the methods we use as reference comparisons to PsgAidRank at the below. That is, for most of these methods, using leave-one-out cross validation to set free-parameter values yielded the best performance — with respect to that of the other k-fold cross validation regimes — for a majority of the relevant comparisons. This is not surprising, especially in light of the small number of queries for many of the corpora. (Recall that TREC8 and WSJ, for example, are used with 50 queries each). Consequently, we present the performance of PsgAidRank, and that of the reference comparisons when using leave-one-out cross validation.

We use reference comparisons that were used above. The first is the relevance model approach, used either to rank the entire corpus (RM3) or only the initially retrieved list (RM3(re-rank)). The second is DocQueryGen∧PsgQueryGen (see Table IV), which is a special case of PsgAidRank that does not utilize document and passage centrality information. This method, as noted above, represents a commonly-used approach for passage-based document ranking [Buckley et al. 1994; Callan 1994; Wilkinson 1994; Cai et al. 2004; Bendersky and Kurland 2008b]. The third reference comparison is DocQueryGen∧DocCent (also termed DocGraph at the above; see Tables IV and VII) [Kurland and Lee 2005]. Recall that DocQueryGen∧DocCent is a special case of PsgAidRank with $\lambda = 1$ — i.e., passage-based information is not used. We present the performance of PsgAidRank and that of the reference comparisons in Table IX.

[9]If two parameter settings yield the same p@5, we choose the one *maximizing* p@10 so as to learn the best possible setting.

Our first observation based on Table IX is that the performance of PsgAidRank is better than that of the initial ranking in almost all reference comparisons (corpus × evaluation measure). Several of these improvements are quite substantial and statistically significant. (Refer, for example, to the performance numbers for AP and TREC8.)

We can also see in Table IX that the performance of PsgAidRank is superior in most relevant comparisons to that of the relevance model implementations. In the very few cases for which a relevance model implementation outperforms PsgAidRank the improvements are not statistically significant. On the other hand, PsgAidRank posts in two cases, p@5 for TREC8 and WT10G, statistically significant improvements over a relevance model implementation. In fact, for these two corpora, the improvements posted by PsgAidRank over RM3, which is used for ranking the entire corpus as is standard, are quite striking.

Another observation that we make based on Table IX is that PsgAidRank outperforms DocQueryGen∧PsgQueryGen in many cases; some of these improvements are also statistically significant. In the very few cases that PsgAidRank is outperformed by DocQueryGen∧PsgQueryGen, the performance differences are not statistically significant. Furthermore, PsgAidRank posts more statistically significant improvements over the initial ranking than DocQueryGen∧PsgQueryGen does. Thus, we see again the importance of using document and passage centrality information — an information type utilized by PsgAidRank but not by DocQueryGen∧PsgQueryGen.

We can also see in Table IX that PsgAidRank outperforms DocQueryGen∧DocCent in almost all reference comparisons. Although these performance differences are statistically significant in a single case, Table IX provides quite a strong evidence to the overall superiority of PsgAidRank with respect to DocQueryGen∧DocCent. A case in point, note that with respect to p@5 — the metric for which performance is optimized in the learning phase — DocQueryGen∧DocCent underperforms the initial ranking upon which re-ranking is performed for three corpora (FR, TREC8, and WSJ)[10]. On the other hand, PsgAidRank underperforms (in terms of p@5) the initial ranking only on WSJ. Furthermore, PsgAidRank improves p@5 over that of the initial ranking in a statistically significant manner for TREC8. Thus, these findings further attest to the merits of using passage-based information in addition to document-based information — specifically, when learning free parameter values across queries — as is done in PsgAidRank in contrast to DocQueryGen∧DocCent.

All in all, perhaps the most important finding based on Table IX is that PsgAidRank is the most effective method in most reference comparisons (refer to the boldfaced numbers). This finding attests to the effectiveness of PsgAidRank, with respect to that of the reference comparisons, when learning free parameter values across queries.

[10]We note that when using 2-fold and 3-fold cross validation, instead of leave-one-out, the p@5 performance of DocQueryGen∧DocCent outperforms that of the initial ranking for TREC8 (50.7 and 50.5, respectively). However, this is not the case for 5-fold, 10-fold, and leave-one-out. Furthermore, for 2-fold and 3-fold the p@5 performance of PsgAidRank for TREC8 (51.2 and 52.4, respectively) still transcends that of DocQueryGen∧DocCent on TREC8.

5. CONCLUSION

We presented a novel language-model-based approach to re-ranking an initially retrieved list so as to improve precision at top ranks. Our model integrates inter-passage, inter-document, passage-query, and document-query similarity information. The precision-at-top-ranks performance of our model is substantially better than that of the initial ranking upon which re-ranking is performed. Furthermore, the performance is superior to that of a standard passage-based document ranking method that does not exploit inter-item similarities. Our model also generalizes and outperforms a recently-proposed re-ranking method that utilizes inter-document similarities, but which does not exploit passage-based information. Finally, our model's performance is superior to that of a state-of-the-art pseudo-feedback-based retrieval approach.

Acknowledgments We thank the anonymous reviewers for their helpful comments. The paper is based upon work supported in part by Israel's Science Foundation under grant no. 890015, by IBM's and Google's faculty research awards, by IBM's SUR award, and by the Center for Intelligent Information Retrieval. Any opinions, findings and conclusions or recommendations expressed in this material are the authors' and do not necessarily reflect those of the sponsoring institutions.

REFERENCES

Abdul-Jaleel, N., Allan, J., Croft, W. B., Diaz, F., Larkey, L., Li, X., Smucker, M. D., and Wade, C. 2004. UMASS at TREC 2004 — novelty and hard. In *Proceedings of the Thirteenth Text Retrieval Conference (TREC-13)*. 715–725.

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

Bendersky, M. and Kurland, O. 2008a. Re-ranking search results using document-passage graphs. In *Proceedings of SIGIR*. 853–854. poster.

Bendersky, M. and Kurland, O. 2008b. Utilizing passage-based language models for document retrieval. In *Proceedings of ECIR*. 162–174.

Brin, S. and Page, L. 1998. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the World Wide Web Conference*. 107–117.

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

Cai, D., Yu, S., Wen, J.-R., and Ma, W.-Y. 2004. Block-based web search. In *Proceedings of SIGIR*. 456–463.

Callan, J. P. 1994. Passage-level evidence in document retrieval. In *Proceedings of SIGIR*. 302–310.

Croft, W. B. and Lafferty, J., Eds. 2003. *Language Modeling for Information Retrieval*. Number 13 in Information Retrieval Book Series. Kluwer.

Denoyer, L., Zaragoza, H., and Gallinari, P. 2001. HMM-based passage models for document classification and ranking. In *Proceedings of ECIR*. 126–135.

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

Diaz, F. and Metzler, D. 2006. Improving the estimation of relevance models using large external corpora. In *Proceedings of SIGIR*. 154–161.

Erkan, G. 2006. Language model based document clustering using random walks. In *Proceedings of HLT/NAACL*.

Erkan, G. and Radev, D. R. 2004. LexPageRank: Prestige in multi-document text summarization. In *Proceedings of EMNLP*. 365–371. Poster.

Golub, G. H. and Van Loan, C. F. 1996. *Matrix Computations*, Third ed. The Johns Hopkins University Press.

Hearst, M. A. and Plaunt, C. 1993. Subtopic structuring for full-length document access. In *Proceedings of SIGIR*. 56–89.

Hussain, M. 2004. Language modeling based passage retrieval for question answering systems. M.S. thesis, Saarland University.

Jiang, J. and Zhai, C. 2004. UIUC in HARD 2004 — passage retrieval using HMMs. In *Proceedings of TREC-13*.

Kaszkiel, M. and Zobel, J. 1997. Passage retrieval revisited. In *Proceedings of SIGIR*. 178–185.

Kaszkiel, M. and Zobel, J. 2001. Effective ranking with arbitrary passages. *Journal of the American Society for Information Science 52,* 4 (November), 344–364.

Kleinberg, J. 1997. Authoritative sources in a hyperlinked environment. Tech. Rep. Research Report RJ 10076, IBM. May.

Krikon, E., Kurland, O., and Bendersky, M. 2009. Utilizing inter-passage and inter-document similarities for re-ranking search results. In *Proceedings of CIKM*. (To appear).

Kurland, O. 2006. Inter-document similarities, language models, and ad hoc retrieval. Ph.D. thesis, Cornell University.

Kurland, O. 2008. The opposite of smoothing: A language model approach to ranking query-specific document clusters. In *Proceedings of SIGIR*.

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

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

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

Lavrenko, V. 2004. A generative theory of relevance. Ph.D. thesis, University of Massachusetts Amherst.

Lavrenko, V., Allan, J., DeGuzman, E., LaFlamme, D., Pollard, V., and Thomas, S. 2002. Relevance models for topic detection and tracking. In *Proceedings of HLT*. 104–110.

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

Lavrenko, V. and Croft, W. B. 2003. Relevance models in information retrieval. See Croft and Lafferty [2003], 11–56.

Liu, X. and Croft, W. B. 2002. Passage retrieval based on language models. In *Proceedings of CIKM*. 375–382.

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

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

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

Mei, Q., Zhang, D., and Zhai, C. 2008. A general optimization framework for smoothing language models on graph structures. In *SIGIR*. 611–618.

Mihalcea, R. 2004. Graph-based ranking algorithms for sentence extraction, applied to text summarization. In *The Companion Volume to the Proceedings of ACL*. 170–173.

Mihalcea, R. and Tarau, P. 2004. TextRank: Bringing order into texts. In *Proceedings of EMNLP*. 404–411. Poster.

Mittendorf, E. and Schäuble, P. 1994. Document and passage retrieval based on hidden Markov models. In *Proceedings of SIGIR*. 318–327.

Murdock, V. and Croft, W. B. 2005. A translation model for sentence retrieval. In *Proceedings of HLT/EMNLP*. 684–695.

Na, S.-H., Kang, I.-S., Lee, Y.-H., and Lee, J.-H. 2008. Completely-arbitrary passage retrieval in language modeling approach. In *Proceedings of AIRS*. 22–33.

Otterbacher, J., Erkan, G., and Radev, D. R. 2005. Using random walks for question-focused sentence retrieval. In *Proceedings of HLT/EMNLP*. 915–922.

Ponte, J. M. and Croft, W. B. 1997. Text segmentation by topic. In *Proceedings of the European Conference on Research and Advanced Technology for Digital Libraries*. 113–125.

Ponte, J. M. and Croft, W. B. 1998. A language modeling approach to information retrieval. In *Proceedings of SIGIR*. 275–281.

Salton, G., Allan, J., and Buckley, C. 1993. Approaches to passage retrieval in full text information systems. In *Proceedings of SIGIR*. 49–58.

Voorhees, E. M. 2005. Overview of the TREC 2005 robust retrieval task. In *Proceedings of the Fourteenth Text Retrieval Conference (TREC)*.

Voorhees, E. M. and Harman, D. K., Eds. 2000. *The Eighth Text REtrieval Conference (TREC-8)*. NIST.

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

Wade, C. and Allan, J. 2005. Passage retrieval and evaluation. Tech. Rep. IR-396, Center for Intelligent Information Retrieval (CIIR), University of Massachusetts.

Wan, X., Yang, J., and Xiao, J. 2008. Towards a unified approach to document similarity search using manifold-ranking of blocks. *Information Processing and Management 44,* 3, 1032–1048.

Wang, M. and Si, L. 2008. Discriminative probabilistic models for passage based retrieval. In *Proceedings of SIGIR*. 419–426.

Wilkinson, R. 1994. Effective retrieval of structured documents. In *Proceedings of SIGIR*. 311–317.

Willett, P. 1985. Query specific automatic document classification. *International Forum on Information and Documentation 10,* 2, 28–32.

Xu, J. and Croft, W. B. 1996. Query expansion using local and global document analysis. In *Proceedings of SIGIR*. 4–11.

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

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

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

A. DOCUMENT AND PASSAGE CENTRALITY

To induce document and passage centrality, we adopt a previously-proposed approach for inducing document centrality [Kurland and Lee 2005].

Let S be a set of documents or passages, and $G = (S, S \times S)$ be the complete directed graph defined over S.

Given $s_1, s_2 \in S$, we assign the following weight to the edge that connects them $(s_1 \rightarrow s_2)$

$$wt(s_1 \rightarrow s_2) \stackrel{def}{=} \begin{cases} p_{s_2}(s_1) & \text{if } s_2 \in Nbhd(s_1; m), \\ 0 & \text{otherwise;} \end{cases}$$

$Nbhd(s_1; m)$ is the m items $s' \in S - \{s_1\}$ that yield the highest $p_{s'}(s_1)$. (Ties are broken by item ID.)

We smooth the edge-weight function using PageRank's [Brin and Page 1998] approach:

$$wt^{[\delta]}(s_1 \rightarrow s_2) = (1 - \delta) \cdot \frac{1}{|S|} + \delta \cdot \frac{wt(s_1 \rightarrow s_2)}{\sum_{s' \in S} wt(s_1 \rightarrow s')};$$

δ is a free smoothing parameter.

Since G with the edge-weight function $wt^{[\delta]}$ constitutes an ergodic Markov chain, a unique stationary distribution exists. The distribution serves as the centrality function $Cent(\cdot)$.