PageRank without Hyperlinks: Structural Reranking using Links Induced by Language Models

OREN KURLAND
Technion
and
LILLIAN LEE
Cornell University

The ad hoc retrieval task is to find documents in a corpus that are relevant to a query. Inspired by the PageRank and HITS (hubs and authorities) algorithms for Web search, we propose a structural reranking approach to ad-hoc retrieval that applies to settings with no hyperlink information. We reorder the documents in an initially retrieved set by exploiting implicit asymmetric relationships among them. We consider *generation links*, which indicate that the language model induced from one document assigns high probability to the text of another. We study a number of reranking criteria based on measures of centrality in the graphs formed by generation links, and show that integrating centrality into standard language-model-based retrieval is quite effective at improving precision at top ranks; the best resultant performance is comparable, and often superior, to that of a state-of-the-art pseudo-feedback-based retrieval approach. In addition, we demonstrate the merits of our language-model-based method for inducing interdocument links by comparing it to previously suggested notions of interdocument similarities (e.g., cosines within the vector-space model). We also show that our methods for inducing centrality are substantially more effective than approaches based on document-specific characteristics, several of which are novel to this study.

Categories and Subject Descriptors: H3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Retrieval models*

General Terms: Algorithms, Experimentation

Additional Key Words and Phrases: Language modeling, PageRank, HITS, hubs, authorities, social networks, high-accuracy retrieval, graph-based retrieval, structural reranking

This article is based on work supported in part by the National Science Foundation under grant number HS-0329064 and CCR-0122581; SRI International under subcontract number 03-000211 on their project founded by the Department of the Interior's National Business Center; and an Alfred P. Sloan Research Fellowship. Any opinions, findings, and conclusions or recommendations expressed are those of the authors and do not affect the views or policies, either expressed or implied, of any sponsoring institutions the U.S. Government, or any other entity. Portions of this work appeared in Kurland and Lee [2005] and Kurland [2006].
Authors' addresses: email: kurland@ie.technion.ac.il; llee@cs.cornell.edu.

DOI 10.1145/1852102.1852104 http://doi.acm.org/10.1145/1852102.1852104

ACM Reference Format:
Kurland, O. and Lee, L. 2010. PageRank without Hyperlinks: Structural reranking using links induced by language models. ACM Trans. Inform. Syst. 28, 4, Article 18 (November 2010), 38 pages. DOI = 10.1145/1852102.1852104 http://doi.acm.org/10.1145/1852102.1852104

1. INTRODUCTION

One of the most basic tasks in the field of text-based information retrieval (IR) is *ad-hoc retrieval*: given a query, automatically find the most relevant documents in a specified corpus. Information retrieval systems capable of achieving high precision at the top ranks of the returned results would be of obvious benefit to human users. Furthermore, for automated systems that are based on ad-hoc retrieval as an intermediate step (e.g., question-answering systems [Voorhees 2002]), high precision at top ranks is also very important.

Crafting retrieval systems capable of obtaining high precision at top ranks remains a key research challenge, but utilizing extra nontextual information can help. Here, we consider links between documents, since they convey potentially useful information about corpus structure (e.g., hyperlink structure in Web settings helps to determine which Web pages are authoritative [Brin and Page 1998; Kleinberg 1999]), and can help us to develop a corpus-based retrieval method that does not depend on external sources (e.g., semantic networks [Voorhees 1993; Shah and Croft 2004]) or user feedback [Ruthven and Lalmas 2003].

As just mentioned, in the Web setting, the PageRank algorithm [Brin and Page 1998] uses explicitly-indicated interdocument relationships to compute which documents are the most central. Here, we consider adapting this idea to corpora in which such explicit links among documents do not exist.

How should we form links in a nonhypertext setting? While previous work in text summarization [Erkan and Radev 2004] and document reranking [Daniłowicz and Baliński 2000; Diaz 2005] utilized cosine-based links induced for pairs of textual items, we draw on research demonstrating the success of using statistical language models (described more fully in Section 3.4) to improve IR performance in general [Ponte and Croft 1998; Croft and Lafferty 2003] and to model interdocument relationships in particular [Kurland and Lee 2004; Liu and Croft 2004; Kurland et al. 2005]. We employ generation links, which are based on the probability assigned by the language model induced from one document to the term sequence comprising another.[1] We thus combine the strengths of two approaches; one is based on language models used both to model textual information within documents and to inferring links between them, and the other induces centrality based on the inferred links.

We note that the analogy between hyperlinks and generation links is not perfect. In particular, one can attribute much of the success of link-based Web-search algorithms to the fact that hyperlinks are often human-provided

[1]While the term, generate, is convenient, we do not think of a generator document or language model as literally creating others. That is, we do not assume an underlying generative model, in contrast to Lavrenko and Croft [2003] and Lavrenko [2004], inter alia. Other work further discusses this issue and proposes alternate terminology (e.g., render) [Kurland et al. 2005].

certifications that two pages are truly related [Kleinberg 1999]. In contrast, automatically-induced generation links are surely a noisier source of information. To compensate, we advocate an approach (used elsewhere as well [Willett 1985; Hearst and Pedersen 1996; Kleinberg 1999; Leuski 2001; Daniłowicz and Baliński 2000; Tombros et al. 2002; Liu and Croft 2004; Baliński and Daniłowicz 2005; Diaz 2005]) that we term *structural reranking*: we use interdocument relationships to compute an ordering not of the entire corpus, but of a possibly unranked set of documents produced by an initial retrieval method. This set should provide a reasonable ratio of relevant to nonrelevant documents, and thus form a good foundation for our algorithms. Note that our approach differs in spirit from pseudo-feedback-based methods [Buckley et al. 1994; Ruthven and Lalmas 2003], which define a model based on the initially retrieved documents expressly in order to rerank the entire corpus. Indeed, since the quality of the initially retrieved results plays a major role in determining the effectiveness of pseudo-feedback-based algorithms [Tao and Zhai 2006], our methods can potentially serve to greatly enhance the input to them.

To compute centrality values for a given generation graph, we propose a number of methods, including variants of PageRank [Brin and Page 1998]. Through an array of experiments, conducted on various TREC datasets [Voorhees and Harman 2005], we show that centrality, as induced by graph-based methods over our generation graphs, and relevance, are connected. In addition, comparisons against numerous baselines show that language-model-based reranking using centrality as a form of document prior is indeed successful at moving relevant documents in the initial retrieval results higher up in the list; the resultant (precision at top ranks) performance is comparable, and often superior, to that of a state-of-the-art pseudo-feedback-based query-expansion method.

Using an additional array of experiments, we study the effect on performance of different properties of the initial list upon which reranking is performed. We then explore an alternative framework for graph formation that is based on a vector-space representation and corresponding similarity measures. In addition, we show that our centrality measures are superior to measures based on document-specific characteristics, several of which are novel to this study.

2. PRELIMINARY DISCUSSION

We start by informally introducing two important notions, document centrality and language models, that will be used throughout Section 3 wherein we present our graph-based methods more formally.

2.1 Document Centrality

In Web retrieval, to determine whether a Web page is a good candidate for answering the information need underlying a user's request, two types of measures (among others) are often used. The first is *textual relevance*—the extent to which a document's content seems to pertain to the user's request. The second is the *centrality* of the document (a.k.a. *authoritativeness*), which is usually estimated using the graph structure of the Web as determined by the hyperlink structure.

Here, we pursue the task of reranking an initial list of documents that was retrieved in response to a query utilizing only the documents' content (and some vocabulary statistics), that is, we assume no hyperlink structure. Our goal is to investigate whether some notion of centrality of documents with respect to the initial list could help in identifying relevant documents.

While on the Web centrality induction is often guided by a hyper-link-induced graph structure (e.g. Brin and Page [1998] and Kleinberg [1999]), and so is the state of affairs in bibliometrics [Garfield 1972; Pinski and Narin 1976], where citations serve as the basis for edges in the citation-graph, it might not be clear at first glance what it means for a document to be central with respect to some document list where no apparent graph structure exists.

We propose to adopt Web-based approaches for centrality induction using a graph defined from the initial document list. We determine links by the textual similarity between documents. Given the constructed graph, centrality definitions such as: "a document is central to the extent that it gets support for centrality from other textually similar central documents" are relatively easy to quantify using graph-based methods. We discuss this centrality definition and others in depth in Section 3. But, while the constructed graph provides convenient technical grounds for inducing centrality in graph terms, an important question is the connection between centrality induced from interdocument-similarity information and relevance.

To address this question we first note that the notion of centrality in an initially retrieved list has already been considered in the past, although it was not usually termed as such, nor were documents the items for which centrality was induced. In pseudo-feedback-based query expansion methods [Buckley et al. 1994], for example, the initially retrieved list is used for deriving a query model with which a reranking of the entire corpus is performed. Some of these methods, such as Rocchio's method [Rocchio 1971] and the relevance model approach [Lavrenko and Croft 2001] use the center of the document list (given some representation and similarity metric) to define a new query model; such a center does not have to be a document in the list, and usually it is indeed not. Other query expansion methods, (e.g., Xu and Croft [1996], Lafferty and Zhai [2001], and Zhai and Lafferty [2001a]), seek central terms in documents in the list, that is, terms that appear in many documents with relatively high weight according to some representation. The common line of reasoning among all these methods is that some notion of centrality with respect to the initial list can help to devise a better representation of the information-need underlying the original query, as the documents in the list were retrieved in response to the query.

Along these lines we hypothesize that central documents that are similar to many other central documents in the list can be considered as good representatives of the query. Thus, identification of such documents can lead to the design of effective reranking approaches of the list, as we will show in this article.

To better understand how centrality might be connected with relevance, we first point out that central documents contain terms from many other central documents in the list and therefore might comprise a good representation of the underlying information need, by virtue of the way the initial list was created. Still, it could be the case that a rather general document that uses many terms that appear in many other documents, is not relevant to the query; we address this issue in Section 3 by considering the query-similarity of a document on top of its centrality status.

Another insight about the connection between centrality and relevance in the initial list can be drawn from the cluster hypothesis [van Rijsbergen 1979]. Indeed, the premise that relevant documents in the list tend to be more similar to each other than to nonrelevant documents, and that they are more similar to each other than nonrelevant documents are to each other, was validated in several studies (e.g. Leuski and Allan [1998]).[2] Thus, we hypothesize that relevant documents will tend to provide centrality support to each other, while nonrelevant documents will spread this support between relevant and nonrelevant documents. Consequently, if there are a reasonable number of relevant documents in the initial list, they will potentially draw the most centrality-support. We present experimental results that support this hypothesis to some degree in Section 5.2.

2.2 Language Models

Statistical language models play an important role in our graph-construction methods. To briefly review the core underlying principle of language-model estimation, consider the following example. (We present in depth our language-model induction techniques in Section 3.4.) Suppose we are given a toy document "hello world hello world morning," and a span of text "hello world." If we assume term-independence (a.k.a. a bag-of-terms representation), then we could say, under some assumptions and conditions, that the probability assigned to the text span by a language model induced from the document is $\frac{2}{5} \times \frac{2}{5}$, as the relative frequency of the terms "hello" and "world" in the document is $\frac{2}{5}$. Often, we will use language-model jargon and say that the probability that the text span was generated from a language model induced from the document is $\frac{2}{5} \times \frac{2}{5}$.

Naturally, if the text span contains a term not appearing in the document from which a language model is induced, the resultant assigned probability will be zero. To address this issue, language models are smoothed using general corpus term-statistics. Furthermore, we can see that long spans of texts will be assigned, in general, smaller probabilities than shorter text spans. We address these issues in Section 3.4.

3. STRUCTURAL RERANKING

Throughout this section, we assume that the following have been fixed: the corpus $\mathcal{C}$ (in which each document has been assigned a unique alphanumeric ID); the query q (composed of a list of terms); the set $\mathcal{D}_{\text{init}} \subseteq \mathcal{C}$ of top documents

[2]The cluster hypothesis [van Rijsbergen 1979] was originally formulated for the entire corpus, rather than for an initially retrieved list.

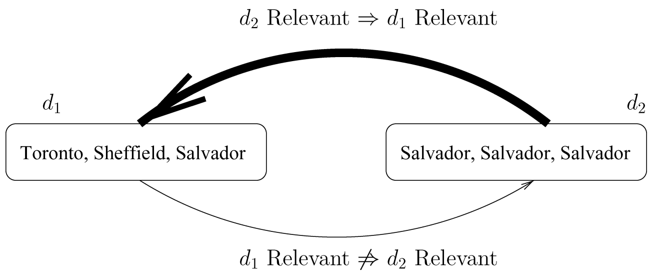
d_2 Relevant $\Rightarrow$ d_1 Relevant
d_1
d_2
Toronto, Sheffield, Salvador
Salvador, Salvador, Salvador
d_1 Relevant $\not\Rightarrow$ d_2 Relevant

Fig. 1. Intuition behind using language models to induce link information. Assuming unsmoothed unigram language models, $p_{d_1}(d_2) = p_{d_1}(\text{“Salvador”})^3 = (1/3)^3$, which is larger than $p_{d_2}(d_1) = 0$ (due to “Sheffield” and “Toronto” not appearing in d_2). Therefore, the support for centrality being transferred from d_2 to d_1 given that d_2 is relevant (thick arrow) is much stronger than the support transferred from d_1 to d_2 (thin arrow) given that d_1 is relevant. This conforms to the intuition that knowing that d_2 is important (central or relevant) would provide strong evidence that d_1 is at least somewhat important, because d_2's importance must stem from the term “Salvador,” which also appears in d_1. However, knowing that d_1 is very important does *not* allow us to conclude that d_2 is, since the importance of d_1 might stem from its first two terms.

returned by some initial retrieval algorithm in response to q (this is the set upon which reranking is performed); and the value of an ancestry parameter, α, that pertains to our graph-construction process.

For each document, $d \in \mathcal{C}$, $p_d(\cdot)$ denotes the smoothed unigram language model induced from d (estimation details appear in Section 3.4). We use g and o to distinguish between a document treated as a generator and a document treated as offspring, that is, something that is generated (details follow).

We use the notation (V, wt) for weighted directed graphs. V is the set of vertices; and $wt : V \times V \to \{y \in \Re : y \geq 0\}$ is the *edge-weight function*. Thus, there is a directed edge between every ordered pair of vertices, but wt may assign zero weight to some edges. We write $wt(v_1 \to v_2)$ to denote the value of wt on edge (v_1, v_2).

3.1 Generation Graphs

Our use of language models to form links can be motivated by considering the following two documents:

d_1: Toronto Sheffield Salvador
d_2: Salvador Salvador Salvador.

Knowing that d_2 is important (central or relevant) would provide strong evidence that d_1 is at least somewhat important, because d_2's importance must stem from the term Salvador, which also appears in d_1. However, knowing that d_1 is very important does not allow us to conclude that d_2 is, since the importance of d_1 might stem from its first two terms. Using language models induced from documents enables us to capture this asymmetry in how centrality is propagated. We allow a document d to receive support for centrality status from a document o only to the extent that $p_d(o)$ is relatively large. If o is not in fact important, the support it provides may not be significant. Indeed, as shown in Figure 1, if for simplicity's sake we induce two unsmoothed unigram language models, $p_{d_1}(\cdot)$ and $p_{d_2}(\cdot)$, from d_1 and d_2 respectively, we get that $p_{d_1}(d_2) = p_{d_1}(\text{“Salvador”})^3 = (1/3)^3$ is larger than $p_{d_2}(d_1) = 0$ (due to “Sheffield” and “Toronto” not appearing in d_2). Therefore, the link induced between d_2 and d_1 should have higher weight than the opposite link, indicating that d_2 transfers more centrality status to d_1 than the other way around.[3] Note that ranking documents by $p_d(q)$, as first proposed by Ponte and Croft [1998], can be considered a variation of our proposed principle: given that a query is central (relevant), rank documents by the support for centrality status that they receive from the query, as captured by $p_d(q)$.

We are thus led to the following definitions.

Definition 1. The *top α generators* of a document $d \in \mathcal{D}_{\text{init}}$, denoted $TopGen(d)$, is the set of α documents $g \in \mathcal{D}_{\text{init}} - \{d\}$ that yield the highest $p_g(d)$, where ties are broken by document ID. (We suppress α in our notation for clarity.)

Definition 2. The *offspring* of a document $d \in \mathcal{D}_{\text{init}}$ are those documents for which d is a top generator, that is, the set $\{o \in \mathcal{D}_{\text{init}} : d \in TopGen(o)\}$.

Note that multiple documents can share offspring, and that it is possible for a document to have no offspring.

We can encode top-generation relationships using either of two generation graphs, $G_U = (\mathcal{D}_{\text{init}}, wt_U)$ and $G_W = (\mathcal{D}_{\text{init}}, wt_W)$, where for $o, g \in \mathcal{D}_{\text{init}}$,

$$wt_U(o \to g) = \begin{cases} 1 & \text{if } g \in TopGen(o), \\ 0 & \text{otherwise;} \end{cases}$$

$$wt_W(o \to g) = \begin{cases} p_g(o) & \text{if } g \in TopGen(o), \\ 0 & \text{otherwise.} \end{cases}$$

Thus, in both graphs, positive-weight edges lead only from offspring to their respective top α generators; but G_U treats edges to the top generators of o uniformly, whereas G_W differentially weights them by the probability their induced language models assign to o.

Several of our algorithms (namely, the direct variants of PageRank) rely on the assumption that the graph satisfies certain connectivity properties with respect to those edges with non-zero weight and that for each $o \in \mathcal{D}_{\text{init}}$, $\sum_{g \in \mathcal{D}_{\text{init}}} wt(o \to g) = 1$, holds. Since G_U and G_W do not satisfy these assumptions, we define *smoothed* versions of them in which all edges, including self-loops have non-zero weight. To be specific, we employ PageRank's [Brin and Page 1998] smoothing technique.

Definition 3. Given an edge-weighted directed graph $G = (\mathcal{D}_{\text{init}}, wt)$ and smoothing parameter $\lambda \in [0, 1)$, the *smoothed* graph $G^{[\lambda]} = (\mathcal{D}_{\text{init}}, wt^{[\lambda]})$ has

[3]The inequality $p_{d_1}(d_2) > p_{d_2}(d_1)$ also holds if smoothed language models are utilized. As noted, our actual method for language-model induction does involve smoothing (details in Section 3.4).

edge weights defined as follows: for every $o, g \in \mathcal{D}_{\text{init}}$,

$$wt^{[\lambda]}(o \to g) = \lambda \cdot \frac{1}{|\mathcal{D}_{\text{init}}|} + (1 - \lambda) \cdot \frac{wt(o \to g)}{\sum_{g' \in \mathcal{D}_{\text{init}}} wt(o \to g')}.$$

Note that the definition is valid ($\sum_{g' \in \mathcal{D}_{\text{init}}} wt(o \to g') \neq 0$), since each document is assigned non-zero generation probability by all documents in $\mathcal{D}_{\text{init}}$ as a result of applying smoothed language models (details in Section 3.4). Furthermore, it is easy to see that $\sum_{g' \in \mathcal{D}_{\text{init}}} wt^{[\lambda]}(o \to g') = 1$, and thus the weights of all edges leading out of any given node in $G^{[\lambda]}$ may be treated as transition probabilities.

With these concepts in hand, we can now phrase our centrality-determination task as follows: given a generation graph, compute for each node (document) how much centrality is transferred to it from other nodes. By our edge-weight definitions, centrality therefore corresponds to the degree to which a document is responsible for "generating" (perhaps indirectly) the other documents in the initially retrieved set. We now consider different ways to formalize this notion of transferrence of centrality.[4]

3.2 Computing Graph Centrality

A straightforward way to define the centrality of a document d with respect to a given graph $G = (\mathcal{D}_{\text{init}}, wt)$ is to set it to d's weighted in-degree, which we call its *influx*:

$$Cen_I(d; G) \stackrel{def}{=} \sum_{o \in \mathcal{D}_{\text{init}}} wt(o \to d). \tag{1}$$

The *Uniform Influx* algorithm sets $G = G_U$, so that the only thing that matters is how many offspring d has; it is thus reminiscent of the journal *impact factor* function from bibliometrics [Garfield 1972], which computes normalized counts of explicit citation links. The *Weighted Influx* algorithm sets $G = G_W$, so that the generation probabilities that d assigns to its offspring are factored in as well.

As previously noted by Pinski and Narin [1976] in their work on influence weights, one intuition not accounted for by weighted in-degree methods is that a document with even a great many offspring should not be considered central (or relevant) if those offspring are themselves very non-central. We can easily modify Equation (1) to model this intuition; we simply scale the evidence from a particular offspring document by that offspring's centrality, thus arriving at the following recursive equation.

$$Cen_{RI}(d; G) \stackrel{def}{=} \sum_{o \in \mathcal{D}_{\text{init}}} wt(o \to d) \cdot Cen_{RI}(o; G), \tag{2}$$

where we also require that $\sum_{d \in \mathcal{D}_{\text{init}}} Cen_{RI}(d; G) = 1$. Unfortunately, for arbitrary G_U and G_W, Equation 2 may not have a unique solution or even any solution at

[4]We present the merits of this framework with respect to an alternative in which one seeks documents that are *central offspring* generated either directly or indirectly by many other documents in $\mathcal{D}_{\text{init}}$, in Section 5.2.2.4.

all under the normalization constraint just given; however, a unique solution is guaranteed to exist for their PageRank-smoothed versions, since in such graphs, the edge weights correspond to the transition probabilities for a Markov chain that is aperiodic and irreducible, and hence has a unique stationary distribution [Grimmett and Stirzaker 2001] that can be computed by a variety of means [Grassmann et al. 1985; Stewart 1994; Golub and Van Loan 1996]. In our experiments, power iteration converged very quickly.

By analogy with these two influx algorithms then, we have the *Recursive Uniform Influx* algorithm, which sets $G = G_U^{[\lambda]}$ and is a direct analog of PageRank [Brin and Page 1998], and the *Recursive Weighted Influx* algorithm, which sets $G = G_W^{[\lambda]}$.

3.3 Incorporating Initial Scores

The centrality scores we presented can be used in isolation as criteria by which to rank the documents in $\mathcal{D}_{\text{init}}$. However, if available, it might be useful to incorporate more information from the initial retrieval engine to help handle cases where centrality and relevance are not strongly correlated. (Recall that the initial retrieval engine participates in any case by specifying the set $\mathcal{D}_{\text{init}}$.) In our experiments, we explore one concrete instantiation of this approach: we apply language-model-based retrieval [Ponte and Croft 1998; Croft and Lafferty 2003] to determine $\mathcal{D}_{\text{init}}$, and consider the following family of reranking criteria:

$$Cen(d; G) \cdot p_d(q), \tag{3}$$

where $d \in \mathcal{D}_{\text{init}}$, Cen is one of the centrality functions defined in the previous section, and $p_d(q)$ is the score that the initial retrieval engine assigns to d. This gives rise to the algorithms *Uniform Influx* + *LM*, *Weighted Influx* + *LM*, *Recursive Uniform Influx* + *LM*, and *Recursive Weighted Influx* + *LM*.

Incidentally, our choosing $p_d(q)$ as the initial score function has the interesting consequence that it suggests interpreting $Cen(d; G)$ as a document *prior*—in fact, Lafferty and Zhai [2001] write, "with hypertext, [a document prior] might be the distribution calculated using the 'PageRank' scheme." We will return to this idea later.

3.4 Estimating Generation Probabilities: Length and Entropy Effects

Generation probabilities form the basis for the graphs on which our algorithms are defined. This section describes our method for estimating these probabilities.

Let $\text{tf}(w \in x)$ denote the term frequency, or number of times the term w occurs in the text or text collection, x. What is often called the *maximum-likelihood estimate* (MLE) of w with respect to x is defined as:

$$\widetilde{p}_x^{MLE}(w) \overset{def}{=} \frac{\text{tf}(w \in x)}{\sum_{w'} \text{tf}(w' \in x)}.$$

Some prior work in language-model-based retrieval [Zhai and Lafferty 2001b] employs a Dirichlet-smoothed version:

$$\widetilde{p}_x^{[\mu]}(w) \overset{def}{=} \frac{\mathrm{tf}(w \in x) + \mu \cdot \widetilde{p}_C^{MLE}(w)}{\sum_{w'} \mathrm{tf}(w' \in x) + \mu};$$

the smoothing parameter μ controls the degree of reliance on relative frequencies in the corpus rather than on the counts in x. Note that smoothing helps to avoid the *zero probability problem*, namely, the assignment of zero probability to unseen terms (see Zhai and Lafferty [2001b] and Zhai and Lafferty [2002] for more details on smoothing language models for information retrieval). Both estimates just described are typically extended to distributions over term sequences by assuming that terms are independent: for an n-term text sequence $w_1 w_2 \cdots w_n$,

$$p_x^{MLE}(w_1 w_2 \cdots w_n) \overset{def}{=} \prod_{j=1}^{n} \widetilde{p}_x^{MLE}(w_j);$$

$$p_x^{[\mu]}(w_1 w_2 \cdots w_n) \overset{def}{=} \prod_{j=1}^{n} \widetilde{p}_x^{[\mu]}(w_j).$$

While in previous work on language models for information retrieval, this Dirichlet-smoothed estimate has been used to assign generation probability to a single query [Croft and Lafferty 2003], using it to estimate generation probabilities when constructing our graphs will have two detrimental effects. First, documents are generally substantially longer than queries, and long term sequences are assigned very low probabilities by bag-of-words models (models assuming term-independence), which leads to numerical (underflow) problems [Lavrenko et al. 2002]. A related issue is that a length bias occurs: longer documents will have lower weights on their outgoing edges than shorter documents.

We adopt another estimation approach, which incorporates the Kullback-Leibler (KL) divergence D between document language models [Kurland and Lee 2004; Kurland et al. 2005]. The KL divergence has previously been used in a number of ways to score documents with respect to a query, for example, Ng [2000] and Lafferty and Zhai [2001]. Unless otherwise specified, for document d and word sequence $s = w_1, \ldots, w_n$ (in our setting, either a document or the query), we set $p_d(s)$ to

$$p_d^{KL,\mu}(s) \overset{def}{=} \exp(-D\left(\widetilde{p}_s^{MLE}(\cdot) \;\middle\|\; \widetilde{p}_d^{[\mu]}(\cdot)\right)), \tag{4}$$

which is equivalent (using some algebraic manipulation) to

$$p_d^{KL,\mu}(s) = \underbrace{(p_d^{[\mu]}(s))^{\frac{1}{|s|}}}_{\text{term A}} \cdot \underbrace{\exp(H(\widetilde{p}_s^{MLE}(\cdot)))}_{\text{term B}},$$

where H is the entropy function: $H(\widetilde{p}_s^{MLE}(\cdot)) \overset{def}{=} -\sum_w \widetilde{p}_s^{MLE}(w) \log \widetilde{p}_s^{MLE}(w)$. Now, observe that term A length-normalizes $p_d^{[\mu]}(s)$ via the geometric mean, and therefore helps ameliorate the previously mentioned problems (length bias and underflow). Additionally, term B raises the generation probability for texts with high-entropy MLE term distributions. High entropy may be correlated with a larger number of unique terms, for example, we get an entropy of 0 for the document "Salvador Salvador Salvador" but log 3 for "Toronto Sheffield Salvador," which in turn, has previously been suggested as a cue for relevance [Singhal et al. 1996; Hiemstra and Kraaij 1999]. Hence, generators of documents inducing high-entropy language models may be good candidates for centrality status. (We hasten to point out, though, that for the algorithms based on smoothed graphs (Definition 3), the entropy term cancels out due to our normalization of edge weights.)

4. RELATED WORK

Work on structural reranking in traditional ad-hoc information retrieval has mainly focused on *query-specific clustering*, wherein one seeks to compute and exploit a clustering of the initial retrieval results [Preece 1973; Willett 1985; Hearst and Pedersen 1996; Leuski 2001; Tombros et al. 2002; Liu and Croft 2004; Kurland 2006; Kurland and Lee 2006; Liu and Croft 2006b, 2006a]. Clusters represent structure within a document set, but do not directly induce an obvious single criterion or principle by which to rank documents; for instance, they have been used to improve rankings indirectly by serving as smoothing mechanisms [Liu and Croft 2004; Kurland 2006], and in *interactive retrieval* settings [Hearst and Pedersen 1996; Leuski 2001] they were utilized for effective visualization of the initial retrieved list.[5] Recently, Kurland and Lee [2006] used our graph-formation principles for creating bipartite graphs wherein documents from the intial list are on one side and query-specific clusters are on the other side. They showed that if centrality is induced over such graphs using the HITS (hubs and authorities) algorithm [Kleinberg 1997], then authoritative documents tend to be relevant and authoritative clusters tend to contain a high percentage of relevant documents.

In a related vein, Baliński and Daniłowicz [2005] and Diaz [2005] apply score regularization to ensure that similar documents within an initial retrieved list receive similar scores. In contrast to our framework, centrality is not introduced as an explicit notion and therefore cannot be explored in its own right.

There has been increasing use of techniques based on graphs induced by implicit relationships between documents or other linguistic items, for example, Salton and Buckley [1988], Hatzivassiloglou and McKeown [1997], Daniłowicz and Baliński [2000], Dhillon [2001], Lafferty and Zhai [2001], Joachims [2003], Erkan and Radev [2004], Mihalcea and Tarau [2004], Pang and Lee [2004], Toutanova et al. [2004], Barzilay and Lapata [2005], Collins-Thompson and Callan [2005], Otterbacher et al. [2005], Zhang et al. [2005], Zhu [2005], Erkan [2006a], and Erkan [2006b]. The work in the domain of text summarization [Erkan and Radev 2004; Mihalcea and Tarau 2004; Erkan 2006b] resembles

[5]Interestingly, some centrality measures have been previously employed to produce clusterings; in Tishby and Slonim [2000], the stochastic-process interpretation of our Equation 2 was utilized to detect structures in the underlying graph, thereby inducing a clustering.

ours, in that it also computes centrality on graphs, although the nodes correspond to sentences or terms rather than documents. Erkan and Radev [2004], Mihalcea and Tarau [2004] and Erkan [2006b] present methods similar to our Recursive Weighted Influx algorithm; the Uniform Influx algorithm is also used in Erkan and Radev [2004] for selecting central sentences. The main contrast between the work of Erkan and Radev [2004] and Mihalcea and Tarau [2004] and ours is that links were not induced by generation probabilities, but by symmetric vector-space similarity measures (Section 5.2.2.3 presents the results of experiments studying the relative merits of our particular choice of link definition). Erkan [2006b], on the other hand, adopted our generation graphs for the task of focused summarization. Our generation graphs have also been used in Erkan [2006a] for creating enhanced document representations for the purposes of summarization. It is also important to note that in Erkan and Radev [2004] and Mihalcea and Tarau [2004], summarization does not depend on a specific user's request and therefore centrality serves as the sole criterion for selecting sentences, while in the ad hoc retrieval setting, one has to handle cases in which relevance and centrality are not strongly correlated. One method for doing so is the technique represented by Equation (3). Alternatively, similarity to a representation of the information need can be used to smooth edge weights [Erkan 2006b].

Otterbacher et al. [2005] and Daniłowicz and Baliński [2000] use interitem similarities to define transition probabilities of a Markov chain for sentence retrieval and document reranking respectively. While the Markov chain approach is similar to our Recursive Weighted Influx method, the transition probabilities are based on symmetric interitem similarity functions, in contrast to our link induction method. Furthermore, these probabilities also incorporate information about similarity to the information need at hand, as in Erkan [2006b], and thus centrality cannot be explored in isolation, as in our algorithms.

Recent work on Web retrieval [Zhang et al. 2005] utilizes asymmetric similarity relationships in the vector space model for link induction. Centrality is induced using the PageRank algorithm, similarly to our Recursive Weighted Influx algorithm. However, since the task at hand is to rank all documents in the corpus (rather than rerank an initially retrieved list), centrality is integrated with similarity to the query to define a relevance scoring function and is not explored in isolation. In Section 5.2.2.3 we evaluate one alternative for utilizing an asymmetric similarity measure within the vector space model, studying its effectiveness both as an isolated ranking criterion and in combination with similarity to the query, following Equation (3).

Our centrality scores constitute a relationship-based reranking criterion that can serve as a bias affecting the initial retrieval engine's scores, as in Equation (3). Alternative biases that are based on individual documents alone have also been investigated. Functions incorporating document or average word length [Hiemstra and Kraaij 1999; Kraaij and Westerveld 2001; Miller et al. 1999] are applicable in our setting; we report on experiments with variants of document length in Section 5.2.2.5. Other previously suggested biases that may be somewhat less appropriate for general domains include document source

[Miller et al. 1999] and creation time [Li and Croft 2003], as well as Web page hyperlink in-degree and URL form [Kraaij et al. 2002].

5. EVALUATION

In what follows, we describe an array of experiments for evaluating the effectiveness of our algorithms in reranking an initial retrieved list to improve precision at top ranks. As part of this array, we compare the performance of our methods with that of a state-of-the-art pseudo-feedback-based query expansion method, namely, the relevance model [Lavrenko and Croft 2001]. We then study the ability to learn the values of the free parameters that our methods incorporate.

In further explorations, we perform a controlled study of the connection between centrality and relevance. As part of this study, we analyze the way relevant and nonrelevant documents are situated within our graphs. We then study the the effect of the initial list size on the performance of our algorithms. In addition, we study the importance of basing our graph formation on a language-modeling framework and generation probabilities by exploring an analogous framework using a vector-space representation and corresponding similarity measures. Furthermore, we demonstrate the importance of the directionality of our induced interdocument links. We also compare our algorithms with previously proposed measures for inducing centrality that are based on document-specific properties, and in doing so explore some new alternatives.

5.1 Experimental Setting

The objective of structural reranking is to (re-)order an initially-retrieved document set, $\mathcal{D}_{\text{init}}$, so as to improve precision at the very top ranks of the final results. Therefore, we employed the following three evaluation metrics: the precision of the top 5 documents (prec@5), the precision of the top 10 documents (prec@10), and the mean reciprocal rank of the first relevant document (MRR) [Shah and Croft 2004].

We are interested in the general validity of the various structural reranking methods we have proposed. We believe that a good way to emphasize the effectiveness (or lack thereof) of the underlying principles is to downplay the role of parameter tuning. Therefore, we made the following design decisions, with the effect that the performance numbers we report are purposely not necessarily the best achievable by exhaustive parameter search.

—The initial ranking that created the set $\mathcal{D}_{\text{init}}$ was built according to the function $p_d^{KL,\mu}(q)$, where the value of μ was chosen to optimize the mean noninterpolated average precision (MAP) of the top 1000 retrieved documents. This is not one of our evaluation metrics, but is a reasonable general-purpose optimization criterion. In fact, results with this initial ranking turned out to be statistically indistinguishable from the results obtained by optimizing with respect to the actual evaluation metrics, although of course they were lower in absolute terms.

—We only optimized settings for α (the ancestry parameter controlling the number of top generators considered for each document) and λ (the edge-weight smoothing factor) with respect to the average precision among the top 5 documents[6] (prec@5) over the given set of queries; not with respect to all three evaluation metrics employed.[7]

The search ranges for the latter two parameters were:

$$\begin{array}{ll} \alpha: & 2, 4, 9, 19, \ldots, |\mathcal{D}_{\text{init}}| - 1 \\ \lambda: & 0.05, 0.1, 0.2, \ldots, 0.9, 0.95. \end{array}$$

As it turned out, for many instances the optimal value of α with respect to precision at 5 was either 4 or 9, suggesting that a relatively small number of generators per document should be considered when constructing the graph. In contrast, λ exhibited substantial variance in optimal value for precision at 5 in some of our datasets. We set $|\mathcal{D}_{\text{init}}|$, the number of initially-retrieved documents, to 50 in all the results reported in the following unless stated otherwise; we study the effect of varying $|\mathcal{D}_{\text{init}}|$ on our algorithms' performance in Section 5.2.2.2.

It is important to point out that our approach of finding optimal free-parameter values with respect to the entire set of tested queries is intended for exploring the potential of the methods we present and the different factors that affect their performance, that is, the potential effectiveness of different ways of utilizing interdocument-similarities for reranking. We believe that such a study should be performed independently from the analysis of whether optimal parameter values generalize from one query to another. As we show in 5.2.1.2, wherein we present performance results for our methods and their reference comparisons with parameter values learned using cross-validation, this interquery parameter-values generalization (or lack thereof) can depend, for example, on the train/test split regime for the set of queries. This holds not only for our methods, but also for the reference comparisons we consider. These findings support previous reports about variability among queries, and consequently, the issues with inferring parameter values, and even appropriate retrieval methods from one query to another. (See Section 5.2.1.2 for elaborated discussion.)

The remaining details of the experimental setting are as follows. We conducted our experiments on the following three TREC corpora,[8] which are

[6]If two parameter settings yield the same prec@5 performance, we choose the one minimizing prec@10 so as to provide a conservative estimate of expected performance. Similarly, if we have ties for both prec@5 and prec@10, we choose the setting minimizing MRR.

[7]The document language model smoothing parameter, μ, was set to 2000 in all our methods and reference comparisons, except for estimating $p_d(q)$, wherein we used the value of μ that was used to create the initial ranking, so as to maintain consistency.

[8]We do not use the AP89 corpus, which was used in the conference version of this article [Kurland and Lee 2005], because for many of the corresponding queries there are no relevant documents in the initial list to be reranked, and, hence, the (average-)performance numbers are dominated by a small number of queries. In addition, some of the performance numbers (for the other three corpora) in this article are different than those in the conference version [Kurland and Lee 2005], as the latter accidentally reflect experiments utilizing a suboptimal choice of $\mathcal{D}_{\text{init}}$.

standard benchmarks in IR research [Voorhees and Harman 2005].

corpus	# of docs	queries	disk(s)
AP	242,918	51-64, 66-150	1-3
TREC8	528,155	401-450	4-5
WSJ	173,252	151-200	1-2

Query 65 was omitted from AP because it had no relevant documents (as judged by TREC's human annotators). All documents and queries (in our case, TREC-topic titles) were stemmed using the Porter [1980] stemmer and tokenized, but no other preprocessing steps were applied. We used the Lemur toolkit[9] for language-model estimation. Statistically significant differences in performance were determined using the two-sided Wilcoxon test at a confidence level of 95%.[10]

Efficiency considerations. It is important to note that the computational overhead incurred by our reranking methods on top of the initial search is not significant. Since the initial list to be reranked consists of a few dozen documents (we demonstrate in Section 5.2.2.2 that reranking is mostly effective with relatively short lists, which is in accordance with other models for reranking [Diaz 2005]), then computing interdocument similarities within the list requires relatively little computational effort. Similar efficiency considerations were taken in work on using clusters of similar documents from the initially retrieved list so as to rerank it (e.g., Willett [1985], Liu and Croft [2004], and Kurland and Lee [2006]), and in other work on graph-based reranking [Baliński and Daniłowicz 2005; Diaz 2005]. Moreover, similarities between snippets (e.g., query-dependent summaries) of documents can potentially serve as proxies for similarities between the full content of documents. Indeed, such an approach was taken in work on clustering the results of Web search engines [Zamir and Etzioni 1998]. Furthermore, computing centrality over graphs with a few dozens nodes using the recursive centrality definitions, which are computationally somewhat more demanding than the nonrecursive definitions, takes only a few iterations of the power method [Golub and Van Loan 1996]. It is also important to note that the size of the corpus (number of documents) has no practical effect on the efficiency of reranking since our methods only rerank the most highly ranked documents from an initial search.

5.2 Results

In the tables that follow, we use the following abbreviations for algorithm names.

[9]www.lemurproject.org.

[10]In what follows, we use alpha-numeric symbols as either superscripts or subscripts so as to denote statistically-significant differences between the performance (per evaluation metric) of two methods; e.g., 0.135^{M_2} or 0.135_{M_2} represents an average performance of 0.135 that is statistically-significantly different from that of method M_2 for the same evaluation metric.

U-In	*Uniform Influx*
W-In	*Weighted Influx*
R-U-In	*Recursive Uniform Influx*
R-W-In	*Recursive Weighted Influx*
U-In + LM	*Uniform Influx + LM*
W-In + LM	*Weighted Influx + LM*
R-U-In + LM	*Recursive Uniform Influx + LM*
R-W-In + LM	*Recursive Weighted Influx + LM*

Table I.
Primary experimental results, showing algorithm performance with respect to our 9 evaluation settings (3 performance metrics × 3 corpora). For each evaluation setting, improvements over the optimized baselines are given in italics; statistically significant differences between our structural reranking algorithms and the initial ranking and optimized baselines are indicated by *i* and *o* respectively; **bold** highlights the best results over all ten algorithms. Notice that even though the structural reranking algorithms were optimized for prec@5 only (and produce the best results for this metric), they still perform well with respect to the other two metrics.

	AP			TREC8			WSJ		
	prec@5	prec@10	MRR	prec@5	prec@10	MRR	prec@5	prec@10	MRR
upper bound	.876	.788	.930	.944	.850	.980	.896	.800	1.000
init. ranking	.457	.432	.596	.500	.456	.691	.536	.484	.748
opt. baselines	.465	.437	.635	.512	.464	.696	.560	.494	.772
U-In	*.513*	*.492*$^{i}_{o}$	*.640*	.500	.442	.622	.512	.472	.673
W-In	*.515*	*.487*^{i}	*.643*	.488	.432	.637	.520	.470	.644$_{o}$
U-In + LM	*.509*	**.494**$^{i}_{o}$	.631	*.528*	**.518**$^{i}_{o}$	.665	.544	.490	.724
W-In + LM	*.511*^{i}	*.486*$^{i}_{o}$	.630	*.516*	.464	**.703**	.560	**.500**	**.787**
R-U-In	*.513*	*.477*	.625	*.520*	.446	.665	.536	.478	.707
R-W-In	*.519*	*.480*	.632	*.524*	.446	.666	.536	.486	.699
R-U-In + LM	*.519*$^{i}_{o}$	*.491*$^{i}_{o}$	**.652**	*.556*	.460	.684	**.576**^{i}	*.496*	.757
R-W-In + LM	**.531**$^{i}_{o}$	*.492*$^{i}_{o}$	.630	**.560**	.460	.676	*.572*^{i}	*.496*	.747

5.2.1 *Primary Evaluations.* Our main experimental results are presented in Table I. The first three rows specify reference-comparison data. The initial ranking was, as just described, produced using $p_d^{KL,\mu}(q)$, with μ chosen to optimize for noninterpolated precision at 1000. The empirical upper bound on reranking, which applies to any algorithm that reranks $\mathcal{D}_{\text{init}}$, indicates the performance that would be achieved if all the relevant documents within the initial fifty were placed at the top of the retrieval list. We also computed an optimized baseline for each metric m and test corpus $\mathcal{C}$; this consists of ranking all the documents (not just those in $\mathcal{D}_{\text{init}}$) by $p_d^{KL,\mu}(q)$, with μ chosen to yield the best m-results on $\mathcal{C}$. As a sanity check, we observe that the performance of the initial retrieval method is always below that of the corresponding optimized baseline (though not statistically distinguishable from it).

The first question we are interested in is the performance of our structural reranking algorithms taken as a whole. As shown in Table I, our methods improve upon the initial ranking in many cases, specifically, roughly 2/3 of the 72 relevant comparisons (8 centrality-based algorithms $\times$ 3 corpora $\times$ 3 evaluation metrics). An even more gratifying observation is that Table I shows (via italics and boldface) that in many cases, our algorithms, even though optimized for precision at 5, can outperform a language model optimized for a different (albeit related) metric, *m*, even when performance is measured with respect to *m*; see, for example, the results for precision at 10 on the AP corpus.

Closer examination of the results in Table I reveals that while our algorithms are effective when applied both to the graph G_W and to G_U, the former is in most cases a better choice when considering prec@5—the metric for which performance was optimized. These results imply that it is in general better to explicitly incorporate generation probabilities into the edge weights of our generation graphs than to treat all the top generators of a document equally.

Another observation we can draw from Table I is that incorporating query-generation probabilities as weights on the centrality scores (see Equation (3)) tends to enhance performance. This can be seen by comparing rows labeled with some algorithm abbreviation "X" against the corresponding rows labeled "X + LM". About 84% of the 36 relevant comparisons exhibit this improvement. Most of the counterexamples occur in settings involving precision at 10 and MRR, for which we did not optimize our algorithms.

Similarly, by comparing "Y"-labeled rows with "R-Y"-labeled ones, we see that in about 70% of the 36 relevant comparisons, it is better to use the recursive formulation of Equation (2), where the centrality of a document is affected by the centrality of its offspring, than to ignore offspring centrality as is done by Equation (1).

Perhaps not surprisingly, then, the Recursive Uniform Influx + LM and Recursive Weighted Influx + LM algorithms, which combine the two preferred features just described (recursive centrality computation and use of the initial search engine's score function) appear to be our best performing algorithms. Working from a starting point below the optimized baselines, they improve the initial retrieval set to yield results that, even at their worst, are not only better than the initial ranking for precision at 5 and 10, but are also nearly statistically indistinguishable from the optimized baselines. Moreover, in one setting (AP, precision at 10) they actually produce statistically significant improvements over the optimized baseline even though they were not optimized for that evaluation metric.[11]

5.2.1.1 *Comparison to Pseudo-Feedback-Based Retrieval*. Our structural reranking methods utilize interdocument relationships in the list, $\mathcal{D}_{\text{init}}$, to find relevant documents that it contains. Pseudo-feedback-based query-expansion methods [Buckley et al. 1994], on the other hand, exploit information from $\mathcal{D}_{\text{init}}$ to define a *query model* to be used for reranking the entire corpus. To compare the two approaches with respect to their ability to attain high precision at top

[11]We note that Recursive Weighted Influx + LM also outperforms the initial ranking in terms of MAP(@50)—although not optimized for this evaluation metric—over AP and WSJ, while the reverse holds for TREC8. The MAP of the initial ranking is .093, 0.175, and 0.225 for AP, TREC8, and WSJ, respectively. The MAP of Recursive Weighted Influx + LM for the three corpora is .098, .171, and .226, respectively.

Table II.
Performance comparison of our recursive weighted Influx + LM Algorithm with a relevance model that is either used to rank all documents in the corpus (Rel Model) or used to only rerank the documents in the initial list $\mathcal{D}_{\text{init}}$ (Rel Model(Rerank)). The best result in a column is boldfaced and statistically significant differences with the initial ranking are marked with i.

	AP			TREC8			WSJ		
	prec@5	prec@10	MRR	prec@5	prec@10	MRR	prec@5	prec@10	MRR
init. ranking	.457	.432	.596	.500	.456	**.691**	.536	.484	**.748**
Rel Model	$.503^i$	$.486^i$	.585	.544	**.494**	.671	$.584^i$	.510	.759
Rel Model(Rerank)	$.511^i$	$.482^i$	.598	.544	.486	.679	**.588**^{i}	**.520**	.748
R-W-In + LM	**.531**^{i}	**.492**^{i}	**.630**	**.560**	.460	.676	$.572^i$	.496	.747

ranks of the retrieved lists, we compare the performance of one of our best performing models, Recursive Weighted Influx + LM, to that of a state-of-the-art pseudo-feedback-based (query expansion) approach—the relevance model [Lavrenko and Croft 2001].

For comparison with Recursive Weighted Influx + LM, we use both the standard relevance model (Rel Model), which is constructed from $\mathcal{D}_{\text{init}}$ and is used to re-rank the entire corpus; and, an implementation denoted Rel Model(Rerank), wherein the relevance model is used to rerank *only* $\mathcal{D}_{\text{init}}$. Lemur's implementation of relevance models was used; see Appendix A for details.

We select the values of the three free parameters on which Rel Model and Rel Model(Rerank) depend (see Appendix A for details) so as to optimize precision at the top 5 documents; recall that our Recursive Weighted Influx + LM algorithm's free parameters—the graph out-degree α and the edge-weight smoothing factor, λ—were also selected so as to optimize precision at 5. The performance results of the relevance models and Recursive Weighted Influx + LM algorithm are presented in Table II.

We can see in Table II that while the performance of Recursive Weighted Influx + LM and that of the relevance models is in general comparable, Recursive Weighted Influx + LM posts better prec@5—the metric for which performance was optimized—for two out of the three corpora. (It is interesting to note that using the relevance model for reranking only the initial list is more effective than using it for ranking all documents in the corpus—as is standard practice—in most relevant comparisons.) We hasten to point out, however, that none of the performance differences between Recursive Weighted Influx + LM and the relevance model implementations is statistically significant, and that all three methods (Recursive Weighted Influx + LM, Rel Model, and Rel Model(Rerank)) post statistically significant performance improvements over the initial ranking for the same reference comparisons.

Performance robustness. Pseudo-feedback-based query expansion approaches are known to post, on average, retrieval performance that is superior to that of using the original query for ranking. Indeed, the results in Table II attest to the effectiveness of pseudo-feedback-based relevance models. However, there are queries for which the performance of using an expanded

Table III. Performance Robustness
The percentage of queries for which the prec@5 of a method is inferior to that of the initial ranking. Best result in a column (i.e., lowest number) is boldfaced.

	AP	TREC8	WSJ
Rel Model	**19.2%**	**16.0%**	8.0%
Rel Model(Rerank)	**19.2%**	**16.0%**	14.0%
R-W-In + LM	**19.2%**	20.0%	**2.0%**

query is inferior to that of using the original query without expansion. This problem is often referred to as the robustness problem of pseudo-feedback-based retrieval [Amati et al. 2004; Cronen-Townsend et al. 2004; Collins-Thompson and Callan 2007; Lee et al. 2008]. The performance results in Table II for the relevance models over TREC8 can potentially attest to such a robustness issue, as none of them are statistically significant better than those of the initial ranking, although they are better on average.

To study the performance robustness of our Recursive Weighted Influx + LM reranking algorithm with respect to that of the relevance models, we report in Table III the percentage of queries for which a method posts prec@5 performance that is inferior to that of the initial ranking. As we can see, the performance of Recursive Weighted Influx + LM is as robust as that of the relevance models in most cases.

All in all, perhaps the most important conclusion that we can draw from the comparison to relevance models is that our graph-based reranking paradigm is a highly effective approach for obtaining high precision at top ranks of the retrieved list. In fact, our algorithms can potentially help to improve the performance of pseudo-feedback-based approaches, if they are used to select the documents considered as (pseudo-)relevant; for example, instead of constructing a relevance model from the top-k initially ranked documents, one can construct the model from the top-k documents in the (re-)ranking produced by our methods.

5.2.1.2 *Learning Parameter Values*. Heretofore our evaluation focused on the potential effectiveness of utilizing interdocument-similarities for reranking using graph-based approaches. To study the performance characteristics of the different methods, we have neutralized issues rising from the values of the free parameters that the methods incorporate by examining the optimal attainable performance over the entire set of queries with respect to these parameters' values. Now, we turn to examine the application of our methods where free parameters of an algorithm are set for a specific query to values determined optimal for other queries, that is, we learn parameter values.

Learning parameter values for a retrieval algorithm per query is inherently a very difficult challenge as each query potentially poses a new task. A case in point: ambiguous queries might call for a different analysis than nonambiguous queries. Furthermore, a retrieval method that is very effective for one query might post quite poor performance for others. For example, Jelinek-Mercer smoothing of document language models is more effective than Dirichlet smoothing for long queries, while the reverse holds for short queries [Zhai and

Lafferty 2001b]. Another example is the difference between content-based and named-page-finding queries in Web retrieval [Zhou and Croft 2007].

Another prominent well-known example of the difficulty in learning parameter values is query expansion. As noted, there are many queries for which the performance of query expansion is inferior to that of using no query expansion. However, automatically determining whether to perform expansion for a given query is a highly difficult and open research problem [Amati et al. 2004; Cronen-Townsend et al. 2004]. Note that this question in the relevance-model case amounts to the choice of the value of the free interpolation parameter, η, in Equation (7) (Appendix A); if $\eta = 0$ then no expansion is performed, while $\eta > 0$ results in utilizing expansion terms. Indeed, there has been quite a lot of work on tackling the sensitivity of expansion methods with respect to free parameter values (e.g., Tao and Zhai [2006] and Winaver et al. [2007]). More generally, there is growing research attention in the information retrieval community to issues that stem from query-variability; for example, refer to the TREC's Robust track [Voorhees 2005], and to work on predicting query difficulty (e.g., Cronen-Townsend et al. [2002] and Yom-Tov et al. [2005]).

Thus, the decision of which queries are to be used as the basis for determining parameter values, that is, the training set, and which queries should serve as the test set, becomes a crucial issue. Another important issue that we have to consider is metric divergence [Azzopardi et al. 2003; Morgan et al. 2004; Metzler and Croft 2005], that is, if learning is based on a specific evaluation metric, then the performance over the test set with respect to other evaluation metrics might be far from optimal.

Having these issues in mind, we have taken the following experimental-design decisions. We have restricted the learning/testing to be performed to using queries for the same corpus. That is, the train-set of queries is for the same corpus as that from which the test-set of queries is chosen. This is highly important so as to avoid intercorpora issues.[12] In addition, instead of learning parameter values with respect to each evaluation measure separately, which is a nonrealistic scenario, we have used prec@5, our main evaluation metric, as the criterion for optimizing performance over the train set.[13] Since each of the three tested collections has a relatively small number of queries, we have taken a cross-validation approach to learning/testing. Furthermore, to study the potential impact of query-variability, we present performance results for three regimes of train/test split of the queries used in a cross-validation manner: leave-one-out (loo), 10-fold, and 2-fold (split). All train/test splits are based on random selection of queries.[14] The performance numbers of our methods, where parameter values are learned, are presented

[12]A good example for an intercorpora issue is passage-based document retrieval. While utilizing passage information is known to be highly effective for corpora containing topically-heterogeneous documents [Callan 1994; Liu and Croft 2002; Bendersky and Kurland 2008], this is not the case for corpora containing homogeneous documents.

[13]In the learning process, if two parameter settings yield the same prec@5, we use prec@10, as the optimization criterion; similarly, if we have ties for both prec@5 and prec@10, we use MRR as the optimization criterion.

[14]The folds in the 10-fold regime were selected to be subsets of those in the 2-fold (split) regime.

Table IV.
Performance of main algorithms wherein the values of the free parameters of all methods (except for those of the initial ranking and optimized baselines) are learned using cross-validation. (Recall that the performance numbers in Table I are obtained by maximizing performance over the entire set of queries.) Learning is performed with one of three regimes for splitting the query-set into learn/test sets: Leave-One-Out (loo), 10-Fold, and 2-Fold (Split). The best result in a column is boldfaced; i and o mark statistically significant differences with the initial ranking and optimized baselines, respectively.

	AP			TREC8			WSJ		
	prec@5	prec@10	MRR	prec@5	prec@10	MRR	prec@5	prec@10	MRR
upper bound	.876	.788	.930	.944	.850	.980	.896	.800	1.000
init. ranking	.457	.432	.596	.500	.456	.691	.536	.484	.748
opt. baselines	.465	.437	.635	.512	.464	**.696**	.560	.494	.772
loo U-In	.513	$.492^i_o$	.640	.500	.442	.622	.512	.472	.673
10fold U-In	.513	$.492^i_o$	.640	.468	.440	.622	.512	.472	.673
split U-In	.513	$.492^i_o$	.640	.500	.442	.622	.512	.472	.673
loo W-In	.515	$.487^i$	.643	.488	.432	.637	.520	.470	$.644_o$
10fold W-In	.515	$.487^i$	.643	.488	.432	.637	.504	.462	$.641_o$
split W-In	.489	.475	.600	.448	.426	.568	.520	.470	$.644_o$
loo U-In + LM	.509	$\mathbf{.494}^i_o$	.631	.528	$\mathbf{.518}^i_o$	.665	$.480^i_o$	$.454^i_o$	$.655^i_o$
10fold U-In + LM	.509	$\mathbf{.494}^i_o$	.631	.496	.480	.666	$.504_o$	.486	$.652^i_o$
split U-In + LM	.491	$.484^i_o$	.633	.516	$.504^i$	.686	$.504_o$	.474	.706
loo W-In + LM	.467	$.483_o$	.620	.456	.470	.668	.560	**.500**	**.787**
10fold W-In + LM	.497	$.491_o$	.637	.460	.458	.636	.536	.492	.756
split W-In + LM	.493	$.488_o$	**.645**	.516	$.496^i$	.645	.532	.490	.745
loo R-U-In	.469	.479	.642	.488	.438	.652	.504	.476	.686
10fold R-U-In	.489	.481	.621	.484	.438	.627	.512	.472	.698
split R-U-In	.462	.445	.609	.508	.430	.663	.508	.490	.661
loo R-W-In	.485	.478	.630	$.424_o$	$.398^i_o$	.561	.492	.478	.663
10fold R-W-In	.493	.482	.632	.464	.426	.592	.488	.474	$.648_o$
split R-W-In	.499	$.482^i$	.626	.492	.470	.632	.504	**.500**	.664
loo R-U-In + LM	.483	.463	.640	.464	.438	.635	$\mathbf{.576}^i$	.496	.757
10fold R-U-In + LM	.505	$.486^i_o$	.639	.496	.454	.658	.548	.486	.731
split R-U-In + LM	.481	$.472^i$	.625	.516	.486	.680	.532	**.500**	.719
loo R-W-In + LM	$\mathbf{.529}^i_o$	$.492^i_o$	.630	**.560**	.460	.676	$.484^i_o$	.472	$.665^i_o$
10fold R-W-In + LM	$.515^i_o$	$.488^i_o$	.633	.528	.454	.662	$.492^i_o$	.492	.694
split R-W-In + LM	.509	$.483^i$	.628	.512	$.462^i$	.663	.516	.484	.747

in Table IV. (Recall that in Table I the performance numbers correspond to the best prec@5 performance with respect to the entire set of queries.) In Table V we present a comparison of our Recursive Weighted Influx + LM algorithm with the relevance models when the free parameter values of all algorithms are learned.

We can see in Table IV that, as expected, the performance numbers when learning parameter values are lower than those in Table I, which reports the best performance with respect to all queries. However, in many cases, the performance of our methods is still better than that of the initial ranking (and that of the optimized baselines), especially for the AP corpus; furthermore, many improvements over the initial ranking for AP are also statistically significant. For TREC8 there are fewer improvements over the initial ranking, although

Table V.
Comparison with relevance models wherein the values of the free parameters of all methods (except for those of the initial ranking) are learned using cross-validation. (Recall that the performance numbers in Table II are obtained by maximizing performance over the entire set of queries.) Learning is performed with one of three regimes for splitting the query-set into learn/test sets: Leave-One-Out (loo), 10-Fold, and 2-Fold (Split). The best result in a column is boldfaced and statistically significant differences with the initial ranking are marked with i; r and e mark statistically significant performance differences of an algorithm under a certain train/test regime with Rel Model and Rel Model(Rerank), respectively, implemented with the same regime; l and f mark statistically significant performance differences of an algorithm and a train/set regime with its implementation with the loo and 10-Fold Regimes, respectively.

	AP			TREC8			WSJ		
	prec@5	prec@10	MRR	prec@5	prec@10	MRR	prec@5	prec@10	MRR
init. ranking	.457	.432	.596	.500	.456	**.691**	.536	.484	.748
loo Rel Model	.499	$.485^{i}$	.582	.508	.494	.662	**.548**	.510	.756
10fold Rel Model	$.477_{l}$	.471	.584	.516	**.506**	.673	.536	.498	.766
split Rel Model	.479	$.458_{l}$	$.617_{lf}$	.516	.494	.660	**.548**	$\mathbf{.518}_{f}$	**.780**
loo Rel Model(Rerank)	$.511^{i}$	$.482^{i}$	$.598^{r}$	.504	.476	.650	$.520^{r}$	**.506**	.721
10fold Rel Model(Rerank)	$.511^{ir}$	$.482^{i}$	$.598^{r}$	.504	.492	.663	$\mathbf{.548}_{l}$	.512	.741
split Rel Model(Rerank)	$.485_{lf}$	$.481^{i}$	.607	.520	.494	$.667^{r}$	**.548**	.520	$.764_{l}$
loo R-W-In + LM	$\mathbf{.529}^{i}$	$\mathbf{.492}^{i}$	.630	$\mathbf{.560}^{re}$	.460	.676	$.484^{ir}$	$.472^{r}$	$.665^{i}$
10fold R-W-In + LM	$.515^{i}$	$.488^{i}$	**.633**	.528	$.454^{r}$	.662	$.492^{ire}$	$.492_{l}$	.694
split R-W-In + LM	.509	$.483^{i}$	.628	.512	$.462^{i}$	.663	.516	$.484^{re}$	$.747_{l}$

the performance of Recursive Weighted Influx + LM is in most cases as good as that of the optimized baselines for prec@5 and prec@10, and better than that of the initial ranking.

For the WSJ corpus, learning parameter values yields in many cases performance inferior to that of the initial ranking—even significantly so in some cases for the Recursive Weighted Influx + LM algorithm. These differences of performance patterns over different corpora can potentially be attributed to the fact that while for AP there are 99 queries, for TREC8 and WSJ there are 50 queries. Hence, learning for AP is based on many more queries than learning for TREC8 and WSJ. Furthermore, perhaps the most important conclusion that we can draw from Table IV is that query-variability potentially plays a crucial role, as the performance numbers for different train/test split regimes can vary substantially. More specifically, no regime consistently dominates the other. A case in point: for AP and TREC8 the Recursive Weighted Influx + LM algorithm performs best with the leave-one-out (loo) regime, while for WSJ the 2-fold (split) regime yields the best performance. In fact, for AP loo results in statistically significant performance improvements over the initial ranking and optimized baselines for prec@5 and prec@10, while split does not; yet, for WSJ split yields performance that is statistically indistiguishable from that of the initial ranking, while loo yields performance that is statistically significantly worse than that of the initial ranking.

We can further observe the impact of the train/test split regime on retrieval performance in Table V, wherein we compare Recursive Weighted Influx + LM with the relevance models. For example, we observe that the split regime can have a significant impact on the performance of the relevance model; for example, for the AP corpus when using the relevance model to rank the entire corpus the 10-fold regime yields p@5 performance that is statistically significantly worse than that of the loo regime, while for TREC8 the 10-fold regime yields p@5 that is better than that of the loo regime. Another observation that we make based on Table V is that Recursive Weighted Influx + LM posts superior performance to that of the relevance models in most cases for all three split regimes over AP, and in many cases for the loo and 10-fold regimes over TREC8 (with the reverse holding for the split regime). On the other hand, the relevance models post better performance over WSJ.

All in all, we conclude that while learning free parameter values in our algorithms can yield very good performance (as is the case for AP), it is still a very challenging task (as the low performance numbers for WSJ attest). We have argued and shown that this challenge characterizes not only our algorithms, but other algorithms as well (specifically, the relevance model). Hence, we intend to further study the issue of effectively setting free-parameter values in our algorithms in future work.

5.2.2 *Further Explorations.* We now turn to further study the characteristics of our reranking methods, and factors that can impact their performance. To neutralize free-parameter value effects in the analysis to follow, we fix the free parameters of the methods to values that optimize prec@5 performance over the entire set of queries, as was the case for Tables I and II.

5.2.2.1 *Centrality and Relevance.* As stated in Section 1, one of the advantages in taking a reranking approach to the retrieval problem is that the ratio of relevant to nonrelevant documents in the initial list, $\mathcal{D}_{\text{init}}$, is in general much higher than that in the entire corpus. However, the percentage of relevant documents in $\mathcal{D}_{\text{init}}$ varies across different queries. Therefore, we now turn to analyze the effect of this percentage on the performance of our centrality-based algorithms. Specifically, we study the connection between centrality in $\mathcal{D}_{\text{init}}$ and relevance, by exploring the performance of the Weighted Influx and Recursive Weighted Influx algorithms—both of which rank documents only by their induced centrality values—when factoring out the differences in the percentage of relevant documents in $\mathcal{D}_{\text{init}}$ (across queries) using the following experimental setting.

For each query, we scan the original ranked list of documents from which $\mathcal{D}_{\text{init}}$ was created from the highest-ranked document to the document at rank 1000, and collect n relevant documents. Queries with fewer than n relevant documents among the top-1000 are discarded; we experimented with $n = 5, 10, 20, 30$, and 40. Then, we perform another pass over the ranked list (top to bottom) and accumulate $50 - n$ nonrelevant documents. $\mathcal{D}_{\text{init}}$ is then the set of 50 collected documents, n of which are relevant. Thus, our list construction is guided by the initial ranking and we can follow our general approach of reranking an initial list of documents that was retrieved in response to a query.

Figure 2 presents the prec@5 performance of our Weighted Influx and Recursive Weighted Influx algorithms when the percentage of relevant documents in

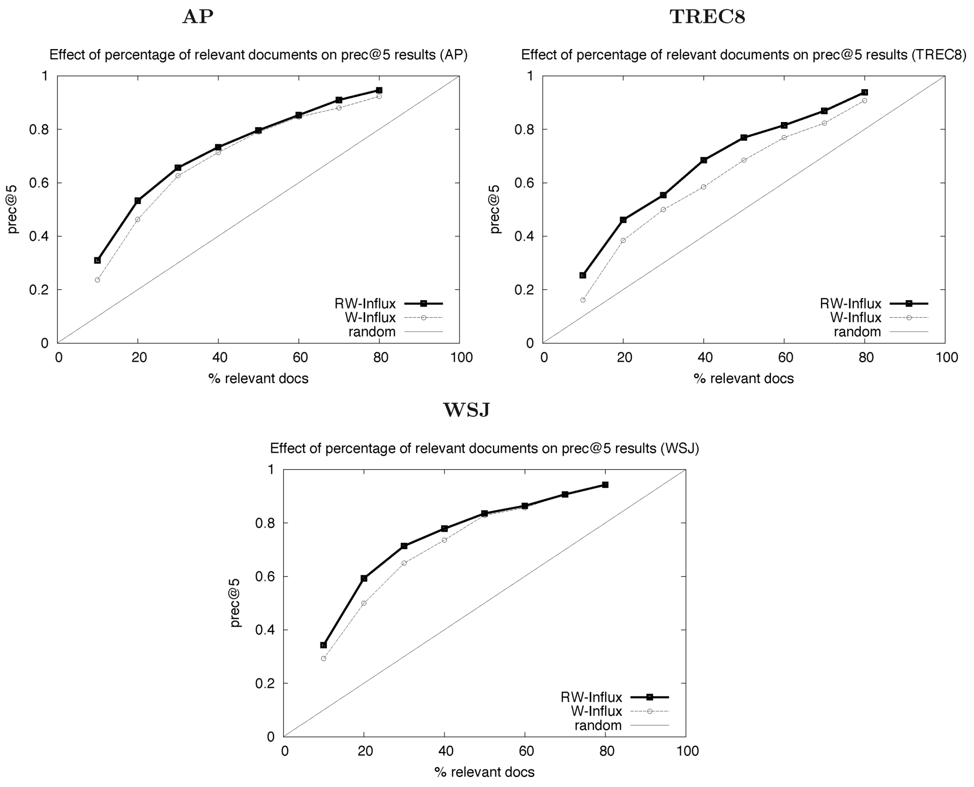

Fig. 2. The effect of the percentage of relevant documents in the initial list, $\mathcal{D}_{\text{init}}$, on the prec@5 performance of the Weighted Influx and Recursive Weighted Influx algorithms; the prec@5 performance of random selection of documents is presented for reference.

$\mathcal{D}_{\text{init}}$ is fixed as described. (We present only the prec@5—the metric for which performance was optimized—numbers to avoid cluttering the figures; prec@10 performance results exhibit the same patterns as those for prec@5.)

Our first observation with respect to Figure 2 is that for all three corpora, centrality as induced by either of the two algorithms is connected with relevance. The performance curves of both algorithms are above the diagonal line, which represents random ordering of documents in $\mathcal{D}_{\text{init}}$. We also note that for both algorithms, the performance is monotonically increasing with respect to the percentage of relevant documents over all tested corpora—a property expected from any reranking algorithm. However, our algorithms' performance on TREC8 is somewhat closer to that of random choice than on WSJ and AP. We attribute this finding to the fact that TREC8 is a much more heterogeneous corpus than the other two, and that the corresponding queries are considered challenging [Hu et al. 2003; Voorhees 2005].

In comparing the performance of the Weighted Influx and Recursive Weighted Influx algorithms in Figure 2 we clearly see that the latter is at least as effective as the former for all tested values of relevant-document percentage and over all three corpora. This finding gives further support to the hypothesis that for determining the centrality of a document, the centrality of its offspring documents should be considered, as was also shown in Section 5.2.1.

The Structure of $\mathcal{D}_{\text{init}}$. An interesting question with respect to the graphs we construct is how relevant and nonrelevant documents are situated within them. More specifically, we would like to know whether the top generators of relevant documents tend to themselves be relevant, and whether the top generators of nonrelevant documents tend to also be nonrelevant. Such an analysis is closely related to van Rijsbergen's cluster hypothesis: *"Closely associated documents tend to be relevant to the same requests"* [van Rijsbergen 1979, chapter 3]. This hypothesis has motivated many cluster-based retrieval approaches (e.g., Jardine and van Rijsbergen [1971] and Croft [1980]) and has also been explored in the reranking setting [Leuski and Allan 1998; Hearst and Pedersen 1996; Tombros 2002].

To perform this analysis, we measure the relative weights on edges from a (non-)relevant document to its top generators that are also (non-)relevant with respect to the total sum of weights on the document's outgoing edges. Indeed, this measure, when applied to the uniform-edge-weight graph, G_U, is exactly Voorhees' [1985] cluster-hypothesis test applied to the reranking setting. Figure 3 presents the values of this estimate when applied to the G_W graph (wherein edge weights represent generation probabilities) constructed with $\alpha = 5$ (we consider for each document its 5 top generators).[15] We present the resultant numbers as a function of the percentage of relevant documents in $\mathcal{D}_{\text{init}}$. We control this percentage as previously described.

Perhaps the most important conclusion based on Figure 3 can be drawn by observing the generation-weight spread when 50% of the documents in $\mathcal{D}_{\text{init}}$ are relevant. In these cases, at least 75% of the generation weight that a relevant document spreads to its top-generators is transferred to relevant documents, while more than 45% of the generation weight that a nonrelevant document spreads is transferred to relevant documents. Therefore, while the set of relevant documents in $\mathcal{D}_{\text{init}}$ keeps centrality-support within the set, the set of nonrelevant documents leaks such support to relevant documents. This observation sheds some light on the connection between centrality (as induced by our methods) and relevance, which was demonstrated by our findings.

5.2.2.2 *The Effect of the Size of* $\mathcal{D}_{\text{init}}$. We posed our centrality-computation techniques as methods for improving the results returned by an initial retrieval engine, and showed that they are successful at accomplishing this goal when the set of top retrieved documents ($\mathcal{D}_{\text{init}}$) is relatively small. (Recall that $|\mathcal{D}_{\text{init}}| = 50$.) We now study the effect of including more documents from the initial ranking in $\mathcal{D}_{\text{init}}$ on the performance of our algorithms. Figure 4 presents the performance results of the Weighted Influx, Recursive Weighted Influx, Weighted Influx + LM and Recursive Weighted Influx + LM algorithms when the number of documents in $\mathcal{D}_{\text{init}}$ is varied. We set $|\mathcal{D}_{\text{init}}|$ to values in $\{50, 100, 500, 1000\}$.

[15]Results for G_U exhibit the same exact patterns as those for G_W and are therefore omitted.

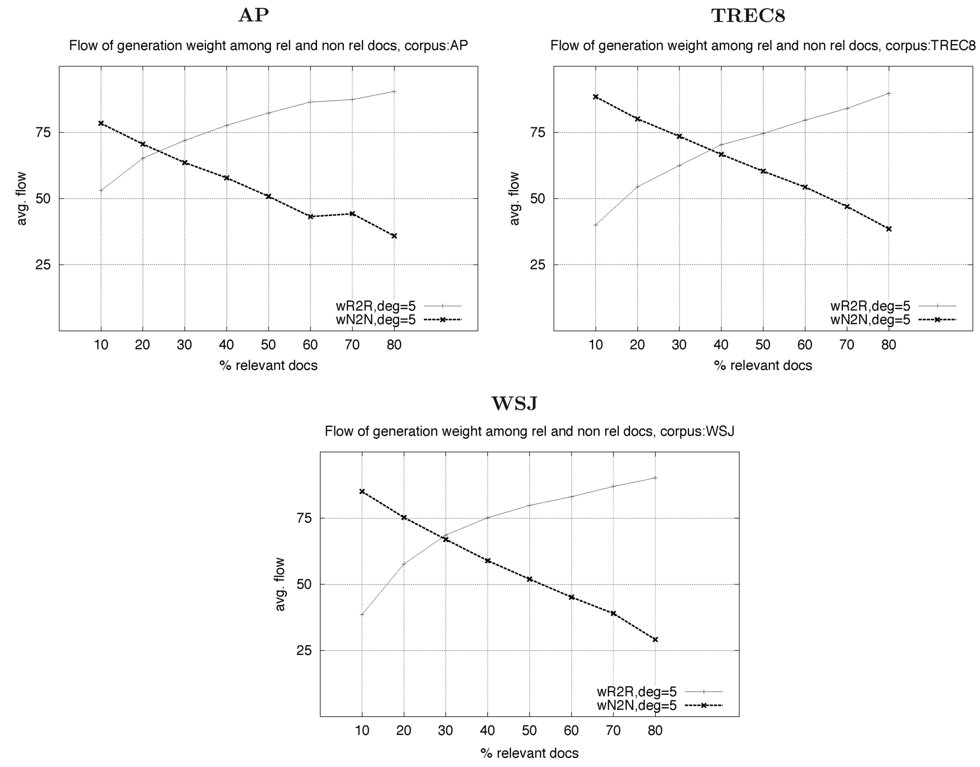

Fig. 3. The average relative spread of generation weight from a relevant document to other relevant documents (denoted "wR2R") and from a nonrelevant document to other nonrelevant documents (denoted "wN2N") as a function of the percentage of relevant documents in $\mathcal{D}_{\text{init}}$; the graph out-degree, α, is fixed at 5.

The first observation that can be made from Figure 4 is that the performance of the Weighted Influx and Recursive Weighted Influx algorithms, which use centrality values as the sole criteria for ranking, quickly degrades with increasing numbers of documents in $\mathcal{D}_{\text{init}}$. We do not see this finding as surprising, since such an increase necessarily results in a severe decrease in the ratio between the number of relevant documents to nonrelevant documents in $\mathcal{D}_{\text{init}}$. As hypothesized in Section 3 and as was shown in Section 5.2.2.1, this ratio can have a major effect on the correlation between centrality and relevance.

We also observe in Figure 4 that the performance decrease of the Weighted Influx + LM and Recursive Weighted Influx + LM algorithms in light of an increase of the size of $\mathcal{D}_{\text{init}}$ is much more gradual than that observed for the respective Weighted Influx and Recursive Weighted Influx algorithms. Furthermore, the Recursive Weighted Influx + LM algorithm posts performance that is better than that of the initial ranking for all tested values of $|\mathcal{D}_{\text{init}}|$ on all three corpora. These performance patterns can be attributed to the initial ranking score embedded in the scoring functions of the Weighted Influx + LM and Recursive Weighted Influx + LM algorithms (see Equation (3) in Section 3.3), which actually plays a dual role when increasing the size of $\mathcal{D}_{\text{init}}$: it lowers

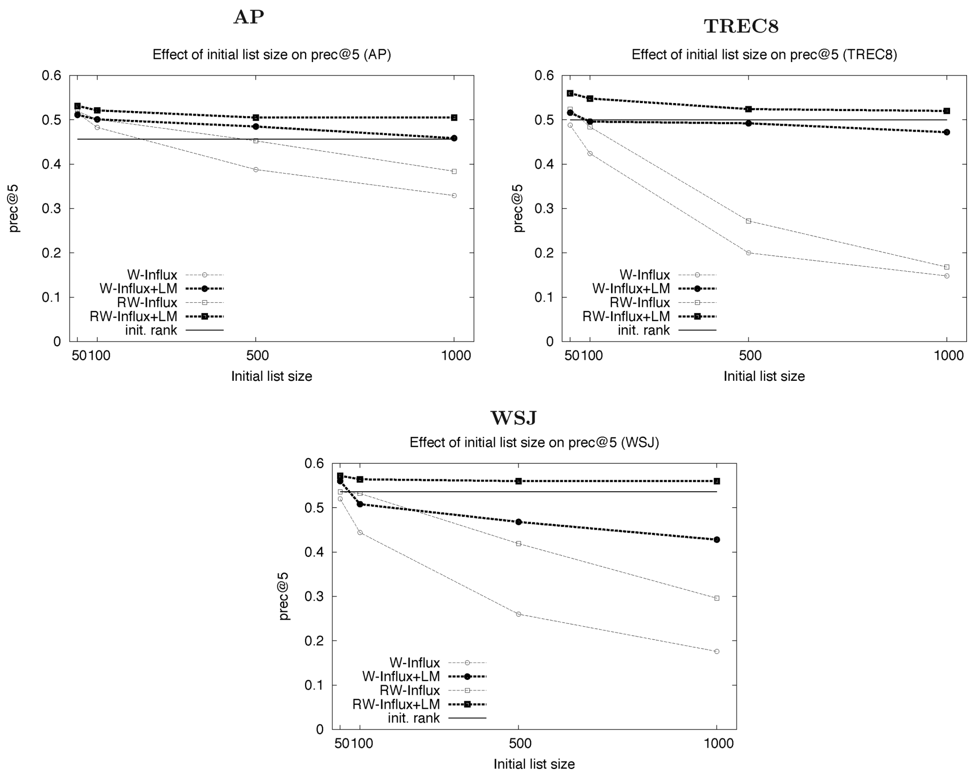

Fig. 4. The effects of varying the size of the initial list $\mathcal{D}_{\text{init}}$ on prec@5 performance.

the final score of documents initially ranked low in the list, which therefore are less likely to be relevant, and helps to handle cases in which centrality and relevance are not strongly correlated (as explained in Section 3.3).

Another point that is evident in Figure 4 is that the recursive algorithms post better performance than that of their nonrecursive analogs for almost all values of $|\mathcal{D}_{\text{init}}|$. This gives further support to the observation that in computing the centrality of a document, the centrality of its offspring documents should be considered as well.

5.2.2.3 *Information Representation and Similarity Measures*. We have advocated the use of generation relationships to define centrality, where these asymmetric relationships are based on language-model probabilities. To compare our choice with previously proposed notions of interdocument relationships, we first distinguish between two aspects. The first is *information representation*; in our framework, documents are represented via their induced unigram language models. Another well-known alternative is the vector-space representation [Salton et al. 1975]. *Similarity measure* is the second aspect we have to consider. Models based on a vector-space representation often use the cosine as a symmetric similarity measure. Indeed, as we mentioned, previous work in summarization [Erkan and Radev 2004] used the cosine to determine centrality in ways very similar to the ones we have considered.

Method	Information representation	Similarity measure ($\text{sim}(d_1,d_2)$)	
L	language model	LM generation probability	$\exp\left(-D\left(\widetilde{p}_{d_1}^{MLE}(\cdot) \,\middle\|\, \widetilde{p}_{d_2}^{[\mu]}(\cdot)\right)\right)$
S	language model	symmetric LM-based measure	$\exp\left(-J\left(\widetilde{p}_{d_1}^{MLE}(\cdot) \,\middle\|\, \widetilde{p}_{d_2}^{[\mu]}(\cdot)\right)\right)$
C	vector space	cosine	$cos(d_1, d_2)$
T	vector space	inner product	$d_1 \cdot d_2$
A	vector space	"asymmetric cosine (inner product)"	$\frac{d_1 \cdot d_2}{\|\|d_2\|\|_2} = cos(d_1, d_2) \times \|\|d_1\|\|_2$

(a) Methods for forming generation graphs. Depending on the information representation approach, d_i ($i = 1, 2$) refers either to the term sequence that d_i is composed of (language model) or to d_i's vectorial representation (vector space). D and J denote KL divergence and J divergence, respectively ($J\,(p \,\|\|\, q) = D\,(p \,\|\|\, q) + D\,(q \,\|\|\, p)$).

		U-In	W-In	U-In+LM	W-In+LM	R-U-In	R-W-In	R-U-In+LM	R-W-In+LM
AP	prec @5	$\boldsymbol{L}CAT$	$\boldsymbol{L}ACT$	$\hat{\boldsymbol{T}}LCA$	$\hat{\boldsymbol{L}}\hat{C}\hat{A}T$	$\hat{\boldsymbol{C}}\hat{A}LT$	$\boldsymbol{L}CAT$	$\hat{\boldsymbol{L}}\hat{C}\hat{A}\hat{T}$	$\hat{\boldsymbol{L}}\hat{C}\hat{A}T$
	prec @10	$\hat{\boldsymbol{L}}CTA$	$\hat{\boldsymbol{L}}CAT$	$\hat{\boldsymbol{L}}\hat{T}CA$	$\hat{\boldsymbol{L}}\hat{A}\hat{C}T$	$\hat{\boldsymbol{C}}\hat{A}LT$	$\hat{\boldsymbol{C}}\hat{A}LT$	$\hat{\boldsymbol{L}}\hat{C}\hat{A}\hat{T}$	$\hat{\boldsymbol{C}}\hat{A}\hat{L}\hat{T}$
	MRR	$\boldsymbol{L}$	$\boldsymbol{L}$	$\boldsymbol{L}$	$\boldsymbol{L}$		$\boldsymbol{L}$	$\boldsymbol{L}CA$	$\boldsymbol{L}CA$
TREC8	prec @5			$\boldsymbol{L}$				$\boldsymbol{L}$	$\boldsymbol{L}CA$
	prec @10			$\hat{\boldsymbol{L}}CA$				$\hat{\boldsymbol{C}}\hat{A}$	
	MRR				$\boldsymbol{T}$				
WSJ	prec @5		$\boldsymbol{C}$	$\boldsymbol{C}\hat{T}A$	$\boldsymbol{A}CT$		$\boldsymbol{C}A$	$\hat{\boldsymbol{T}}\hat{C}\hat{A}\hat{L}$	$\boldsymbol{T}CA\hat{L}$
	prec @10				$\boldsymbol{C}A$				$\hat{\boldsymbol{C}}\hat{A}$
	MRR			$\hat{\boldsymbol{T}}$	$\boldsymbol{L}$			$\hat{\boldsymbol{C}}\hat{A}T$	

(b) Comparison of the different methods specified in Table (a) for graph formation. In each entry, methods that post a relative improvement of at least 5% over the initial ranking are ordered left to right in nonascending order of performance (the best performing method in each entry, that is, the leftmost one, is boldfaced). A hat ("ˆ") marks methods that post a statistically significant improvement with respect to the initial ranking.

method	best	better than init. ranking	significantly better than init. ranking
L	**47.2%**	63.9%	16.7%
S	1.4%	26.4%	0%
C	36.1%	**72.2%**	**18.1%**
T	12.5%	56.9%	11.1%
A	2.8%	70.8%	**18.1%**

(c) Summary of the relative performance of methods in Table (a). For each method, the percentage of cases for which it performs the best, outperforms the initial ranking (by any margin) and significantly outperforms the initial ranking is presented. Percentages refer to the 72 relevant comparisons: 8 centrality-based algorithms × 3 corpora × 3 evaluation metrics. Bold: highest percentage per column.

Fig. 5. Comparison of different methods for defining generation graphs.

Figure 5 presents a comparison of the different methods we experimented with to define generation graphs. We focus on unigram language models and tf.idf vectors for document representation, as they represent two approaches that have formed the basis for numerous approaches in information retrieval. While there are a huge number of similarity measures one can think of, we focus on a few simple but representative choices, rather than attempt to exhaustively search the enormous space of possible models of similarity.

Table (a) of Figure 5 provides the specification of the methods we compare, focusing on differences in choice of representation and similarity measure. The first two methods are based on a smoothed unigram language-model representation. The L method is the one we advocated throughout this article, language models with generation probabilities (details in Section 3.4). Method S utilizes the J divergence [Jeffreys 1946], resulting in a symmetric variant of the generation probabilities in L; for two probability distributions p and q over terms, $J(p \,\|\|\, q) = D(p \,\|\|\, q) + D(q \,\|\|\, p)$.

The next three methods utilize the vector-space representation based on tf.idf weights. In C, the cosine of the angle between the vector representations of d_1 and d_2 determines similarity, whereas in method T we use the inner product between the respective vectors. Method A presents an asymmetric variant of the previous two measures (recall that $\cos(d_1, d_2) = \frac{d_1 \cdot d_2}{||d_1||_2 \cdot ||d_2||_2}$). It is interesting to note that this measure incorporates length normalization of d_2, which is similar in spirit to the normalization embedded in our estimates of language-model generation probabilities (refer back to Section 3.4 for details). To run the evaluation for methods S, C, T, and A we simply modify Definition (1) to use the corresponding similarity measure as the basis for determining the edge weights of our graphs.

Note that the fact that a measure is symmetric does not imply that edges (v_1, v_2) and (v_2, v_1) get the same weight even in our nonsmoothed graphs—document d_1 being a top generator of d_2 with respect to the measure does not imply the reverse. It should also be observed that the language-model weights on centrality scores (the $p_d(q)$ term in Equation (3), on which the $+$LM algorithms are based) were not replaced with the similarity measure values, which makes sense since we want our comparison to focus on the effect of different means of computing graph-based centrality.

Tables (b) and (c) in Figure 5 depict the relative performance of the different methods. In Table (b), for each choice of algorithm, evaluation measure, and dataset, we present the methods in nonascending order of performance (ordered left to right); a method is presented only if it posts a 5% relative improvement or more over the initial ranking with respect to the specific evaluation metric. In addition, Table (c) provides a summary of the relative performance of the different methods. For each method, we present as a percentage (out of the 72 relevant comparisons—8 centrality-based algorithms $\times$ 3 corpora $\times$ 3 evaluation metrics) the relative number of times it, (1) performs the best, (2) posts improvement over the initial ranking, and (3) significantly improves over the initial ranking.

As can be seen in Table (c), the L method (language-model generation probabilities) is the best performing method in a majority of the relevant comparisons. Indeed, in Table (b) we see that on AP and TREC8, L dominates the other methods. Additional pairwise comparisons between the different methods attest to the superiority of L.

Table (c) (Figure 5) also shows that both the C (cosine) and A (its asymmetric variant) methods are very effective, as can be seen by the percentage of times they significantly improve on the initial ranking. Pairwise comparisons between the two showed that neither substantially outperforms the other.

In comparing symmetric with asymmetric measures, we first see in Tables (b) and (c) that our original proposal of asymmetric generation probabilities (L) is much more effective than its symmetric version (S). When comparing the inner product (T) with its asymmetric variant (A), we see that the latter is more effective with respect to improvements over the initial ranking (and their significance). Furthermore, pairwise comparisons of the two methods give further support to the superiority of the asymmetric variant (A). However, when comparing C and A (which is C's asymmetric variant) via a pairwise comparison, neither of the two methods is superior to the other. (Although according to the best-performing criterion one might suggest that C is superior.) It is important to point out, however, that our methods S and A are not necessarily the most effective ones for (a-)symmetrizing other measures.

Overall, while language-model generation probabilities indeed seem to be an attractive choice compared to other interdocument relationships considered in past literature, we believe that the important message emerging from our findings is that the overall reranking approach is a flexible and effective paradigm that can incorporate different types of interdocument relationships when appropriate.

5.2.2.4 *Central Generators vs. Central Offspring*. So far, our use of language-model generation probabilities for graph formation was based on the idea of searching for central generators in the initial list, $\mathcal{D}_{\text{init}}$, that is, seeking documents with language models that assign high probability to terms in many other central documents. (Refer back to the discussion in Section 3.1 and to the example in Figure 1.) To validate the importance (or lack thereof) of this approach of utilizing generation probabilities, we study an alternative graph formation method that is based on searching for central offspring documents—documents with term sequences that are assigned high probability by many documents either directly or indirectly.

To explore a central-offspring search approach, we first flip the directionality of the similarity estimate that we have used; for documents $d_1, d_2 \in \mathcal{D}_{\text{init}}$ we define the offspring-directed similarity:

$$p_{d_1}^{off}(d_2) \stackrel{def}{=} p_{d_2}(d_1). \tag{5}$$

We then use this estimate in Definition (1) (Section 3) to construct graphs that describe centrality-status propagation from documents to their top offspring. Note that the resulting graphs do not necessarily reflect an edge inversion of the original graphs, since the fact that g is a top generator of o does not imply that o is a top offspring of g. (Recall our discussion in Section 3 with regard to the asymmetry embedded in our link induction method.) Also, we note that both types of graphs have an α out-degree parameter.

Thus, the Uniform Influx, Weighted Influx, Recursive Uniform Influx and Recursive Weighted Influx algorithms implemented over the new graphs rank documents in $\mathcal{D}_{\text{init}}$ by their induced offspring-centrality. However, for the +LM algorithms (see Equation (3) in Section 3.3) we still use $p_d(q)$ as bias on the centrality value, as we are only interested in testing the effectiveness of the offspring-search approach.

In Table VI we compare the performance numbers of our reranking algorithms when implemented either on the original graphs that describe flow from offspring to generators—the performance numbers are those that were presented in Table I—with the performance numbers of their implementation on the new graphs that describe flow from generators to offspring. (The latter results are denoted with an [off] prefix.)

The message arising from Table VI is clear: it is much better to use our original proposal of searching for central generators than to use the searching for central offspring approach; for all algorithms, it is almost always the case that the implementation over the original graphs results in substantially

Table VI.
Reranking documents by their centrality as generators versus their centrality as offspring (denoted with the [off] prefix). *Italics* mark the best performance in a block (algorithm × corpus × evaluation metric) and boldface highlights the best performance in a column. Statistically significant differences with the initial ranking are marked with i.

	AP			TREC8			WSJ		
	prec@5	prec@10	MRR	prec@5	prec@10	MRR	prec@5	prec@10	MRR
init. rank	.457	.432	.596	.500	.456	.691	.536	.484	.748
U-In	*.513*	*.492*^{i}	*.640*	*.500*	*.442*	*.622*	*.512*	*.472*	*.673*
[off]U-In	.384^i	.385^i	.496^i	.348^i	.322^i	.543^i	.400^i	.382^i	.504^i
W-In	*.515*	*.487*^{i}	*.643*	*.488*	*.432*	*.637*	*.520*	*.470*	*.644*
[off]W-In	.388^i	.372^i	.519^i	.340^i	.334^i	.537^i	.400^i	.392^i	.492^i
U-In + LM	*.509*	**.494**^{i}	*.631*	*.528*	**.518**^{i}	.665	*.544*	*.490*	.724
[off]U-In + LM	.457	.432	.596	.508	.458	*.710*	*.544*	.468	*.756*
W-In + LM	*.511*^{i}	*.486*^{i}	*.630*	*.516*	*.464*	*.703*	*.560*	**.500**	**.787**
[off]W-In + LM	.436	.428	.555	.504	.444	.685	.536	.460	.691
R-U-In	*.513*	*.477*	*.625*	*.520*	*.446*	*.665*	*.536*	*.478*	*.707*
[off]R-U-In	.408	.405	.513	.352^i	.322^i	.539^i	.412^i	.396^i	.524^i
R-W-In	*.519*	*.480*	*.632*	*.524*	*.446*	*.666*	*.536*	*.486*	*.699*
[off]R-W-In	.410	.403	.530	.344^i	.334^i	.527^i	.404^i	.386^i	.506^i
R-U-In + LM	*.519*^{i}	*.491*^{i}	**.652**	*.556*	*.460*	.684	**.576**^{i}	*.496*	*.757*
[off]R-U-In + LM	.461	.431	.610^i	.520	.458	**.711**	.556	.482	.726
R-W-In + LM	**.531**^{i}	*.492*^{i}	*.630*	**.560**	.460	.676	*.572*^{i}	*.496*	*.747*
[off]R-W-In + LM	.463	.442	.595	.504	*.462*	*.693*	.560	.480	.683

better performance than that obtained using the implementation over graphs constructed with the estimate from Equation (5). (That is, italics markups that indicate which of the two graph-formation approaches is more effective almost always appear in rows corresponding to the original implementation of our algorithms.)

5.2.2.5 *Nonstructural Reranking*. So far, we have discussed the use of graph-based centrality as a reranking criterion, the idea being that relationships between documents can serve as an additional source of information. Our best empirical results seem to be produced by using the weighted formulation given in Equation (3) from Section 3.3:

$$Cen(d; G) \cdot p_d(q).$$

Since in this equation $Cen(d; G)$ can be regarded as a "prior" on documents, it is natural to ask whether other previously-proposed biases on generation probabilities might prove similarly useful. The comparison is especially interesting because these biases have tended to be isolated-document heuristics; we thus refer to their use as a replacement for $Cen(d; G)$ as nonstructural reranking.

Document length has been employed several times in the past to model the intuition that longer texts contain more information [Hiemstra and Kraaij 1999; Kraaij and Westerveld 2001; Miller et al. 1999]. We refine this hypothesis to disentangle several distinct notions of information: the number of tokens in a document, the distribution of these tokens, and the number of types

Table VII.
Comparison between our use of language-model-based structural-centrality scores in Equation (3) vs. nonstructural reranking heuristics. For each evaluation setting, *italics* mark improvements over the default baseline of uniform centrality scores, stars (*) indicate statistically significant differences with this default baseline, and **Bold** highlights the best results over all eight algorithms.

	AP			TREC8			WSJ		
	prec@5	prec@10	MRR	prec@5	prec@10	MRR	prec@5	prec@10	MRR
uniform (= init)	.457	.432	.596	.500	.456	.691	.536	.484	.748
W-In	*.511∗*	*.486∗*	**.630**	*.516*	*.464*	*.703*	*.560*	**.500**	**.787**
R-W-In	**.531∗**	**.492∗**	*.630*	**.560**	*.460*	.676	**.572∗**	*.496*	.747
length	.416	.414	.551	.472	.414	.642	.480	.446	.694
log(length)	.453	.432	*.606*	.496	*.468*	*.692*	*.552*	.484	.717
entropy	*.461*	.425	*.608*	.496	*.468*	**.717∗**	*.544*	*.486*	.722
uniqTerms	.420	.413	.560	.492	.442	*.712*	.508	.456	.698
log(uniqTerms)	*.459*	.423	*.608*	.496	**.472**	*.700*	*.544*	*.490*	.723

("Salvador Salvador Salvador" contains three tokens but only one type). Thus, as substitutions for centrality in this expression, we consider not only document length, but also the entropy of the term distribution and the number of unique terms, the latter statistic having served as the basis for pivoted unique normalization in Singhal et al. [1996]. As baseline, we took the initial retrieval results; note that doing so corresponds to using a uniform bias, or equivalently, using no bias at all.

As can be seen in Table VII, taking the log of token or type count is an improvement over using the raw frequencies, often yielding above-baseline performance. The entropy is more effective than raw frequency of either tokens or types, and in one case leads to the best performance overall. However, in the majority of settings, structural reranking gives the highest accuracies.

6. CONCLUSION

For search engines, obtaining high precision at the top ranks of a retrieved list of documents is of utmost importance. In this article, we have adapted ideas from Web retrieval to settings with no hyperlink information to improve this precision. We have proposed and evaluated a number of methods for structural reranking (reranking an initially retrieved list based on interdocument similarities) using inter-document generation relationships based on language models. Our experimental results demonstrate the effectiveness of our methods in improving performance over that of the initial ranking upon which reranking is performed; moreover, one of our most effective methods posts performance that favorably compares with that of a state-of-the-art pseudo-feedback-based query expansion approach. Further analysis revealed that generation relationships seem more effective within our centrality-computation framework than relationships based on vector-space proximity, and that using interdocument relationships seems to be a promising alternative to employing the isolated-document heuristics we implemented (several of which were novel to this study).

APPENDIX

A. RELEVANCE MODEL

The relevance model approach [Lavrenko and Croft 2001] is based on the assumption that there is an underlying relevance language model $\mathcal{R}$ that generates the the terms both in the query and in the relevant documents. We follow Lavrenko and Croft [2003] and estimate $\mathcal{R}$ from documents in $\mathcal{D}_{\text{init}}$ using the Independent Identically Distributed approach; the resultant relevance model is known as RM1.

Specifically, we use Jelinek-Mercer smoothing to estimate the probability assigned by a language model induced from document $d \in \mathcal{D}_{\text{init}}$ to term w:

$$\widetilde{p}_d^{\,JM[\beta]}(w) \overset{def}{=} \beta \widetilde{p}_d^{\,MLE}(w) + (1-\beta)\widetilde{p}_{\mathcal{C}}^{\,MLE}(w).$$

(β is a free parameter; refer back to Section 3.4 for details about the MLE estimate.)

We can then define the relevance language model $\mathcal{R}$ as:

$$\widetilde{p}_{\mathcal{R}}(w;\beta) \overset{def}{=} \sum_{d_i \in \mathcal{D}_{\text{init}}} \widetilde{p}_{d_i}^{\,JM[\beta]}(w) \cdot p(d_i|q), \tag{6}$$

where for document $d \in \mathcal{D}_{\text{init}}$ we write (assuming $q = q_1, \ldots, q_l$, where l is the query length):

$$p(d|q) = \frac{p(d)\prod_j \widetilde{p}_d^{\,JM[\beta]}(q_j)}{\sum_{d_i \in \mathcal{D}_{\text{init}}} p(d_i)\prod_j \widetilde{p}_{d_i}^{\,JM[\beta]}(q_j)},$$

and set $p(d_i)$ to a constant, thereby assuming a uniform distribution over documents.

While $\widetilde{p}_{\mathcal{R}}(\cdot;\beta)$ is a probability distribution defined over the entire vocabulary, it is a common practice to clip it by using only the γ terms to which the highest probability is assigned so as to result in improved performance [Connell et al. 2004; Cronen-Townsend et al. 2004; Metzler et al. 2005; Diaz and Metzler 2006]. We denote the resultant clipped model—obtained after normalization is performed to yield a valid probability distribution[16]—as $\ddot{p}_{\mathcal{R}}(\cdot;\beta,\gamma)$.

An additional step that has recently been suggested [Abdul-Jaleel et al. 2004; Diaz and Metzler 2006] for preventing query drift [Mitra et al. 1998] is to anchor the relevance model to the original query by interpolation; the resultant interpolated relevance model (RM3) is defined as:

$$\ddot{p}_{\mathcal{IR}}(w;\beta,\gamma,\eta) \overset{def}{=} (1-\eta)\widetilde{p}_q^{\,MLE}(w) + \eta\, \ddot{p}_{\mathcal{R}}(w;\beta,\gamma), \tag{7}$$

where η is a free interpolation parameter. We can then rank all the documents in the corpus by their KL divergence from the interpolated relevance model $D(\ddot{p}_{\mathcal{IR}}(\cdot;\beta,\gamma,\eta) \;||\; \widetilde{p}_d^{\,[\mu]}(\cdot))$ [Lavrenko and Croft 2003]; we refer to this implementation as Rel Model. (μ is set to 2000 as in all our algorithms.)

[16]We used Lemur's implementation of the relevance model. As the implementation does not perform normalization of term probabilities after clipping, we introduced such normalization. We note that both approaches of normalizing and not-normalizing have merits.

Since our structural reranking algorithms rerank only the initial list, $\mathcal{D}_{\text{init}}$, we also examine the performance of using the interpolated relevance model for reranking only documents in $\mathcal{D}_{\text{init}}$ and refer to this implementation as Rel Model(Rerank).

We select the values of the free parameters on which Rel Model and Rel Model(Rerank) are dependent, namely β, γ, and η from $\{0.1, 0.3, 0.5, 0.7, 0.9, 1\}$, $\{25, 50, 75, 100, 500, 1000, 5000, \mathit{ALL}\}$ ("ALL" refers to all terms in the vocabulary, that is, no term clipping is performed) and $\{0.1, \ldots, 1\}$ respectively.

ACKNOWLEDGMENTS

We thank James Allan, Jamie Callan, Bruce Croft, Carmel Domshlak, Jon Kleinberg, Fernando Pereira and the anonymous reviewers for valuable discussions and comments. We also thank Carnegie Mellon University for its hospitality during the academic year 2004-2005.

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 13th Text Retrieval Conference (TREC-13)*. 715–725.

Amati, G., Carpineto, C., and Romano, G. 2004. Query difficulty, robustness, and selective application of query expansion. In *Proceedings of European Conference on IR Research (ECIR)*. 127–137.

Azzopardi, L., Girolami, M., and van Rijsbergen, K. 2003. Investigating the relationship between language model preplexity and IR precision-recall measures. In *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR)*. 369–370. Poster.

Baliński, J. and Daniłowicz, C. 2005. Re-ranking method based on inter-document distances. *Inform. Process. Manag. 41,* 4, 759–775.

Barzilay, R. and Lapata, M. 2005. Collective content selection for concept-to-text generation. In *Proceedings of the Human Language Technology/Empirical Methods in Natural Language Processing Conference (HLT/EMNLP)*. 331–338.

Bendersky, M. and Kurland, O. 2008. Utilizing passage-based language models for document retrieval. In *Proceedings of European Conference on IR Research (ECIR)*. 162–174.

Brin, S. and Page, L. 1998. The anatomy of a large-scale hypertextual Web search engine. In *Proceedings of the 7th International 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 the 3rd Text Retrieval Conference (TREC-3)*. 69–80.

Callan, J. P. 1994. Passage-level evidence in document retrieval. In *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR)*. 302–310.

Collins-Thompson, K. and Callan, J. 2005. Query expansion using random walk models. In *Proceedings of the 14th International Conference on Information and Knowledge Management (CIKM)*. 704–711.

Collins-Thompson, K. and Callan, J. 2007. Estimation and use of uncertainty in pseudo-relevance feedback. In *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR)*. 303–310.

Connell, M., Feng, A., Kumaran, G., Raghavan, H., Shah, C., and Allan, J. 2004. UMass at TDT. TDT2004 System Description.

Croft, W. B. 1980. A model of cluster searching based on classification. *Inform. Syst. 5*, 189–195.

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

CRONEN-TOWNSEND, S., ZHOU, Y., AND CROFT, W. B. 2002. Predicting query performance. In *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR)*. 299–306.

CRONEN-TOWNSEND, S., ZHOU, Y., AND CROFT, W. B. 2004. A language modeling framework for selective query expansion. Tech. rep. IR-338, Center for Intelligent Information Retrieval, University of Massachusetts.

DANIŁOWICZ, C. AND BALIŃSKI, J. 2000. Document ranking based upon Markov chains. *Inform. Proc. Manag. 41,* 4, 759–775.

DHILLON, I. 2001. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the 17th ACM Special Interest Group on Knowledge Discovery and Data Mining (SIGKDD) Conference*. 269–274.

DIAZ, F. 2005. Regularizing ad hoc retrieval scores. In *Proceedings of the 14th International Conference on Information and Knowledge Management (CIKM)*. 672–679.

DIAZ, F. AND METZLER, D. 2006. Improving the estimation of relevance models using large external corpora. In *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR)*. 154–161.

ERKAN, G. 2006a. Language model based document clustering using random walks. In *Proceedings of the Human Language Technology/North American Chapter of the Association for Computational Linguistics (HLT/NAACL)*.

ERKAN, G. 2006b. Using biased random walks for focused summarization. In *Proceedings of Document Understanding Conference (DUC)*.

ERKAN, G. AND RADEV, D. R. 2004. LexRank: Graph-based lexical centrality as salience in text summarization. *J. Artif. Intell. Res. 22*, 457–479.

GARFIELD, E. 1972. Citation analysis as a tool in journal evaluation. *Science 178*, 471–479.

GOLUB, G. H. AND VAN LOAN, C. F. 1996. *Matrix Computations*, Third ed. The Johns Hopkins University Press.

GRASSMANN, W. K., TAKSAR, M. I., AND HEYMAN, D. P. 1985. Regenerative analysis and steady state distributions for Markov chains. *Oper. Res. 33,* 5, 1107–1116.

GRIMMETT, G. R. AND STIRZAKER, D. R. 2001. *Probability and Random Processes*, Third ed. Oxford Science Publications.

HATZIVASSILOGLOU, V. AND MCKEOWN, K. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of the 35th Association for Computational Linguistics (ACL)/8th European Association for Computational Linguistics (EACL)*. 174–181.

HEARST, M. A. AND PEDERSEN, J. O. 1996. Reexamining the cluster hypothesis: Scatter/Gather on retrieval results. In *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR)*. 76–84.

HIEMSTRA, D. AND KRAAIJ, W. 1999. Twenty-One at TREC7: Ad hoc and cross-language track. In *Proceedings of the 7th Text Retrieval Conference (TREC-7)*. 227–238.

HU, X., BANDHAKAVI, S., AND ZHAI, C. 2003. Error analysis of difficult TREC topics. In *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR)*. 407–408. Poster.

JARDINE, N. AND VAN RIJSBERGEN, C. J. 1971. The use of hierarchic clustering in information retrieval. *Inform. Storage Retrieval 7,* 5, 217–240.

JEFFREYS, H. 1946. An invariant form for the prior probability in estimation problems. *Proceedings of the Royal Society of London. Series A, Mathematical and Physical Sciences 186,* 1007, (Sept.) 453–461.

JOACHIMS, T. 2003. Transductive learning via spectral graph partitioning. In *Proceedings of the International Conference on Machine Learning (ICML)*. 290–297.

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

KLEINBERG, J. 1999. Authoritative sources in a hyperlinked environment. *J. ACM 46*, 604–632.

KRAAIJ, W. AND WESTERVELD, T. 2001. TNO-UT at TREC9: How different are Web documents? In *Proceedings of the 9th Text Retrieval Conference (TREC-9)*. 665–671.

KRAAIJ, W., WESTERVELD, T., AND HIEMSTRA, D. 2002. The importance of prior probabilities for entry page search. In *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR)*. 27–34.

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

KURLAND, O. AND LEE, L. 2004. Corpus structure, language models, and ad hoc information retrieval. In *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR)*. 194–201.

KURLAND, O. AND LEE, L. 2005. PageRank without hyperlinks: Structural re-ranking using links induced by language models. In *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR)*. 306–313.

KURLAND, O. AND LEE, L. 2006. Respect my authority! HITS without hyperlinks utilizing cluster-based language models. In *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR)*. 83–90.

KURLAND, O., LEE, L., AND DOMSHLAK, C. 2005. Better than the real thing? Iterative pseudo-query processing using cluster-based language models. In *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR)*. 19–26.

LAFFERTY, J. D. AND ZHAI, C. 2001. Document language models, query models, and risk minimization for information retrieval. In *Proceedings of the ACM Special Interest Group on Information Retrieval (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 the Human Language Technology Conference (HLT)*. 104–110.

LAVRENKO, V. AND CROFT, W. B. 2001. Relevance-based language models. In *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR)*. 120–127.

LAVRENKO, V. AND CROFT, W. B. 2003. Relevance models in information retrieval. In W. B. Croft and J. Lafferty (Eds)., *Language Modeling for Information Retrieval*, Number 13 in Information Retrieval BookSeries, Kluwer. 11–56.

LEE, K.-S., CROFT, W. B., AND ALLAN, J. 2008. A cluster-based resampling method for pseudo-relevance feedback. In *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR)*. 235–242.

LEUSKI, A. 2001. Evaluating document clustering for interactive information retrieval. In *Proceedings of the 10th International Conference on Information and Knowledge Management (CIKM)*. 33–40.

LEUSKI, A. AND ALLAN, J. 1998. Evaluating a visual navigation system for a digital library. In *Proceedings of the 2nd European Conference on Research and Advanced Technology for Digital Libraries (ECDL)*. 535–554.

LI, X. AND CROFT, W. B. 2003. Time-based language models. In *Proceedings of the 12th International Conference on Information and Knowledge Management (CIKM)*. 469–475.

LIU, X. AND CROFT, W. B. 2002. Passage retrieval based on language models. In *Proceedings of the 11th International Conference on Information and Knowledge Management (CIKM)*. 375–382.

LIU, X. AND CROFT, W. B. 2004. Cluster-based retrieval using language models. In *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR)*. 186–193.

LIU, X. AND CROFT, W. B. 2006a. 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. 2006b. Representing clusters for retrieval. In *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR)*. 671–672. Poster.

METZLER, D. AND CROFT, W. B. 2005. A Markov random field model for term dependencies. In *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR)*. 472–479.

METZLER, D., DIAZ, F., STROHMAN, T., AND CROFT, W. B. 2005. Using mixtures of relevance models for query expansion. In *Proceedings of the 14th Text Retrieval Conference (TREC)*.

MIHALCEA, R. 2004. Graph-based ranking algorithms for sentence extraction, applied to text summarization. In *The Companion Volume to the Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics*. 170–173.

MIHALCEA, R. AND TARAU, P. 2004. TextRank: Bringing order into texts. In *Proceedings of the Empirical Methods of Natural Language Processing (EMNLP)*. 404–411. Poster.

MILLER, D. R. H., LEEK, T., AND SCHWARTZ, R. M. 1999. A hidden Markov model information retrieval system. In *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR)*. 214–221.

MITRA, M., SINGHAL, A., AND BUCKLEY, C. 1998. Improving automatic query expansion. In *Proceedings of Special Interest Group on Information Retrieval (SIGIR)*. 206–214.

MORGAN, W., GREIFF, W., AND HENDERSON, J. 2004. Direct maximization of average precision by hill-climbing, with a comparison to a maximum entropy approach. Tech. rep. 04-0367, The MITRE Corporation.

NG, A. Y., ZHENG, A. X., AND JORDAN, M. I. 2001. Stable algorithms for link analysis. In *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR)*. 258–266.

NG, K. 2000. A maximum likelihood ratio information retrieval model. In *Proceedings of the 8th Text Retrieval Conference (TREC-8)*. 483–492.

OTTERBACHER, J., ERKAN, G., AND RADEV, D. R. 2005. Using random walks for question-focused sentence retrieval. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*. 915–922.

PANG, B. AND LEE, L. 2004. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the Association for Computational Linguistics (ACL)*. 271–278.

PINSKI, G. AND NARIN, F. 1976. Citation influence for journal aggregates of scientific publications: Theory, with application to the literature of physics. *Inform. Proc. Manag. 12*, 297–312.

PONTE, J. M. AND CROFT, W. B. 1998. A language modeling approach to information retrieval. In *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR)*. 275–281.

PORTER, M. F. 1980. An algorithm for suffix stripping. *Program 14,* 3, 130–137.

PREECE, S. E. 1973. Clustering as an output option. In *Proceedings of the American Society for Information Science*. 189–190.

ROCCHIO, J. J. 1971. Relevance feedback in information retrieval. In *The SMART Retrieval System: Experiments in Automatic Document Processing*, G. Salton, Ed. Prentice Hall, 313–323.

RUTHVEN, I. AND LALMAS, M. 2003. A survey on the use of relevance feedback for information access systems. *Knowl. Eng. Rev. 18,* 2, 95–145.

SALTON, G. AND BUCKLEY, C. 1988. On the use of spreading activation methods in automatic information retrieval. In *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR)*. 147–160.

SALTON, J., WONG, A., AND YANG, C. S. 1975. A vector space model for automatic indexing. *Commun. ACM 18,* 11, 613–620.

SHAH, C. AND CROFT, W. B. 2004. Evaluating high accuracy retrieval techniques. In *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR)*. 2–9.

SINGHAL, A., BUCKLEY, C., AND MITRA, M. 1996. Pivoted document length normalization. In *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR)*. 21–29.

STEWART, W. J. 1994. *Introduction to the Numerical Solution of Markov chains*. Princeton University Press.

TAO, T. AND ZHAI, C. 2006. Regularized esitmation of mixture models for robust pseudo-relevance feedback. In *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR)*. 162–169.

TISHBY, N. AND SLONIM, N. 2000. Data clustering by Markovian relaxation and the information bottleneck method. In *Advances in Neural Information Processing Systems (NIPS) 14*. 640–646.

TOMBROS, A. 2002. The effectiveness of hierarchic query-based clustering of documents for information retrieval. Ph.D. thesis, Department of Computing Science, University of Glasgow.

TOMBROS, A., VILLA, R., AND VAN RIJSBERGEN, C. 2002. The effectiveness of query-specific hierarchic clustering in information retrieval. *Inform. Proc. Manag. 38,* 4, 559–582.

TOUTANOVA, K., MANNING, C. D., AND NG, A. Y. 2004. Learning random walk models for inducing word dependency distributions. In *Proceedings of the International Conference on Machine Learning (TCML)*.

VAN RIJSBERGEN, C. J. 1979. *Information Retrieval*, second ed. Butterworths.

VOORHEES, E. M. 1985. The cluster hypothesis revisited. In *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR)*. 188–196.

VOORHEES, E. M. 1993. Using WordNet to disambiguate word senses for text retrieval. In *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR)*. 171–180.

VOORHEES, E. M. 2002. Overview of the TREC 2002 question answering track. In *The 11th Text Retrieval Conference (TREC-11)*. 115–123.

VOORHEES, E. M. 2005. Overview of the TREC 2005 robust retrieval task. In *Proceedings of the 14th Text Retrieval Conference (TREC)*.

VOORHEES, E. M. AND HARMAN, D. K. 2005. *TREC: Experiments and Evaluation in Information Retrieval*. The MIT Press.

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

WINAVER, M., KURLAND, O., AND DOMSHLAK, C. 2007. Towards robust query expansion: Model selection in the language model framework to retrieval. In *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR)*. 729–730.

XU, J. AND CROFT, W. B. 1996. Query expansion using local and global document analysis. In *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR)*. 4–11.

YOM-TOV, E., FINE, S., CARMEL, D., AND DARLOW, A. 2005. Learning to estimate query difficulty: including applications to missing content detection and distributed information retrieval. In *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR)*. 512–519.

ZAMIR, O. AND ETZIONI, O. 1998. Web document clustering: a feasibility demonstration. In *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR)*. 46–54.

ZHAI, C. AND LAFFERTY, J. 2002. Two-stage language models for information retrieval. In *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR)*. 49–56.

ZHAI, C. AND LAFFERTY, J. D. 2001a. Model-based feedback in the language modeling approach to information retrieval. In *Proceedings of the Conference on Information and Knowledge Management (CIKM)*. 403–410.

ZHAI, C. AND LAFFERTY, J. D. 2001b. A study of smoothing methods for language models applied to ad hoc information retrieval. In *Proceedings of the ACM Special Interest Group on Information Retrieval (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 the ACM Special Interest Group on Information Retrieval (SIGIR)*. 504–511.

ZHOU, Y. AND CROFT, B. 2007. Query performance prediction in Web search environments. In *Proceedings of the ACM Special Interest Group on Information Retrieval (SIGIR)*. 543–550.

ZHU, X. J. 2005. Semi-supervised learning with graphs. Ph.D. thesis, Carnegie Mellon University.

Received March 2008; revised October 2009; accepted March 2010