PageRank without hyperlinks: Structural re-ranking using links induced by language models

Oren Kurland [1,3]
kurland@cs.cornell.edu

Lillian Lee [1,2,3]
llee@cs.cornell.edu

1. Computer Science Department, Cornell University, Ithaca NY 14853, U.S.A.
2. Language Technologies Institute, Carnegie Mellon University, Pittsburgh PA 15213, U.S.A.
3. Computer Science Department, Carnegie Mellon University, Pittsburgh PA 15213, U.S.A.

ABSTRACT

Inspired by the PageRank and HITS (hubs and authorities) algorithms for Web search, we propose a *structural re-ranking* approach to ad hoc information retrieval: we reorder the documents in an initially retrieved set by exploiting asymmetric relationships between them. Specifically, we consider *generation links*, which indicate that the language model induced from one document assigns high probability to the text of another; in doing so, we take care to prevent bias against long documents. We study a number of re-ranking 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.

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

General Terms: Algorithms, Experimentation

Keywords: language modeling, PageRank, HITS, hubs, authorities, social networks, high-accuracy retrieval, graph-based retrieval, structural re-ranking

1. INTRODUCTION

Information retrieval systems capable of achieving high precision at the top ranks of the returned results would be of obvious benefit to human users, and could also aid pseudo-feedback approaches, question-answering systems, and other applications that use IR engines for pre-processing purposes [31, 35, 32]. But crafting such systems remains a key research challenge.

The PageRank Web-search algorithm [1] uses explicitly-indicated inter-document relationships as an additional source of information beyond textual content, computing which documents are the most *central*. Here, we consider adapting this idea to corpora in which explicit links between documents do not exist.

How should we form links in a non-hypertext setting? While previous work in summarization has applied PageRank to cosine-based links [4], we draw on research demonstrating the success of using *language models* to improve IR performance in general [30, 2] and to model inter-document relationships in particular [16]. Specifically, 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] Our use of such links echoes the standard language-model-based ranking principle, first introduced in [30], that a document is relevant to the extent that its corresponding language model assigns high probability to the query. However, given that we are working with multiple documents rather than a single query, we employ a technique that compensates for length bias in estimating generation probabilities.

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 certifications that two pages are truly related [13]. In contrast, automatically-induced generation links are surely a noisier source of information. To compensate, we advocate an approach (used elsewhere as well [39, 10, 13, 20, 37, 22]) that we term *structural re-ranking*: we use inter-document 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 non-relevant documents, and thus form a good foundation for our algorithms. Note that our approach differs in spirit from pseudo-feedback-based methods [31], which define a model based on the initially retrieved documents expressly in order to re-rank 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 [35], 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 [1] and HITS (a.k.a. hubs and authorities) [13]. Comparisons on various TREC datasets against numerous baselines (including use of cosine-based links and re-ranking em-

[1] While the term "generate" is convenient, we do not think of a "generator" document or language model as literally "creating" others. Other work further discusses this issue and proposes alternate terminology (e.g., "render") [17].

SIGIR'05, August 15–19, 2005, Salvador, Brazil.

ploying only document-specific characteristics) show that language-model-based re-ranking 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.

2. STRUCTURAL RE-RANKING

Throughout this section, we assume that the following have been fixed: the corpus $\mathcal{C}$ (in which each document has been assigned a unique numerical ID); the query q; the set $\mathcal{D}_{\text{init}} \subseteq \mathcal{C}$ of top documents returned by some initial retrieval algorithm in response to q (this is the set upon which re-ranking 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 2.4). 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 below).

We use the notation (V, wt) for weighted directed graphs: V is the set of vertices and $wt : V \times V \rightarrow \{y \in \Re : y \geq 0\}$ 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 $wt(v_1 \rightarrow v_2)$ to denote the value of wt on edge (v_1, v_2).

2.1 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 (i.e., central or relevant) would provide strong evidence that d_1 is at least somewhat important. 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.) Note that ranking documents by $p_d(q)$, as first proposed by Ponte and Croft [30], can be considered a variation of this principle.

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 that* d *is a top generator of, i.e., 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 $o, g \in \mathcal{D}_{\text{init}}$,

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

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

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.

Some of our algorithms require "smoothed" versions of these graphs, in which all edges (including self-loops) have non-zero weight, to work correctly. To be specific, we employ PageRank's [1] 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 edge weights defined as follows: for every* $o, g \in \mathcal{D}_{\text{init}}$.

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

The weights of all edges leading out of any given node in $G^{[\lambda]}$ sum to 1 and thus 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 (i.e., 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.

2.2 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*:

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

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 [5], 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 in their work on *influence weights* [29], 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:

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

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 all under the normalization constraint just given; however, a unique solution *is* guaranteed to exist for their PageRank-smoothed versions.[2] By analogy with the two influx algorithms given above, then, we have the ***Recursive Uniform Influx*** algorithm, which sets $G = G_U^{[\lambda]}$ and is a direct analog of PageRank, and the ***Recursive Weighted Influx*** algorithm, which sets $G = G_W^{[\lambda]}$.

2.3 Incorporating Initial Scores

The centrality scores presented above 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 it 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 [30, 2] to determine $\mathcal{D}_{\text{init}}$, and consider the following family of re-ranking criteria:

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

where $d \in \mathcal{D}_{\text{init}}$ and Cen is one of the centrality functions defined in the previous section. 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 initial score function has the interesting consequence that it suggests interpreting $Cen(d; G)$ as a document "prior" — in fact, Lafferty and Zhai write, "with hypertext, [a document prior] might be the distribution calculated using the 'PageRank' scheme" [18]. We will return to this idea later.

2.4 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 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

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

Some prior work in language-model-based retrieval [22, 40] employs a *Dirichlet-smoothed* version:

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

the smoothing parameter μ controls the degree of reliance on relative frequencies in the corpus rather than on the counts in x. Both estimates just described are typically extended to distributions over term sequences by assuming that terms are independent: for an n-term text sequence $w_1 w_2 \cdots w_n$,

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

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

Another estimation approach, which we adopt, incorporates the Kullback-Leibler divergence D between document language models [16, 17] (see also previously proposed ranking principles [26, 18]): unless otherwise specified, for document d and word sequence s (in our setting, either a document or the query), we set $p_d(s)$ to

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

Equation 4 has some useful properties. We can show that

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

where H is the entropy function. Now, observe that for both $p_x^{MLE}(\cdot)$ and $p_x^{[\mu]}(\cdot)$, longer text sequences tend to be assigned lower probabilities; this would correspond to an unmotivated reduction of weights for edges out of long documents in the graph G_W. However, Term A length-normalizes $p_d^{[\mu]}(s)$ via the *geometric mean*, which has helped ameliorate numerical problems in previous work [19]. 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 [33, 11]. 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.)

3. RELATED WORK

Work on structural re-ranking in traditional ad hoc information retrieval has mainly focused on *query-dependent clustering*, wherein one seeks to compute and exploit a clustering of the initial retrieval results [39, 10, 20, 37, 22]. 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 [22]. Interestingly, some centrality measures have been previously employed to produce clusterings [36].

There has been increasing use of techniques based on graphs induced by implicit relationships between documents or other linguistic items [9, 3, 12, 4, 24, 28, 38]. The work in the domain of text summarization [4, 24] most resembles ours, in that it also computes centrality on graphs (although the nodes correspond to sentences or terms instead of documents). Perhaps the main contrast with our work is that links were not induced by generation probabilities; Section 4.2 presents the results of experiments studying the relative merits of our particular choice of link definition.

[2] The edge weights correspond to the transition probabilities for a Markov chain that is aperiodic and irreducible, and hence has a unique stationary distribution [8] that can be computed by a variety of means [34, 6, 7]. In our experiments, power iteration converged very quickly.

Our centrality scores constitute a relationship-based re-ranking 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 [11, 14, 25] are applicable in our setting; we report on experiments with (variants of) document length in Section 4.2. Other previously suggested biases that may be somewhat less appropriate for general domains include document source [25] and creation time [21], and webpage hyperlink in-degree and URL form [15].

4. EVALUATION

4.1 Experimental Setting

The objective of structural re-ranking 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) [32].

We are interested in the general validity of the various structural re-ranking 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 non-interpolated average precision 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 precision among the top 5 documents, not with respect to all three evaluation metrics employed.

The search ranges for the latter two parameters were:

$$\alpha: \quad 4, 9, 19, \ldots, |\mathcal{D}_{\text{init}}| - 1$$
$$\lambda: \quad 0, 0.05, 0.1, 0.2, \ldots, 0.9, 0.95$$

As it turned out, for many instances (except for the *Weighted Influx* algorithm), 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 results reported below (similar performance patterns were obtained when $|\mathcal{D}_{\text{init}}| = 100$).

The remaining details are as follows. We conducted our experiments on the following four TREC corpora:

corpus	# of docs	queries	disk(s)
AP89	84,678	1-46,48-50	1
AP	242,918	51-64, 66-150	1-3
WSJ	173,252	151-200	1-2
TREC8	528,155	401-450	4-5

(AP89 is a subset of AP containing articles just from the year 1989). All documents and queries (in our case, TREC-topic titles) were stemmed using the Porter stemmer and tokenized, but no other pre-processing steps were applied. We used the Lemur toolkit [27] for language-model estimation. Statistically-significant differences in performance were determined using the two-sided Wilcoxon test at a confidence level of 95%.

4.2 Results

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

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*

4.2.1 *Primary evaluations*

Our main experimental results are presented in Table 1. The first three rows specify reference-comparison data. The initial ranking was, as described above, produced using $p_d^{KL,\mu}(q)$ with μ chosen to optimize for non-interpolated precision at 1000. The *empirical upper bound on structural re-ranking*, which applies to any algorithm that re-ranks $\mathcal{D}_{\text{init}}$, 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: note that these bounds indicate that the initial rankings for AP89 are quite worse than those for the other three corpora. 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 how our structural re-ranking algorithms taken as a whole do. As shown in Table 1, our methods improve upon the initial ranking in many cases, specifically, roughly 2/3 of the 96 relevant comparisons (8 centrality-based algorithms × 4 corpora × 3 evaluation metrics). An even more gratifying observation is that Table 1 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 1 reveals that in about 60% of the 48 relevant comparisons, our algorithms not only are at least as effective when applied to the graph G_W as when applied to G_U, but often yield better performance results; the comparison between *Recursive Weighted*

	AP89			AP			WSJ			TREC8		
	prec@5	prec@10	MRR	prec@5	prec@10	MRR	prec@5	prec@10	MRR	prec@5	prec@10	MRR
upper bound	63.7	53.1	75.5	87.6	78.8	93.0	89.6	80.0	100.0	94.4	85.0	98.0
init. ranking	28.3	26.5	52.3	45.7	43.2	59.6	54.8	48.4	76.2	50.0	45.6	69.1
opt. baselines	30.0	27.4	**54.3**	46.5	43.9	63.5	56.0	49.4	77.2	51.2	46.4	**69.6**
U-In	29.6	*27.8*	39.5 $_{o}$	*50.9*	*49.0* $^{i}_{o}$	**66.3**	50.0	46.6	66.7	50.0	45.0	62.0
W-In	*31.3*	*29.6*	46.8	*51.3*	*48.7* ^{i}	*64.4*	52.0	47.8	63.3 $_{o}$	49.2	43.4	63.7
U-In+LM	**33.5**	27.0	46.5	*51.3* ^{i}	**49.4** $^{i}_{o}$	63.2	*56.4*	49.2	73.6	*52.8*	**52.0** $^{i}_{o}$	66.6
W-In+LM	*31.7*	*27.6*	48.4	*51.1* ^{i}	*48.4* $^{i}_{o}$	63.0	*57.2*	*50.0*	77.2	*51.6*	*49.6* ^{i}	64.5
R-U-In	*31.3*	*28.9*	46.4	*51.5*	*48.9* ^{i}	63.4	53.6	*49.6*	68.5	*52.0*	44.6	66.5
R-W-In	*32.2*	*29.6*	40.5 $_{o}$	*52.1* ^{i}	*49.1* $^{i}_{o}$	*63.9*	54.0	49.2	70.2	*52.4*	44.6	66.5
R-U-In+LM	*33.0*	*29.3*	45.8	*52.1* $^{i}_{o}$	*49.2* $^{i}_{o}$	*64.3*	**58.8** ^{i}	**51.0** ^{i}	**78.6**	*55.6*	46.0	68.4
R-W-In+LM	**33.5**	**29.8**	46.0	**52.9** $^{i}_{o}$	*49.0* $^{i}_{o}$	62.6	**58.8** ^{i}	*50.6*	**78.6**	**56.0**	45.8	67.6

Table 1: Primary experimental results, showing algorithm performance with respect to our 12 evaluation settings (3 performance metrics $\times$ 4 corpora). For each evaluation setting, improvements over the optimized baselines are given in italics; statistically significant differences between our structural re-ranking 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 re-ranking 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.

Influx (R-W-In) and *Recursive Uniform Influx* (R-U-In) is a good example. These results imply that it is a bit 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 1 is that adding in 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 80% of the 48 relevant comparisons exhibit this improvement. Most of the counterexamples occur in settings involving precision at 10 and MRR, which we did not optimize our algorithms for.

Similarly, by comparing "Y"-labeled rows with "R-Y"-labeled ones, we see that in about 70% of the 48 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 clearly better than the initial ranking for precision at 5 and 10, but are also merely 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.

It is interesting to note that the *relative* performance of our algorithms does not seem to depend strongly on the quality of the initial ranking, in the following sense. The average percentage of relevant documents among the 50 that are initially retrieved is 21%, 35.5%, 33.3% and 30.3% for AP89, AP, WSJ and TREC8, respectively, but the relative improvements for precision at 5 and 10 that our algorithms achieve with respect to the initial ranking are almost always higher on AP89 than on WSJ or TREC8.

4.2.2 Links based on the vector-space model

We have advocated the use of generation relationships to define centrality, where these asymmetric relationships are based on language-model probabilities. However, other inter-document relationships have been previously exploited in information retrieval. Perhaps the most well-known is *vector-space proximity*, with the cosine frequently used as (symmetric) closeness metric; indeed, as mentioned above, previous work in summarization [4] has used the cosine to determine centrality in ways very similar to the ones we have considered. It is thus important to examine whether the performance improvements we have achieved can be reproduced, or even surpassed, by the use of vector-space-based links rather than language-model-based generation links.

To run this evaluation, we simply modified Definition 1 and all eight of our structural re-ranking algorithms to use the cosine of the angle between log tf.idf document vectors, rather than language-model probabilities, to form the basis for determining the edge weights of our graphs. (Note that the fact that the cosine is symmetric does not imply that edges (v_1, v_2) and (v_2, v_1) get the same weight even in our non-smoothed graphs — document d_1 being a top "generator" of d_2 with respect to the cosine does not imply the reverse.) It should be observed that the language-model weights on centrality scores (i.e., the $p_d(q)$ term in Equation 3, on which the "*+LM*" algorithms are based) were *not* replaced with cosine values, which makes sense since we want our comparison to focus on the effect of different means of computing graph-based centrality.

Table 2 depicts the relative performance differences between using our language-model-based graphs and graphs induced using vector-space proximity in the manner just described. For each choice of algorithm, evaluation measure, and dataset, we indicate which formulation, if any,

		U-In	W-In	U-In+LM	W-In+LM	R-U-In	R-W-In	R-U-In+LM	R-W-In+LM
	prec @5	□	□				□		
AP89	prec @10		♦						♦
	MRR	□	□	□		□	□	□	
	prec @5	♦	♦	♦	♦	♦	♦	♦	♦
AP	prec @10	♦	♦	♦	♦	♦	♦	♦	♦
	MRR	♦	♦		♦		♦	♦	
	prec @5						♦		
WSJ	prec @10					♦	♦		
	MRR		□						
	prec @5	♦	♦			♦	♦	♦	♦
TREC8	prec @10	♦	♦		♦		♦		
	MRR	□			□				

Table 2: Structural re-ranking based on language models (LM) vs. structural re-ranking based on cosine-measured vector-space proximity (VEC). We indicate the settings in which the relative difference was at least 5% with either a "♦" (LM superior) or a "□" (VEC superior).

	AP89			AP			WSJ			TREC8		
	prec@5	prec@10	MRR	prec@5	prec@10	MRR	prec@5	prec@10	MRR	prec@5	prec@10	MRR
uniform (= init)	28.3	26.5	52.3	45.7	43.2	59.6	54.8	48.4	76.2	50.0	45.6	69.1
W-In	*31.7*	*27.6*	48.4	*51.1*∗	*48.4*∗	**63.0**	*57.2*	*50.0*	*77.2*	*51.6*	**49.6∗**	64.5
R-W-In	**33.5**	**29.8**	46.0	**52.9∗**	**49.0∗**	*62.6*	**58.8∗**	**50.6**	**78.6**	**56.0**	*45.8*	67.6
length	*29.1*	24.3	50.8	41.6	41.4	55.3	44.4∗	42.4∗	64.6∗	47.2	41.4	64.2
log(length)	*30.4*	*27.0*	*52.5*	45.3	43.2	*60.6*	*57.2*	*49.0*	69.8∗	49.6	*46.8*	*69.2*
entropy	*30.0*	26.5	**52.6**	*46.1*	42.5	*60.8*	*56.8*	*48.6*	71.1∗	49.6	*46.8*	**71.7∗**
uniqTerms	27.4	24.8	52.3	42.0	41.3	56.2	50.0	44.6	68.8	49.2	44.2	*71.2*
log(uniqTerms)	*30.4*	*27.0*	*52.5*	*45.9*	42.3	*60.8*	*57.2*	*49.0*	70.0∗	49.6	*47.2*	*70.0*

Table 3: Comparison between our use of language-model-based structural-centrality scores in Equation 3 vs. non-structural re-ranking 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.

resulted in at least 5% relative improvement with respect to the other. As can be seen, in at least three of our four corpora, our language-modeling approach seems to be a more effective basis for determining document centrality than the vector-space/cosine. We hasten to point out, though, that in most instances, vector-space proximity yielded better performance than the corresponding baselines (the results are omitted since the precise numerical comparison does not yield additional information); this finding provides further support to the idea that the overall structural re-ranking approach is a flexible and effective paradigm that can incorporate different types of inter-document relationships when appropriate.

4.2.3 Inducing centrality with the HITS algorithm

One well-known alternative method for computing centrality in a graph is the HITS algorithm [13], originally proposed for Web search. There has been some work utilizing it for text summarization in non-Web domains as well [23]. The reason we have not yet discussed it in detail is that it differs conceptually from our proposed algorithms in an important way: two different notions of centrality are identified, represented by *hub* and *authority* scores. While the concepts of hubs and authorities are highly suitable for Web-search scenarios, it is less clear whether it is useful in our setting to distinguish between the two.

As a preliminary investigation, we experimented with using hub and authority scores as measures of centrality on the generation graphs we built. Space constraints preclude a detailed discussion, but the results may be summarized as follows. We found that authority scores yielded better performance than hub scores, and that the results were generally at least as good as or better than those for the optimized baselines. However, they were slightly inferior in several cases to those of the corresponding influx algorithms. Thus, it seems that our method for graph construction can support a variety of different algorithms, but that the HITS-style hubs/authorities distinction may not be effective for the task we have addressed.

4.2.4 Non-structural re-ranking

So far, we have discussed the use of graph-based centrality as a re-ranking 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 2.3:

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

Since, as noted above, 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 "non-structural re-ranking".

Document length has been employed several times in the past to model the intuition that longer texts contain more information [11, 14, 25]. 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* ("Salvador Salvador Salvador" contains three tokens but only one type). Thus, as substitutions for centrality in the above expression, we consider not only document length, but also the entropy of the term distribution and the number of unique terms (used as the basis for pivoted unique normalization in [33]). 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 3, 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 two cases leads to the best performance overall. However, in the majority of settings, structural re-ranking gives the highest accuracies.

4.2.5 Re-ranking vs. ranking

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. But one can ask whether it is necessary to restrict our attention to an initial pool $\mathcal{D}_{\text{init}}$; that is, would we expect similarly good results if we based our generation graphs on the entire corpus? As it happens, preliminary experiments with the *Recursive Uniform Influx+LM* and *Recursive Weighted Influx+LM* algorithms on two *full* corpora (AP89 and LA combined with FR) showed that one would be better off sticking with the standard language-modeling approach if no pre-filtering of documents is available.

We do not see this finding as surprising, for our intuition is that in the re-ranking case, there is a more direct connection between centrality and relevance since we can assume that relevant documents comprise a reasonable fraction of the initial retrieval results.

5. CONCLUSION

We have proposed and evaluated a number of methods for structural re-ranking using inter-document generation relationships based on language models. Our main experiments showed that even *non-optimized* instantiations of our overall approach yield results rivaling those of *optimized* baselines. Further analysis revealed that generation relationships seem more effective within our centrality-computation framework than relationships based on vector-space proximity do, and that using inter-document relationships seems to be a promising alternative to employing the isolated-document heuristics we implemented (several of which were novel to this study). Based on our results, we believe that exploring other methods for combining statistical language models and explicitly graph-based techniques is a fruitful line for future research.

Acknowledgments. We thank James Allan, Bruce Croft, Carmel Domshlak, Jon Kleinberg, Fernando Pereira and the anonymous reviewers for valuable discussions and comments. We also thank CMU for its hospitality during the year. This paper is based upon work supported in part by the National Science Foundation under grant no. IIS-0329064 and CCR-0122581; SRI International under subcontract no. 03-000211 on their project funded 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 necessarily reflect the views or official policies, either expressed or implied, of any sponsoring institutions, the U.S. government, or any other entity.

6. REFERENCES

[1] Sergey Brin and Lawrence Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of the 7th International World Wide Web Conference*, pages 107–117, 1998.

[2] W. Bruce Croft and John Lafferty, editors. *Language Modeling for Information Retrieval*. Number 13 in Information Retrieval Book Series. Kluwer, 2003.

[3] Inderjit Dhillon. Co-clustering documents and words using bipartite spectral graph partitioning. In *Proceedings of the Seventh ACM SIGKDD Conference*, pages 269–274, 2001.

[4] Güneş Erkan and Dragomir R. Radev. LexRank: Graph-based lexical centrality as salience in text summarization. *Journal of Artificial Intelligence Research*, 22:457–479, 2004.

[5] Eugene Garfield. Citation analysis as a tool in journal evaluation. *Science*, 178:471–479, 1972.

[6] Gene H. Golub and Charles F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, third edition, 1996.

[7] Winfried K. Grassmann, Michael I. Taksar, and Daniel P. Heyman. Regenerative analysis and steady state distributions for Markov chains. *Operations Research*, 33(5):1107–1116, 1985.

[8] Geoffrey R. Grimmett and David R. Stirzaker. *Probability and Random Processes*. Oxford Science Publications, third edition, 2001.

[9] Vasileios Hatzivassiloglou and Kathleen McKeown. Predicting the semantic orientation of adjectives. In *Proceedings of the 35th ACL/8th EACL*, pages 174–181, 1997.

[10] Marti A. Hearst and Jan O. Pedersen. Reexamining the cluster hypothesis: Scatter/Gather on retrieval results. In *Proceedings of SIGIR*, 1996.

[11] Djoerd Hiemstra and Wessel Kraaij. Twenty-One at TREC7: Ad hoc and cross-language track. In *Proceedings of the Seventh Text Retrieval Conference (TREC-7)*, pages 227–238, 1999.

[12] Thorsten Joachims. Transductive learning via spectral graph partitioning. In *Proceedings of ICML*, 2003.

[13] Jon Kleinberg. Authoritative sources in a hyperlinked environment. *Journal of the ACM*, 46:604–632, 1999.

[14] Wessel Kraaij and Thijs Westerveld. TNO-UT at TREC9: How different are web documents? In *Proceedings of the Ninth Text Retrieval Conference (TREC-9)*, pages 665–671, 2001.

[15] Wessel Kraaij, Thijs Westerveld, and Djoerd Hiemstra. The importance of prior probabilities for entry page search. In *Proceedings of SIGIR*, pages 27–34, 2002.

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

[17] Oren Kurland, Lillian Lee, and Carmel Domshlak. Better than the real thing? Iterative pseudo-query processing using cluster-based language models. In *Proceedings of SIGIR*, 2005.

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

[19] Victor Lavrenko, James Allan, Edward DeGuzman, Daniel LaFlamme, Veera Pollard, and Steven Thomas. Relevance models for topic detection and tracking. In *Proceedings of the Human Language Technology Conference (HLT)*, pages 104–110, 2002.

[20] Anton Leuski. Evaluating document clustering for interactive information retrieval. In *Proceedings of the tenth International Conference on Information and Knowledge Managment (CIKM)*, pages 33–40, 2001.

[21] Xiaoyan Li and W. Bruce Croft. Time-based language models. In *Proceedings of the 12th International Conference on Information and Knowledge Managment (CIKM)*, pages 469–475, 2003.

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

[23] Rada Mihalcea. 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*, pages 170–173, 2004.

[24] Rada Mihalcea and Paul Tarau. TextRank: Bringing order into texts. In *Proceedings of EMNLP*, pages 404–411, 2004. Poster.

[25] David R. H. Miller, Tim Leek, and Richard M. Schwartz. A hidden Markov model information retrieval system. In *Proceedings of SIGIR*, pages 214–221, 1999.

[26] Kenney Ng. A maximum likelihood ratio information retrieval model. In *Proceedings of the Eighth Text Retrieval Conference (TREC-8)*, pages 483–492, 2000.

[27] Paul Ogilvie and Jamie Callan. Experiments using the LEMUR toolkit. In *Proceedings of the Tenth Text Retrieval Conference (TREC-10)*, pages 103–108, 2001.

[28] Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the ACL*, pages 271–278, 2004.

[29] Gabriel Pinski and Francis Narin. Citation influence for journal aggregates of scientific publications: Theory, with application to the literature of physics. *Information Processing and Management*, 12:297–312, 1976.

[30] Jay M. Ponte and W. Bruce Croft. A language modeling approach to information retrieval. In *Proceedings of SIGIR*, pages 275–281, 1998.

[31] Ian Ruthven and Mounia Lalmas. A survey on the use of relevance feedback for information access systems. *Knowledge Engineering Review*, 18(2):95–145, 2003.

[32] Chirag Shah and W. Bruce Croft. Evaluating high accuracy retrieval techniques. In *Proceedings of SIGIR*, pages 2–9, 2004.

[33] Amit Singhal, Chris Buckley, and Mandar Mitra. Pivoted document length normalization. In *Proceedings of SIGIR*, pages 21–29, 1996.

[34] William J. Stewart. *Introduction to the numerical solution of Markov chains*. Princeton University Press, 1994.

[35] Tao Tao and ChengXiang Zhai. A two-stage mixture model for pseudo feedback. In *Proceedings of the 27th SIGIR*, pages 486–487, 2004. Poster.

[36] Naftali Tishby and Noam Slonim. Data clustering by Markovian relaxation and the information bottleneck method. In *Advances in Neural Information Processing Systems (NIPS) 14*, pages 640–646, 2000.

[37] Anastasios Tombros, Robert Villa, and C.J. van Rijsbergen. The effectiveness of query-specific hierarchic clustering in information retrieval. *Information Processing and Management*, 38(4):559–582, 2002.

[38] Kristina Toutanova, Christopher D. Manning, and Andrew Y. Ng. Learning random walk models for inducing word dependency distributions. In *Proceedings of the International Conference on Machine Learning*, 2004.

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

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