A Probabilistic Relevance Propagation Model for Hypertext Retrieval

Azadeh Shakery
Department of Computer Science
University of Illinois at Urbana-Champaign
Illinois 61801
shakery@uiuc.edu

ChengXiang Zhai
Department of Computer Science
University of Illinois at Urbana-Champaign
Illinois 61801
czhai@cs.uiuc.edu

ABSTRACT

A major challenge in developing models for hypertext retrieval is to effectively combine content information with the link structure available in hypertext collections. Although several link-based ranking methods have been developed to improve retrieval results, none of them can fully exploit the discrimination power of contents as well as fully exploit all useful link structures. In this paper, we propose a general relevance propagation framework for combining content and link information. The framework gives a probabilistic score to each document defined based on a probabilistic surfing model. Two main characteristics of our framework are our *probabilistic view* on the relevance propagation model and propagation through *multiple sets of neighbors*. We compare eight different models derived from the probabilistic relevance propagation framework on two standard TREC Web test collections. Our results show that all the eight relevance propagation models can outperform the baseline content only ranking method for a wide range of parameter values, indicating that the relevance propagation framework provides a general, effective and robust way of exploiting link information. Our experiments also show that using multiple neighbor sets outperforms using just one type of neighbors significantly and taking a probabilistic view of propagation provides guidance on setting propagation parameters.

Categories and Subject Descriptors

H.3.3. [**Information Storage and Retrieval**]: Information Search and Retrieval—*Retrieval Models*

General Terms

Algorithms, Experimentation

Keywords

Content and link ranking, hypertext retrieval model, probabilistic relevance propagation, web information retrieval

CIKM'06, November 5–11, 2006, Arlington, Virginia, USA.

1. INTRODUCTION

Hypertext Retrieval, the task of searching for information in a hypertext collection, has been around for a while. A key characteristic that distinguishes the search task in a hypertext collection from a traditional retrieval task is the existence of link information in the former one. Although the primary goal of creating links is to guide a user to other parts of the collection, the link information can also be exploited to improve the search accuracy. The existence of this extra information makes it inappropriate to use traditional information retrieval methods, which do the retrieval task based on the content only, to do the search task.

The early works on the hypertext retrieval task were more on the literature side. Some researchers have used bibliographic citation methods to determine relationships among documents in scientific papers [32, 16, 37]. They have used different citation methods in this direction, namely direct citation, bibliographic coupling - the sharing of one or more references by two documents - and co-citation. Modha and Spangler [25] have proposed a clustering algorithm that clusters hypertext documents using words, out-links and in-links, Chakrabrti et al. [6] have developed a technique called "spectral filtering" for discovering high-quality topical resources in hyperlinked corpora and Ray Larson [22] has applied co-citation analysis methods to the World Wide Web to produce clusterings of the WWW sites that have topical similarities.

Currently with the fast growth and popularity of the World Wide Web, the search task on this huge collection of hypertext data has gained much attention. The problem of hypertext retrieval on the Web has been studied extensively. Several link-based ranking methods have been developed to improve retrieval results [20, 27, 2, 5, 3, 26, 18, 30, 14, 36, 4, 39, 24, 38, 41, 43, 19, 40, 1, 29].

Although these algorithms have been shown to improve the performance over some baseline approaches, it remains a challenging research question what is the best way to exploit the content information and the link information to maximize search accuracy. These works appear to have adopted five strategies for combining content and link information: (1) Using the query as a filter to select documents and rank them according to link-based scores (e.g., PageRank [27] and HITS [20]); (2) Computing a weighted combination of topic-specific PageRank scores, where the weights are determined by the query (Topic-sensitive PageRank [18]); (3) Using the query to compute the relevance value of each document and regulating the influence of nodes in HITS using these value (e.g. ARC [5], Bharat and Henzinger[2]); (4) Using the query to compute the relevance value of each document and propagate these values through links (Intelligent Surfer [30], [36], [29]). (5) Using sitemap links to propagate term frequencies ([38], [29]). Unfortunately, none of these combination methods can fully exploit

the discrimination power of contents as well as fully exploit all useful link structures. Despite the importance of link information, the contents of documents are clearly the most *direct* evidence regarding whether a document is relevant to a user's interest. Thus presumably, contents of the documents should be the main basis for ranking them. In this sense, among the five strategies, only the last two are close to fully exploiting the content information to improve ranking. However, the intelligent surfer only considers the in-links of a document, the "relevance propagation" method only considers direct in-links or out-links and the "term propagation" method only considers parent-child links in a sitemap. Each of these methods only considers *one* type of *explicit* neighbors and none of them fully take advantage of all the available link information. But intuitively, all neighbors can be potentially exploited; for example, both out-links and in-links may be useful as we will show in our experiments. Besides, for the propagation methods, there exist no principled framework to do the propagation. For example, the content scores can be transformed using any monotonic function without affecting the ranking, but such transformation would presumably affect the propagation. How should we transform the scores to achieve the best propagation results?

In this paper, we propose a general *probabilistic relevance propagation* framework for combining content and link information, which can fully take advantage of content information and the link structure in a principled way and can unify most existing link-based ranking algorithms. The basic idea of probabilistic relevance propagation is to first compute a content-based relevance probability score for each document using the query, and then propagate the probabilities through different groups of neighbors. We exploit the content information as a basis for finding the probability of the relevance of a document to a query and use the link structure to define different groups of neighbors to propagate the probabilities through.

After propagation, unlike [29], our model gives us a probabilistic score for each document defined based on a probabilistic surfing model. Moreover, our model supports using multiple types of neighbors, which is shown to outperform the results of using a single type of neighbor. On the other hand, the probabilistic interpretation of the model suggests that we should transfer the content-based retrieval scores to probabilities of relevance, which is shown to be beneficial in our experiments. The probabilistic interpretation also provides guidance on how to set various parameters in the propagation model.

In some sense, our work resembles previous work on spreading activation [7, 33, 12, 13, 34, 15, 28, 35, 23, 11] as both involve propagating values through a network/graph. The main difference, however, is that in these spreading activation methods, the number of steps for propagating the weights is predefined and is a small value in most of the cases, while our framework is an iterative process which iterates until the ranks converge to a limit.

We derive several special instances of the general probabilistic relevance propagation framework and show that probabilistic relevance propagation is a very general mechanism that allows us to recover most of the major existing algorithms as special cases. Moreover, it also naturally suggests several new algorithms that can combine content and link information.

In our experiments, we evaluated several propagation algorithms and the experiment results show that: (1) Using relevance propagation to combine link information and content information for scoring can improve retrieval accuracy over using only content for scoring. (2) Using multiple sets of neighbors for propagation outperforms using a single neighbor set. (3) Using probabilities to control the effect of different groups of neighbors helps. (4) Using probabilities to control the influence of each document in a neighbor set helps.

The rest of the paper is organized as follows: We present our relevance propagation framework and derive several special cases in section 2. We discuss the experiment results in section 3 and conclude in section 4 finally.

2. A PROBABILISTIC RELEVANCE PROPAGATION FRAMEWORK

Given a query, intuitively, a good result document is one whose content is related to the query topic and which is surrounded by other good documents; i.e. located in the center of a subset of the collection relevant to the query. Thus in order to maximize ranking accuracy, we need to consider the relevance of the document to the query as well as the relevance of its neighbors.

In this section, we propose a general *probabilistic relevance propagation* framework for combining relevance values of different groups of neighbors in a principled way. The basic idea of probabilistic relevance propagation is to first use the query to compute a content-based self relevance probability score for each document and then propagate the scores through the neighbors.

2.1 Probabilistic Relevance Propagation Framework

In this framework, we allow different types of neighbors to influence the quality score of a document. Figure 1 shows a sample

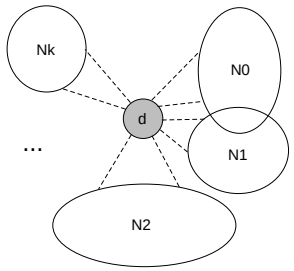


Figure 1: A typical documents d and its neighbors

document d surrounded by k different groups of neighbors. The neighbor sets are not necessarily mutually exclusive. In-links, Out-links, the single document itself and the whole set of documents are a few examples of potential neighbor sets.

Think of a random surfer surfing the Web looking for documents related to a given query q. At each step, the surfer being in a document, selects a group of neighbors surrounding the document and jumps to a document in that group. The surfer keeps doing this iteratively, jumping to neighbor documents looking for documents relevant to query q. The final score of each page is equal to the stationary probability of the surfer visiting the page.

Formally, the probability of the surfer being in each document is defined as:

$$p(x) = \sum_{i=1}^{k} \alpha_i \sum_{d \in D} p(d) p_i(d \rightarrow x)$$

$$\sum_{i=1}^{k} \alpha_i = 1 \ , \quad \sum_{x \in D} p_i(d \rightarrow x) = 1$$

where D is the set of all documents, α_i indicates the probability of choosing a particular type of neighbor set when leaving the current

document and p_i is the probability of visiting a particular page in the chosen neighbor set. Note that $p_i(a \to b)$ is positive only if b is a neighbor of a, otherwise $p_i(a \to b) = 0$.

In order to compute the scores, for each neighbor set, we construct a matrix M_i where $M_i(m,n) = p_i(d_m \to d_n)$. The probability scores can be computed using matrix multiplication: $\vec{P} = M^T\vec{P}$ where $\vec{P}$ is the vector of the probability values and $M = \Sigma_{i=1}^{k}\alpha_i M_i$. The probability values are computed iteratively through matrix multiplications in a very similar way as any of the existing link-based scoring algorithms. Clearly, efficient matrix multiplication methods can be used to further speed up the scoring. The final scores will be the values of the stationary probability distribution. To ensure reachability to each document, we generally would include the whole set of documents as one special set of neighbors in our propagation framework. Thus by the Ergodicity theorem for Markov chains [17], we know that the Markov chain defined by such a transition matrix M must have a unique stationary probability distribution.

In this framework, we identify three groups of probabilistic parameters:

- *Hyper-Relevance Probability* $p(d)$*:* Defined for each document indicating the probability of visiting the document.
- *Neighbor Set Selection Probability* α*:* Defined for each group of neighbors indicating the probability of choosing a particular type of neighbor set when leaving the current document.
- *Navigation Probability* $p_i(d \to x)$*:* Defined for each document in a specific group indicating the probability of visiting a particular page in the chosen neighbor set.

2.2 Special Cases

By setting α_is to different values and instantiating p_i's with specific functions, we can easily obtain many special cases of our general relevance propagation framework. In particular, the framework can recover most existing link-based algorithms. Table 1 shows a family of relevance propagation algorithms which are covered by our general framework.

As can be seen from the table, PageRank and its extensions are special cases of the framework. The HITS algorithm is not directly a special case, since it does not satisfy the probability property. But with minor changes, i.e. normalization of the weights, it will be a special case. In this table, we include the normalized version of HITS as well as the normalized version of its extensions.

2.3 Parameter Estimation

In our framework, we have identified three groups of probabilistic parameters: content relevance probabilities, neighbor set selection probabilities and navigation probabilities. The content relevance probability of a document can be estimated based on its relevance score given by any content-based retrieval method. Neighbor set selection probabilities and navigation probabilities can be either set to uniform or estimated based on content-based relevance scores. In this subsection, we show how to estimate the parameters. We compare different estimation methods in the following section.

2.3.1 Content Relevance Probabilities

In our probabilistic framework we should convert the original content scores to probabilities. The specific conversion method is inevitably dependent on the specific content scoring method, but with some training data, we may use techniques such as logistic regression to do the conversion. If the original retrieval model is a probabilistic model, we have some natural analytical way to transform the scores. As an example of this transformation, here we show how we compute relevance probabilities from Okapi scores and from language model(LM) scores.

• Okapi

Having the Okapi scores, our goal is to normalize the scores to find the *relevance probabilities* of the documents to the query. We use logistic regression to normalize the scores [31]:

The Okapi score is $a * X + b$, where X is the log odds of relevance, i.e., $\log(p(rel)/(1-p(rel))$. So, to recover the probability $p(rel)$, we have $p(rel) = \exp(x)/(1+\exp(x))$. Given a score s, we have $s = aX + b$, or $X = (s-b)/a$. Thus, the normalization formula should be $p(rel) = \exp((s-b)/a)/(1+\exp((s-b)/a))$.

In order to set a and b, we assume that the minimum score *min* corresponds to a very small probability δ. We also assume that the maximum score *max* corresponds to $p(rel) = \Delta$. Solving these equations will give us values for a and b:

$$a = \frac{min - max}{\log(\frac{\delta}{1-\delta}) - \log(\frac{\Delta}{1-\Delta})}$$

$$b = \frac{max \times \log\frac{\delta}{1-\delta} - min \times \log\frac{\Delta}{1-\Delta}}{\log\frac{\delta}{1-\delta} - \log\frac{\Delta}{1-\Delta}}$$

• Language Modeling Approach

In the language modeling approach, we score a document D w.r.t. a query Q by $s = \log p(Q|D)$ [42]. Thus we can do an exponential transformation to recover the probability of relevance. That is, $p(rel) \propto p(Q|D)p(D) \propto exp(s)$ (assuming uniform p(D)).

2.3.2 Neighbor Set Selection Probabilities

The easiest way to estimate neighbor set selection probabilities is uniform estimation, counting all the neighbor sets to be equal, i.e. $\alpha_i = \frac{1}{k}$.

But obviously this is not the best we can do. Our framework suggests to use relevance scores for Neighbor set probability estimation. We get our intuition for defining neighbor set selection probabilities from the surfer model. In the surfer model, in each step, the surfer should decide on the neighbor set he wants to jump to. Intuitively, the surfer will select the neighbor set based on the average relevance of the documents in the neighbor set, the higher the average relevance, the more probable the surfer will select that group. Using this intuition, we set α_i using:

$$\alpha_i \propto \frac{1}{|N_i|}\sum_{X \in N_i} rel(X)\,, \qquad \sum \alpha_i = 1$$

2.3.3 Navigation Probabilities

Like neighbor set selection probabilities, the navigation probabilities are most easily estimated through uniform estimation. But intuitively, estimating the probabilities using relevance values should give better results.

We define navigation probabilities based on the content relevance probabilities of target pages. The higher the probability of the relevance of the target page, the higher the probability of navigating the link: $p(d \to x) \propto p(x)$.

2.4 Summary

The proposed framework provides a general probabilistic interpretation of relevance-based propagation through multiple sets of neighbors. It can unify most existing link-based ranking algorithms, making it possible to compare the assumptions made in each specific algorithm. It also makes it possible to systematically explore

Table 1: Probabilistic Relevance Propagation Algorithms

Method	k	Neighbors	α_is	p_is
PageRank[27]	2	N_0: Set of all documents N_I: Set of In-links	$\alpha_0 > 0$ const. $\alpha_I > 0$ const.	$P_0(d \to x) = \frac{1}{N}$ $P_I(d \to x) = \frac{1}{\lvert OUT(d)\rvert}$
Topic-Sensitive PR[18]	2	N_0: Set of all documents N_I: Set of In-links	$\alpha_0 > 0$ const. $\alpha_I > 0$ const.	$P_0(d \to x) = \begin{cases} \frac{1}{\lvert C_j\rvert} & \text{if } d \text{ in ODPC}^1(c_j) \\ 0 & \text{o.w.} \end{cases}$ $P_I(d \to x) = \frac{1}{\lvert OUT(d)\rvert}$
Intelligent Surfer[30]	2	N_0: Set of all documents N_I: Set of In-links	$\alpha_0 > 0$ const. $\alpha_I > 0$ const.	$P_0(d \to x) = \frac{Rel(x)}{\sum_{k \in D} Rel(k)}$ $P_I(d \to x) = \frac{Rel(x)}{\sum_{d \to k} Rel(k)}$
Authorities **Normalized HITS[20]**	1	N_{CC}: Set of Co-Citations	$\alpha_{CC} = 1$	$P_{CC}(d \to x) \propto$ $\begin{cases} \lvert\text{IN}(d)\rvert & \text{if } d = x \\ \#\text{Common Parents} & \text{o.w.} \end{cases}$
Hubs	1	N_{CR}: Set of Co-References	$\alpha_{CR} = 1$	$P_{CR}(d \to x) \propto$ $\begin{cases} \lvert\text{OUT}(d)\rvert & \text{if } d = x \\ \#\text{Common Children} & \text{o.w.} \end{cases}$
Authorities **Normalized Weighted HITS[2]**	1	N_{CC}: Set of Co-Citations	$\alpha_{CC} = 1$	$P_{CC}(d \to x) \propto$ $\text{Rel}(x) \times \begin{cases} \lvert\text{IN}(d)\rvert & \text{if } d = x \\ \#\text{Common Parents} & \text{o.w.} \end{cases}$
Hubs	1	N_{CR}: Set of Co-References	$\alpha_{CR} = 1$	$P_{CR}(d \to x) \propto$ $\text{Rel}(x) \times \begin{cases} \lvert\text{OUT}(d)\rvert & \text{if } d = x \\ \#\text{Common Children} & \text{o.w.} \end{cases}$
Authorities **Normalized ARC[5]**	1	N_{CC}: Set of Co-Citations	$\alpha_{CC} = 1$	$P_{CC}(d \to x) \propto$ $\text{Rel}(\text{anchor}(x)) \times \begin{cases} \lvert\text{IN}(d)\rvert & \text{if } d = x \\ \#\text{Common Parents} & \text{o.w.} \end{cases}$
Hubs	1	N_{CR}: Set of Co-References	$\alpha_{CR} = 1$	$P_{CR}(d \to x) \propto$ $\text{Rel}(\text{anchor}(x)) \times \begin{cases} \lvert\text{OUT}(d)\rvert & \text{if } d = x \\ \#\text{Common Children} & \text{o.w.} \end{cases}$
Authorities **Normalized Randomized HITS[26]**	2	N_0: Set of all documents N_{CC}: Set of Co-Citations	$\alpha_0 > 0$ const. $\alpha_{CC} > 0$ const.	$P_0(d \to x) = \frac{1}{N}$ $P_{CC}(d \to x) \propto \begin{cases} \lvert\text{IN}(d)\rvert & \text{if } d = x \\ \#\text{Common Parents} & \text{o.w.} \end{cases}$
Hubs	2	N_0: Set of all documents N_{CR}: Set of Co-References	$\alpha_0 > 0$ const. $\alpha_{CR} > 0$ const.	$P_0(d \to x) = \frac{1}{N}$ $P_{CR}(d \to x) \propto \begin{cases} \lvert\text{OUT}(d)\rvert & \text{if } d = x \\ \#\text{Common Children} & \text{o.w.} \end{cases}$

the algorithm space and compare different components of algorithms. Moreover, taking a strict probabilistic view of propagation provides guidance on how to normalize content scores and how to set other propagation parameters to optimize retrieval accuracy, as will be shown later in the paper.

3. COMPARISON OF RELEVANCE PROPAGATION ALGORITHMS

We have done some experiments to evaluate the performance of our proposed models. In this section, we present our experiment results.

3.1 Experiment Design

3.1.1 Data Set and Baseline Methods

As our data set, we used the ".GOV" test collection, which is an 18 gigabyte, 1.25 million document 2002 partial crawl of the .gov domain used in TREC-2002, TREC-2003 and TREC-2004 experiments for topic distillation [8, 9, 10]. We used two sets of queries in our experiments: (1) 50 "topic distillation" topics created by NIST for TREC-2003 and (2) 75 "topic distillation" topics created by NIST for TREC-2004. The topics are keyword queries for which key resources exist within the .GOV collection.

An important advantage of using this set of data is that it is created carefully for the purpose of evaluating Web retrieval algorithms with a significant number of judgments available for quantitatively comparing different methods.

In our experiments, we used two baseline methods: Okapi and Language Modeling approach. Since our exploration is orthogonal to the use of anchor text and many other heuristics which are known to improve the performance, we preferred not to enter these heuristics in our baseline. Despite this, we already have a very strong baseline compared to the reported results in TREC2003 [9] and TREC2004 [10]. We expect the performance to be further improved when we use other heuristics on top of our method.

3.1.2 Neighbor Sets

In our experiments, we compare the performance of using two types of neighbors: The set of documents which have links to the document(IN) and the set of documents which are linked from the document(OUT). There also exist a universal neighbor set N_0 which contains all the documents in the collection. Selecting this universal neighbor set to jump to is equivalent to jumping to a random page.

3.1.3 Content Relevance Probabilities

The probabilistic relevance propagation framework allows us to use any content-based retrieval algorithm from which we can com-

pute the relevance probabilities. In our experiments, we try two baseline methods: Okapi and language modeling approach and compute the relevance probabilities from these relevance scores.

3.1.4 Neighbor Set Selection and Navigation Probabilities

As mentioned earlier, α_is are the parameters which indicate the probability of choosing a particular type of neighbor when leaving the current document. In our experiments, we follow one of the two approaches: either manually set α_i to different values from 0 to 1 or automatically set α_i using neighbor set selection probabilities.

Navigation probabilities on the other hand indicate the probability of visiting a particular page in a group. In our experiments, we use two different estimations of these probabilities: Uniform estimation ("Uni") and relevance based estimation ("Wt").

3.2 Result Analysis

3.2.1 Effectiveness of Exploiting Link Information

The first research question we want to answer is whether applying probabilistic relevance propagation on top of a content-based retrieval method would improve the performance. Most existing studies of link-based scoring algorithms focus on comparing different link-based algorithms without comparing link-based algorithms with scoring using only contents. The Web Track of TREC has seen some evaluation of effectiveness of exploiting link information to improve content-based scoring, but the results are not quite conclusive due to the many uncontrolled factors.

To answer the first research question, we compare the performance of using two types of neighbors: in-links and out-links. (Note that we also have the universal neighbor set). We will have:

$$\begin{aligned} p(x) &= \alpha_0 \Sigma_{d \in D} p(d) p_0(d \rightarrow x) \\ &+ \alpha_I \Sigma_{d \in IN} p(d) p_I(d \rightarrow x) \\ &+ \alpha_O \Sigma_{d \in OUT} p(d) p_O(d \rightarrow x) \end{aligned}$$

where α_0 is the probability of randomly jumping to a page, α_I is the probability of jumping to an in-link and α_O is the probability of jumping to an out-link. Jumping probabilities can either be uniform (considering all the members to be equal) or weighted based on relevance probabilities. We also consider the combination of the two types of neighbors. This gives us eight combinations, which we compare with the content-only baseline in tables 2 and 3. The numbers in parenthesis show the percent of improvement over the baseline methods. We use precision at 10 and average precision for comparison. The shown results are the best performances achieved by these methods through tuning the parameter α manually in the probabilistic relevance propagation model; we will analyze the sensitivity later.

From tables 2 and 3, we can make the following observations:

1. On both query sets, both types of neighbors can outperform the baseline significantly.
2. Weighted propagation of probabilities outperform uniform propagation.
3. The combination of different types of neighbors outperform using any single neighbor set.
4. We get significant improvement using both Okapi and LM baselines.

Overall, we see that the probabilistic relevance propagation framework is reasonable and all these specific derived algorithms can help improve search results.

Table 2: Combining Link and Content - Okapi Baseline

Method	TREC-2003	
	Prec@10	**Avg. Prec**
Baseline	0.108	**0.121**
Uni-IN	0.118(9.3%)	**0.145(20%)**
Wt-IN	0.128(18.5%)	**0.144(19%)**
Uni-OUT	0.118(9.3%)	**0.151(24.8%)**
Wt-OUT	0.122(13%)	**0.163(34.7%)**
Uni-IN Uni-OUT	0.126(16.7%)	**0.168(38.8%)**
Uni-IN Wt-OUT	0.132(22.2%)	**0.166(37.2%)**
Wt-IN Uni-OUT	0.138(27.8%)	**0.173(43%)**
Wt-IN Wt-OUT	0.138(27.8%)	**0.179(47.9%)**
	TREC-2004	
	Prec@10	**Avg. Prec**
Baseline	0.129	**0.093**
Uni-IN	0.18(39.5%)	**0.125(34.4%)**
Wt-IN	0.181(40.3%)	**0.125(34.4%)**
Uni-OUT	0.156(20.9%)	**0.112(20.4%)**
Wt-OUT	0.157(21.7%)	**0.113(21.5%)**
Uni-IN Uni-OUT	0.179(38.8%)	**0.124(33.3%)**
Uni-IN Wt-OUT	0.184(42.6%)	**0.127(36.6%)**
Wt-IN Uni-OUT	0.18(39.5%)	**0.125(34.4%)**
Wt-IN Wt-OUT	0.188(45.7%)	**0.127(36.6%)**

3.2.2 Effectiveness of Combining Different Groups of Neighbors

In tables 4 and 5, we compare the results of using only one type of neighbor with the results when we consider multiple groups of neighbors. We did a Wilcoxon signed rank test to see if the improvement on Average Precision is statistically significant. In these tables we compare the best results for each type of neighbor. Statistically significant improvements are distinguished by a star($*$).

As the tables show, combining different groups of neighbors improves the performance over using a single set of neighbors. Potentially, we can improve the performance by adding new types of neighbors, e.g. co-citations (documents which have at least one common parent with the document) and co-references (documents which have at least one common child with the document).

3.2.3 Content Score Transformation

The probabilistic framework suggests that we should convert the original content scores to probabilities. In our experiments, we use Okapi and LM methods as our baseline and transform the scores to probabilities using logistic regression and exponential transformation respectively.

Figures 2 and 3 compare the performance of probabilistic transformation with the performance of the original raw score propagation as done in all the previous work. As can be seen, the performance is much better when we use probabilistic transformation.

3.2.4 Comparison of Estimation Methods

• Relevance-Based Estimate of α Improves over Uniform Estimate.

In one set of experiments, we tried to set αs automatically based on the average relevance values of neighbors. Table 6 compares the results of relevance-based estimation of α with uniform estimation. As the table shows, in most of the cases relevance-based estimation gives better results. Note that these results are completely automatic; i.e. we do not have to tune any parameters. Thus these improvements are very encouraging. These results also confirm that using multiple neighbor sets improves over using just a single neighbor set.

Table 3: Combining Link and Content - LM Baseline

Method	TREC-2003	
	Prec@10	**Avg. Prec**
Baseline	0.092	**0.099**
Uni-IN	0.118(28.3%)	**0.135(36.4%)**
Wt-IN	0.118(28.3%)	**0.142(43.4%)**
Uni-OUT	0.106(15.2%)	**0.129(30.3%)**
Wt-OUT	0.11(19.6%)	**0.135(36.3%)**
Uni-IN Uni-OUT	0.118(28.3%)	**0.150(51.5%)**
Uni-IN Wt-OUT	0.122(32.6%)	**0.142(43.4%)**
Wt-IN Uni-OUT	0.126(37%)	**0.145(46.5%)**
Wt-IN Wt-OUT	0.128(39.1%)	**0.144(45.5%)**
	TREC-2004	
	Prec@10	**Avg. Prec**
Baseline	0.129	**0.095**
Uni-IN	0.165(27.9%)	**0.113(18.9%)**
Wt-IN	0.167(29.5%)	**0.115(21.1%)**
Uni-OUT	0.141(9.3%)	**0.107(12.6%)**
Wt-OUT	0.144(11.6%)	**0.11(15.8%)**
Uni-IN Uni-OUT	0.16(24%)	**0.115(20.8%)**
Uni-IN Wt-OUT	0.163(26.4%)	**0.116(21.1%)**
Wt-IN Uni-OUT	0.164(27.1%)	**0.117(23.2%)**
Wt-IN Wt-OUT	0.167(29.5%)	**0**.119(25.3%)

• Relevance-Based Estimate of $p_i(d \rightarrow x)$ Improves over Uniform Estimate.

In table 7, we compare the results of uniformly setting the navigation weights vs. estimating them based on relevance scores. As the table shows, relevance based estimation improves the performance in most of the cases.

3.2.5 Sensitivity Analysis

We have so far only looked at the best performance using each method. We now turn to the question about how sensitive each method is to the setting of the parameter α, which controls the amount of influence from the neighbors. To answer this research question, we compute an "optimal range" of parameter values for each method, which is defined as the interval of parameter values for which a method outperforms the baseline. Table 8 shows the optimal ranges for four of our algorithms.

We see that, in general, the optimal range is wide for most methods, indicating that exploiting these groups of neighbors for relevance propagation is useful. The uniform methods are generally more sensitive to the setting of α, which indicates that using weighted methods is more robust.

In figures 2 and 3, we show the complete picture of the sensitivity of these methods with prec@10 and average precision.

4. CONCLUSIONS

In this paper, we proposed a general *probabilistic relevance propagation* framework for combining content and link information in a principled manner to fully take advantage of query-based content scoring and link structures. The framework can unify most existing link-based ranking algorithms and can also suggest several interesting new algorithms through different propagation strategies.

Following the probabilistic relevance propagation framework, we systematically compared eight specific relevance propagation models on two TREC test collections for Web retrieval.

Our results show that all the eight relevance propagation models that we tested can outperform the baseline content only ranking method for a wide range of parameter values, indicating that

Table 4: Using Multiple Neighbors vs. a Single Neighbor Set-TREC-2003

Multi Neighbor Sets	Single Neighbor Set		Improvement
Okapi Baseline			
Uni-IN Uni-OUT 0.168	Uni-IN	0.145	15.9% *
	Uni-OUT	0.151	11.3%
Uni-IN Wt-OUT 0.166	Uni-IN	0.145	14.5% *
	Wt-OUT	0.163	1.8%
Wt-IN Uni-OUT 0.173	Wt-IN	0.144	20.1% *
	Uni-OUT	0.151	14.6%
Wt-IN Wt-OUT 0.179	Wt-IN	0.144	24.3% *
	Wt-OUT	0.163	9.8%
LM Baseline			
Uni-IN Uni-OUT 0.150	Uni-IN	0.135	11.1% *
	Uni-OUT	0.129	16.3%
Uni-IN Wt-OUT 0.142	Uni-IN	0.135	5.2% *
	Wt-OUT	0.135	5.2%
Wt-IN Uni-OUT 0.145	Wt-IN	0.142	2.1%
	Uni-OUT	0.129	12.4% *
Wt-IN Wt-OUT 0.144	Wt-IN	0.142	1.4%
	Wt-OUT	0.135	6.7% *

Table 5: Using Multiple Neighbors vs. a Single Neighbor Set-TREC-2004

Multi Neighbor Sets	Single Neighbor Set		Improvement
Okapi Baseline			
Uni-IN Uni-OUT 0.124	Uni-IN	0.125	-
	Uni-OUT	0.112	10.7% *
Uni-IN Wt-OUT 0.126	Uni-IN	0.125	0.8%
	Wt-OUT	0.113	11.5%
Wt-IN Uni-OUT 0.125	Wt-IN	0.125	-
	Uni-OUT	0.112	11.6% *
Wt-IN Wt-OUT 0.127	Wt-IN	0.125	1.6%
	Wt-OUT	0.113	12.4% *
LM Baseline			
Uni-IN Uni-OUT 0.115	Uni-IN	0.113	1.8%
	Uni-OUT	0.107	7.5% *
Uni-IN Wt-OUT 0.116	Uni-IN	0.113	2.7%
	Wt-OUT	0.11	5.5% *
Wt-IN Uni-OUT 0.117	Wt-IN	0.115	1.7%
	Uni-OUT	0.107	9.3% *
Wt-IN Wt-OUT 0.119	Wt-IN	0.115	3.5%
	Wt-OUT	0.11	8.1% *

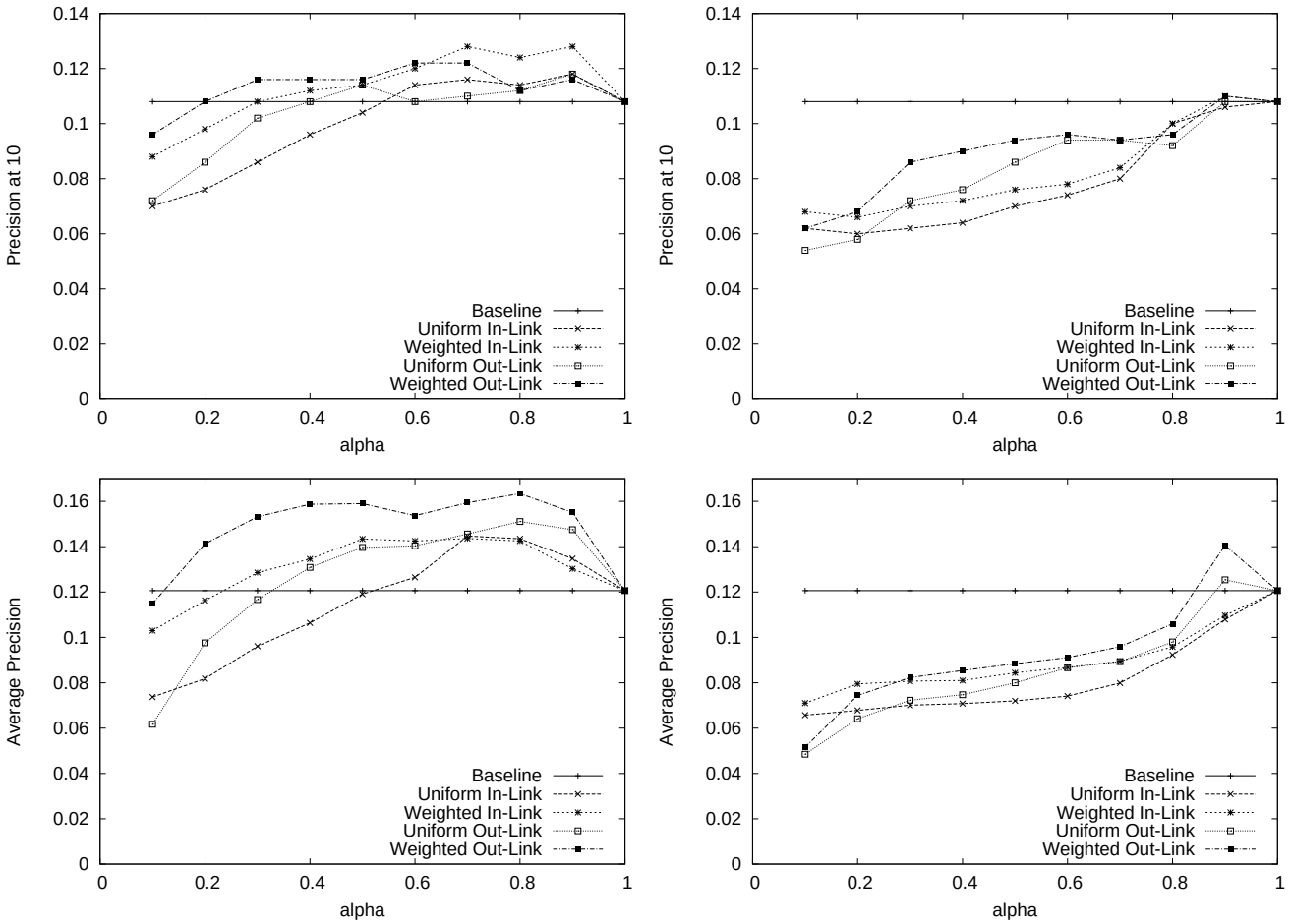
the relevance propagation framework provides a general, effective and robust way of exploiting link information to improve hypertext search accuracy.

While the previous work all uses just one type of neighbor for propagation, we have shown that using multiple neighbor sets outperforms using just one type of neighbors significantly. We have also shown that taking a probabilistic view of propagation provides guidance on setting propagation parameters, that using content scores to estimate the probabilities of relevance improves the performance and that relevance based estimation of the parameters helps us improve the results.

There are several interesting directions for further research:

1. Our framework naturally accommodates the use of anchor text through estimating navigation parameters based on anchor text. It is interesting to see how this estimation compares with our current estimation methods.

2. We have shown that in-links and out-links are useful for relevance propagation and can outperform the "content only"



(a) Results using relevance probabilities

(b) Results using raw content scores

Figure 2: TREC-2003 results with Okapi baseline

Table 6: Effectiveness of Neighbor Set Selection Probability Estimation (α)

Method	Uniform Estimate		Relevance Estimate	
	Prec@10	**Avg Prec**	Prec@10	**Avg Prec**
Baseline	0.108	**0.1206**	0.108	**0.1206**
Uni-IN	0.104	**0.1191**	0.114	**0.1438**
Wt-IN	0.114	**0.1434**	0.124	**0.1496**
Uni-OUT	0.114	**0.1397**	0.114	**0.1563**
Wt-OUT	0.116	**0.159**	0.118	**0.1556**
Uni-IN Uni-OUT	0.116	**0.1352**	0.118	**0.1586**
Uni-IN Wt-OUT	0.124	**0.1578**	0.122	**0.16**
Wt-IN Uni-OUT	0.124	**0.1677**	0.13	**0.1699**
Wt-IN Wt-OUT	0.128	**0.175**	0.138	**0.1694**

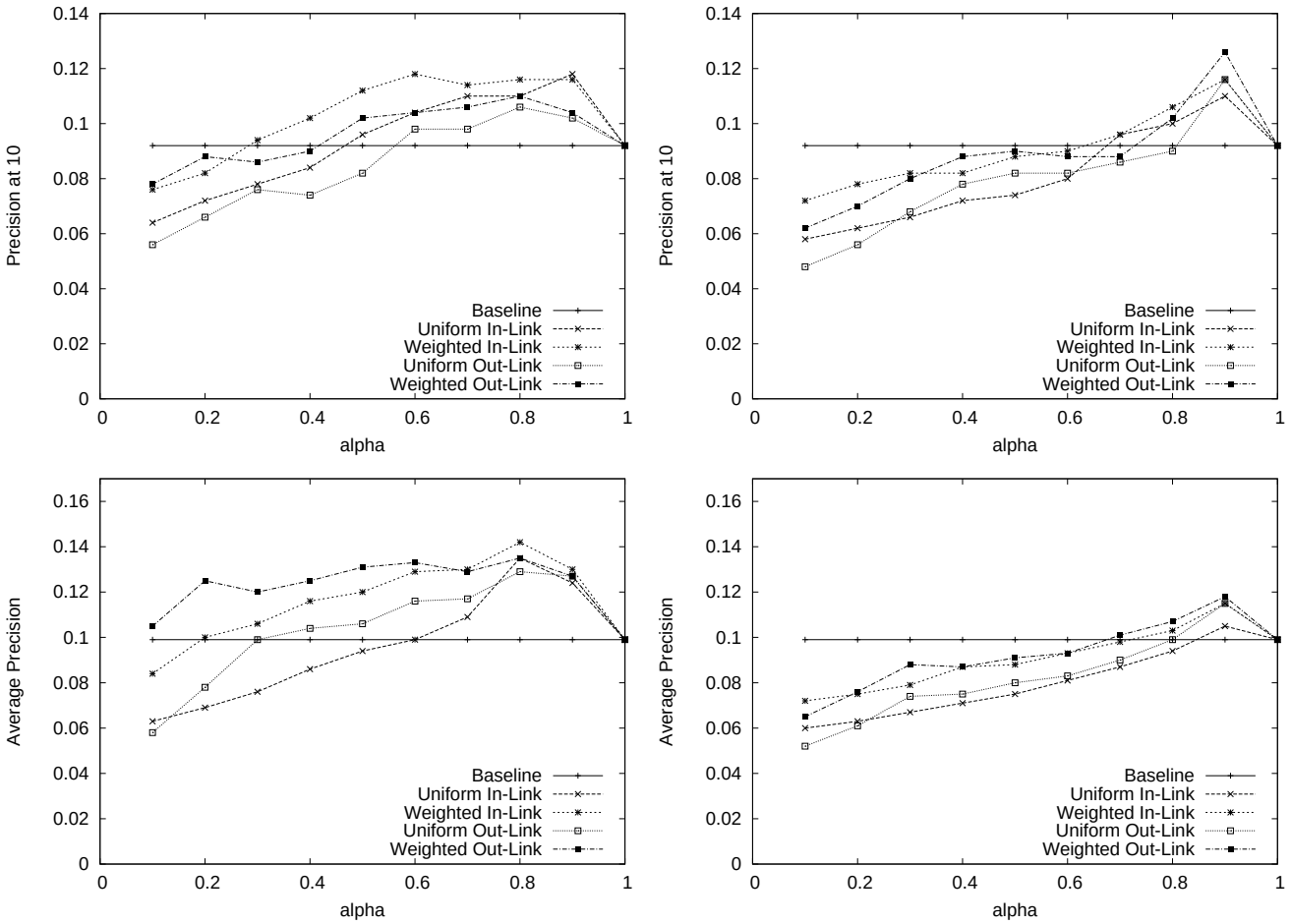
Table 7: Navigation Probability Estimation - TREC-2003

Okapi Baseline				
Neighbor Set	Uniform Estimate		Relevance Estimate (Impr.)	
	Prec@10	**Avg Prec**	Prec@10	**Avg Prec**
IN	0.118	**0.145**	0.128(8.5%)	**0.144(-)**
OUT	0.118	**0.151**	0.122(3.4%)	**0.163(8.1%)**
IN & OUT	0.126	**0.168**	0.138(9.5%)	**0.179(6.5%)**
LM Baseline				
Neighbor Set	Uniform Estimate		Relevance Estimate (Impr.)	
	Prec@10	**Avg Prec**	Prec@10	**Avg Prec**
IN	0.118	**0.135**	0.118(-)	**0.142(5.2%)**
OUT	0.106	**0.129**	0.11(3.8%)	**0.135(4.7%)**
IN & OUT	0.118	**0.150**	0.128(8.5%)	**0.144(-)**

baseline. It would be interesting to try other kinds of neighbors, e.g. co-citations and co-references to see if they can further improve the performance.

3. Other than the neighbor sets derived from the explicit link structure of the Web, we can also define other types of neighbors. In general, the framework allows us to define any set of documents with a specific characteristic as a neighbor set. As an example, we can define the set of pages with similar content as a neighbor set [21]. It is interesting to see if exploiting these types of neighbors can further improve the retrieval accuracy.

4. The probabilistic relevance propagation framework is a general hypertext retrieval framework that can be applicable to any hypertext retrieval environment. For example, we may apply the algorithms we studied here to literature search where the links represent citations.



(a) Results using relevance probabilities

(b) Results using raw content scores

Figure 3: TREC-2003 results with LM baseline

Table 8: Ranges of α Values for Improving Baselines

Method	Okapi Baseline			
	TREC-2003		TREC-2004	
	Prec @ 10	**Avg. Prec**	Prec @ 10	**Avg. Prec**
Uni-IN	[0.6, 0.9]	**[0.6, 0.9]**	[0.3, 0.9]	**[0.3, 0.9]**
Wt-IN	[0.3, 0.9]	**[0.3, 0.9]**	[0.3, 0.9]	**[0.2, 0.9]**
Uni-OUT	[0.4, 0.9]	**[0.4, 0.9]**	[0.6, 0.9]	**[0.4, 0.9]**
Wt-OUT	[0.2, 0.9]	**[0.2, 0.9]**	[0.3, 0.9]	**[0.2, 0.9]**
	Language Model Baseline			
Uni-IN	[0.5, 0.9]	**[0.6, 0.9]**	[0.6, 0.9]	**[0.6, 0.9]**
Wt-IN	[0.3, 0.9]	**[0.2, 0.9]**	[0.3, 0.9]	**[0.4, 0.9]**
Uni-OUT	[0.6, 0.9]	**[0.3, 0.9]**	[0.7, 0.9]	**[0.6, 0.9]**
Wt-OUT	[0.5, 0.9]	**[0.1, 0.9]**	[0.6, 0.9]	**[0.3, 0.9]**

5. ACKNOWLEDGMENTS

This work is in part supported by the National Science Foundation under award numbers 0425852, 0347933, and 0428472.

6. REFERENCES

[1] V. Anh and A. Moffat. Melbourne university 2004: Terabyte and web tracks. In *Proceedings of the TREC Conference*, 2004.

[2] K. Bharat and M. R. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In *SIGIR '98: Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 104–111, New York, NY, USA, 1998. ACM Press.

[3] K. Bharat and G. A. Mihaila. When experts agree: using non-affiliated experts to rank popular topics. *ACM Trans. Inf. Syst.*, 20(1):47–58, 2002.

[4] A. Borodin, G. O. Roberts, J. S. Rosenthal, and P. Tsaparas. Link analysis ranking: algorithms, theory, and experiments. *ACM Trans. Inter. Tech.*, 5(1):231–297, 2005.

[5] S. Chakrabarti, B. Dom, D. Gibson, J. Kleinberg, P. Raghavan, and S. Rajagopalan. Automatic resource list compilation by analyzing hyperlink structure and associated text. In *Proceedings of the 7th International World Wide Web Conference*, 1998.

[6] S. Chakrabarti, B. Dom, D. Gibson, S. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Spectral filtering for resource discovery. In *ACM SIGIR workshop on Hypertext Information Retrieval on the Web*, 1998.

[7] P. R. Cohen and R. Kjeldsen. Information retrieval by constrained spreading activation in semantic networks. *Inf. Process. Manage.*, 23(4):255–268, 1987.

[8] N. Craswell and D. Hawking. Overview of the trec-2002 web track. In *Proceedings of the TREC Conference*, 2002.

[9] N. Craswell and D. Hawking. Overview of the trec-2003 web track. In *Proceedings of the TREC Conference*, 2003.

[10] N. Craswell and D. Hawking. Overview of the trec-2004 web track. In *Proceedings of the TREC Conference*, 2004.

[11] F. Crestani and P. L. Lee. Searching the web by constrained spreading activation. *Inf. Process. Manage.*, 36(4):585–605, 2000.

[12] W. B. Croft, T. J. Lucia, and P. R. Cohen. Retrieving documents by plausible inference: a priliminary study. In *SIGIR '88: Proceedings of the 11th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 481–494, New York, NY, USA, 1988. ACM Press.

[13] W. B. Croft, T. J. Lucia, J. Cringean, and P. Willett. Retrieving documents by plausible inference: an experimental study. *Inf. Process. Manage.*, 25(6):599–614, 1989.

[14] B. D. Davison. Toward a unification of text and link analysis. In *SIGIR '03: Proceedings of the 26th annual international ACM SIGIR conference on Research and development in informaion retrieval*, pages 367–368, New York, NY, USA, 2003. ACM Press.

[15] H. P. Frei and D. Stieger. The use of semantic links in hypertext information retrieval. *Inf. Process. Manage.*, 31(1):1–13, 1995.

[16] E. Garfield. Citation indexes for science. *Science*, 129, 1955.

[17] G. Grimmett and D. Stirzaker. Probability and random processes. In *Oxford University Press*, 1989.

[18] T. H. Haveliwala. Topic-sensitive pagerank: A context-sensitive ranking algorithm for web search. *IEEE Transactions on Knowledge and Data Engineering*, 15(4):784–796, 2003.

[19] J. Kamps, G. Mishne, and M. de Rijke. Language models for searching in web corpora. In *Proceedings of the TREC Conference*, 2004.

[20] J. M. Kleinberg. Authoritative sources in a hyperlinked environment. *J. ACM*, 46(5):604–632, 1999.

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

[22] R. Larson. Bibliometrics of the world wide web: An exploratory analysis of the intellectual structure of cyberspace. In *Annual Meeting of the American Society of Information Science*, 1996.

[23] M. Marchiori. The quest for correct information on the Web: Hyper search engines. *Computer Networks and ISDN Systems*, 29(8–13):1225–1236, 1997.

[24] F. Mathieu and M. Bouklit. The effect of the back button is a random walk: Application for pagerank. In *Proceedings of the 13th International World Wide Web Conference*, 2004.

[25] D. S. Modha and W. S. Spangler. Clustering hypertext with applications to web searching. In *HYPERTEXT '00: Proceedings of the eleventh ACM on Hypertext and hypermedia*, pages 143–152, New York, NY, USA, 2000. ACM Press.

[26] A. Y. Ng, A. X. Zheng, and M. I. Jordan. Stable algorithms for link analysis. In *SIGIR '01: Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 258–266, New York, NY, USA, 2001. ACM Press.

[27] L. Page, S. Brin, R. Motwani, and T. Winograd. The pagerank citation ranking: Bringing order to the web. Technical report, Stanford Digital Library, 1998.

[28] P. Pirolli, J. Pitkow, and R. Rao. Silk from a sow's ear: Extracting usable structures from the web. In *Proc. ACM Conf. Human Factors in Computing Systems, CHI*. ACM Press, 1996.

[29] T. Qin, T.-Y. Liu, X.-D. Zhang, Z. Chen, and W.-Y. Ma. A study of relevance propagation for web search. In *SIGIR '05: Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 408–415, New York, NY, USA, 2005. ACM Press.

[30] M. Richardson and P. Domingos. The intelligent surfer: Probabilistic combination of link and content information in pagerank. In *Advances in Neural Information Processing Systems*, 2002.

[31] S. Robertson. Threshold setting and performance optimization in adaptive filtering. *Information Retrieval*, 5(2-3):239–256, 2002.

[32] G. Salton. Associative document retrieval techniques using bibliographic information. *J. ACM*, 10(4):440–457, 1963.

[33] G. Salton and C. Buckley. On the use of spreading activation methods in automatic information retrieval. Technical report, Ithaca, NY, USA, 1988.

[34] J. Savoy. Bayesian inference networks and spreading activation in hypertext systems. *Inf. Process. Manage.*, 28(3):389–406, 1992.

[35] J. Savoy. Ranking schemes in hybrid boolean systems: a new approach. *J. Am. Soc. Inf. Sci.*, 48(3):235–253, 1997.

[36] A. Shakery and C. Zhai. Relevance propagation for topic distillation uiuc trec 2003 web track experiments. In *Proceedings of the TREC Conference*, 2003.

[37] H. Small. Co-citation in the scientific literature: a new measure of the relationship between two documents. *The American Society of Information Science*, 24, 1973.

[38] R. Song, J. R. Wen, S. M. Shi, T. Y. Xin, G. M. abd Liu, T. Qin, X. Zheng, J. Y. Zhang, G. R. Xue, and W. Y. Ma. Microsoft research asia at web track and terabyte track of trec 2004. In *Proceedings of the TREC Conference*, 2004.

[39] M. Sydow. Random surfer with back step. In *Proceedings of the 13th International World Wide Web Conference*, 2004.

[40] T. Tomiyama, K. Karoji, T. Kondo, Y. Kakuta, and T. Takagi. Meiji university web, novelty and genomics track experiments. In *Proceedings of the TREC Conference*, 2004.

[41] H. Zaragoza, N. Craswell, M. Taylor, S. Saria, and S. Robertson. Microsoft cambridge at trec 13: Web and hard tracks. In *Proceedings of the TREC Conference*, 2004.

[42] C. Zhai and J. Lafferty. Model-based feedback in the language modeling approach to information retrieval. In *CIKM '01: Proceedings of the tenth international conference on Information and knowledge management*, pages 403–410, New York, NY, USA, 2001. ACM Press.

[43] Z. Zhou, Y. Guo, B. Wang, X. Cheng, H. Xu, and G. Zhang. Trec 2004 web track experiments at cas-ict. In *Proceedings of the TREC Conference*, 2004.