

# A Study of Relevance Propagation for Web Search

Tao QIN<sup>2\*</sup>, Tie-Yan LIU<sup>1</sup>, Xu-Dong ZHANG<sup>2</sup>, Zheng CHEN<sup>1</sup>, Wei-Ying MA<sup>1</sup>

<sup>1</sup>Microsoft Research Asia, No.49 Zhichun Road, Haidian District, Beijing 100080, P.R. China

<sup>2</sup>Dept. Electronic Engineering, Tsinghua University, Beijing, 100084, P.R. China

<sup>1</sup>{t-tyliu, zhengc, wyma}@microsoft.com

<sup>2</sup>qinshita099@mails.tsinghua.edu.cn, zhangxd@tsinghua.edu.cn

## ABSTRACT

Different from traditional information retrieval, both content and structure are critical to the success of Web information retrieval. In recent years, many relevance propagation techniques have been proposed to propagate content information between web pages through web structure to improve the performance of web search. In this paper, we first propose a generic relevance propagation framework, and then provide a comparison study on the effectiveness and efficiency of various representative propagation models that can be derived from this generic framework. We come to many conclusions that are useful for selecting a propagation model in real-world search applications, including 1) sitemap-based propagation models outperform hyperlink-based models in sense of both effectiveness and efficiency, and 2) sitemap-based term propagation is easier to be integrated into real-world search engines because of its parallel offline implementation and acceptable complexity. Some other more detailed study results are also reported in the paper.

## Categories and Subject Descriptors

H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval; H.5.4 [Information Interfaces and Presentation]: Hypertext/Hypermedia

## General Terms

Algorithms, Experimentation, Theory.

## Keywords

Relevance Propagation, Hyperlink Based Score Propagation, Hyperlink Based Term Propagation, Sitemap Based Score Propagation, Sitemap Based Term Propagation

## 1. INTRODUCTION

Different from traditional information retrieval (IR), the Web contains both content and link structures that have provided many new dimensions for exploring better IR techniques. In the early days, people analyze web content and structure independently. Typical approaches such as [10][11][14] use TF-IDF [3] of the query term in the page to compute a relevance score, and use hyperlinks to compute a query-independent importance score (e.g.

PageRank [19]). And then these two scores are combined to rank the retrieved documents. Such a methodology has brought the first major improvement in Web IR [6]. In recent years, some new methodologies that explore the inter-relationship between content and link structures have been introduced and produced some exciting results. Roughly speaking, these methods can be divided into two categories: one is to enhance link analysis with the assistance of content information [2][5][13][17]; the other is relevance propagation, which propagates content information with the assistance of Web structure [6][18][22][23].

For the first category, HITS [17] and topic-sensitive PageRank [13] are the representatives. HITS first constructs a query specific sub-graph, and then computes the authority and hub scores on this sub-graph to rank the documents. Topic-sensitive PageRank calculates a set of PageRank vectors, each element of which is biased to one of the predefined topics. Generally speaking, these methods conduct link analysis on a sub-graph which is sampled from the whole Web graph by considering the content of the web pages.

For the second category, many relevance propagation methods were proposed to refine the content of web pages by propagating content-based attributes through web structure. For example, [6][18] propagate anchor text from one page to another to expand the feature set of web pages. [22] propagates the relevance score of a page to another page through the hyperlink between them. [23] propagates query term frequency from child pages to parent pages in the sitemap tree. Experiments on TREC benchmark showed that both methods in [22][23] can improve the retrieval accuracy<sup>1</sup>.

Comparatively speaking, the first category has been a long-studied research topic, and many works have been done on detailed aspects of content-dependent link analysis [1][4][8][9][13][15][17]. However, the second category, although attracted much attention in recent years, has not yet been studied comprehensively. In this paper, we explore more on the second category and study how effective and efficient the relevance propagation methods are and whether they are practical for real-world Web search applications.

We will give a comprehensive study of relevance propagation technologies for Web information retrieval. Firstly, we extend the existing works [22][23] to come out a generic framework, and then show that the various existing propagation models can actually be derived from it. Secondly, we conduct both theoretical

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

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

Copyright 2005 ACM 1-59593-034-5/05/0008...\$5.00.

<sup>1</sup>This work was performed at Microsoft Research Asia

and experimental evaluations over these models to answer the following questions:

- 1) Do these relevance propagation methods improve web search accuracy?
- 2) Which method is more effective and efficient?
- 3) Which method is more feasible to be applied in real-world search engines?

Based on the experiments on two different benchmarks (TREC .Gov2002 collection and MSN.com) and some theoretic analysis, we found that relevance propagation through parent-child relationship in the sitemap tree is more effective than through hyperlinks, and term propagation is more feasible for real-world implementation than relevance score propagation. These conclusions are very meaningful and informative for model selection in real-world search engines.

The organization of this paper is as follow. In Section 2, we review some existing relevance propagation methods. In Section 3, we discuss the generic relevance propagation framework and propagation methods. Then we study the effectiveness and efficiency of relevance propagation models in Section 4. Conclusions are given in Section 5.

## 2. RELATED WORKS

As mentioned in the introduction, many relevance propagation methods have been proposed to enhance relevance weighting [6][18][22][23]. In this section, we will briefly review two latest and representative works.

Shakery, et al [22] found that two factors are important to relevance weighting: the relevance score of a web page and the relevance scores of the pages that have links to/from that page. Motivated by this observation, they propagate the relevance score of a page to its neighbors in the link graph. They define a so-called hyper relevance score for each page as a function of three variables: its self-relevance score, a weighted sum of the hyper relevance scores of all the pages that point to it (denoted by in-link pages), and a weighted sum of the hyper relevance scores of all the pages that it points to (denoted by out-link pages). Based on these definitions, they proposed a relevance score propagation model as below:

$$h^{k+1}(p) = \alpha s(p) + \beta \sum_{p_i \rightarrow p} h^k(p_i) \omega_l(p_i, p) + \gamma \sum_{p \rightarrow p_j} h^k(p_j) \omega_o(p, p_j) \quad (1)$$

where  $\alpha + \beta + \gamma = 1$

where  $h^i(p)$  is the hyper relevance score of page  $p$  after the  $i$ -th iteration,  $s(p)$  is the original relevance score of page  $p$ ,  $\omega_l$  and  $\omega_o$  are weighting functions for both in-link and out-link pages respectively. Note that  $h^0(p) = s(p)$ .

For practical implementation, they derive three simplified cases, which are shown in Table 1.

As can be seen, the hyper relevance scores are computed iteratively. When the iteration process converges, the final hyper relevance scores will be used for relevance ranking. Experiments [22] show that the above score propagation models generally perform better than without propagation. 1~2% improvements are observed on the Web Track of TREC 2002. However, the amount

of improvement is sensitive to the document collection and the tuning of parameters [22].

**Table 1. Three cases of the relevance score propagation model**

Special case	Model formulation
weighted in-link	$h^{k+1}(p) = \alpha s(p) + (1 - \alpha) \sum_{p_i \rightarrow p} h^k(p_i) \omega_l(p_i, p) \quad (2)$ <p>where <math>\omega_l(p_i, p) \propto s(p)</math></p>
weighted out-link	$h^{k+1}(p) = \alpha s(p) + (1 - \alpha) \sum_{p \rightarrow p_j} h^k(p_j) \omega_o(p, p_j) \quad (3)$ <p>where <math>\omega_o(p, p_j) \propto s(p_j)</math></p>
uniform out-link	$h^{k+1}(p) = \alpha s(p) + (1 - \alpha) \sum_{p \rightarrow p_j} h^k(p_j) \quad (4)$

Different from [22], [23] proposes a sitemap-based feature propagation method. They first construct a sitemap for each website based on URL analysis, and then propagate query term frequency along the parent-child relationship in the sitemap tree as follow:

$$f_t'(p) = (1 + \alpha) f_t(p) + \frac{(1 - \alpha)}{|Child(p)|} \sum_{q \in Child(p)} f_t(q) \quad (5)$$

where  $f_t(p)$  and  $f_t'(p)$  are the occurrence frequencies of term  $t$  in page  $p$  before and after propagation,  $q$  is the child page of  $p$ ,  $\alpha$  is a weight which controls the contribution of the child pages to their parent. As can be seen, [23] actually propagates term frequency based on sitemap, and so we rename it as sitemap-based term propagation.

After the propagation of term frequency, any relevance weighting algorithm can be adopted on the refined features to rank web pages. Their experiments with a variation of BM2500 model got one of the best results on the Web Track of TREC 2004 [11], which shows that the sitemap-based term propagation method could significantly boost the retrieval performance.

Although the aforementioned two works were developed independently and seem not correlated to each other, however, as shown in the next section, they can actually be considered as two special cases of one generic propagation framework. This encourages us to further study whether we can derive some other models from this generic framework, and which of them is more effective and efficient for real-world applications. We will provide some detailed comparisons and evaluation of different propagation models in the following sections.

## 3. A GENERIC PROPAGATION FRAMEWORK

In this section, we will generalize the two propagation models introduced in the previous section to a generic framework. For this purpose, we will first extend the sitemap-based term propagation model [23] to an iterative version.

### 3.1 An Iterative Version of the Sitemap-Based Term Propagation Model

At the first glance, the two relevance propagation methods defined in (1) and (5) seem very different: the final hyper relevance score in (1) is calculated iteratively, while only one-step computation is

needed in (5). However, with a little modification, (5) could also be extended to an iterative version:

$$f_i^{k+1}(p) = (1 + \alpha)f_i^0(p) + \frac{(1 - \alpha)}{|Child(p)|} \sum_{q \in Child(p)} f_i^k(q) \quad (6)$$

where  $f_i^i(p)$  is the occurrence frequency of term  $t$  in page  $p$  after the  $i$ -th iteration, and  $f_i^0(p)$  is the original occurrence frequency of term  $t$  in page  $p$ . Actually, (5) is just the simplified case of (6) with only one step of iteration.

So far, (6) has been very similar to (2)(3) and (4): the right hand is composed of the original relevance of page  $p$ , and the relevance iteratively propagated from other pages. The only difference is the coefficient of linear combination. To bridge the gap, (6) can be further modified as follows:

$$f_i^{k+1}(p) = \alpha f_i^0(p) + (1 - \alpha) \sum_{q \in Child(p)} f_i^k(q) \frac{1}{|Child(p)|} \quad (7)$$

In fact, the above modification is necessary because the coefficients in (6) will make the iterative computation un-converged (i.e., after each iteration, the information is doubled because  $(1 + \alpha) + (1 - \alpha) = 2$ ). On the other hand, the modification does not change the basic idea of (5). In the following discussions, we will directly call (7) by the iterative version of sitemap-based term propagation model.

### 3.2 The Generic Relevance Propagation Framework

As shown in the previous section, the formulations of the relevance score propagation model (2)(3)(4) and the sitemap-based term propagation model (7) are quite similar. This encourages us to propose a generic relevance propagation framework on top of them:

$$c^{k+1}(p) = g(c^0(p), c^k(N_p)) \quad (8)$$

where  $c^0(p)$  represents the original relevance of page  $p$ ,  $c^i(p)$  represents the refined relevance of page  $p$  after the  $i$ -th iteration, and  $N_p$  represents the pages in the neighborhood of page  $p$ . There could be many choices of function  $g$ . Similar to the propagation methods in [22] [23], we adopt linear combination for simplicity.

$$c^{k+1}(p) = \alpha c^0(p) + (1 - \alpha) \sum_{q \in N_p} c^k(q) \omega(p, q) \quad (9)$$

It is easy to find that in the score propagation model (1),  $h^i(p)$  corresponds to  $c^i(p)$ ,  $s(p)$  corresponds to  $c^0(p)$ , and  $N_p$  contains the pages which point to or are pointed to by  $p$ . That is, the relevance is represented by score, and the structure is hyperlink graph. In this regard, we rename the relevance score propagation models by hyperlink-based score propagation model (or HS model). Similarly, in the iterative sitemap-based term propagation model,  $f_i^i(p)$  and  $f_i^0(p)$  correspond to  $c^i(p)$  and  $c^0(p)$  respectively, and  $N_p$  represents the child pages of  $p$  in the sitemap tree. That is, the relevance is represented by term frequency and the structure is the sitemap tree. This is consistent with its name, sitemap-based term propagation model (or ST model in brief). Table 2 shows the relationship between the two existing models and the generic framework.

**Table 2. The Generic Propagation Framework**

	Score-level	Term-level
<b>Hyperlink</b>	Hyperlink based score propagation [22]	?
<b>Sitemap</b>	?	Sitemap based term propagation [23]

### 3.3 Two New Propagation Models

From Table 2, one can easily think of two new models for relevance propagation (corresponding to the two question marks): hyperlink-based term propagation model (or HT model) and sitemap-based score propagation model (or SS model).

Similar to the ST model, the HT model needs to propagate the frequency of query term in a web page before adopting relevance weighting algorithms to rank the documents. While similar to the HS model, the HT model also has three special cases:

**Table 3. Special cases of the HT model**

Special case	Model formulation
weighted in-link	$f_i^{k+1}(p) = \alpha f_i^0(p) + (1 - \alpha) \sum_{p_i \rightarrow p} f_i^k(p_i) \omega_{in}(p_i, p)$ (10) where $\omega_{in}(p_i, p) \propto f_i^0(p)$
weighted out-link	$f_i^{k+1}(p) = \alpha f_i^0(p) + (1 - \alpha) \sum_{p \rightarrow p_j} f_i^k(p_j) \omega_{out}(p, p_j)$ (11) where $\omega_{out}(p, p_j) \propto f_i^0(p_j)$
uniform out-link	$f_i^{k+1}(p) = \alpha f_i^0(p) + (1 - \alpha) \sum_{p \rightarrow p_j} f_i^k(p_j)$ (12)

Symmetrically, similar to the HS model, the SS model first computes a relevance score for each page, and then propagates it to calculate hyper relevance score. And similar to the ST model (7), sitemap tree is used for the propagation:

$$h^{k+1}(p) = \alpha s(p) + (1 - \alpha) \sum_{q \in Child(p)} h^k(q) \frac{1}{|Child(p)|} \quad (13)$$

In summary, we proposed a generic relevance propagation framework and derived four propagation models as its special cases. In the following, we will evaluate each of them in terms of effectiveness, efficiency and the potential application in real world search engines.

## 4. EMPIRICAL EVALUATIONS

In this section, experiments were conducted to evaluate the performance and efficiency of relevance propagation models. We first introduce the experimental settings and some implementation issues, and then present experimental results and discussions.

### 4.1 Experimental Settings

To avoid the corpus bias, two different data collections were used in our experiments. One is the ‘‘GOV’’ corpus, which is crawled from the .gov domain in early 2002. This corpus has been used as the data collection of Web Track since TREC 2002. There are totally 1,053,110 pages with 11,164,829 hyperlinks in it. The other data set is the ‘‘MSN’’ corpus, which is crawled from msn.com. This corpus has 2,218,428 pages with 29,163,922 hyperlinks.

When conducting the experiments on the “.GOV” corpus, we used the topic distillation task in the Web Track of TREC 2003 and 2004 as our query sets (with 50 and 75 queries respectively). For simplicity, we denote these two query sets by TD2003 and TD2004. The ground truths of these tasks are provided by the TREC committee. When doing experiments on the “MSN” corpus, we use a self-defined query set with 100 queries in total. The ground truths are labeled by human beings<sup>2</sup>.

As the baseline, we adopted BM2500 [20] as the relevance weighting function in our experiments:

$$relevance = \sum_{T \in Q} \omega \frac{(k_1 + 1)tf(k_3 + 1)qtf}{(K + tf)(k_3 + qtf)} \quad (14)$$

where  $Q$  is a query consisting of terms  $T$ ;  $tf$  is the occurrence frequency of the term  $T$  within the web page,  $qtf$  is the frequency of the term  $T$  within the topic from which  $Q$  was derived, and  $\omega$  is the Robertson/Sparck Jones weight [21] of  $T$  in  $Q$ .  $K$  is calculated by

$$K = k_1 \left( (1-b) + b \times dl / avdl \right) \quad (15)$$

where  $dl$  and  $avdl$  denote the page length and the average page length. In our experiments, we set  $k_1 = 2.5$ ,  $k_3 = 1000$ ,  $b = 0.8$ .

According to the Web Track of TREC, we returned 1000 search results by each model for evaluation. And we used mean average precision (MAP) and precision at 10 (P@10) as evaluation criterions [3][10].

## 4.2 Implementation Issues

Following [22], the hyperlink-based propagations are not applied on all the web pages. Instead, for each query, we construct a working set. The flowchart of the working set construction is shown in Figure 1(a). For a query, we first retrieve the relevant set which contains all the pages that have at least one query term. Then we rank all the pages in the relevant set with BM2500, and choose 1000 pages with highest relevance scores as the core set. Then, we find the set of pages that point to the core set (denoted by the citing set), and the set of pages that are pointed to by the core set (denoted by the cited set). After that, we construct the working set as (16), which is illustrated by the gray part in Figure 1(b).

$$\text{Working Set} = (\text{Core Set} \cup \text{Citing Set} \cup \text{Cited Set}) \cap \text{Relevant Set} \quad (16)$$

Our implementation of the SS and ST models is similar to [23]. That is, we first generate the sitemap tree for each website in the corpus based on URL analysis<sup>3</sup>. Then for each query, we rank all the relevant documents with BM2500 and select top 10000 pages as the working set to conduct the propagation.

Besides, another important implementation issue about the sitemap-based propagation models is that they can actually be implemented in a non-iterative manner, although they are defined with an iterative form. The reason is that there is no loop in the sitemap tree [12] so that there is no propagation circle in the SS

and ST models. If we carry out the propagation from the leaf node to the root of the sitemap tree in a bottom-up manner, we can avoid iterative computation but come up with the same propagation result.

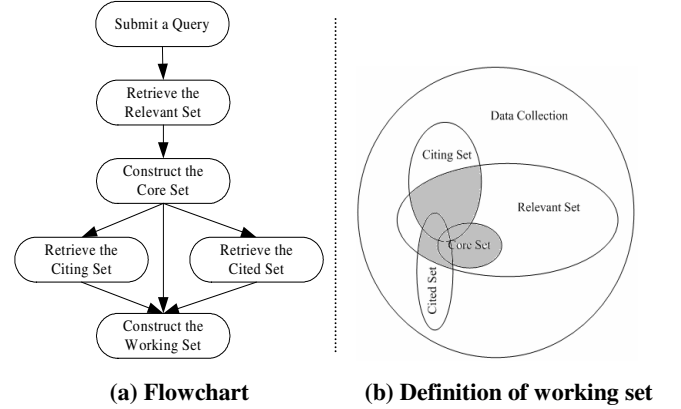


Figure 1. Illustration of working set construction

Let’s see an example in Figure 2. This is a website with only 4 pages. Table 4 shows the propagation results according to the iterative version of the SS model (where  $\beta = 1 - \alpha$ , and  $s_i$  is the relevance score of page  $p_i$  before propagation). From this table we can see that it takes 2 iterations to converge because the scores after the 2nd and the 3rd iterations are the same.

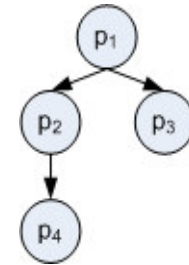


Figure 2. A website with 4 pages

Table 4. Score propagation result of Figure 2

Iteration	$p_1$	$p_2$	$p_3$	$p_4$
0	$s_1$	$s_2$	$s_3$	$s_4$
1	$\alpha s_1 + \beta(s_2 + s_3)$	$\alpha s_2 + \beta s_4$	$\alpha s_3$	$\alpha s_4$
2	$\alpha s_1 + \alpha \beta(s_2 + s_3) + \alpha \beta \beta s_4$	$\alpha s_2 + \alpha \beta s_4$	$\alpha s_3$	$\alpha s_4$
3	$\alpha s_1 + \alpha \beta(s_2 + s_3) + \alpha \beta \beta s_4$	$\alpha s_2 + \alpha \beta s_4$	$\alpha s_3$	$\alpha s_4$

Now let us consider a non-iterative implementation of the SS model. This time, we first update the score of page  $p_4$  and get  $h^1(p_4) = \alpha s_4$ ; after that, we update  $p_2$  and  $p_3$  by  $h^1(p_2) = \alpha s_2 + \alpha \beta s_4$  and  $h^1(p_3) = \alpha s_3$ ; for the last step we update  $p_1$ , and get  $h^1(p_1) = \alpha s_1 + \alpha \beta(s_2 + s_3) + \alpha \beta \beta s_4$ . Although we did not update all the pages in each step, the resulting scores are the same as those produced by the iterative implementation. With such a bottom-up strategy, we need to propagate the relevance score (or term frequency) of a page to its parent *only once*, which can greatly improve the efficiency of sitemap-based propagation models. Note that although we avoid iteration, the final scores we get are the converged results of (13) with many steps of iteration.

<sup>2</sup> We returned 30 results for each query and ask 6 users to label the relevance and irrelevance of these results.

<sup>3</sup> For the details of sitemap generation, please refer to [12].

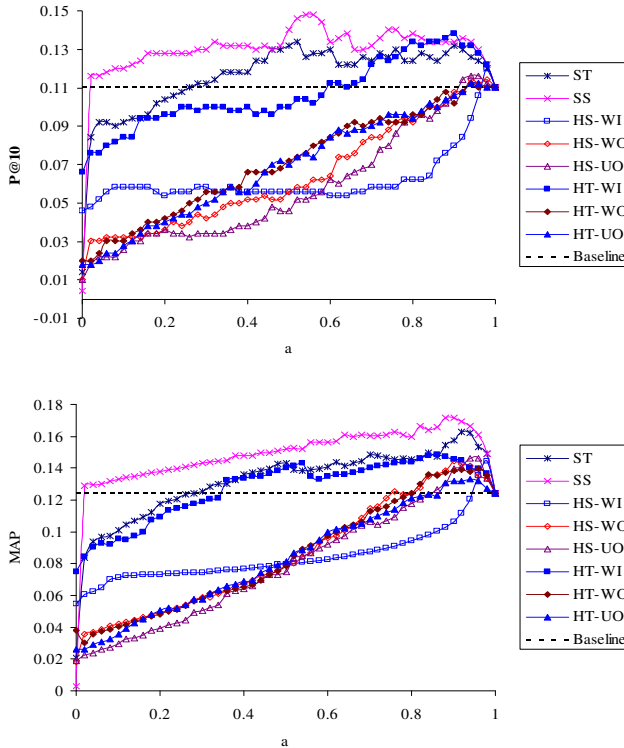
### 4.3 Effectiveness Evaluation

With the preparations in subsection 4.1 and 4.2, we tested all the four propagation models and collected the retrieval performance for evaluation. Before presenting the results, we first give the abbreviations of these models in Table 5 for ease of reference.

**Table 5. Algorithms and their abbreviations**

Models		Abbreviation
Hyperlink-based score propagation model	weighted in-link	HS-WI
	weighted out-link	HS-WO
	uniform out-link	HS-UO
Hyperlink-based term propagation model	weighted in-link	HT-WI
	weighted out-link	HT-WO
	uniform out-link	HT-UO
Sitemap-based score propagation model		SS
Sitemap-based term propagation model		ST

Figure 3 shows the performance of the four models (total 8 algorithms) on the “.GOV” corpus with the TD2003 query set. As can be seen, all models (with proper parameters) boosted the retrieval performance against the baseline (the baseline corresponds to  $\alpha=1$ , which MAP is 0.124 and P@10 is 0.110). Among these models, SS got the best result; ST and HT-WI had similar performance and in the second position; HS-WI produced the worst performance and the other four algorithms performed similarly with only marginal improvement over the baseline.



**Figure 3. Performance on the “.GOV” corpus with TD2003**

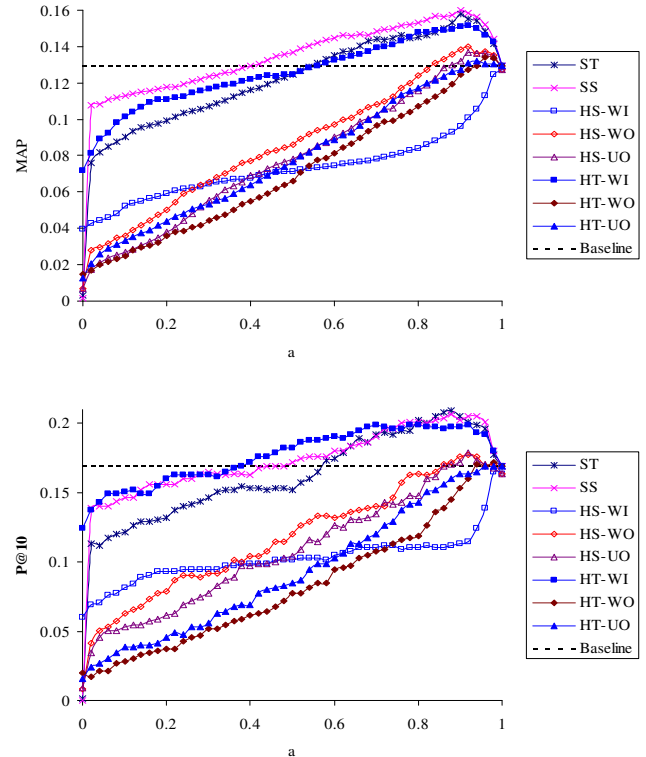
To gain more understanding of the performance comparison, we further list in Table 6 the best MAP and P@10 of each algorithm. From this table, we can find that the sitemap-based models outperformed the hyperlink-based models by much. For example,

the best MAP of the SS model was 0.1719, which was not only 40% better than the baseline, but also won the best result reported in TREC 2003 by over 10%. Similarly, the best P@10 of the SS model was 0.148, which outperformed the baseline and the best result of TREC 2003 by 30% and 15% respectively. Comparatively speaking, almost none of the hyperlink-based models outperformed the best results reported in TREC 2003.

**Table 6. Best performance of each algorithm**

Algorithms	Best MAP	Best P@10
Baseline	0.1240	0.110
Best Result in TREC 2003	0.1543	0.128
HS-WI	0.1443	0.120
HS-WO	0.1442	0.114
HS-UO	0.1461	0.116
HT-WI	0.1488	0.138
HT-WO	0.1394	0.112
HT-UO	0.1332	0.112
SS	0.1719	0.148
ST	0.1628	0.134

Figure 4 shows the performance of the four models on the “.GOV” corpus with the TD2004 query set. We can draw very similar conclusions from this figure to those from Figure 3. A little difference is that HS-WI performed better with TD2004 than with TD2003. This time it was almost as good as the two sitemap-based models.



**Figure 4. Performance on the “.GOV” corpus with TD2004**

Figure 5 shows the performance on the “.MSN” corpus. Again, we can come to similar conclusions to the above two experiments. The difference is that the ST model was much better than the SS model on this corpus, unlike what it performed on the “.GOV”

corpus. Besides, HT-WI did not perform as well as on the “.GOV” corpus. Actually, these differences are understandable because the “.MSN” corpus is quite different from the “.GOV” corpus: 1) This corpus was crawled from a commercial website but not from the governmental websites; 2) The query set for the “.MSN” corpus is very different from those for the “.GOV” corpus. The TD2003 and TD2004 query sets are for topic distillation; however, the queries for the “.MSN” corpus are for general purpose<sup>4</sup>. Although with these differences, the experimental results still tell us that relevance propagation can boost the performance, no matter over what kind of data sets and with what kind of query sets.

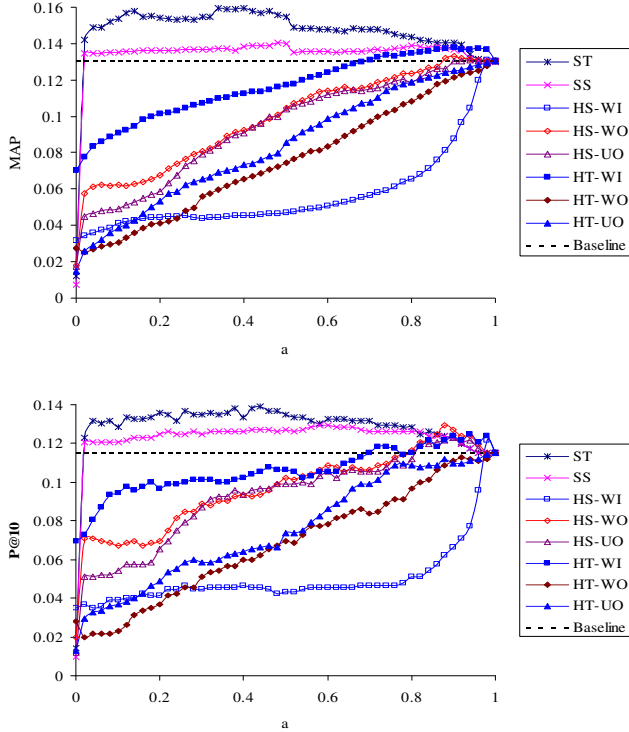


Figure 5. Performance on the “.MSN” corpus

Besides the peak of the performance curve, the robustness of an algorithm is also an important factor for its effectiveness. We list in Table 7 the range of  $\alpha$  in which each algorithm got better result than the baseline. From this table we can see that the SS model was the most robust, which won the baseline with most values of  $\alpha$ . The ST and HT-WI models were the second winners, which boosted the retrieval performance with about half of the possible  $\alpha$  values. Comparatively, the remaining algorithms were not so robust, since they resulted in improvements only when  $\alpha$  was closed to 1.

To summarize the above experiments, we can draw the following conclusions:

- 1) In general, relevance propagation can boost the search performance with proper parameter settings;

<sup>4</sup> When asking the users to label, we have not given them any specific guidelines. So the labeling results just reflect their general intensions.

- 2) The sitemap-based models are more effective and robust than the hyperlink-based models;
- 3) The two sitemap-based models have similar performance. Among the hyperlink-based models, the HT-WI model performs best while HS-WI performs worst.

Table 7. Range of  $\alpha$  with improvement over the baseline

Algorithms	GOV with TD2003		GOV with TD2004		MSN	
	MAP	P@10	MAP	P@10	MAP	P@10
HS-WI	(0.94, 1)	(0.96, 1)	(0.96, 1)	(0.96, 1)	(0.96, 1)	(0.96, 1)
HS-WO	(0.76, 1)	(0.94, 1)	(0.84, 1)	(0.86, 1)	(0.88, 1)	(0.80, 1)
HS-UO	(0.84, 1)	(0.92, 1)	(0.86, 1)	(0.86, 1)	(0.90, 1)	(0.82, 1)
HT-WI	(0.36, 1)	(0.60, 1)	(0.56, 1)	(0.38, 1)	(0.68, 1)	(0.70, 1)
HT-WO	(0.80, 1)	(0.94, 1)	(0.94, 1)	(0.94, 1)	(0.98, 1)	(0.98, 1)
HT-UO	(0.88, 1)	(0.94, 1)	(0.92, 1)	(0.92, 1)	(0.98, 1)	(0.98, 1)
SS	(0.02, 1)	(0.02, 1)	(0.40, 1)	(0.44, 1)	(0.02, 1)	(0.02, 1)
ST	(0.28, 1)	(0.28, 1)	(0.54, 1)	(0.56, 1)	(0.02, 1)	(0.02, 1)

## 4.4 Efficiency Evaluation

In the previous section, we investigate the effectiveness of the relevance propagation models. However, for real-world applications, efficiency is another important factor besides effectiveness. In this regard, we evaluate the efficiency of the four models in this section to see their potential of being used in search engines.

Roughly speaking, typical architecture of a search engine has three components [3][6]: crawler, indexer, and searcher. If we want to integrate relevance propagation technologies into search engine, we should consider these three components. Clearly, we could only embed relevance propagation into the second or third component. Since the search engine indexes the Web offline, and implement the search operation online, we will discuss the efficiency of relevance propagation for the online case and offline case respectively.

### 4.4.1 Online Complexity

Due to the algorithm descriptions, all the relevance propagation models have two kinds of computations. The first one is to retrieve the relevant pages and rank them by relevance weighting functions. Actually this is also needed by existing search engines. The second is the additionally-introduced complexity, including working set construction, relevance propagation and so on. This will be the major concern when integrating these models into the search engines. In this regard, we will focus on the analysis of these additional computations in this section.

According to the model formulation and the implementation issues, we can get the following estimations on the online complexity of the relevance propagation models. Note that the time complexity we estimate here is for one query.

1. Since the SS model needs to propagate the relevance score of a page to its parent *only once* due to the non-iterative implementation, if we denote the time complexity of propagating an entity (the relevance score or a term frequency) from a page to its parent is  $c_s$ , the time complexity of the SS model can be represented by  $w c_s$ , where  $w$  is the size of the working set.
2. For the ST model, we need to propagate the frequency of each query term from a page to its parent. Suppose each

page in the working set has  $q$  terms to propagate on average<sup>5</sup>, then the complexity of ST model is  $qwc_s$ .

3. For each step of iteration in the HS models, we need to propagate the relevance score of a page along its in-link or out-link in the sub graph of the working set. Note that the source and destination pages of the hyperlink should be both in the working set, and so the average numbers of in-links and out-links per page are equal to each other. We denote this number by  $l$ . If we further use  $c_h$  to indicate the time complexity of propagating an entity from a page to another page along hyperlinks, we can get that the complexity of each step of iteration in the HS models is  $wlc_h$ . If it takes  $t$  iterations for the propagation to converge, the overall complexity will be  $twlc_h$ . And so the complexity for the three cases of HS model is all  $twlc_h$ .
4. Similar to the analysis of the HS models, we can get the complexity of the three cases of HT model is all  $twlc_h$ .

To verify the above theoretical analysis, we logged the time usage of the eight algorithms on a PC with 1.5GHz CPU and 2GB memory. To save space, here we only list the statistical data on the “GOV” corpus with TD2004 in Table 8. The second column summarizes the theoretical time complexity. The third column shows the average size of the working set. As can be seen, the size of the working set for the HS and HT models is about 7 times the size of the core set (which is predefined as 1000 in our experimental settings). The average number of hyperlinks per page is 11 as shown in the fourth column, which is consistent with previous reports [3][16]. The fifth column shows the average iteration numbers. Note that the SS and ST models do not need any iteration, so we set  $t = 1$ . The sixth column shows the average number of query term per page. An interesting finding is that the pages in the working set of the ST model contain more terms than that of the HT model on average. This is reasonable because the working set of the ST model consists of the top 10000 most relevant pages, but the working set of the HT model is made up of the top 1000 most relevant pages and some other pages in the citing and cited sets (which might not be very relevant). So generally speaking, those pages in the working set of the ST model will be more relevant than those of the HT model. Therefore the average number of query terms per page of the ST model will be larger. The last column shows the average CPU time per query.

**Table 8. Time complexity on GOV with TD2004**

Algorithm	Time Complexity	average $w$	average $l$	average $t$	average $q$	average CPU time (ms)
HS-WI	$twlc_h$	6796.5	11.0	7.4	-	47.9
HS-WO	$twlc_h$	6796.5	11.0	6.5	-	36.5
HS-UO	$twlc_h$	6796.5	11.0	6.6	-	39.8
HT-WI	$twlc_h$	6796.5	11.0	9.1	1.5	54.0
HT-WO	$twlc_h$	6796.5	11.0	11.1	1.5	63.3
HT-UO	$twlc_h$	6796.5	11.0	8.9	1.5	51.6
SS	$wc_s$	10000.0	-	1	-	1.9
ST	$qwc_s$	10000.0	-	1	3	8.3

By analyzing Table 8, we can come to the following conclusions

- 1) The sitemap-based models are more efficient than the hyperlink-based models;
- 2) The score-level propagation models are faster than those term-level models;
- 3) The numbers in Table 8 is quite in accordance with our theoretical analysis.

#### 4.4.2 Offline Complexity

Since a real search engines should handle hundreds of queries per second [3][6], it will be very difficult to implement these propagation techniques online according to Table 8. So offline implementation is much more preferred if we want to apply them in real-world applications.

Search engines usually build offline invert and forward indices to store the information of each term (including frequency, position and so on) in web pages [3][6]. Then it is easily understood that term-level propagation models can well match this mechanism and we only need to refine the offline index files. To illustrate it, let us take the ST model for example. Suppose the child pages of page  $p$  contain a particular word, and we need to propagate the occurrence frequency of this word to page  $p$ . If  $p$  already contains this particular word, we only need to modify its frequency; while if  $p$  does not contain the word, we need to add its ID to the forward index [6] of page  $p$ , and then update its term frequency. Comparatively, the score-level propagation models could hardly be integrated into search engines, because scores do not exist in the offline indices but are dependent on the online relevance ranking algorithm used in the search engine.

Although term-level propagation has the potential to be integrated into search engines, whether they can finally survive will depend on their efficiencies. As we know, the update cycle of the indices in state-of-the-art search engines is very short, if these propagation models can not fit the need of such frequent update, they still can not be used. To gain more understanding of this issue, we further conduct an efficiency analysis. Note that this time the working set will be the whole web page collections.

For the ST model, since we need to propagate all the unique words in each page to its parent, the time complexity will be  $\bar{q}nc_s$ , where  $\bar{q}$  is the average number of unique words per page, and  $n$  is the total number of pages in the data corpus. According to [3][6], the average size of a web page is about 5~6K bytes. Therefore it is reasonable to assume that  $\bar{q} \approx 100$ . Furthermore, as we know, typical search engines currently indexes 5-8 billion pages. So we let  $n=8 \times 10^9$ . Based on Table 8, we know that it will cost about  $8.3/3 \approx 2.8$  ms to propagate one word on a graph with 10000 pages. Then accordingly, we can estimate that it will take about  $2.8 \times 8 \times 10^9 \times 100 / 10000$  ms (or 62.2 hours, or 2.6 days) to re-index the 8-billion pages. Obviously this is an acceptable time complexity.

For the HT model, with similar analysis to the ST model, we know that its time complexity is  $\bar{q}tnlc_h$ . According to [3][16], the average number of hyperlinks per page is about 10 in real Web. This number is very similar to that in Table 8. So for simplicity, we assume that the hyperlink number and iteration number in real Web are all equal to those in Table 8 (although we can expect that with larger working set, the convergence will be slower), we can estimate the lower-bound time usage for the HT model to re-index the 8-billion pages:  $(50/1.5) \times 8 \times 10^9 \times 100 / 6800 = 3.9 \times 10^9$  ms (or

<sup>5</sup> To be noted, since some pages in the working set may not contain all the terms in the query,  $q$  is not equal to the number of terms in the query.

1083 hours, or 45 days). Clearly, so long a time to build index is unacceptable for real search engine application and we must use parallel computation. However, as the propagation of HT happens on the whole Web, it is non-trivial to decompose the computations into local tasks. Comparatively, since the ST model only propagates within websites, the computations can be more easily divided and conquered.

Based on the above discussions and derivations, we can come to the following conclusions:

- 1) Score-level propagation is very difficult to implement offline;
- 2) The time complexity of the ST model for offline implementation is acceptable; however the time complexity of the HT model is out of tolerance;
- 3) The ST model is easy for parallel implementation while the parallel implementation of the HT model is non-trivial.

## 5. CONCLUSIONS

In this paper, we conducted a comprehensive study on relevance propagation in Web information retrieval. In particular, a generic relevance propagation framework was proposed and shown that it can be used to derive many existing propagation models. Then we investigated the effectiveness and efficiency of those propagation models with both theoretical analysis and experimental verifications. The following conclusions are drawn from our study:

- 1) Generally speaking, relevance propagation can boost the performance of web information retrieval;
- 2) Sitemap-based propagation models outperform the hyperlink-based propagation models in sense of both effectiveness and efficiency;
- 3) Score-level propagation and term-level propagation have similar effectiveness, but the former is more efficient than the latter in on-line implementations. However, score-level propagation is not practical for real-world search engines because it can not be implemented offline;
- 4) Term-level propagation models can be implemented offline. Among them, sitemap-based model is the best because of its computation can be performed in parallel.

Overall, sitemap-based term propagation model is a good choice for search applications in terms of effectiveness, efficiency and offline parallel implementation.

## 6. REFERENCES

- [1] Amento, B., Terveen, L., and Hill, W. Does "Authority" Mean Quality? Predicting Expert Quality Ratings of Web Pages. In Proc. ACM SIGIR 2000, pages 296--303.
- [2] Amitay, E., Carmel, D., Darlow, A., Lempel, R., and Soffer, A. Topic Distillation with Knowledge Agents, in the 11th TREC, 2002.
- [3] Baeza-Yates, R., Ribeiro-Neto, B. Modern Information Retrieval, Addison Wesley, 1999.
- [4] Bharat, K., and Henzinger, M. R. Improved Algorithms for Topic Distillation in a Hyperlinked Environment. In Proceedings of the ACM-SIGIR, 1998.
- [5] Bharat, K., and Mihaila, G. A. When Experts Agree: Using Non-affiliated Experts to Rank Popular Topics. In 10th WWW, 2001.
- [6] Brin, S., and Page, L. The Anatomy of a Large Scale Hypertextual Web Search Engine, Proc. 7th WWW, 1998.
- [7] Broder, A. A Taxonomy of Web Search. SIGIR Forum 36(2), 2002.
- [8] Chakrabarti, S. Integrating the Page Object Model with hyperlinks for enhanced topic distillation and information extraction, In the 10th WWW, 2001.
- [9] Chakrabarti, S., Joshi, M., and Tawde, V. Enhanced Topic Distillation Using Text, Markup Tags, and Hyperlinks, In Proceedings of the 24th ACM SIGIR, 2001, pp. 208-216.
- [10] Craswell, N., Hawking, D. Overview of the TREC 2003 Web Track, in the 12th TREC, 2003.
- [11] Craswell, N., Hawking, D. Overview of the TREC 2004 Web Track, in the 13th TREC, 2004.
- [12] Feng, G., Liu, T. Y., Zhang, X. D., Qin, T., Gao, B., Ma, W. Y. Level-Based Link Analysis, in the 7th APWeb, 2005.
- [13] Haveliwala, T.H. Topic-Sensitive Pagerank. In Proc. of the 11th WWW, 2002.
- [14] Hawking, D. Overview of the TREC-9 Web Track, in the 9th TREC, 2000.
- [15] Ingongngam, P., and Rungsawang, A. Report on the TREC 2003 Experiments Using Web Topic-Centric Link Analysis, in the 12th TREC, 2003.
- [16] Kamvar, S. D., Haveliwala, T. H., Manning, C. D., Golub, G. H. Exploiting the Block Structure of the Web for Computing PageRank, In Proc. of the 13th WWW, 2003.
- [17] Kleinberg, J. Authoritative Sources in a Hyperlinked Environment, Journal of the ACM, Vol. 46, No. 5, pp. 604-622, 1999.
- [18] MCBRYAN, O. GENVL and WWW: Tools for Taming the Web. In Proceedings of the 1st WWW, 1994.
- [19] Page, L., Brin, S., Motwani, R., and Winograd, T. The PageRank Citation Ranking: Bringing Order to the Web, Technical report, Stanford University, Stanford, CA, 1998.
- [20] Robertson, S. E. Overview of the Okapi Projects, Journal of Documentation, Vol. 53, No. 1, 1997, pp. 3-7.
- [21] Robertson, S. E., and Sparck Jones, K. Relevance Weighting of Search Terms, Journal of the American Society of Information Science, Vol. 27, No. May-June, 1976, pp. 129-146.
- [22] Shakery, A., Zhai, C. X. Relevance Propagation for Topic Distillation UIUC TREC 2003 Web Track Experiments, in the 12th TREC, 2003.
- [23] Song, R., Wen, J. R., Shi, S. M., Xin, G. M., Liu, T. Y., Qin, T., Zheng, X., Zhang, J. Y., Xue, G. R., and Ma, W. Y. Microsoft Research Asia at Web Track and Terabyte Track of TREC 2004, in the 13th TREC, 2004.