Exploiting the Hierarchical Structure for Link Analysis*

Gui-Rong Xue[1] Qiang Yang[3] Hua-Jun Zeng[2] Yong Yu[1] Zheng Chen[2]

[1]Computer Science and Engineering
Shanghai Jiao-Tong University
Shanghai 200030, P. R. China
grxue@sjtu.edu.cn,
yyu@cs.sjtu.edu.cn

[2]Microsoft Research Asia
49 Zhichun Road
Beijing 100080, P. R. China
{hjzeng, zhengc}@microsoft.com

[3]Department of Computer Science
Hong Kong University of Science and Technology
Clearwater Bay,Kowloon, Hong Kong
qyang@cs.ust.hk

ABSTRACT

Link analysis algorithms have been extensively used in Web information retrieval. However, current link analysis algorithms generally work on a flat link graph, ignoring the hierarchal structure of the Web graph. They often suffer from two problems: the sparsity of link graph and biased ranking of newly-emerging pages. In this paper, we propose a novel ranking algorithm called Hierarchical Rank as a solution to these two problems, which considers both the hierarchical structure and the link structure of the Web. In this algorithm, Web pages are first aggregated based on their hierarchical structure at directory, host or domain level and link analysis is performed on the aggregated graph. Then, the importance of each node on the aggregated graph is distributed to individual pages belong to the node based on the hierarchical structure. This algorithm allows the importance of linked Web pages to be distributed in the Web page space even when the space is sparse and contains new pages. Experimental results on the .GOV collection of TREC 2003 and 2004 show that hierarchical ranking algorithm consistently outperforms other well-known ranking algorithms, including the PageRank, BlockRank and LayerRank. In addition, experimental results show that link aggregation at the host level is much better than link aggregation at either the domain or directory levels.

Categories and Subject Descriptors

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

General Terms: Algorithms, Performance, Experimentation.

Keywords: Link Analysis, Hierarchical Web graph, Hierarchical Random Walk Model.

1. INTRODUCTION

Link analysis algorithms play key roles in Web search systems. They exploit the fact that the Web link structure conveys the relative importance of Web pages. For example, Google's PageRank [24] is a widely applied algorithm, which can be described as the stationary probability distribution of a certain random walk on the Web link graph - the graph whose nodes are the individual Web pages and directed edges are the hyperlink between pages. However, the existing link analysis algorithms often suffer from two problems: sparsity of link-graph and biased ranking of newly-emerging pages [10][14]. The first problem is caused by the fact that the link distribution of the Web graph generally satisfies the power law [15] and the sparse link matrix makes most of the pages unable to obtain any importance ranking at all [11][16]. Additionally, such link distribution makes the distribution of PageRank scores follow a power law [28]. The second problem implies that a newly emerging Web page has too few in-links to obtain a reasonable importance scores [10].

We suggest in this paper using the inherent hierarchical structure of the Web, which is embedded in URLs, to solve the problems. For example, in URL http://www.cs.berkeley.edu/Research/Projects/, we might expect to find some project-related information about research in the computer science department of UC Berkeley. In [26], Simon argued that all the system is likely to be organized as a hierarchical structure. The World Wide Web is a good example of the hierarchical organization [5]. From a local view of the Web, a Web site is organized as a hierarchical structure where the hierarchical information is represented by the directory structure. From a global view of the WWW, the whole Web is also organized as a hierarchical structure, in which the first level provides the top-level domains (such as berkeley.edu). Subsequently, the following levels contain the virtual hosts, the virtual folders and the Web pages. Furthermore, as discussed in [13][23], the link structure of Web is closely related to such hierarchical structure too.

In this paper, we take the link structure of the Web as well as the hierarchical structure of the Web into consideration in page importance calculation. Different from the traditional flat link graph, we construct a new Web-link graph which consists of two layers, i.e. the upper-layer graph and lower-layer graph, as shown in Figure 1. A novel random walk model is defined, which assumes that a user seeks for information by starting from the upper-layer and either random jump to another upper-layer node or follows the hierarchical links down to the lower-layer. Based on this random walk model, we propose a new link analysis algorithm called *Hierarchical Rank* to calculate the importance of the Web pages. This algorithm allows the importance of a supernode that is computed at the upper-layer graph to be propagated down to the Web pages in the lower-layer graph. We show that the link distribution in the upper-layer graph is much denser than that in general link graph. By exploiting this fact, the propagation step can solve the new page issue more satisfactorily.

In this paper, we conduct experiments on the .GOV collection of

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

*This work was conducted while the first author was doing internship at Microsoft Research Asia. Qiang Yang is supported by a Hong Kong RGC Grant CA03/04.EG01.

TREC 2003 and 2004. The experiment results show that our hierarchical rank consistently outperforms existing ranking algorithms at the flat link graph. Furthermore, we demonstrate that the aggregation at host level works much better than other two aggregation level, i.e. domain level and directory level.

The rest of this paper is organized as follows. In Section 2, we review the recent works on the link analysis and Web graph analysis. Then, we present the characteristics about the Web graph in Section 3. In Section 4, we propose our hierarchical ranking algorithm to take the link structure and hierarchical structure into consideration. Our experimental results are presented in Section 5. Finally, we summarize our main contributions and discuss the future works in Section 6.

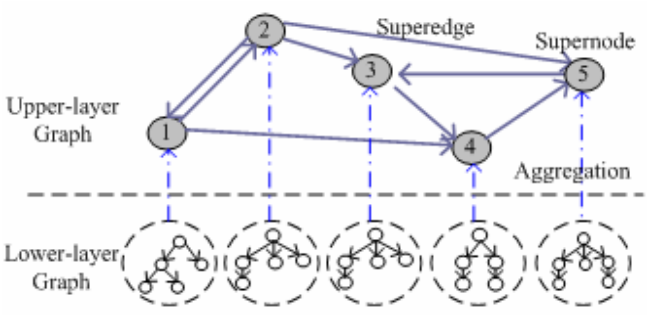

Figure 1. Hierarchical Structure of a Web Graph

2. PREVIOUS WORK

There are two types of works in recent research area, one is working on "link analysis", and the other is working on "Web structure analysis".

2.1 Previous Work on Link Analysis

The link analysis technology has been widely used to analyze the Web pages' importance, such as HITS [20] and PageRank [7][24]. PageRank is a core algorithm of Google which measures the importance of Web pages. It models the users' browsing behaviors as a random surfing model, which assumes that a user either follows a link from the current page or jumps to a random page in the graph. The PageRank of a page p_i is then computed by the following equation:

$$PR(p_i) = \frac{\varepsilon}{n} + (1-\varepsilon) \times \sum_{l_{j,i} \in E} PR(p_j) / \text{outdegree}(p_j) \quad (1)$$

where ε is a dampening factor, which is usually set between 0.1 and 0.2; n is the number of nodes in G; and out-degree(p_j) is the number of the edges leaving page p_j, i.e., the number of hyperlinks in page p_j. The PageRank could be computed by an iterative algorithm and corresponds to the primary eigenvector of a matrix derived from adjacency matrix of the available portion of the Web. Some extended link analysis algorithms to PageRank and HITS are proposed recently, such as [4][8][9][17]. But most of these works only consider the "flat" Web link structure and assign the same weight to the hyperlink. In [18], the author divides the Web graph into different blocks. According to the hyperlink that links between the different blocks or among the same block, they assign the different weight to the hyperlinks to compute the PageRank and evaluate the performance for search.

Recently, there are also several works to consider the aggregated graph analysis. A. Z. Broder et al. [6] proposed an efficient PageRank approximation on the aggregated graph. Y. Asano [1] proposes to find the directory-based site and then to efficiently find the Web community. Recently, Wu et al. [27] proposed a two-layer Layered Markov Model for decentralized Web ranking. In our experiments presented later in the paper, Wu et. al's algorithm is compared as the name "LayerRank". Since the algorithm pays much attention on the intra-link and does not consider the inter-link when computing the importance of the page layer, its performance is not better than the PageRank. As we discussed, the above algorithms only consider the individual Web pages as the elements while the inherent hierarchical structure of the Web are not taken into account. In this paper, we propose a combination of the host structure and the link analysis and show that this integration could improve the performance of ranking results.

2.2 Previous Work on Web Graph Structure

There are many research works on exploiting the hierarchical structure of the Web. Nadav et al. [12] proposed that hyperlinks tend to exhibit a "locality" that is correlated to the hierarchical structure of URLs, and that many features of the organization of information in the Web are predictable from the knowledge of the hierarchical structure. Kamvar et al. [19] proposed to utilize the hierarchical structure of the Web graph to accelerate the computation of PageRank.

There also exists some work that model the Web by considering the hierarchical structure of the Web. A hierarchical model of the Web was previously suggested by Laura et. al [21]. In their model, every page that enters the graph is assigned a constant value for the abstract "region" the page belongs to; the page is allowed to link only to other pages in the same region. Nadav et al. [13] described a Web-graph model to integrate the link structure and the explicit hierarchical structure together, which reflects a social division by the organization.

3. HIERARCHICAL WEB GRAPH

We first list the key terminologies used in the following discussion through an example URL: http://cs.stanford.edu/research/ index.htm, as shown in Table 1.

Table 1. Terminology

Term	Example: cs.stanford.edu/research/index.htm
Domain	**stanford.edu**
Host	**cs.stanford.edu**
Directory	**cs.stanford.edu/research/**
Page	**cs.stanford.edu/research/index.htm**

As mentioned in Section 1, the Web is organized as a hierarchical structure. In this paper, we propose a two-layer model of the Web. The upper-layer graph is an aggregated link graph which consists with supernodes and superedges, in which each supernode (such as domain, host and directory) aggregates the pages from the same supernode and superedges among supernodes are also aggregated from the underlying links of pages. The lower-layer graph is the hierarchical tree structure, in which each node is the individual Web page in the supernode, and the edges are the hierarchical links between the pages. (See Figure 1).

Within the hierarchical structure, the Web contains the Web pages, the directories, the hosts as well as the domains. Thus, the whole Web graph could be abstracted according to a hierarchical structure, such as a page level, a directory level, a host level and a domain level. According to the different levels, the hyperlinks at each level can be divided into two types: *intra-links* and *inter-links*. For example, when we construct the abstract Web graph at the directory level, the Web pages in the same directory are organized as a super node and the hyperlinks that link two Web pages in the directory are called intra-links. Likewise, the hyperlinks that link two Web pages in different super nodes are called the inter-links. Furthermore, the hyperlink of a page could be two types: one is *inlink* that link to the page and the other is *outlink* that links from the page. According to the analysis in [18][22], the intra-link plays less value than the inter-link when computing the PageRank at different level. However, the work still does not exploit the hierarchical structure inherent in the Web graph.

To investigate the hierarchical structure of the Web in detail, we gather the following statistics. We take all the hyperlinks in the .GOV collection, and count how many of the links are "Intra-link" and "Inter-link" at different abstract level.

Table 2. Link Distribution over .GOV Collection at Different Abstraction Levels

Level	Intra-Link	Inter-Link
Domain	7,342,031 (97%)	227,322 (3%)
Host	6,506,578 (86%)	1,062,775 (14%)
Directory	2,956,566 (39%)	4,612,787 (61%)
Page	0 (0%)	7,569,353 (100%)

As shown in Table 2, the percentage of the intra-link and the inter-link at four different levels is different. Take the result of the host level as an example, there are about 86% of all links were intra-links to a host, which tend to show a high degree of locality [13].

Another observation is interesting. We have also found that when links are inter-links between two different hosts, they tend to link to the top level of the host. These structure features of the Web have also shown up in other ways [19], allowing very high levels of compression for the link graph and enabling a block-oriented approach to accelerate the convergence of PageRank.

Based on the analysis, we construct a two-layer model of the Web, which is described in detail as follows.

3.1 Two-Layer Hierarchical Graph Construction

We use the $G=(V, E)$ to represent the directed graph of the Web, where the V and E refer to the vertex and edge set respectively. If a page p is represented by a graph vertex, we will use p to refer to the page as well as to the vertex.

Let $S=\{S_1, S_2, \ldots, S_m\}$ be a partition on the vertex set of G (by definition, this is also a partition of all the pages in the Web). We define the following types of directed graphs.

Upper-layer graph. An upper-layer graph contains m vertices called supernodes, one for each element of the partition. Supernodes are linked to each other using directed edges called superedges. Superedges are created based on the following rule: there is a directed superedge $\overrightarrow{E_{i,j}}$ from S_i to S_j if there is at least one page in S_i that links to some page in S_j. Furthermore, the upper-layer graph is a weighted graph where the weight of the edges from S_i to S_j is the number of the links from pages in S_i to pages S_j.

Lower-layer graph. In order to capture the essence of the structure within a supernode, we represent all the individual pages $\{p_0, p_1, \ldots, p_n\}$ in a supernode S with a hierarchical tree structure by considering the URL properties. Thus, all the pages in one supernode are organized by the URL relationship. For example, a supernode begins with a unique root node, which is entry page of supernode. Then the pages in the first level provide the children nodes of the root page, and so on. The structure in Figure 2 is generated by this method at the level of the host.

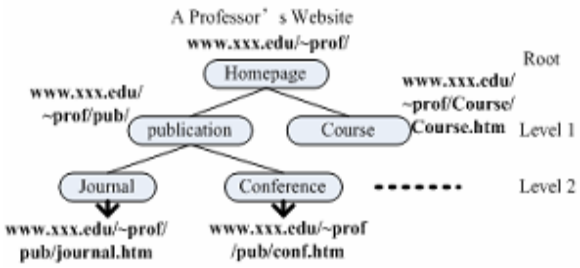

Figure 2. Hierarchical Structure of a Supernode

4. HIERARCHICAL RANKING

4.1 Hierarchical Random Walk Model

According to the inherent hierarchical structure of the Web, we first consider a surfing model that considers both the link structure and the hierarchical structure of the Web.

The hierarchical random walk model is as follows:

1. At the beginning of each browsing session, a user randomly selects the supernode.

2. After the user finished reading a page in a supernode, he may select one of the following three actions with a certain probability:

   (a). Going to another page within current supernode, following the hierarchical link structure of the supernode.

   (b). Jumping to another supernode that is linked by current supernode.

   (c). Ending the browsing.

According to the hierarchical random walk model, we can apply a two-stage computation of the hierarchical rank: first stage calculates the importance of supernodes according to the surfing behaviors among the supernodes and second stage calculates the importance of pages according to the surfing behaviors inside a supernode.

In the first stage, a user selects a supernode randomly, and jumps to another supernode randomly according to the superedges. The supernode surfing behavior in our model is exactly the same as the inter-page surfing behavior in PageRank. Thus, we can obtain the importance of supernodes by performing the PageRank algorithm on the supernode-level weighted graph.

In the second stage, we deal with page-level surfing inside a supernode. Since the interest of any page can be traced back to the root of the supernode, and the interest will dissipate when

propagating among pages inside a supernode, it is analogously like a *Dissipative Heat Conductance* (DHC) model [29] to describe the behaviors of users. In the DHC model, we place a single heat source with temperature SI_i at the root of each supernode S_i, where SI_i is the importance of the supernode S_i. The heat dissipates when it propagates alone with the tree structure inside the supernode P_j. If we keep temperature of the entry point constant, the temperature will converge after propagation for a time. The final temperature of each page gives the importance of that page. We give details in the following sections.

4.2 Calculating Supernode Importance

We represent the upper-layer graph as a matrix. Suppose that the Web contains m supernodes, the $m \times m$ adjacency matrix is denoted by A and the entries $A[i, j]$ stands for the weighted link from supernode i to supernode j. The adjacency matrix A is used to compute the importance of each supernode S_i, denoted as SI_i. In an "ideal" form, SI_i can be calculated by the sum of the importance of all the hosts that point to host i:

$$SI_i = \sum_{j:l_{ji} \in E} SI_j \cdot A[j,i] \tag{2}$$

However, in practice, many supernodes have no in-links (or the weight of them is 0), and the eigenvector of the above equation is mostly zero. Therefore, the basic equation (3) is modified to obtain a "*random walk model*" to deal with this issue. When browsing a supernode, with the probability 1-ε, a user randomly chooses one of the links on the current supernode and jumps to the supernode it links to. With a probability ε, the user "resets" by jumping to another supernode uniformly from the collection of supernodes. Therefore, the supernode importance formula is modified as Equation (4):

$$SI_i = \frac{\varepsilon}{n} + (1-\varepsilon) \sum_{j:l_{ji} \in E} SI_j \cdot A[j,i] \tag{3}$$

Or in a matrix form:

$$\overrightarrow{SI} = \frac{\varepsilon}{n} \vec{e} + (1-\varepsilon) A \overrightarrow{SI} \tag{4}$$

where $\vec{e}$ is the vector of all 1's, and ε ($0<\varepsilon<1$) is a parameter. In our experiment, we also set ε to 0.15.

4.3 Calculating Page Importance

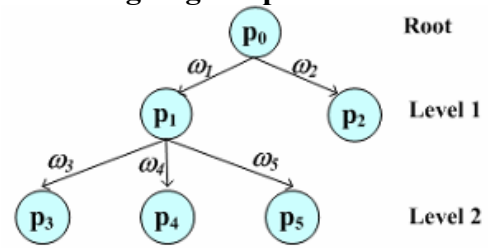

Figure 3. Hierarchical Weight Structure

After obtaining the importance of a supernode, the intuitive method for assigning the importance to a page is the method proposed in [14], in which the importance of the pages in the supernode is equal to the importance of the supernode they belong to. Here we propose that the pages' importance scores should be calculated according to the importance of the supernode as well as the hierarchical structure in the supernode. In this section, we introduce a *Dissipative Heat Conductance* (DHC) model to compute the importance of the pages in a hierarchical structure.

4.3.1 Constructing Weighted Tree Structure

Several factors could affect the calculation of the page's importance, such as whether a page is an index page or a content page, as well as the number of external links from the outside of the supernodes. We formulate a hierarchical tree structure of the supernode as a directed weighted tree structure as shown in Figure 3. In the weighted tree structure, each edge is from to the parent node to its child nodes and also the edge is weighted by the properties of the Web pages.

Here a function is given to calculate the weight of page related to his parent page. Given a page p_j in the supernode S_i, the weight ω_j that the page p_j to its parent node is calculated as:

$$\omega_j = \theta \cdot link(p_j) + (1-\theta) index(p_j) \tag{5}$$

Index is used to assign different weights to the index pages and other pages. Here we also use the hierarchical rule to judge whether the page is a index page or not. If the URL of the page contains the characteristics, such as *index*, *default* or the URL is ended with "/", the page is the index page.

$$index(p_j) = \begin{cases} 1 & if \ \ p_j \ \ is \ index \ \ page \\ \alpha & if \ \ p_j \ \ is \ other \ \ page \end{cases} \tag{6}$$

The parameter α is a factor between 0 and 1. Its importance is explained later.

Link is defined to calculate the percentages of the inlinks that the page p_j has. Formally, the function is defined as follows:

$$link(p_j) = \beta \frac{OIL(p_j)}{\sum_{p_k \in S_i} OIL(p_k)} + (1-\beta) \frac{IIL(p_j)}{\sum_{p_k \in S_i} IIL(p_k)} \tag{7}$$

where β is the factor to give the inter-link and the intra-link with the different weight. $IIL(p_j)$ is the number of the intra-hyperlink to the page p_j, while $OIL(p_j)$ is the number of inter-hyperlink to the page p_j.

4.3.2 Calculating Page Importance by DHC

Based on the hierarchical weighted structure, a page's importance in a supernode can be calculated recursively from the root page down to the bottom pages by using *DHC* algorithm. Actually, in this paper, we use the equations 8, which is simplifications of *DHC*, to calculate the page's importance based on the hierarchical weight structure. Each page p_j gets a value w_{ij} that shows how important the page p_j is in the supernode S_i.

$$w_{ij} = \prod_{p_k \in \{\text{nodes from } n_j \text{ to root}\}} \gamma \times \omega_k \tag{8}$$

where the parameter γ is a *heat dissipative* factor.

Thus, we get a matrix $W_{m \times n}$ that each entry w_{ij} is value of page p_j importance in supernode S_i. Obviously, if page p_j does not belong to supernode S_i, w_{ij} is equal to 0. Here the weight of root page is set to 1.

Finally, the importance of a page p_j on the whole Web graph, denoted as PI_j, is calculated as follows:

$$PI_j = SI_i \times w_{ij} \tag{9}$$

Or in a matrix form:

$$\overrightarrow{PI} = \overrightarrow{SI} \cdot W \qquad (10)$$

where the page p_j belongs to supernode S_i, SI_i is the importance of the supernode S_i. The importance of the root page is equal to the importance of supernode. An example is shown in Figure 4.

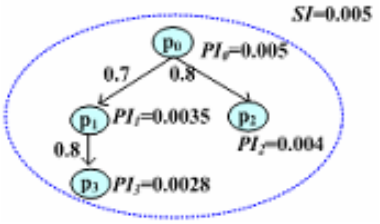

Figure 4. Example of Page Importance Calculation

5. EXPERIMENTS

We designed the experiment on the TREC .GOV dataset. The queries of the topic distillation in TREC 2003 and TREC 2004 are used in our experiments.

5.1 TREC .GOV Data Set

Table 3. TREC .GOV Dataset

		TREC 2003	TREC 2004
Number of Queries		50	75
Relevant Pages/Query		10.32	21.32
BM2500	P@10	0.112	0.2053
	MAP	0.1285	0.1517

The collection of TREC .GOV consists of 1,247,753 Web pages from a fresh crawl of the Web pages made in early 2002. Among them, 1,053,372 are text/html, which are used in our experiment. There are totally 7,569,353 hyperlinks in the collection. In the experiment, we used BM2500 [25] as the relevance weighting function. The mean average precision on TREC 2003 is 0.1285 and the P@10 is 0.112. Compared with the best result of TREC 2003 anticipants (with MAP of 0.1543 and P@10 of 0.1280), this baseline is reasonable. While the performance of BM2500 on TREC 2004 is that the mean average precision and the P@10 are 0.1517 and 0.2053, respectively. Details are shown in Table 3.

5.2 Evaluation Metrics

In order to measure the retrieval performance of different ranking algorithms, we take P@10 and Means Average Precision (MAP) as the evaluation metrics which is widely used in TREC and the details could be found in [3].

As we discussed in Section 1, the sparseness problem and new pages problem are two major problems in current link analysis. We also compare the effectiveness of different ranking algorithms on solving the two problems by KDist [19] distance. Consider two partially ordered lists of web pages τ_1 and τ_2, KDist(τ_1, τ_2) is the probability that τ_1' and τ_2' disagree on the relative ordering of a randomly selected pair of distinct nodes. In this work, we just compared the lists containing the same sets of Web pages, so that KDist is identical to Kendall's τ distance.

5.3 Experimental Methods

We describe four existing ranking methods to compare with our proposed hierarchical ranking algorithm.

5.3.1 PageRank

We implemented the PageRank algorithm based on the link matrix deduced from traditional page level link analysis.

5.3.2 Weighted PageRank (WeightRank)

In [18], the authors divide the whole graph into blocks. Then the hyperlinks between the different blocks and the hyperlinks among the same blocks are assigned different weights when computing the important of the Web pages. Their experiments verified that the host level could achieve the higher performance than the domain level and the directory level. In this paper, we also compare with the host level partition.

5.3.3 Two-Layer PageRank (LayerRank)

As we reviewed in Section 2, a two-layered ranking algorithm [27] was proposed. One layer is host-layer PageRank and the other layer is document-layer PageRank inside a host. Then, a weighted product between them is applied to obtain the final global ranking for all the Web pages.

5.3.4 Block-Level PageRank (BlockRank)

We also conduct an experiment to compare with the block-level PageRank [12], which divide the whole Web page into different semantic blocks. Then the PageRank algorithm is performed on the block-level.

5.3.5 Hierarchical Ranking

Hierarchical ranking corresponds to our proposed algorithm, which interpolates the link structure and the hierarchical structure together. Three level abstractions are proposed to partition the Web space into a domain level, a host level and a directory level. Accordingly, we define corresponding ranking as *DomainRank*, *HostRank* and *DirectoryRank*, respectively.

In the following test, we chose the top 2000 results according to the BM2500 score. Then we combine the relevance with the importance as follows:

$$\lambda f_{relevance}(p)+(1-\lambda) f_{importance}(p) \qquad (11)$$

The function f can be score-based or the order-based of the page p in the results. The score-based function is the relevance score or the importance score, while the order-based function is the page's position in the list which is sorted by relevance score or by importance score.

For each ranking method, we both evaluate the two functions on the TREC 2003 and take the higher performance function as the combining method.

5.4 Experimental Results

Several parameters for our experiments are fixed in the following experiments, i.e. θ=0.6, α=0.6, β=0.4 and γ=0.8. We used the score-based linear combination for the hierarchical ranking. Through tuning on TREC 2003 for the best p@10 precision, we set the combining parameter λ to 0.85, 0.6 and 0.73 for DomainRank, HostRank and DirectoryRank, respectively. For the methods of PageRank, LayerRank, WeightRank and BlockRank, we take the parameters that achieve the best performance on these ranking.

5.4.1 Performance at Different Hierarchical Levels

The partitioned Web graph could be aggregated into different abstraction levels, the domain, host, and directory levels, which correspond to three ranking algorithms: DomainRank, HostRank

and DirectoryRank, respectively. We make the comparison on TREC 2003 and TREC 2004 to show which level gives a higher performance for each algorithms. Figure 5 shows that the host-level abstraction is the best choice for the hierarchical link analysis. The performance decreased at the directory-level link analysis since the navigation links within the host do affect the performance of link analysis. Additionally, the domain-level link analysis also can not achieve the high performance. By observation on the link data, some useful inter-links between the hosts which are recommendation hyperlink, cannot take into account when computing the DomainRank. So the rest experiments are all conducted at the host level. That is, we compare other algorithms with the HostRank.

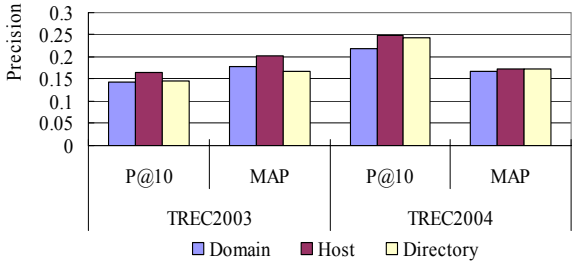

Figure 5. Performance of Different Levels

5.4.2 Overall Performance

As can be seen in Figure 6, all ranking algorithms achieve higher performance than the baseline of relevance function, which confirms that the link analysis on the TREC data could work well.

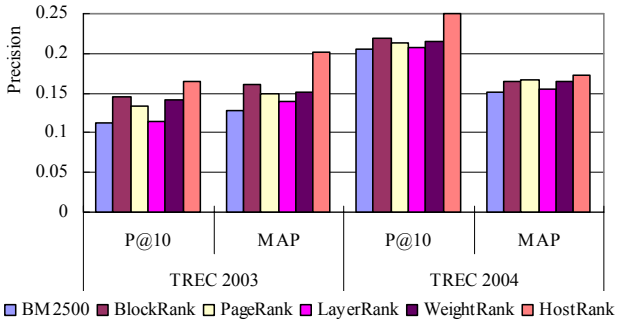

Figure 6. Performance of Different Ranking on TREC

In the topic distillation task of TREC 2003, our proposed HostRank algorithm significantly outperforms BM2500, BlockRank, PageRank, LayerRank and WeightRank on average precision by 57.4%, 25.5%, 36.2%, 46.0%, and 33.1%, respectively. Also, on the measure of P@10, HostRank significantly outperforms BM2500, BlockRank, PageRank, LayerRank and WeightRank by 46.4%, 12.3%, 22.4%, 43.9%, and 15.5%, respectively.

In the topic distillation task of TREC 2004, our proposed HostRank algorithm significantly outperforms BM2500, BlockRank, PageRank, LayerRank and WeightRank on average precision by 14.3%, 5.9%, 4.6%, 11.8%, and 5.7%, respectively. Meanwhile, on the measure of P@10, HostRank significantly outperforms BM2500, BlockRank, PageRank, LayerRank and WeightRank by 21.4%, 14.0%, 16.9%, 20.6%, and 16.1%, respectively.

From results on the queries of topic distillation task of TREC 2003 and 2004. Our proposed hierarchical ranking algorithm consistently achieves highest performance than the well-known link analysis algorithms. Integrating the hierarchical structure and the link structure of the Web exactly improve the performance of retrieval. LayerRank algorithm, which performs two-layer PageRank calculation, can not achieve better performance than PageRank and WeightRank. Since the algorithm pays much attention on the intra-links of the host while these hyperlinks contain less information. WeightRank algorithm can also acquire higher performance than the PageRank and LayerRank algorithms. By assigning different weight to the intra- and inter-links of the Web graph, the effect by the navigation link can be leveraged to some extent. BlockRank algorithm can achieve higher performance than PageRank, WeightRank and LayerRank for the method provide more semantic link graph than on the page level.

Table 4. P-value for Different Ranking Results

		Block Rank	Page Rank	Layer Rank	Weight Rank	Host Rank
TREC 2003	P@10	0.015	0.041	0.028	0.021	**0.003**
	MAP	0.041	0.097	0.021	0.027	**0.002**
TREC 2004	P@10	0.324	0.335	0.433	0.292	**0.006**
	MAP	0.162	0.164	0.552	0.458	**0.028**

To understand whether these improvements are statistically significant, we performed t-tests and the results are shown in Table 4. Compared to the baseline BM2500, all the t-test results show that HostRank significantly improves the search results.

5.4.3 Ranking on Sparseness Data

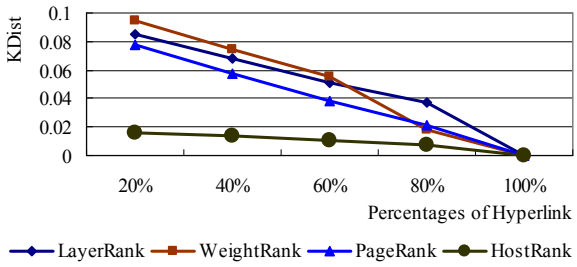

Figure 7. KDist of Sparse Link Graph
(The lower value shows the higher performance)

The density of a link graph can have a significant impact on the performance of link analysis. To show the performance of our ranking function, we conducted an experiment to simulate the phenomenon of the sparseness of link graph and compare the performance about four rank algorithms: PageRank, WeightRank, LayerRank and HostRank.

In Figure 7 we empirically analyze how KDist between ranking orders on the sparse link graph and the whole link graph evolves when the density of link graph becomes from slight to strong. In this experiment, we randomly select 20%, 40%, 60%, 80% and 100% of the whole hyperlinks to represent different degree of how tightly the link graph is. The results show that the degree of how tightly the link graph is has different impact on the final ranking of the different link analysis algorithms. When the link graph becomes denser, the ranking orders of all link analysis algorithms tend to the final orders. As seen from the Figure 7, the KDist of HostRank algorithm is below that of the other algorithm, which means that the sparseness has the least impact on the ranking order of the HostRank algorithm.

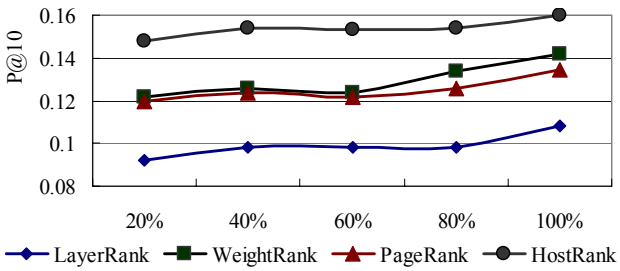

Figure 8. Performance of Sparse Link on TREC 2003

We also conduct the experiments by utilizing the different ranking to show the performance on retrieval. As seen in Figure 8, the HostRank both achieve the highest performance and least impact on the sparse link graph. P@10 of HostRank is decreased about 1% when the hyperlink number changed from 100% to 20% on TREC 2003, while P@10 of other three algorithms is decreased about 2%.

5.4.4 Ranking of New Pages

As we mentioned in Section 1, our methods can alleviate the new pages biased problem [10]. To show the performance of our ranking function, we conducted an experiment to simulate the phenomenon of the new pages and compare the performance about four rank algorithms: PageRank, WeightRank, LayerRank and HostRank.

Table 5. KDist Measurement on Ranking of New Pages
(The lower value shows the higher performance)

Method	KDist	Method	KDist
LayerRank	0.072	PageRank	0.044
WeightRank	0.053	HostRank	**0.0159**

First, we randomly select 10,000 pages with different rank values as the test pages. Then, we remove 90% of the hyperlinks that linked to the 10,000 pages. In this way, we constructed a new Web graph with the remaining hyperlinks. We performed four algorithms on the modified Web graph and then measured the KDist to show the effect on the new pages. The results are shown in Table 5.

The average distance KDist is 0.0159 for the HostRank in the .GOV dataset. Notice that it is lower. This means that the ordering induced by HostRank on the partial graph is very close to HostRank on the full graph and the new pages could also get its more approximate ranking by the HostRank algorithm.

5.5 Parameter Tuning

In this section, the experiments are conducted on the 50 queries of topic distillation at TREC 2003 to show how the parameters affect the performance of our proposed hierarchical ranking algorithms.

5.5.1 Combining Parameters Selection

We conduct several experiments on the queries of topic distillation at TREC 2003 by host level ranking algorithm to show the performance on different parameters. Each parameter is tuned independently.

In Equation 6, the parameter α is used to show a trade-off between the index page and the non-index page. As shown in Figure 9-1, when we set α to 0.6, the hierarchical rank could achieve the highest performance.

Meanwhile, we also perform the experiment to show the different weight on the performance between intra-inlink and inter-inlink. As shown in Figure 9-2, experimental result also verifies that inter-inlink should give higher weight than intra-inlink and the ranking could get the highest performance when β in equation 7 is set to 0.4.

In Equation 5, we linearly combine the link function and index function by the parameter θ. As shown in Figure 9-3, the combination parameter is set to 0.6, which implies that the index function is more important than the link function which is also confirmed by the experiment in Section 5.5.2.

The last experiment is on tuning the parameter γ in the Equation 8. As shown in Figure 9-4, the hierarchical rank could achieve the highest performance when the *heat dissipative* factor is set to 0.8. Lower or higher value could not achieve the higher performance.

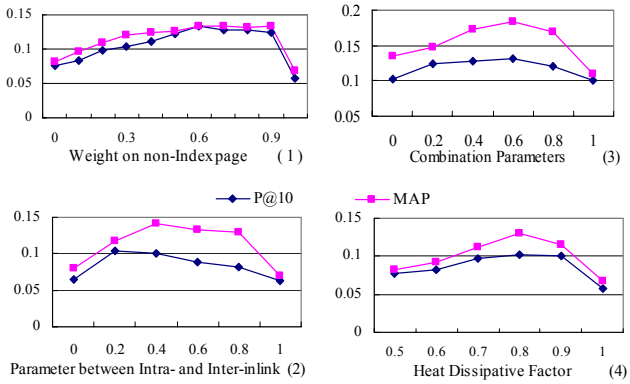

Figure 9. Performance of Different Propagation Parameters

5.5.2 Comparison on Individual Parameters

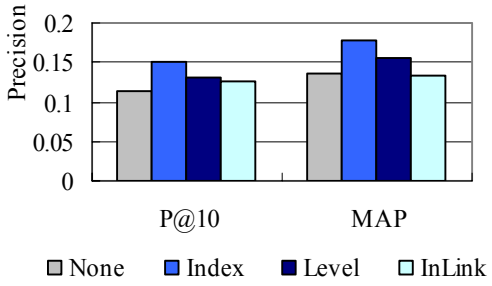

Figure 10. Performance of Individual Parameters

As described in Equation 5, 6, and 7, there are three factors that affect the calculation of a pages' importance: the depth of the pages in the tree structure (Level), the in-link distribution of the pages (InLink) and whether the page is an index page (Index). In this section, we conduct experiments on the queries of topic distillation track at TREC 2003 to show how the parameters affect the performance of hierarchical ranking algorithm individually. The baseline method directly sets the importance of the host to the pages in the host. We denote its method as "None".

As shown in Figure 10, these features have impacts on the performance of retrieval. The factor of whether a page is an entry page is the most important in the topic-distillation task. Then the level of the page also shows great improvement on the performance. The in-link feature shows little improvement, which also testify that the intra-links are less valuable for link analysis.

6. CONCLUSIONS AND FUTURE WORK

Considering both the hierarchical structure and the link structure of the Web, we proposed a Hierarchical Random Walk Model, which approximates the behaviors of the users' surfing the Web. Based on the model, we presented a hierarchical ranking algorithm to calculate the importance of Web pages. The ranking algorithm can significantly improve the performance of Web search, efficiently alleviate the sparse link problem and assign the reasonable rank to the newly-emerging Web pages.

In this work, we only perform the experiments on the TREC .GOV collection, which might be different in the other domain. In future work, we will conduct the experiments on the large scale Web collection to evaluate our algorithms.

7. REFERENCES

[1] Y. Asano, Applying the Site Information to the Information Retrieval from the Web. In Proc. of IEEE Conference of WISE, 2002.

[2] R. Baeza-Yates, F. Saint-Jean, and C. Castillo. Web Dynamics, Age and Page Quality. In Proc. of SPIRE 2002, LNCS, Springer, Lisbon, Portugal, September 2002.

[3] R. Baeza-Yates and B. Ribeiro-Neto. Modern Information Retrieval. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 1999.

[4] K. Bharat and M. R. Henzinger. Improved Algorithms for Topic Distillation in a Hyperlinked Environment. In Proc. of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, July 1998.

[5] K. Bharat, B. W. Chang, M. R. Henzinger, and M. Ruhl. Who Links to Whom: Mining Linkage between Web Sites. In the Proc. of 1st International Conference on Data Mining (ICDM) p51-58， 2001.

[6] A. Broder, and R. Lempel. Efficient PageRank Approximation via Graph Aggregation. In Proc. of 13th International World Wide Web Conference, May 2004.

[7] S. Brin and L. Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. In Proc. of 7th International World Wide Web Conference, May 1998.

[8] 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 Proc. of the 7th International World Wide Web Conference, May 1998.

[9] S. Chakrabarti, M. Joshi, and V. Tawde. Enhanced Topic Distillation using Text, Markup Tags, and Hyperlinks. In Proceedings of the 24th Annual International ACM SIGIR conference on Research and Development in Information Retrieval , ACM Press, 2001, pp. 208-216.

[10] J. Cho and S. Roy. Impact Of Search Engines On Page Popularity. In the Proc. of 13th World Wide Web Conference, May 2004.

[11] B. D. Davison. Recognizing Nepotistic Links on the Web. In Artificial Intelligence for Web Search, pages 23--28. AAAI Press, July 2000.

[12] D. Cai, X. F. He, J. R. Wen and W.Y. Ma. Block-level Link Analysis. The 27th Annual International ACM SIGIR conference on Research and Development in Information Retrieval (SIGIR'2004), July 2004.

[13] N. Eiron and K. S. McCurley. Locality, Hierarchy, and Bidirectionality on the Web. In the Workshop on Web Algorithms and Models, 2003.

[14] N. Eiron, K. S. McCurley, and J. A. Tomlin. Ranking the Web Frontier. In Proceedings of the 13th International World Wide Web Conference, pages 309--318. ACM Press, 2004.

[15] M. Faloutsos, P.Faloutsos, and C.Faloutsos. On Powerlaw Relationships of the Internet Topology. In ACM SIGCOMM, pages 251-262, 1999.

[16] C. Gurrin and A. F. Smeaton. Replicating Web Structure in Small-Scale Test Collections. Information Retrieval, 2004.

[17] T. H. Haveliwala. Topic-Sensitive PageRank. In Proc. of the 11th Int. World Wide Web Conference, May 2002.

[18] X. M. Jiang, G. R. Xue, H. J. Zeng, Z. Chen, W.-G. Song, and W.-Y. Ma. Exploiting PageRank Analysis at Different Block Level. In the Proc. of Conference of WISE 2004.

[19] S. D. Kamvar, T. H. Haveliwala C. D. Manning and G. H. Golub. Exploiting the Block Structure of the Web for Computing PageRank. In Proc of 12th International World Wide Web Conference, 2003.

[20] J. Kleinberg, Authoritative Sources in a Hyperlinked Environment. Journal of the ACM, Vol. 46, No. 5, pp. 604-622, 1999.

[21] L. Laura, S. Leonardi, G. Caldarelli, and P. D. L. Rios. A Multi-Layer Model for the Web Graph. In 2nd International Workshop on Web Dynamics, Honolulu, 2002.

[22] C. Monz, J. Kamps, and M. de Rijke. The University of Amsterdam at TREC 2002.

[23] E. Ravasz and A.L.Barabasi. Hierarchical Organization in Complex Networks. PHYSICAL REVIEW, E67,026112, 2003.

[24] L. Page, S. Brin, R. Motwani, and T. Winograd. The PageRank Citation Ranking: Bringing Order to the Web. Technical report, Stanford University, Stanford, CA, 1998.

[25] S. E. Robertson. Overview of the Okapi Projects. Journal of Documentation, Vol. 53, No. 1, 1997.

[26] H. A. Simon. The Sciences of the Artificial. MIT Press, Cambridge, MA, 3rd edition, 1981.

[27] J. Wu, K. Aberer. Using a Layered Markov Model for Decentralized Web Ranking. EPFL Technical Report IC/2004/70, August 19, 2004.

[28] G. Pandurangan, P. Raghavan, and E. Upfal. Using PageRank to Characterize Web Structure. In 8th Annual International Computing and Combinatorics Conference, 2002.

[29] D. Zhou, J., Weston, A. Gretton, O. Bousquet, and B. Scholkopf. Ranking on Data Manifolds. Advances in NIPS 16. Cambridge, MA: MIT Press, 2004.