Block-level Link Analysis

Deng Cai[1*] Xiaofei He[2*] Ji-Rong Wen[*] Wei-Ying Ma[*]

[*]Microsoft Research Asia
Beijing, China
{jrwen, wyma}@microsoft.com

[1]Tsinghua University
Beijing, China
cai_deng@yahoo.com

[2]Department of Computer Science
University of Chicago
xiaofei@cs.uchicago.edu

ABSTRACT

Link Analysis has shown great potential in improving the performance of web search. PageRank and HITS are two of the most popular algorithms. Most of the existing link analysis algorithms treat a web page as a single node in the web graph. However, in most cases, a web page contains multiple semantics and hence the web page might not be considered as the atomic node. In this paper, the web page is partitioned into blocks using the vision-based page segmentation algorithm. By extracting the page-to-block, block-to-page relationships from link structure and page layout analysis, we can construct a semantic graph over the WWW such that each node exactly represents a single semantic topic. This graph can better describe the semantic structure of the web. Based on block-level link analysis, we proposed two new algorithms, Block Level PageRank and Block Level HITS, whose performances we study extensively using web data.

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, Performance, Human Factors

Keywords

Web information retrieval, VIsion-based Page Segmentation, Graph Model, Link Analysis

1. INTRODUCTION

Traditional information retrieval techniques can give poor results on the Web, with its vast scale and highly variable content quality. Recently, however, it was found that Web search results might be much improved by using the information contained in the link structure between pages. PageRank [19] and HITS [16] are two of the most popular algorithms. A number of extensions to these two algorithms are also proposed, such as [1][2][6][7][8][13][17].

SIGIR'04, July 25–29, 2004, Sheffield, South Yorkshire, UK.

PageRank simulates a random walk on the link graph and assigns each page a score of importance. Different from PageRank, HITS assigns two scores to a page, authority score and hub score. Hubs and authorities exhibit a mutually reinforcing relationship.

All these link analysis algorithms are based on two assumptions: (a) the links convey human endorsement. If there exists a link from page A to page B and these two pages are authored by different people, then the first author found the second page valuable. Thus the importance of a page can be propagated to those pages it links to. (b) pages that are co-cited by a certain page are likely related to the same topic. However, these two assumptions do not hold in many cases. A typical example is the web page at http://news.yahoo.com (Figure 1) which contains multiple semantics (marked with rectangles with different colors) and many links only for navigation and advertisement (the left region). In this case, the importance of each page may be mis-calculated by PageRank and topic drift may happen in HITS.

These two problems are caused by the fact that a single web page often contains multiple semantics and the different parts of the web page have different importance in that page. Thus, from the perspective of semantics, a web page should not be the smallest unit. The hyperlinks contained in different semantic blocks usually point to the pages of different topics. Naturally, it is more reasonable to regard the semantic blocks as the smallest units of information. Recently some works were proposed to overcome these two problems [2][9][14][15][18]. But they still consider the web page as the unit of information.

In this paper, we proposed two novel link analysis algorithms called Block Level PageRank (BLPR) and Block Level HITS (BLHITS) which treat the semantic blocks as information units. By using vision-based page segmentation (VIPS) algorithm [4][5], we extract page-to-block and block-to-page relationships and then construct a page graph and a block graph. Based on this graph model, the new link analysis algorithms are capable of discovering the intrinsic semantic structure of the web. The above two assumptions become reasonable in block level link analysis algorithms. Thus, the new algorithms can improve the performance of search in web context.

The rest of this paper is organized as follows. In Section 2, we describe the VIPS page segmentation algorithm. In Section 3, we describe how to build the graph models. Link analysis algorithms using the new graph model are given in Section 4. Some experimental evaluations are provided in Section 5. Finally, we give concluding remarks and future work in Section 6.

Figure 1: Part of a sample web page (news.yahoo.com). Clearly, this page is made up of different semantic blocks (with different color rectangle). Different blocks have different importances in the page. The links in different blocks point to the pages with different topics.

2. VISION-BASED PAGE SEGMENTATION

The VIsion-based Page Segmentation (VIPS) algorithm [4][5] aims to extract the semantic structure of a web page based on its visual presentation. Such semantic structure is a tree structure; each node in the tree corresponds to a block. Each node will be assigned a value (Degree of Coherence) to indicate how coherent of the content in the block based on visual perception, the bigger is the DoC value, the more coherent is the block. The VIPS algorithm makes full use of page layout structure. It first extracts all the suitable blocks from the html DOM tree, and then it finds the separators between these blocks. Here, separators denote the horizontal or vertical lines in a web page that visually cross with no blocks. Based on these separators, the semantic tree of the web page is constructed. Thus, a web page can be represented as a set of blocks (leaf nodes of the semantic tree). For details, see [5]. Compared with DOM based methods, the segments obtained by VIPS are much more semantically aggregated [24]. Noisy information, such as navigation, advertisement, and decoration can be easily removed because they are often placed in certain positions of a page. Contents with different topics are distinguished as separate blocks.

3. BLOCK LEVEL WEB GRAPH

In this section, we describe how to construct a block level web graph. Like page-to-page graph model, the block-to-block model might be useful for many web based applications, such as web image retrieval and web page categorization, but in this paper our primary purpose is using link analysis to improve web information retrieval. Our graph model is induced from two kinds of relationships, *i.e.* ***block-to-page*** and ***page-to-block***. We begin with some definitions. Let P denote the set of all the web pages, $P = \{p_1, p_2, \ldots, p_k\}$, where k is the number of web pages. Let B denote the set of all the blocks, $B = \{b_1, b_2, \ldots, b_n\}$, where n is the number of blocks. It is important to note that, for each block there is only one page that contains that block. $b_i \in p_j$ means the block i is contained in the page j.

3.1 Block-Based Link Structure Analysis

In this section, we describe the block-based link structure using matrix notations. Let Z denote the block-to-page matrix with dimension $n \times k$. Z can be formally defined as follows:

$$Z_{ij} = \begin{cases} 1/s_i & \text{if there is a link from block } i \text{ to page } j \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where s_i is the number of pages that block i links to. Z_{ij} can also be viewed as a probability of jumping from block i to page j. The block-to-page relationship gives a more accurate and robust representation of the link structures of the Web. It is important to note that, traditional link analysis algorithm like HITs [16] does not distinguish between links in different semantic blocks. It may cause the problem of topic drifting.

From Figure 1, we can see the links in different blocks point to the pages with different topics. In this example, one link points to a page about entertainment and another link points to a page about sports.

3.2 Page Layout Analysis

The page-to-block relationships are obtained from page layout analysis. Let X denote the page-to-block matrix with dimension k × n. As we have described above, each web page can be segmented into blocks. Thus, X can be naturally defined as follows:

$$X_{ij} = \begin{cases} 1/s_i & \text{if } b_j \in p_i \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

where s_i is the number of blocks contained in page i. The above formula assigns equal importance value to each block in a page. It is simple but less practical. Intuitively, some blocks with big size and centered position are probably more important than those blocks with small size and margin position. This observation leads to the following formula,

$$X_{ij} = \begin{cases} f_{P_i}(b_j) & \text{if } b_j \in p_i \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

where f is a function which assigns to every block b in page p an importance value. Specifically, the bigger $f_p(b)$ is, the more important the block b is. f is empirically defined below,

$$f_p(b) = \beta \frac{\text{size of block } b \text{ in page } p}{\text{dist. from the center of } b \text{ to the center of screen}} \quad (4)$$

where β is a normalization factor to make the sum of $f_p(b)$ to be 1, i.e.

$$\sum_{b \in p} f_p(b) = 1 \quad (5)$$

Note that, $f_p(b)$ can also be viewed as a probability that the user is focused on the block b when viewing the page p.

Some more sophisticated definitions of f can be formulated by considering the background color, fonts, etc. Also, f can be learned from some pre-labeled data (the importance value of the blocks can be defined by people) as a regression problem by using learning algorithms such as SVM [20], neural networks, etc. By incorporating more advanced block importance models, such as the schemes described in [21], we expect that a better result might be achieved.

3.3 Building Graph Models

In the last two subsections, we have constructed two affinity matrices, *i.e.* block-to-page and page-to-block. Based on these two matrices, we can build two graph models, *i.e. page graph* $G_P(V_P, E_P, W_P)$ and *block graph* $G_B(V_B, E_B, W_B)$. For each graph, V is the set of the nodes (page, block, respectively), E is the set of edges linking two nodes, W is a weight matrix defined on the edges. We begin with the page graph.

3.3.1 Page Graph

When constructing a graph, we essentially define a weight matrix on the edges. W_P can be simply defined as follows. $W_P(i, j)$ is 1 if page i links to page j, and 0 otherwise. This definition is pretty simple yet has been widely used as the first step to many applications, such as PageRank [19], HITS [16], community mining [11][12], etc. However, based on our previous discussions, different blocks in a page have different importance. Therefore, those links in blocks with high importance value should be more important than those in blocks with low importance value. In other words, a user might prefer to follow those links in important blocks. This consideration leads to the following definition of W_P,

$$W_P(\alpha,\beta) = \sum_{b\in\alpha} f_\alpha(b) Z(b,\beta), \quad \alpha,\beta \in P \tag{6}$$

or

$$W_P = XZ \tag{7}$$

X is a $k \times n$ page-to-block matrix and Z is a $n \times k$ block-to-page matrix, thus W_P is a $k \times k$ page-to-page matrix.

Here we provide a simple analysis of our definition of W_P from the probabilistic viewpoint. Let's consider $W_P(\alpha, \beta)$ as a probability $Prob(\beta|\alpha)$ of jumping from page α to page β. Since page α is composed of a set of blocks, we have

$$Prob(\beta \mid \alpha) = \sum_{b\in\alpha} Prob(\beta \mid b)\, Prob(b \mid \alpha) \tag{8}$$

where $Prob(\beta|b)$ is actually $Z(b, \beta)$ and $Prob(b|\alpha)$ is $f_\alpha(b)$.

Finally, it would be interesting to see under what conditions our definition of W_P reduces to the ordinary definition. This occurs when the function $f(b)$ is defined as the number of links contained in block b.

3.3.2 Block Graph

The block graph is constructed over the blocks. Let's first consider a jump from block a to block b. Suppose a user is looking at block a. In order to jump to the block b, he first jumps to page β which contains block b, and then he focuses his attention on block b. Thus, a natural definition of W_B is as follows,

$$\begin{aligned} W_B(a,b) &= Prob(b \mid a) \\ &= \sum_{\gamma\in P} Prob(\gamma \mid a)\, Prob(b \mid \gamma) \\ &= Prob(\beta \mid a)\, Prob(b \mid \beta) \\ &= Z(a,\beta) X(\beta,b), \quad a,b \in B \end{aligned} \tag{9}$$

or

$$W_B = ZX \tag{10}$$

where W_B is a $n \times n$ matrix. By definition, W_B is clearly a probability transition matrix. However, there is still one limitation of this definition such that it is unable to reflect the relationships between the blocks in the same page. Two blocks are likely related to the same topics if they appear in the same page. This leads to a new definition,

$$W_B = (1-t)ZX + tD^{-1}U \tag{11}$$

where t is a suitable constant. D is a diagonal matrix, $D_{ii} = \sum_j U_{ij}$. U_{ij} is zero if block i and block j are contained in different pages; otherwise, it is set to the *DOC* (degree of coherence, see [5] for details) value of the smallest block containing both block i and block j. It is easy to check that the sum of each row of $D^{-1}U$ is 1. Thus, W_B can be viewed as a probability transition matrix such that $W_B(a, b)$ is the probability jumping from block a to block b.

4. BLOCK LEVEL LINK ANALYSIS ALGORITHMS

PageRank and HITS are two of the most popular link analysis algorithms. However, both of them ignore the fact that a single web page might contain multiple semantics and the different parts of the web page might have different importances. The different hyperlinks in a page might point to the pages with different semantics. Therefore, the computed importances of web pages might not accurately reflect the web structure in the sense of semantics. In this section, we introduce two new link analysis algorithms, *i.e. block level PageRank* and *block level HITS*, which are based on the graph model described in previous section.

4.1 Block Level PageRank

Block Level PageRank (BLPR) is similar to the original PageRank algorithm in spirit. The key difference between them is that, traditional PageRank algorithm models web structure in the page level while BLPR models web structure in the block level.

Let A denote the weight matrix of graph G built in previous section. We first construct a probability transition matrix M by renormalizing each row of A to sum to 1. One then imagines a random web surfer who at each time step is at some web page, and decides which page to visit on the next step at follows: with probability $1-\varepsilon$, he randomly picks one of the hyperlinks on the current page, and jumps to the web page it links to; with probability ε, he "resets" by jumping to a web page picked uniformly and at random from the collection. Here, ε is a suitable parameter. This process defines a Markov Chain on the web pages, with transition matrix $\varepsilon U+(1-\varepsilon)M$, where U is the transition matrix of uniform transition probability (U_{ij} = 1/n, for all i,j). The vector of PageRank scores is then defined to be the stationary distribution of this Markov Chain, i.e., the left eigenvector of the transition matrix.

Mathematically, BLPR can be computed as follows:

$$(\varepsilon U + (1-\varepsilon)M)^T \mathbf{p} = \mathbf{p} \quad (12)$$

where $\mathbf{p}$ is a vector whose i^{th} element is the block level PageRank of the i^{th} web page.

Note that, in our method, the Markov Chain is induced from the graph model described in previous section. Therefore, the computed block level PageRank can reflect the semantic structure of the web to some extent. For example, those web pages pointed by many advertisement hyperlinks might not be assigned a large importance value since the advertisement hyperlinks are always in the less important blocks. Specifically, those noisy hyperlinks play little role in computing block level PageRanks.

4.2 Block Level HITS

Different from PageRank which assigns only one value to each page, HITS assigns two values to each page (authority value and hub value). Hubs and authorities exhibit a mutually reinforcing relationship.

As we discussed before, there are always multiple semantic regions in one page. Some hyperlinks such as banners, navigation panels, and advertisements in a page do not convey human endorsement. Thus equally mutually reinforcing all the links in a page might not be suitable.

Based on our block level graph of the web presented in previous section, we proposed a Block Level HITS (BLHITS) algorithm. In BLHITS, the authority hub reinforcing idea is the same as the original HITS. The main difference is that in BLHITS, a page will have only authority score and a block will have only hub score.

The authority-hub reinforcing relationship in our block level HITS is described in equation (13). A denotes the vector of authority values for pages and H denotes vector of hub values for blocks. Z is the block to page matrix we discussed in section 3:

$$A = Z^T H \qquad H = ZA \quad (13)$$

Bharat and Henzinger [2] addressed the three problems in original HITS algorithm developed by Kleinberg, i.e. a "links-only" approach: *Mutually Reinforcing Relationships* Between Hosts (where certain arrangements of documents 'conspire" to dominate the computation), *Automatically Generated Links* (where no human's opinion is expressed by the link), and *Non-relevant Documents* (where the graph contains documents not relevant to the query topic). They assign each edge of the graph an authority weight and a hub weight to solve the first problem and combining connectivity and content analysis to solve the latter two.

Chakrabarti et al [7][8] addressed another problem in HITS that regarding the whole page as a hub is not suitable because a page always contains multiple regions in which the hyperlinks point to different topics. They proposed to *dis-aggregate* hubs into coherent regions by segmenting the DOM tree of a HTML page, and showed great improvement over the original HITs. The idea of splitting hub into several coherent regions is similar to our Block-Level HITS method. In Chakrabarti's work, the web-pages were segmented based on the text distribution (consider the similarity between region and query) [8] or Minimum Description Length [7]. It is computed on-line and thus it is time consuming. In our method, we use the VIPS [5] algorithm, which has proved to be very effective in getting a semantic partition of webpage [24]. So all the computation can be done off-line which can greatly speed up the search. Moreover, Chakrabarti's algorithm only considers splitting a mixed hub, and the different hubs (regions) in a page are equally treated. In our Block-Level HITS algorithm, the importance values of different parts of the page are treated differently. Thus, the links in these hubs are treated differently, which can affect the authority-hub reinforcing process.

We list below the main differences between our BLHITS algorithm and the traditional HITS algorithm:

1. The analysis of BLHITS is at block level. It regards the hyperlinks as from blocks to pages, while HITS regards the hyperlinks as from pages to pages.
2. The root set in BLHITS is made up of top ranked blocks rather than top ranked pages in HITS. When a query is submitted to our system, we first retrieve the top ranked pages. The top ranked blocks are then extracted from these top ranked pages. In this step, those noisy blocks (such as advertisement block) are excluded. In our system, all the pages are pre-indexed at block level, so we can directly get the top ranked blocks without any extra computation.
3. When expanding the root set, we only consider the out-links contained in top ranked blocks. HITS expands all the links in the pages, which inevitably introduce noisy pages into the base set. Similarly, we only add those blocks which contain links link to the pages in the root set rather than the whole pages to the root set.
4. B&H [2] proved that combining the connectivity and content analysis (pruning those nodes according the relevance of the node with an expanded query) is very effective in enhancing the performance of HITS. In our BLHITS, the nodes are blocks, so the relevance measure is between blocks and expanded query, which makes more sense.
5. B&H [2] assigns authority weight and hub weight for each edge in the graph to solve the mutually reinforcing relationships problem. But this only occurs when there exists k edges from documents on a first host to a single document on a second host and exists t edges from a single document on a first host to a set of documents on a second host. Besides implementing this heuristics, we extend the idea of authority weight and hub weight of the edge, i.e. we multiply an additional weight (deduced from the importance weight of the block in original page) for each edge. This weight is calculated by the ratio between the importance value of the block containing this link and the maximum block importance value in that page. So this weight is 1 for the most important block in the page. In this way, those links which do not convey author endorsement will have little effect in computation.

5. EXPERIMENTS

In this section, several experiments were performed to compare our proposed block level link analysis algorithms, *i.e.* Block Level PageRank and Block Level HITS, to the traditional PageRank and HITS algorithms.

5.1 Experiment framework

We chose the topic distillation task in web track of TREC 2003 as the benchmark of the algorithms. The data set used in this task is ".GOV", which was crawled from .gov Web sites in the year of 2002. It contains 1,247,753 documents. 1,053,372 of them are text/html files, which were used in our experiments. The task aims to find the key entry page of some topic. Here key entry page must obey the following two rules:

1. It is principally devoted to the topic
2. It is not part of a larger site which is also principally devoted to the topic

For example, for the topic of 'science', the following websites might be considered as key resources:

www.nsf.gov/	National Science Foundation
science.nasa.gov/	Science @ NASA
www.science.gov/	Government Science Portal
www.house.gov/science/welcome.htm	House Committee on Science

But the page 'www.house.gov' fails on the first rule while the page 'www.nsf.gov/home/bio/' fails on the second rule.

So this task is quite different from traditional ad hoc retrieval task in that only relying on the relevance of the content might not work. Link analysis can provide some extra useful information for ranking. This situation is much like the real world web search. So this corpus and queries are very suitable in evaluating different link analysis algorithms.

There are totally 50 queries. The number of relevant pages (based on human judgment) for each query ranged from 1 to 86 with average of 10.32. Among them, 31 queries have less than 10 relevant pages, so the average P@10 is a little bit low.

Only the "title" field of each query is used for retrieval. All the pages in the dataset were partitioned using VIPS and indexed at block level [23], the Page to Block and Block to Page matrix are constructed.

5.1.1 Relevance weighting

In our experiments, each document includes the text of the html page and the anchor texts from other page. And we use BM2500 [20] as the relevance weighting function. It is of the form:

$$\sum_{T \in Q} w \frac{(k_1+1)tf\,(k_3+1)qtf}{(K+tf)(k_3+qtf)} \qquad (14)$$

where Q is a query containing key terms T, tf is the frequency of occurrence of the term within a specific document, qtf is the frequency of the term within the topic from which Q was derived, and w is the Robertson/Sparck Jones weight of T in Q. It is calculated by

$$\log \frac{(r+0.5)/(R-r+0.5)}{(n-r+0.5)/(N-n-R+r+0.5)} \qquad (15)$$

where N is the number of documents in the collection, n is the number of documents containing the term, R is the number of documents relevant to a specific topic, and r is the number of relevant documents containing the term. In our experiment, the R and r are set to zero. In (14), K is calculated by

$$k_1((1-b)+b\times dl/avdl) \qquad (16)$$

where dl and $avdl$ denote the document length and the average document length. In our experiments, we tune the k_1= 4.2, k_3=1000, b = 0.8 to achieve the best baseline (We took the result of using relevance score only as our baseline). The mean average precision is 0.1285 and the P@10 is 0.112. Compared with the best result of TREC2003 participants (with MAP of 0.1543 and P@10 of 0.1280), this baseline is reasonable.

5.1.2 PR and BLPR weighting

To compare to PageRank, we implement the PageRank algorithm based on the link matrix deduced from traditional page level link analysis. And we calculate the BLPR (block level PageRank) based on the link matrix we discussed in section 3.

These two ranks are calculated off-line, and stored for combination with relevance rank.

5.1.3 HITS and BLHITS weighting

HITS algorithm is query dependant, so we can not calculate a unique rank off-line.

For comparison, we implemented the page-level HITS algorithm described in [16], the size of the root set, i.e. t, is 200, the in-link parameter d is 50, and we also discard those intrinsic links (links between pages with the same domain name), which is a very simple heuristic but very effective [16]. Moreover, based on the work of Bharat and Henzinger [2], we modify the basic algorithm on two aspects:

1. We eliminated mutually reinforcing relationship between hosts. That is, if there are k edges from pages on a first website to a single page on a second website, we assign each edge an *authority weight* of *1/k*. If there are t edges from a single document on a first host to a set of documents on a second host, we give each edge a *hub weight* of *1/t*.
2. We combined connectivity and content analysis. After we obtained the base set, we expand the original query using traditional query expansion techniques [10], and then we compute the relevance score of all the documents in the base set with the expanded query, and finally prune the nodes from the neighborhood graph with Median Weight threshold.

More details and effectiveness analysis can be referred to [2]. In our Block Level HITS, we implement the algorithm described in section 4.2.

5.2 Results on PR & BLPR

In this subsection, we compare the PageRank (PR) and Block-Level PageRank (BLPR) algorithms on web search.

5.2.1 Intuitive result

We calculated the PR value and BLPR value for each page offline. Although it is not easy to judge which one is better simply based on these two values, we can get some interesting information from the top ranked pages. Figure 2(a) shows 15 pages with the highest PR value and Figure 2(b) shows 15 pages with the highest BLPR value.

1	G00-05-4074066	http://www.usgs.gov/
2	G00-08-3906771	http://www.nasa.gov/
3	G00-06-0690672	http://www.usda.gov/
4	G00-54-0168623	http://www.usgs.gov/privacy.html
5	G00-09-2318516	http://www.doi.gov/
6	G01-90-0139455	http://www.bnl.gov/bnlweb/security_notice.html
7	G01-08-1822984	http://naca.larc.nasa.gov/readme.html
8	G00-10-2171731	http://www.nws.noaa.gov/
9	G07-50-3922430	http://ar.inel.gov/home.html
10	G01-43-2031819	http://www.usgs.gov/accessibility.html
11	G00-01-3584374	http://firstgov.gov/
12	G00-06-3965004	http://ds.usda.gov/
13	G01-61-3538562	http://www.usgs.gov/mail.html
14	G00-13-0831825	http://ds.usda.gov/h
15	G00-48-1523058	http://www.usgs.gov/disclaimer.html

(a) Top 15 PR pages in .GOV data set

1	G00-05-4074066	http://www.usgs.gov/
2	G00-08-3906771	http://www.nasa.gov/
3	G00-06-0690672	http://www.usda.gov/
4	G00-54-0168623	http://www.usgs.gov/privacy.html
5	G00-10-2171731	http://www.nws.noaa.gov/
6	G00-02-1372443	http://www.nara.gov/
7	G00-02-3781964	http://www.cdc.gov/
8	G00-04-1013476	http://www.ca.gov/
9	G00-04-0880016	http://www.abag.ca.gov/
10	G01-92-0844584	http://www.ca.gov/state/portal/myca_homepage.jsp
11	G00-01-0423383	http://access.wa.gov/
12	G00-09-2318516	http://www.doi.gov/
13	G01-74-0536144	http://my.ca.gov/state/portal/myca_homepage.jsp
14	G00-01-3584374	http://firstgov.gov/
15	G00-00-1711483	http://www.lib.noaa.gov/

(b) Top 15 BLPR pages in .GOV data set

Figure 2: Comparison the top 15 pages in PR list and BLPR list

In the top 15 PageRank list, we can see that it is unreasonable for the six pages (ranked 4, 6, 7, 10, 13 and 15) to get such high ranks. In contrast, in top 15 BLPR list, only 1 page (ranked 4) gets unreasonably high rank. This is because that BLPR treat the different parts of the pages with different importances, so the impacts of the links which do not confer the recommendation (navigational link, advertisement links) are restrained.

5.2.2 Results on TREC2003

In this test, we combined the relevance rank with PageRank (or Block Level PageRank). We chose the top 2000 results according to the relevance score, and then we sorted these 2000 results according to their PR values or BLPR values. Thus we get two ranking lists. One is according to the relevance and the other is according to importance (PR or BLPR). We combine them as follows:

$$\alpha \cdot rank_{relevance}(d) + (1-\alpha) \cdot rank_{importance}(d) \qquad (17)$$

We selected top (with least combined rank value) 1000 results from the combined list for evaluation. The average precision and P@10 variation with alpha are shown in figure 3 and figure 4. All the curves converge to the baseline when $\alpha = 1$.

As can be seen, PR combined with relevance got the highest performance when alpha is 0.94. The average precision is 0.1485 and P@10 is 0.136. BLPR combined with relevance get the highest performance when alpha is 0.92. The average precision is 0.1610 and P@10 is 0.14. In both figures, the BLPR curves are above the PR curves, which means BLPR is always better than PR.

To understand whether these improvements are statistically significant, we performed various t-tests with 95% confidence level. For the P@10 improvement, compared to baseline, both the PR-Combination and BLPR-Combination are significant (p-value is 0.0416 and 0.0279, respectively). For the average precision improvement, compared to baseline, BLPR-Combination is significant (p-value is 0.0406) but PR-Combination fails (p-value is 0.096). All these t-test results show that both PageRank and Block Level PageRank are useful in improving the web search. Moreover, the Block Level PageRank is better than PageRank.

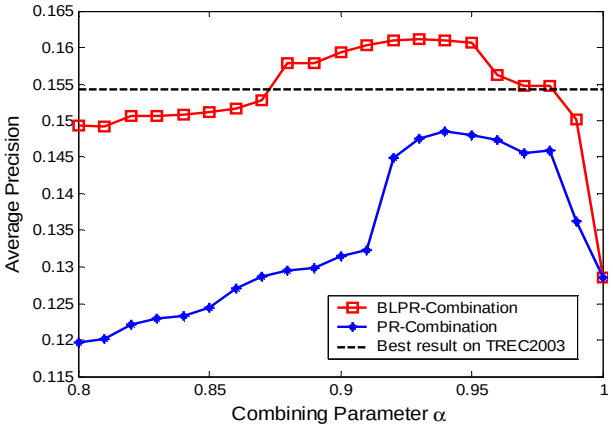

Figure 3: Average precision given alpha for PR-Combination and BLPR-Combination on TREC2003 (The black dashed line denotes the best result on TREC2003)

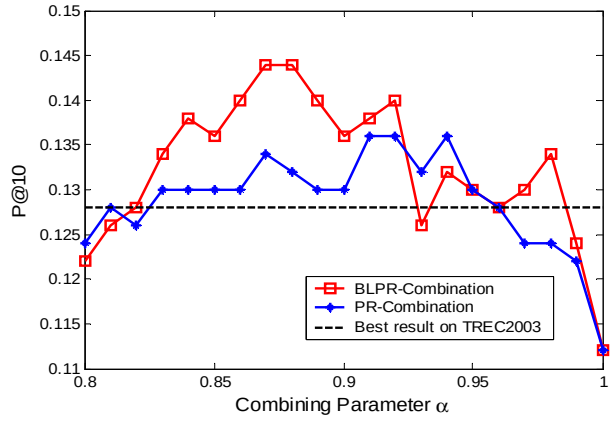

Figure 4: P@10 for PR-Combination and BLPR-Combination on TREC2003 (The black dashed line denotes the best result on TREC2003)

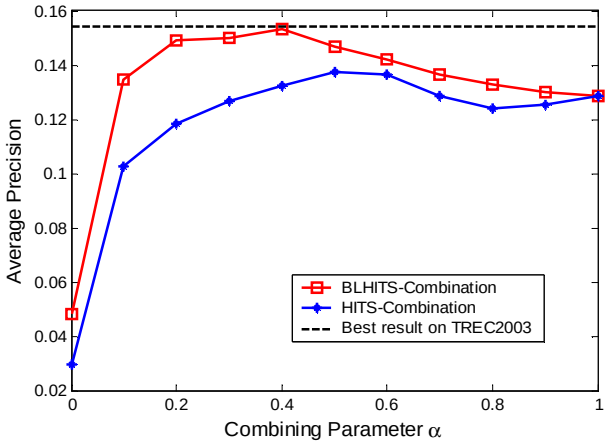

Figure 5: Average precision for HITS-Combination and BLHITS-Combination on TREC2003 (The black dashed line denotes the best result on TREC2003)

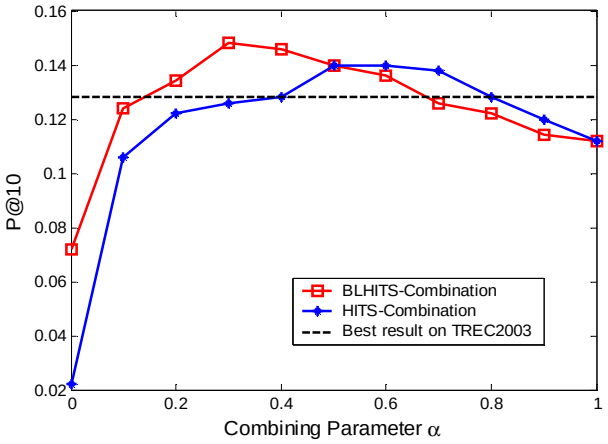

Figure 6: P@10 for HITS-Combination and BLHIT-Combination on TREC2003 (The black dashed line denotes the best result on TREC2003)

5.3 Results on HITS & BLHITS

HITs algorithm is query dependant. So different from PageRank, we can use authority rank of the page to get a result list. And we can also combine the authority rank of a page with relevance rank.

The combination method is the same as equation 17 by replacing the rank of importance with rank of authority. Different from PageRank combination, not all the top ranked 2000 relevant pages have authority rank. For these pages, we ranked them at the bottom of the authority rank list. Similarly, those appeared in authority rank list but not in relevance top 2000 list are ranked at the bottom of the relevance rank list.

The average precision and P@10 variation with alpha are shown in figure 5 and figure 6. All the curves converge to the baseline when $\alpha = 1$, and all the curves converge to the HITS or BLHITS self when $\alpha = 0$.

From these two figures, we can see that either HITS or BLHITS, performed worse than baseline, either on average precision or P@10. When combined with the relevance rank, both HITS-Combination and BLHITS-Combination outperformed baseline which only used relevance rank.

The average precision and P@10 of BLHITS is 0.048 and 0.072, respectively. This result achieved 60% and 227% improvements over HITS on average precision and P@10 (0.030 and 0.022 respectively). The best result of BLHITS-Combination was achieved when alpha is 0.4. (average precision is 0.1533 and P@10 is 0.146), while HITS-Combination achieved its best result when alpha is 0.5 (average precision is 0.1376 and P@10 is 0.14). Based on average precision, as alpha varies, the curve of BLHITS-Combination is always above the HITS-Combination curve. When based on P@10, the peaks of HITS-Combination curve and BLHITS-Combination curve are different. BLHITS-combination achieved its best result with a smaller alpha than HITS-combination. This indicates that BLHITS played more roles in search.

We performed several t-tests with 95% confidence level to see whether these improvements are statistically significant. We first compared HITS and BLHITS directly. Both the average precision and P@10 improvements are significant since the p-value is 0.0158 and 0.00000746 respectively. This indicates that, as to finding authoritative sources, Block Level HITS always outperformed original page level HITS.

When combined with relevance rank, both the HITS-Combination method and BLHITS-Combination method outperformed the baseline. For the P@10 improvement, both of these two combination methods are significant (p-value is 0.0256 for HITS-Combination and 0.0058 for BLHITS-Combination). But for average precision improvement, both of these methods are failed. (p-values are 0.0904 and 0.3259 for BLHITS-Combine and HITS-Combine respectively)

As another kind of link analysis algorithm, HITS (BLHITS) also showed great potential to improve web search. Also, experimental results showed Block Level HITS is better than Page Level HITS.

Table 1. Search results comparison on TREC 2003

Methods	P@10	AvP
Baseline	0.112	0.1285
PageRank Combination	0.136	0.1485
Block Level PageRank Combination	0.14	0.161
HITS	0.022	0.03
Block Level HITS	0.072	0.048
HITS Combination	0.14	0.1376
Block Level HITS Combination	0.146	0.1533

6. CONCLUSIONS AND FUTURE WORK

In this paper, we addressed the problem that a web page always contains multiple semantics while traditional link analysis algorithms ignored this fact. Based on web page segmentation (VIPS) techniques, we treat the web page as a set of blocks and the links are from blocks to pages rather than from pages to pages. From the page to block relationship (page layout analysis) and block to page relationship (link analysis), we can construct a new page to page graph and block to block graph. Based on these new graphs,

we implement Block Level PageRank and Block Level HITS algorithms. Experiments show that Block Level PageRank outperforms PageRank and Block Level HITS outperforms HITS.

Several questions remain to be investigated in our future work.

1. Within our framework of analysis, the block-to-block graph is induced as a by-product. In fact, we can also compute BlockRank from this graph. It is interesting to find out how this rank can help web search.
2. Some advanced block importance models were proposed recently [21], it is interesting to see how block level link analysis algorithm can get benefits from such models.

7. ACKNOWLEDGMENTS

We are grateful to Kaihua Zhu for the implementation of Block Level HITS algorithm and many valuable discussions.

8. REFERENCES

[1] B. Amento, L. Terveen, and W. Hill. Does "authority" mean quality? predicting expert quality ratings of web documents. In Proc. *ACM SIGIR 2000*, pages 296--303.

[2] K. Bharat and M. R. Henzinger. Improved algorithms for topic distillation in a hyperlinked environment. In Proceedings of *the ACM-SIGIR*, 1998.

[3] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine", In *The Seventh International World Wide Web Conference*, 1998.

[4] D. Cai, S. Yu, J.-R. Wen, and W.-Y. Ma, Extracting content structure for web pages based on visual representation, *Proc.5th Asia Pacific Web Conference*, Xi'an China, 2003.

[5] D. Cai, S. Yu, J.-R. Wen, and W.-Y. Ma, VIPS: a vision-based page segmentation algorithm, *Microsoft Technical Report*, MSR-TR-2003-79, 2003.

[6] 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 Int. World Wide Web Conference*, May 1998.

[7] S. Chakrabarti, Integrating the Document Object Model with hyperlinks for enhanced topic distillation and information extraction, In the *10th International World Wide Web Conference*, 2001.

[8] 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.

[9] Brian D. Davison. Recognizing nepotistic links on the Web. In Artificial Intelligence for Web Search, pages 23--28. AAAI Press, July 2000.

[10] N. E. Efthimiadis, Query Expansion, In *Annual Review of Information Systems and Technology*, Vol. 31, 1996, pp. 121-187.

[11] G. Flake, S. Lawrence, L. Giles, and F. Coetzee, Self-organization and identification of web communities, *IEEE Computer*, pp. 66-71, 2002.

[12] D. Gibson, J. Kleinberg, and P. Raghavan. Inferring web communities from link topology. In Proceedings of *the 9th ACM Conference on Hypertext and Hypermedia (HYPER-98)*, pages 225--234, New York, June 20--24 1998. ACM Press.

[13] T.H. Haveliwala. Topic-sensitive pagerank. In Proc. of *the 11th Int. World Wide Web Conference*, May 2002.

[14] N. Jushmerick. Learning to remove Intemet advertisements. Proc. of 3 rd International Conf. On Autonomous Agents, 1999

[15] H. Kao, S. Lin, J. Ho and M. Chen, Entropy-Based Link Analysis for Mining Web Informative Structures. *CIKM'02*, 2002.

[16] J. Kleinberg, Authoritative sources in a hyperlinked environment, *Journal of the ACM*, Vol. 46, No. 5, pp. 604-622, 1999.

[17] R. Lempel and S. Moran, The stochastic approach for link-structure analysis (SALSA) and the TKC effect, Proc. *9th International World Wide Web Conference*, 2000.

[18] Joel C. Miller, Gregory Rae, Fred Schaefer. Modifications of Kleinberg's HITS algorithms Using Matrix Exponentiation and Web Log Records, in: Proc. of *the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2001.

[19] 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.

[20] S. E. Robertson, Overview of the okapi projects, *Journal of Documentation*, Vol. 53, No. 1, 1997, pp. 3-7.

[21] R. Song, H. Liu, J. R. Wen, W. Y. Ma, "Learning Block Importance Models for Web Pages," *Proc. 13th World Wide Web Conference*, New York, 2004.

[22] V. Vapnik, The nature of statistical learning theory, Springer, New York, 1995.

[23] J. R. Wen, R. Song, D. Cai, K. Zhu, S. Yu, S. Ye, and W.-Y. Ma, Microsoft Research Asia at the web track of TREC 2003, in *the twelfth Text Retrieval Conference* (TREC 2003), 2003.

[24] S. Yu, D. Cai, J.-R. Wen, and W.-Y. Ma, Improving pseudo-relevance feedback in web information retrieval using web page segmentation, *Proc. 12th World Wide Web Conference*, Budapest, Hungary, 2003.