ABSTRACT

The purpose of this project was to improve the Hypertext-Induced Topic Selection (HITS)-based algorithms on Web documents. The HITS algorithm is a very popular and effective algorithm to rank Web documents based on the link information among a set of documents.

In this project, to improve the HITS algorithm, the base document is collected from multiple search engines, and different weights are assigned adaptively depending on the rankings from Web search engines, expert agreement, and content analysis.
# TABLE OF CONTENTS

Abstract ............................................................................................................................... ii  
Table of Contents ............................................................................................................... iii  
List of Figures .................................................................................................................... iv  
1. Introduction and Background ..................................................................................... 1  
   1.1 Previous Related Research .................................................................................. 2  
2. Improvement of Weighted HITS-based Algorithm ...................................................... 8  
3. Research ...................................................................................................................... 10  
   3.1 New Weighted HITS-Based Algorithm .................................................................. 10  
   3.2 Comparing the Performance of the HITS-based Algorithms ............................... 13  
4. Evaluation and Results ............................................................................................... 16  
5. Future Work ............................................................................................................... 26  
6. Conclusion ................................................................................................................. 26  
Bibliography and References ......................................................................................... 28
LIST OF FIGURES

Figure 1.1. The Base Set ................................................................. 3
Figure 1.2. The Average relevance scores of 28 queries computed by BHITS and WBHITS ......................................................... 5
Figure 1.3. The average Relevance Scores of 28 Queries Computed by Google and LinbaCrawler .................................................. 6
Figure 2.1. The Outline of This Project ........................................... 9
Figure 3.1. Example Text for the Cover Set ..................................... 11
Figure 4.1. RootLink Array Subscript Structures ............................. 17
Figure 4.2. BackLink Array Subscript Structures ............................. 17
Figure 4.3. Simplified Link Scenario in the Base Set ....................... 18
Figure 4.4. Expanded Links Scenario in the Base Set ..................... 18
Figure 4.5. HITS-matrix which Represents Figure 4.3 .................... 19
Figure 4.6. HITS-matrix which Represents Figure 4.4 .................... 20
Figure 4.7. User Interface for Query “Parallel Architecture” ............ 21
Figure 4.8. The Average Relevance Authority Scores of 6 Queries Computed by HITS, WHITS, and Google .................................... 22
Figure 4.9. The Average Relevance Hub Scores of 6 Queries Computed by HITS, and WHITS ......................................................... 23
1. INTRODUCTION AND BACKGROUND

For information retrieval from the Internet, using a search engine is one of the most common tasks performed by computer users among the general public or computer scientists. However, with the quick growth rate of the Internet, most search engines have several limitations. First, each of the search engines can only index a fraction of all the available Web pages, for reasons such as robot exclusions or a lack of connection links. Secondly, those search engines have difficulty keeping their indexed information updated. Furthermore, the quality of documents returned from different search engines may differ widely for the same query because of the different index methods, different coverage, and different ranking algorithms. A meta search approach has the potential to provide improved search capability for users. Meta search engines merge the returned lists from different search engines, and present them to the user in the order of relevance. Meta search engines can provide larger overall Web coverage and better overall document rankings than a single search engine can. Meta search engines will enhance the effectiveness of information retrieval on the Internet.

The most important decision made by a meta search engine is how to order the results retrieved from multiple search engines. The order of the documents is critical because users tend to examine only the top 10 or 20 documents and rarely read more than the top 50 documents. Analyzing the hyperlink structure among WWW pages gives us a way to rank the documents as well as helps to find quality documents related to the topic of the query.
1.1 Previous Related Research

The network structure of a hyperlinked environment can be a rich source of information about the content of the environment, provided we have an effective means for understanding it [Kleinberg 1997]. Kleinberg developed the hypertext-induced topic selection (HITS) algorithm which extracts information from the link structure. It is a very popular and effective algorithm to rank Web documents based on the link information among a set of documents. The underlying assumption of the HITS algorithm is that a Web page serves two purposes: to provide information and to provide links relevant to a topic. This gives two ways to categorize a Web page. A Web page is an authority on a topic if it provides good information, and it is a hub if it provides links to good authorities. The algorithm is based on the relationship that exists between authorities and hubs. When a Web document links to another Web document, there is good reason for the hyperlink. Web documents referenced most frequently are considered ‘better’ authorities and therefore more important. Hubs and authorities exhibit a mutually reinforcing relationship: a better hub points to many good authorities, and a better authority is pointed to by many good hubs.

Figure 1.1 shows how to build a base document set. A root set of Web documents is selected from Web search engines based on a query, then the root set is expanded to the base set which contains the neighborhood Web documents that either point to Web documents in the root set (back pages), or are pointed to by Web documents in the root set (forward pages).
A root set, $R$, is fetched from a search engine; then, for each document $r \in R$, a set of documents $L$ that point to $r$ and another set of documents $L'$ that are pointed to by $r$ are added to the set, $I$, as $R$’s neighborhood. For a document $i \in I$, let $a_i$ and $h_i$ be the authority score and hub score respectively. To begin the algorithm, $a_i$ and $h_i$ are initialized to 1. While the values have not converged, the algorithm iteratively proceeds as follows [Kleinberg 1997]:

1. For all $i' \in I$ which points to $i$,

\[
a_i = \sum_{i'} h_{i'}.
\] (1.1)
2. For all $i' \in I$ which is pointed to by $i$, 

$$h_i = \sum_{i'} a_i$$  \hspace{1cm} (1.2) 

3. Normalize $a_i$ and $h_i$ values so that 

$$\sum_i a_i = \sum_i h_i = 1.$$  \hspace{1cm} (1.3) 

Kleinberg showed that the algorithm will eventually converge, but the bound on the number of iterations is unknown. In practice, the algorithm converges quickly [Kleinberg 1997].

Documents that have high authority scores are expected to have relevant content, whereas documents with high hub scores are expected to contain URL links to relevant content.

Since Kleinberg’s algorithm is only based on linkage information, it will not return the right results in some cases. For example, Bharat identified that mutually reinforcing relationships between hosts give undesirable weight to the opinion of a single person [Bharat 1998]. In some cases, a set of documents on one host point to a single document on a second host. This drives up the hub scores of the documents on the first host and the authority score of the documents on the second host.

The problem can be solved by assigning smaller weights to the documents from the same host. Bharat improved Kleinberg’s HITS algorithm by giving a document an authority weight of $1/k$ if there are $k$ documents on a first host which link to a single document on a second host, and a hub weight of $1/l$ of there are $l$ links from the document on the first host to a set of documents on a second host. With this improvement, the Eq.(1.1) and Eq.(1.2) are modified as follows [Bharat 1998]:
1. For all \( i' \in I \) which points to \( i \),

\[
a_i = \sum_{i'} h_{i'} \cdot (\text{auth}_\text{wt of } a_i)
\]  \hspace{1cm} (1.4)

2. For all \( i' \in I \) which is pointed to by \( i \),

\[
h_i = \sum_{i'} a_{i'} \cdot (\text{hub}_\text{wt of } h_i)
\]  \hspace{1cm} (1.5)

Bharat showed that the simple modification of the HITS algorithm achieved remarkably better precision, while further precision can be obtained by adding content analysis.

Although Bharat’s improved HITS (BHITS) generally works well, there are cases where it generates bad results. Li identified that if a root page has few in-links and a large number of out-links which point to documents on different hosts, the results based on BHTIS are not satisfactory [Li 2002]. Bharat only considered to give lesser weights to the documents from same host, but usual Web pages with small in-links and large out-links point to documents on different hosts. By preventing BHITS form converging to small-in-large-out Web documents, Li improved BHITS algorithm.

![Figure 1.2 The Average Relevance Scores of 28 queries computed by BHITS and WBHITS [Li 2002]](image-url)
The experimental results show that Li’s weighted HITS-based Algorithm (WBHITS) outperforms BHITS in terms of the average relevance score on 28 queries (see Figure 1.2). The relevance scores were scaled between 0 and 10, with 0 representing the most irrelevant and 10 most relevant.

However, in Li’s previous work [Li 2002], the weights are a fixed number when the link spam was found in the base set, and it may not generate the best results.

Also, Li developed a new type of Meta search engine, LinbaCrawler, which returns not only authority documents, which are relevant to the query of interest, but also hub documents, which may cover different subtopics of the query [Li 2004].

Figure 1.3 shows that LinbaCrawler can find better results than Google in general. The relevance score was scaled between 0 and 10, with 0 representing the most irrelevant and 10 most relevant. But LinbaCrawler did not consider link spam and forward documents, and can be further improved.

Figure 1.3 The average relevance scores of 28 queries computed by Google and LinbaCrawler [Li 2004]
In this project, to improve the HITS algorithm, the base document set will be collected from multiple search engines, and different weights will be assigned adaptively depending on the rankings from the Web search engines, expert agreement, and content analysis.
2. IMPROVEMENT OF WEIGHTED HITS-BASED ALGORITHM

As stated previously, this research project seeks to improve the HITS-based algorithms’ precision by assigning the adaptive weight on Web pages in the base set. The scope of this project can be narrowed down to the following two steps:

1. Improving the HITS-based algorithm.

2. Comparing the performance of the different algorithms.

2.1 Improving HITS-based algorithm

This project improves the HITS-based algorithms by assigning different weights to the document depending on the ranking from the Web search engines, expert agreement, and content analysis. Since several search engines are used to form the base set, a way to combine the results from the different search engines is required. Expert agreement can be one of those solutions [Oztekin 2002]. For the content analysis, cover density ranking (CDR) is used to produce more relevant results [Clarke 2000].

2.2 The way to compare the different algorithms

The performance of the new weighted HITS-based algorithm has been statistically compared to the performance of the current HITS-based algorithms. The evaluation metrics employed in experiments is probability of win ($P_{\text{win}}$), which has been applied to evaluate heuristic methods in machine learning [Wah 1995]. Details of the probability of win can be found in section 3.2. It measures in a domain-independent fashion whether one re-ranking algorithm performs statistically better than another.

The outline of this project is shown in Figure 2.1. The queries are sent to multiple search engines. The retrieved links from these search engines form the root set. The Web
documents in the root set can be pointed to by another Web document and can contain links which point to other Web documents.

The root set is expanded to form the base set which included the root set and the Web documents which point to the documents in the root set and pointed to by the documents in the root. Each document in the base set has an authority score value and a hub score value. Those scores are calculated by the new weighted HITS-based algorithm. The documents with high authority scores contain the relevant topics for the query. The documents with high hub scores point to the documents which contain the relevant topics for the query.

**Figure 2.1 The Outline of This Project**

The root set is expanded to form the base set which included the root set and the Web documents which point to the documents in the root set and pointed to by the documents in the root. Each document in the base set has an authority score value and a hub score value. Those scores are calculated by the new weighted HITS-based algorithm. The documents with high authority scores contain the relevant topics for the query. The documents with high hub scores point to the documents which contain the relevant topics for the query.
3. RESEARCH

3.1 New Weighted HITS-Based Algorithm

By applying the expert agreement scores ($e$), and CDR scores ($c$), the Eq.(1.4) and Eq.(1.5) are modified as follows:

1. For all $i' \in I$ which points to $i$,

$$a_i = (1 - \alpha) + \sum_{i'} h_{i'} \cdot c_{i'} \cdot e_i \cdot \alpha$$  \hspace{1cm} (3.2)

2. For all $i' \in I$ which is pointed to by $i$,

$$h_i = (1 - \alpha) + \sum_{i'} a_{i'} \cdot c_{i'} \cdot e_i \cdot \alpha$$  \hspace{1cm} (3.3)

In Eq.(3.2), $(1 - \alpha)$ is the estimated hub score of unseen documents, which point to document $i$. A base set document may be pointed to by or point to many documents that are not included in the base set. In this algorithm, those unseen documents are counted by assigning a damping factor $\alpha$, $0 \leq \alpha \leq 1$.

3.1.1 Expert Agreement

The order of the links returned by the search engines plays an important role in deciding the relevance level of the link to the query topic. A link which is on the high ranking on the query result will have more possibilities of containing more relevant topics for the query than the links ranked very low. Since multiple search engines were used to obtain the base set, the way of ranking the links becomes more complicated. There are two ways to rank the links from multiple search engines:

1. Only consider the best rank and ignore the other results from the other search engines.
2. Combine all the results from several search engines.

The first method can be implemented easily, but a document occurring in several search engines can be more relevant or important than the ones occurring in just one engine at similar ranks [Oztekin 2002]. For instance, if a link has third, second, second, and third ranks in four different search engines, respectively, it may be a better link than the one that has first or second rank in one search engine only. To improve the ranking of this type of document, the “Agreement” scheme that is described below can be implemented [Oztekin 2002].

Expert agreement scores are calculated by simply adding the reciprocals of the display positions for each document. Document A, which has the 3rd and 4th display positions in two different search engines respectively, may be a more relevant document than document B that has the 2nd display position in one engine only. Because the expert agreement score of A is $\frac{1}{3} + \frac{1}{4} = \frac{7}{12}$, while that of B is $\frac{1}{2}$.

3.1.2 Cover Density Ranking (CDR)

CDR [Clarke 2000] was developed to better meet users’ expectations – a document containing most or all of the query terms should be ranked higher than a document containing fewer terms, regardless of the frequency of term occurrence. The following (Figure 3.1) illustrates the concept of a cover set:

```
The Master of Science with a major in Computer Science is designed to prepare graduate professionals who can apply the necessary knowledge of computing to information requirements of organizations in business, government, industry and education. [TAMUCC 2004]
```

Figure 3.1 Example Text for the Cover set
In Figure 3.1, superscripts indicate term positions. Each term set,

\[ T_1 = \{ \text{“master”, “science”} \} , \]
\[ T_2 = \{ \text{“master”, “education”} \} \]

has the cover set,

\[ W_1 = \{ (2, 4) \} , \]
\[ W_2 = \{ (2, 36) \} . \]

In CDR, the results of phrase queries are ranked in the following two steps [Clarke 2000]:

1. Documents containing one or more query terms are ranked by coordination level, i.e., a document with a larger number of distinct query terms ranks higher. The documents are thus sorted into groups according to the number of distinct query terms each contains, with the initial ranking given to each document based on the group in which it appears.

2. The documents at each coordination level are ranked to produce the overall ranking. The score of the cover set \( w = \{ (p_1, q_1), (p_2, q_2), \ldots, (p_n, q_n) \} \) is calculated as follows:

\[
    S(w) = \sum_{j=1}^{n} I(p_j, q_j) \quad \text{and}
\]

\[
    I(p_j, q_j) = \begin{cases} 
        \frac{\lambda}{q_j - p_j + 1} & \text{if } q_j - p_j + 1 > \lambda, \\
        1 & \text{otherwise}. 
    \end{cases}
\]

Where \((p_j, q_j)\) is an ordered pair over a document, called cover, specifying the shortest interval of two distinct terms in the document [Clarke 2000]. \( p_j \) is the position of one
term, \( q_j \) is the position of another term, and \( q_j \) is assumed to be larger than \( p_j \). \( \lambda \) is a constant and is set to 4 in this experiment because usually the snippet of text for each document is pretty short, Covers of length \( \lambda \) or shorter are given score 1, and longer covers are assigned scores less than 1 in proportion to the inverse of their lengths.

By applying Eq.(3.1) to the example text in Figure 3.1, the score of the cover set \( T_1 \) becomes 1 and the score of the cover set \( T_2 \) becomes \( \frac{4}{35} \). As a result, the example text in Figure 3.1 will have almost 9 times more relevance score for \( T_1 \) than \( T_2 \).

To adapt CDR to the Web document, we need to find out how many distinct query terms a document has and rank the documents with more distinct terms higher. This version of the CDR method will compute the relevance scores of documents in two steps:

1. Documents are scored according to the regular CDR method. Each document belongs to a coordination level group and has a score within that group.

2. The scores are normalized to range \((0, 1]\) for documents containing only one term, to range \((1, 2]\) for documents containing two different terms, and so on, so forth.

The benefit of this method is that it not only considers the number of distinct terms in a document, but also how these distinct terms appeared in the document, such as how close they are.

### 3.2 Comparing the Performance of the HITS-based algorithms

The method of comparing different algorithms’ performance is the same one as that used in [Li 2004]. To compare re-ranking algorithms bearing different performance distributions across different kinds of queries, a performance metric that is independent
of the actual distributions is needed. The statistical metric, probability of win \( P_{\text{win}} \), measures statistically how much better (or worse) the sample mean of one hypothesis, \( \mu_1 \), is as compared to that of another, \( \mu_2 \) [Wah 1995]. It resembles the significance level in general hypothesis testing, but there are two major differences. First, only one hypothesis \{H: \mu_1 > \mu_2 \} is specified, without the alternative hypothesis. Further, in contrast to hypothesis testing, acceptance confidence is not given in advance but is evaluated based on sample values. The advantage of \( P_{\text{win}} \) is that it considers both the mean and the variance of the performance data.

Under the assumption that the performance values of re-ranking algorithms are independent and normally distributed, two re-ranking algorithms, \( R_1 \) and \( R_2 \), can be compared based on the probability of win. First, we calculate the difference of performance values between the two algorithms, \( P_1 - P_2 \), on the \( N \) sample queries. In comparing precision of re-ranking algorithms, the performance values are relevance scores.

Assuming that the sample mean of \( P_1 - P_2 \) is \( \hat{\mu} \) and the sample variance is \( \hat{\sigma}^2 \), then \( P_{\text{win}} \) is defined as follows:

\[
P_{\text{win}} = F_{t} \left( N-1, \frac{\hat{\mu}}{\sqrt{\hat{\sigma}^2/N}} \right)
\]

Where \( F_{t} (\nu, x) \) is the cumulative distribution function of Student’s \( t \)-distribution with \( \nu \) degrees of freedom. \( P_{\text{win}} \), with a value in \([0,1]\), is the probability that the true
performance (population mean) of $R_1$ is better than that of $R_2$. The closer the $P_{\text{win}}$ value approaches 1 (or 0), the higher the confidence that $\mu_1$ is better than $\mu_2$.

When $N \to \infty$, we have

$$P_{\text{win}} \approx \Phi \left( \frac{\hat{\mu}}{\sqrt{\hat{\sigma}^2 / N}} \right)$$ (3.5)

Where $\Phi$ is the standard cumulative normal distribution function [Devore 1982].
4. EVALUATIONS AND RESULTS

To test the new algorithm, three search engines were used. Those were Google [Google 2005], MSN [MSN 2005], and Yahoo! [Yahoo 2005]. All of these search engines are very popular with the general public. Each search engine provides the service to retrieve the documents which have the link back to the original search result. Also, each search engine has their own robot-crawled Web document database.

The queries are selected from the previous research [Bharat 1998], [Li 2002], and [Li 2004]. They are (1)vintage car, (2)Zen Buddhism, (3)rock climbing, (4)Thailand tourism, (5)gulf war, and (6)table tennis. Each query was sent to each of three search engines in parallel by using threads. The top 30 Web documents were selected from each search engine’s query results to form a root set. The URL, and snippet content from the search engine were saved in two dimensional arrays, rootLink[i][j] and rootContent[i][j].

The crawler visits a Web page, reads it, and then follows links to other pages within the site. The full html content of each Web document was retrieved and saved in a two dimensional array, rootContentF[i][j]. After that, the root set was expanded to the base set by adding the back Web pages and forward Web pages.

To obtain the forward page set, the content of a Web page which is in the root set was scanned and the Web links were extracted and saved as forward links in a three dimensional array, forwardLink[i][j][k]. The back link documents (which have links to the document in the root set) were gained through search engines. Those are saved in a three dimensional array, backLink[i][j][k]. Each link was crawled and the full html content was saved in the array and it was used for content analysis. The link information was used to calculate the expert agreement scores.
CDR scores and Expert agreement scores were combined and the result was used as weight for the document.

In the previous research [Li 2004], the relationships of backlink and rootlink were gained from the array structure’s subscript. The first subscript of the rootLink array represents which search engine was used to obtain it. The second subscript of the rootLink array shows the search engine results rank. (see Figure 4.1)

![Figure 4.1 RootLink Array Subscript Structures](image1)

In the backLink array structure, the first two subscripts have same meaning as with the rootLink array’s two subscripts. (see Figure 4.2)

![Figure 4.2 BackLink Array Subscript Structures](image2)
Among Web pages in the base set, if a backLink array’s first two subscripts match the rootLink array’s two subscripts, the backLink Web pages have a link which point to the rootLink Web page. (see Figure 4.3)

But this approach has a limitation. It can not track the relationships among the backlink Web pages and forwardlink Web pages. For example, in Figure 4.3, if the Web page #8 has a link which points to the Web page #7, the relationship between the Web page #7 and the Web page #8 can not be tracked by the array subscript structure. Figure 4.3 Simplified Links Scenario in the Base Set become Figure 4.4 Expanded Links Scenario in the Base Set, when the link relationship among backLink and forwardLink Web pages was considered. In Figure 4.4, the backLink Web page #5 is linked to another backLink Web page #6, and the forwardLink Web page #9 is link to the backLink Web page #6.

Figure 4.3 Simplified Links Scenario in the Base Set
The method which is using array structure’s subscript can not reflect the complex real WWW environment. To solve this problem, the HITS-matrix was designed. (see Figure 4.5 and Figure 4.6 ) The following is the pseudo code for how to build the HITS-matrix.

```plaintext
float[][ ] matrixH = new float[totalLinkNum][totalLinkNum];

for( int x=0; x < totalLinkNum ; x++)
{
    for (int y=0; y < totalLinkNum ; y++)
    {
        if ( WebPageFullText[x].indexOf(WebPageURL[y]) != -1)
            matrixH[x][y] = 1;
        else
            matrixH[x][y] = 0;
    }
}
```
Full html contents for each Web page in the base set were saved in an array, WebPageFullText[]. Each Web page’s URL in the base set was saved in an array, WebPageURL[]. Each Web document’s full html content was scanned for matching all links saved in the base set by using indexOf() method in Java's String class. If the link exists in the Web document, the value 1 was assigned.

In the HITS-matrix, Each Web document in the base set was indexed vertically in two dimensional arrays. The link information of each Web document in the base set was indexed horizontally in two dimensional arrays. The value 1 indicates that the Web link appears in the corresponding Web page.

![Figure 4.5 HITS-matrix which represents Figure 4.3](image)

Figure 4.5 HITS-matrix which represents Figure 4.3
In Figure 4.5, the HITS-matrix was constructed by using the link information from Figure 4.3. The HITS-matrix, in Figure 4.6, was formed by using the link information from Figure 4.4. This shows the rank of relevance score can be changed when the links among backLink Web pages and forwardLink Web pages are considered. Since the real WWW structure is complex, Figure 4.4 and 4.6 are much better representative than Figure 4.3 and 4.5. The weight score which were calculated by the CDR score and expert agreement is multiplied to the corresponding HITS matrix value to gain the authority and hub score.

The merged results are displayed in a two column table with authority documents in the left and hub documents in the right. Each document consists of a URL, a title and a short summary. The documents are presented to the user in sorted order with the best one at the top. A typical user interface is shown in Figure 4.7 for query “parallel architecture”.

Figure 4.6 HITS-matrix which represents Figure 4.4
To compare the performance of the new algorithm, two query result pools were built. One is formed by the top 10 authority documents and the second one by the top 10 hub documents of 6 queries. Then, evaluators personally visited the two query result pools, and manually scored them in a scale between 0 and 10, with 0 representing the most irrelevant and 10 most relevant. An authority document received a high score if it contained both useful and comprehensive information about the query. A hub page was given a high score if it had many links which lead to a relevant message because evaluators were encouraged to follow outgoing link and browse a page’s neighborhood. The broken links and the documents that were written in languages evaluators did not understand were skipped. The average of six scores was taken as the final score for a document and the average of 10 authority or hub documents’ scores as the final evaluation score of a query.
The average relevance authority scores of the HITS-based algorithm, the WHITS-based algorithm, and Google on 6 queries are plotted in Figure 4.8, which demonstrates that the WHITS-based algorithm outperformed both the HITS algorithm and Google. The average relevance hub scores of the HITS-based algorithm and the WHITS-based algorithm on 6 queries are plotted in Figure 4.9, which demonstrates that the WHITS-based algorithm improves the HITS-based algorithm in the hub documents area also.

Figure 4.8 The Average Relevance Authority Scores of 6 queries computed by HITS, WHITS, and Google
Figure 4.9 The Average Relevance Hub Scores of 6 queries computed by HITS, and WHITS

Based on the scores of authority and the score of hub on 6 queries, the performance of three methods, the HITS-base algorithm, the WHITS-based algorithm, and Google were compared by using statistical metrics: probability of win ($P_{\text{win}}$). A $P_{\text{win}}$ value larger than 0.5 means a higher chance to be better, so the algorithm with $P_{\text{win}}$ values larger than 0.5 against any other methods is the best one. The results are shown in the table 4.1 and table 4.2. In table 4.1, the values in the first row are the statistical comparison results of the HITS-based algorithm against other methods including itself. It shows the HITS-based algorithm is better than Google, but worse than the WHITS-based algorithm. The WHITS-based algorithm is the better than Google and slightly better than the HITS-based algorithm. As a conclusion, the WHITS-based algorithm is the best among the three methods.
Table 4.1 Statistical Performance Comparison of Evaluation Scores of Authority documents of HITS, WHITS, and Google search engine

<table>
<thead>
<tr>
<th></th>
<th>HITS</th>
<th>WHITS</th>
<th>Google</th>
</tr>
</thead>
<tbody>
<tr>
<td>HITS</td>
<td>0.50</td>
<td>0.39</td>
<td>0.99</td>
</tr>
<tr>
<td>WHITS</td>
<td>0.61</td>
<td>0.50</td>
<td>0.99</td>
</tr>
<tr>
<td>Google</td>
<td>0.01</td>
<td>0.01</td>
<td>0.50</td>
</tr>
</tbody>
</table>

Table 4.2 Statistical Performance Comparison of Evaluation Scores of Hub documents of HITS, and WHITS

<table>
<thead>
<tr>
<th></th>
<th>HITS</th>
<th>WHITS</th>
</tr>
</thead>
<tbody>
<tr>
<td>HITS</td>
<td>0.50</td>
<td>0.38</td>
</tr>
<tr>
<td>WHITS</td>
<td>0.62</td>
<td>0.50</td>
</tr>
</tbody>
</table>

In the Table 4.2, statistical performance comparison of evaluation scores of Hub documents of the HITS-based algorithm, and the WHITS-based algorithm shows the WHITS-based algorithm is better than the HITS-based algorithm. It means the results from the WHITS-based algorithm will contain the more relevant information than the HITS-based algorithm.
5. FUTURE WORK

In this project, the experimental results support the hypothesis that by assigning adaptive weight to the HITS-base algorithm, the search results become more relevant to the query topic.

However, the algorithm’s improvement can be measured by accuracy and efficiency. In this project, a new algorithm returns more relevant documents at high ranking. Therefore, the accuracy of the algorithm was obtained. But, it took three or four hours to get the results of a query. Thus, the efficiency of the algorithm is problematic. The next steps of this project must be the implementation for increasing the efficiency of the WHITS-based algorithm.
6. CONCLUSION

The HITS algorithm is a very popular and effective algorithm to rank Web documents based on the link information among a set of documents. The algorithm presumes that a good hub is a document that points to many others, and a good authority is a document that many documents point to. Hubs and authorities exhibit a mutually reinforcing relationship: a better hub points to many good authorities, and a better authority is pointed to by many good hubs. To run the algorithm, a base set was collected from several search engines’ query results and their neighbor documents. Each document in the base set has an authority score and hub score. Those scores were calculated by the new weighted HITS-based algorithm. The documents with high authority scores contain the more relevant topics for the query. The documents with high hub scores point to the documents which contain the relevant topics for the query.

In this project, by adopting cover density ranking [Clarke 2000] and expert agreement [Oztekin 2002], the HITS-based algorithm was improved with the new authority weight and hub weight.
BIBLIOGRAPHY AND REFERENCES


