Adaptive Hierarchical Surrogate for Searching Web with Mobile Devices

Wookey Lee, Sanggil Kang, Seungkil Lim, Myong-Keun Shin, and Young-Kuk Kim* Member, IEEE

Abstract — This paper proposes a new web-page search mechanism suitable for mobile devices, called an adaptive hypermedia search. It utilizes hierarchical structure of hypermedia objects for handheld devices, such as cellular phones and PDAs, which have usually limited resources. We developed a tree-filtering algorithm and a Top_K algorithm that can be used to provide search recommendations for mobile devices. In the experimental section, we implement our system in Windows Mobile 5.0 SDK environment and show that our method can save mobile resource in terms of web-page search time. Also, we show the resource savings according to different wireless technologies such as WiBro, HSDPA, and Wi-Fi1.

Index Terms — Hypermedia, Web Information Retrieval, Mobile Devices, PDA, Top-k algorithm.

I. INTRODUCTION

Mobile access to web-pages with small handheld computers such as cellular phones and PDAs has been at the center of developing issues in mobile environment since early twenty-first century. Especially, a handheld device equipped with a browser and a wireless connection can enable to utilize Internet at anytime, anywhere [1, 2]. However, low bandwidth, small storage capacity, limited battery life, and slow CPU speed has been serious obstacle for the development of searching web-pages through mobile devices.

Traditional search engines commonly used in desktop environment have usually ignored the hyperlinks information even though considering the number of hyperlinks to derive ranks for web hypermedia objects when providing search results, any information on relevance or linkage among web hypermedia objects of the web site is lost. In considering the limitations of resource capabilities, it is indispensable for mobile devices to be supported by a structural surrogate. By using the structure, the limitation of mobile resource can be solved to some extent.

This paper focuses on devising a new web search mechanism for the mobile devices by utilizing the hierarchical structure of a web site. We call this type of search a structured web search, whose results or search recommendations from the structured web search contain hyperlinks information on web hypermedia objects not simply listing web hypermedia objects. These systems are able to provide adaptive navigation and even adaptive contents which may be changed according to some kind of user and service model. The adaptation can filter out unnecessary hyperlinks to a web-page according to its hierarchical structure, which can play an importance role on relieving mobile resource limitation.

In the experimental section, we implement our system in Windows Mobile 5.0 SDK environment and show that our method can save mobile resource in terms of web-page search speed. Also, we show the resource saving according to different wireless technologies such as WiBro (IEEE 802.16e, mobile WiMAX), HSDPA [13, 14], and Wi-Fi [15].

This paper is organized as follows. Section II introduces background and related work. Section III describes overall architecture of the proposed system. In Section IV, we implement and demonstrate the mobile web search engine. Also, we show the performance of our system by comparing with conventional web search engine. Then we summarize research contributions with some concluding remarks in Section V.

II. BACKGROUND AND RELATED WORK

Assume that a user starts visiting an example web site, given in Fig. 1, by selecting a web object, A.html, from a list of URLs (Uniform Resource Locators) given by a search engine. Note that the rectangles and the arrows in the figure represent web objects and hyperlinks, respectively. If the user perceives the selected web object (that is, A.html) does not contain relevant contents to his or her search interests and/or the user needs to crawl within the web site, the user may move to one of the linked web objects to the selected web object (in this example, A1.html or A2.html). Otherwise the user may use some supporting tools available in this web site to find the relevant web contents. Most frequently used tools include a search engine (called intranet search engine) and a site map inside the web site.

Although the user can submit a similar search query again to the intranet search engine, the query results returned are another list of web objects in the web site. The user has no choice but to select one web object from the list and to move to one of the linked web objects again and again until the user finds relevant contents to for search interests.
In this research, we suggest one method for determining weights for arcs that represent weights for hyperlinks corresponding directed graph for a web site, which basically uses weights for web hypermedia objects that are weights for URLs of the directed graph. Methods for measuring weights for web hypermedia objects shortly, URL weights, can be classified into two typical types. The first one, called an arc-based URL weight measurement method, uses hyperlinks structure of web sites to compute weights for web hypermedia objects. A well-known method included in this type is PageRank method [3, 4]. The second one, called as URL-based URL weight measurement method, focuses on keywords in a web hypermedia object in determining the weights for web hypermedia objects. The tf-idf (term frequency and inverse document frequency) method is included in this type of URL weight measurement method.

The above two measurement approaches are in some sense orthogonal since the former measurement methods only exploited the hyperlink structure without explicit consideration of keywords or contents of web hypermedia objects, whereas the latter ones use them explicitly not considering the hyperlink structure. We call the URL weights obtained from the arc-based measurement methods as global weight since the URL weights are computed considering whole structures of hyperlinks. On the other hand, the URL weights obtained with the URL-based measurement methods need user’s query terms. We call this URL weight a local weight since these methods (including cosine measure, tf-idf measure, probability measure, fuzzy measure, etc) use only local information. These two global and local weights can be merged into one integrated weight for web hypermedia searching like Google. Note that it is an open problem to select the best combination of the global and local weight for effective searching.

Notice that the PageRank (or HITS) algorithms have been proved to converge, as a linear combination of the two already converged results (i.e., \( R(i) \) and \( R(j) \)) with fixed parameters that can trivially be shown to be converged in Equation (1). Also, notice that other methods for determining global weights such as HITS by Kleinberg [3], arc co-citation method [5], or the anchor text method [6, 7], and local weights such as cosine measures from the vector space model [8-10]; can also be used to compute arc weights with the equation (3). In addition, many other similarity measures including Euclidean distance, co-citation measures, or support vector machine [11, 12] also have potentials to be used as the arc weights of the directed digraph. Fig. 2 illustrates the schematic architecture of mobile web search system in a server.

### III. OVERALL ARCHITECTURE OF MOBILE WEB SEARCH ENGINE

#### A. System Architecture

The mobile web search engine is consisted of four parts such as Mobile User Interface (MUI), Domain Decision Module (DDM), Weight Computing Module (WCM), and Adaptive Search Module (ASM). The MUI receives queries from user and sends the search result to the user through a graphic user interface (GUI).

#### B. Domain Decision Module

In order to use the rooted directed spanning tree as a dynamic search tree for a mobile web structuring model, we should solve “which rooted directed spanning tree is the ‘best one’ if there are several rooted directed spanning trees with the same root URL, and how the ‘best’ can be found.” Since
the arc weights may be considered as the degree of relevance between two linked web hypermedia objects in a web site, we can choose a rooted directed spanning tree that has maximum sum of arc weights as the best rooted directed spanning tree for the mobile web structuring model. In other words, we use the sum of arc weights as a comparison criterion when determining the best dynamic search tree.

Now we describe a mathematical model, called the mobile web structuring model, to find the best rooted directed spanning tree for a given search domain that is, a directed sub-graph with arc weights. The structured search domain is determined dynamically based on the current location of a user in a web site, so it can be said ‘adaptive’. In other words, the adaptive search domain is an induced sub-graph by the set of URLs \( A^* \subseteq N \) that includes a root URL that is determined by the current location of a user in a web site and all URLs that can be reachable from the root URL via any directed paths starting from the root URL. Also, arc weights for the structured search domain are assumed to compute with Equation (1) given in the previous section. Note that arc weights are dependent on the structured search domain. In other words, the weight of a directed arc can have a different value according to the given structured search domain.

C. Weight Computing Module

We suggest one method for measuring arc weights using the global and local URL weights. The following equation shows the arc weight measurement method, called a bidirectional arc weight measurement.

\[
w_{(i,j)} = p \cdot R(i) + (1-p) \cdot R(j) \tag{1}\]

In (1), an integrated URL weight for URL \( i \), \( R(i) \), is computed with the following equation, where \( R^{\text{global}}(i) \) and \( R^{\text{local}}(i) \) are the global and local weights for URL \( i \), respectively.

\[
R(i) = R^{\text{global}}(i) \cdot R^{\text{local}}(i) \tag{2}\]

where \( R^{\text{global}}(i) = R^{PP}(i) \) and \( R^{\text{local}}(i) = R^{\text{bid}}(i) \).

As shown in (1), the arc weight is computed as a linear combination of the weights for two linked URLs \( i \) and \( j \). Here, the two URLs weights, \( R(i) \) and \( R(j) \), are obtained by the global and local weights for the URLs as given in (2). The arc weight \( w(i,j) \) can be used as a measure for the degree of relevance between two linked web hypermedia objects (that are source URL and destination URL) of the arc \((i,j)\) over a given keyword vector. Therefore, it can be assumed that the magnitude of the arc weight represents how close these two URLs are. So, the generalized arc weights can consider both weights for the source and destination URLs when computing weights for directed arcs. Note that it is worthwhile to find best-suited value of \( p \) by comparing effectiveness for searching relevant web hypermedia objects in a web site with various \( p \) values, which can be another issue to be exploited. We will skip this, because it is beyond the scope of this research.

The arc weights for the directed tree-graph can be computed as follows.

\[
\max \{ \sum w(i,j)x(i,j) \} \quad \text{for} \quad i,j \in A^*, \ j > 1, (i,j) \in A^* \tag{3}\]

subject to

\[
\sum x(i,0) = 0 \quad \text{for} \quad i \in A^*, (i,0) \in A^* \tag{4}\]

where \( i, j = \{0, 1, \ldots, n\} \) are indices for URLs included in structured search domain. The root URL is denoted as \( 0 \). The \( N^* \) is the set of all URLs and the \( A^* \) is the set of all directed arcs. The \( w(i,j) \) is the arc weight of directed arc \((i,j)\) that represents the degree of relevance between two linked URLs \( i \) and \( j \). The \( x(i,j) \) is decision variable that equals 1 if directed arc \((i,j)\) is selected to make rooted spanning tree, and 0 otherwise.

\[
\sum x(i,j) = 0, \quad \text{for all} \quad i,j \in N \tag{5}\]

where \( T_c \) is an un-reachable set for mobile devices from URL \( i \) to \( j \).

\[
\sum x(i,j) = 1 \quad \text{for} \quad i \in N^*, (i,j) \in A^*, j \neq 0 \tag{6}\]

\[
x(i,0) = 0 \quad \text{for all} \quad i \in N \tag{7}\]

The basic strategy to eliminate directed circuits using the constraints (7) and (8) are quite simple that the whole paths of the circuit should be less than the size of the circuit.

\[
x(i,j) + \sum_{k=1}^{m-1} x(j, i, j) \leq m \quad \text{if} \quad m = 1
\]

\[
\sum_{k=1}^{m-1} x(j, i, j) + x(j, i) \leq m \quad \text{if} \quad 2 \leq m \leq |N^*| - 1
\tag{8}\]

For all directed circuits starting from any URL \( i \) and returning to the same URL through \( m \) intermediate URLs (here, a directed circuit can be represented as an ordered list of URLs). Fig. 3 is an example of the structured search domain with arc weights computed from the above algorithm. Here, URL0 is a root.

\[
\begin{array}{c}
\text{URL0} \\
0.73 \\
0.72 \\
0.76 \\
1.00 \\
\hline
\text{URL1} \\
1.6 \\
2.22 \\
2.28 \\
2.26 \\
1.00 \\
\end{array}
\]

Fig. 3. Example structured search domain with URL0 as a root.
D. Adaptive Search Module

Dynamic search tree (that is, rooted directed spanning tree) obtained by the web structuring model could be a complicated structure since a typical web site has more than hundreds or thousands of web hypermedia objects and hyperlinks. In order to use dynamic search tree as a convenient and effective searching tool, there is a requirement to develop efficient applications that have capabilities of recommending search directions for a user in a web site. In this research, we devise two filtering algorithms for the purpose of providing the most relevant web hypermedia objects according to the current location of a user in a web site. Basic idea of the two algorithms is to narrow down the size of dynamic search tree obtained by solving the web site structuring model given in previous section. We call the down-sized dynamic search tree as filtered search tree. The first filtering algorithm as seen in Fig. 4, named as filter_tree algorithm, generates filtered search tree using a filtering threshold, whereas the second algorithm, named as Top_K_tree algorithm, reduces the size of dynamic search tree by selecting most relevant k web hypermedia objects. Listed below is the notation used in these algorithms.

DST: dynamic search tree obtained with web site structuring model
FST: filtered search tree
N*: set of all URLs included in DST
C: set of all URLs included in FST
\( u, v, n \): URLs in N*
\( \text{succ}(v) \): the set of URLs immediately succeeding v
\( f \): filtering threshold

---

Input: DST, f; output: FST
1: filter_tree(n, f)
   2: if |\( \text{succ}(n) \)| == 0 then
   3:     return;
   4:   }
   5: else
   6:     \( C = \text{succ}(n) \); // set of the child URLs of n
   7:     for each u, where \( u \in C \) {
   8:         filter_tree(u, f);
   9:         if then
10:             delete the URL u
11:         else {
12:             add \( \text{succ}(u) \) to C;
13:             for each v where \( v \in \text{succ}(u) \)
14:                 create an arc from URL u to URL v with \( w(n, v) \);
15:                 delete URL u and a arc from URL n to URL u;
16:             } // else
17:         } // for
18:     } // else
19: } // filter_tree()

---

The core of the filter_tree algorithm is that if the weight of a directed arc is less than the filtering threshold (line 9), then both the directed arc and the destination URL of the directed arc are deleted (lines 11-12). Additionally, instead of the deleted URL and directed arc, source URL of the deleted directed arc is connected to each of succeeding URLs of the deleted URL (lines 13-15). The algorithm finds a leaf URL by the lines 1-4 of the algorithm and then a decision will be done whether that leaf URL and corresponding directed arc will be filtered out or not. Since the algorithm is executed upwards from the leaf URL, or if any, to the siblings, and finally to the root URL of the dynamic search tree recursively, the complexity of this filtering algorithm is \( O(n) \).

---

Fig. 5 shows two different filtered search trees (FSTs) for a given dynamic search tree (DST). For Fig. 5(a), the DST without filtering threshold (i.e., with \( f = 0.0 \) ) is trivially equal to FST. When filtering threshold \( f \) is set to 1.0, directed arcs (1, 3) and (2, 7) with corresponding URLs 3 and 7 are not included in FST since the arc weights for the directed arcs, and are filtered out as shown in Fig. 5(b). When filtering threshold \( f \) is set to 1.5, a deeper cut will be followed. All the URLs and directed arcs are deleted except for URLs 0, 4, 5, 8 and directed arcs (0, 4), (4, 5) and (4, 8). The resulting FST is shown in Fig. 5(c).

A more general case is that a user does not know the range of \( f \) value, so it requires only the Top-k web hypermedia objects. It should be the most relevant to the current location (that is, current web hypermedia object) of the user in a web site, so the Top_K_tree algorithm as seen in Fig. 6 can be used to support the user’s searching requests.

For the top k algorithm, named Top_K_tree, at first, all directed arcs included in DST are sorted in descending order of arc weights (lines 2-3). From the sorted directed arcs, the
The implementation environment is Windows Mobile 5.0 SDK (Standard Development Kit) for Pocket PC and Visual Studio 2005. Windows Mobile 5.0 SDK extends Visual Studio 2005 so that we can implement and test application software targeting Windows Mobile 5.0 based Pocket PC devices. Also, the Windows Mobile 5.0 emulator images were used to test our system locally. We also used ActiveSync 4.0 for synchronizing data between Windows-XP Professional and Pocket PC devices.

### B. Demonstration of Our System

Adaptive search mechanism for hyperlinked objects is elucidated step by step using information on real web site. We assume that the current location of a user in an e-learning web site is given by selecting a web hypermedia object from a returned ranked list. Detailed explanations are as follows.

Preprocessing Step: Search query and corresponding root URL decision: A PDA user submits a search query with the three keywords (hurricane, kids, safety) to which a search site is given by selecting a web hypermedia object from the returned ranked list. Detailed explanations are as follows.

Preprocessing Step: Search query and corresponding root URL decision: A PDA user submits a search query with the three keywords (hurricane, kids, safety) to which a search system gives back an available list of URLs. The user selects a URL from the returned URLs. Selected web hypermedia object becomes the root URL of structured search domain as well as the current location of the user in the web site. The root can be changed dynamically, wherever the user selects the current URL as a root.

### IV. IMPLEMENTATION

#### A. Implementation Environment

We implemented the proposed mobile web search engine on Pocket PC based on Windows Mobile as shown in Fig. 7.

### TABLE 1

<table>
<thead>
<tr>
<th>URL-id</th>
<th>URL</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td><a href="http://hurricane.lsu.edu">http://hurricane.lsu.edu</a></td>
</tr>
<tr>
<td>1</td>
<td><a href="http://hurricane.lsu.edu/navbar.htm">http://hurricane.lsu.edu/navbar.htm</a></td>
</tr>
<tr>
<td>2</td>
<td><a href="http://hurricane.lsu.edu/home_page.htm">http://hurricane.lsu.edu/home_page.htm</a></td>
</tr>
<tr>
<td>57</td>
<td><a href="http://hurricane.lsu.edu/undergrad_minor.htm">http://hurricane.lsu.edu/undergrad_minor.htm</a></td>
</tr>
<tr>
<td>58</td>
<td><a href="http://hurricane.lsu.edu/graduate_minor.htm">http://hurricane.lsu.edu/graduate_minor.htm</a></td>
</tr>
<tr>
<td>69</td>
<td><a href="http://hurricane.lsu.edu/research_reports.htm">http://hurricane.lsu.edu/research_reports.htm</a></td>
</tr>
<tr>
<td>102</td>
<td><a href="http://hurricane.lsu.edu/activities.htm">http://hurricane.lsu.edu/activities.htm</a></td>
</tr>
<tr>
<td>103</td>
<td><a href="http://hurricane.lsu.edu/facilities.htm">http://hurricane.lsu.edu/facilities.htm</a></td>
</tr>
<tr>
<td>134</td>
<td><a href="http://hurricane.lsu.edu/before.htm">http://hurricane.lsu.edu/before.htm</a></td>
</tr>
<tr>
<td>135</td>
<td><a href="http://hurricane.lsu.edu/during.htm">http://hurricane.lsu.edu/during.htm</a></td>
</tr>
<tr>
<td>136</td>
<td><a href="http://hurricane.lsu.edu/after.htm">http://hurricane.lsu.edu/after.htm</a></td>
</tr>
</tbody>
</table>
Step 1: We define the structured search domain in Decision Domain Module: With the selected root URL, in the server system, a structured search domain, that is directed sub-graph $G^*$ is induced by the set of URLs, $N^*$, that includes both the root URL itself and all URLs in the web site in the web site that can be reachable from the root URL via directed arcs starting from the root URL. Web hypermedia objects and its corresponding URL ids included in structured search domain for the selected root URL. In TABLE I, URL-id “0” represents the root URL.

Step 2: The arc weights are computed in Weight Computing Module: Arc weights for the directed sub-graph ($G^*$) built in Step 1 are computed using the method explained in the previous section.

Step 3: We find the best adaptive search tree in Adaptive Search Module: With the directed sub-graph and corresponding arc weights, we can find the best rooted directed spanning tree that is, optimal adaptive search tree using the web site structuring model.

Step 4: We generate Top-k adaptive search tree: The Top-k algorithm deriving from a filtering algorithm can be applied to obtain Top-k search trees. Fig. 8 shows the resulting adaptive search tree when Top-10 results are required.

We also applied our method for web search to three wireless technologies such as WiBro(IEEE 802.16e, mobile WiMAX), HSDPA [13, 14], and Wi-Fi [15] to show the mobile resource saving of our method in terms of speed in second. As seen in Fig. 10, the speed gain of WiBro and Wi-Fi is less than HSDPA because they are high speed. As seen in TABLE II, for URL-id = 1 and 2, no speed gain is made because they are directly linked to the root URL. Except them, we obtained the speed gain for reaching to the URLs for three wireless technologies.

We can conclude that if our method is applied to for webpage search through mobile devices then the search speed can be improved. For example, the URL-id = 102 is reached via URL-id = 1 and 5 from root URL in the original web structure. For our system, the URL-id = 102 is reached via URL-id = 1 only. For this case, we can save 1 depth by applying our method. For URL-id = 58, it can be reached via URL-id = 1, 3, 14, 22, 56, and 57 in the original web structure while via URL-id = 1 in our system. So, we can reach the destination webpage with skipping 5 web-pages, which results in saving mobile resource. The amount of saving resource can be measured in byte. The size of the web-pages we have to pass through for reaching URL-id = 58 in the original structure is 1,623, 5,053, 20,580, 20,543, 17,623, 11,351, 29,083,
respectively. The total size can be obtained by summing those sizes and the default URL size 655, which is 116,511 bytes. For our system, the total size is 31362 obtained from summation of 1,623, 29,083, and 655. Thus, we can save 851,150 by using our method. TABLE II shows the resource summation sizes and the default URL size 655, which is 116,511 bytes. The total size can be obtained by summing those sizes.

For our system, the total size is 31362 obtained from summation of 1,623, 29,083, and 655. Thus, we can save 851,150 by using our method. TABLE II shows the resource summation sizes and the default URL size 655, which is 116,511 bytes. The total size can be obtained by summing those sizes.

V. CONCLUSION

We proposed and implemented the mobile web search system using Top_K-tree algorithm. The system includes three parts: Domain Decision Module, Weight Computing Module, and Adaptive Search Module. Also, we demonstrated our algorithm in a real mobile environment using specific URLs. In the experiment, it was shown that our method can be suitable for mobile web search in terms of saving mobile resource.

Contributions of this study can be summarized as follows. First, we applied the optimization techniques embedded onto mobile web information retrieval systems. Second, we devised a new mobile web search mechanism that explicitly uses an arc weight measurement contained in the hierarchical surrogate from which a mobile user can easily navigate hypermedia contents. Finally, a tree filtering and a Top-k tree algorithm are generated in order to elucidate the new web search mechanism for mobile devices.

This research can be extended to several directions. We need to implement the mobile structured search engine mechanism suggested in this approach. In addition, we can develop user interfaces with metadata that apply the notion of the structured search. Finally, we can test many of weight measurement methods including conventional information retrieval measures to evaluate the performance of the structured web search and to find best-suited weight measures for our approach.

ACKNOWLEDGMENT

This research was supported by the Ministry of Information and Communication, Korea, under the College Information Technology Research Center Support Program, grant number IITA-2006-C1090-0603-0031.

REFERENCES


Wookey Lee has received his BS, MS, and Ph.D. in Seoul National University, Korea and he currently is a professor at Inha University, Incheon, Korea. He finished MSE in Carnegie-Mellon University, USA and he has been a visiting scholar in ES, UBC, Canada. He has been a PC member in WAIM, IJWAS, IEEE DEST, etc. He got the best paper award in KORMS(’04). He is editor-in-chief in the Journal of Information Technology and Architecture. His research interests are Mobile and Multimedia DB, Web IR, Web Structuring, Data Warehousing, etc.

Sanggil Kang received the M.S. and Ph.D. degrees in Electrical Engineering from Columbia University and Syracuse University, USA in 1995 and 2002, respectively. He is currently an Assistant Professor in Computer and Information Engineering at InHA University, Korea. His research interests include Mobile Computing, Artificial Intelligence, Multimedia Systems, Inference Systems, Time Series, etc.
Seung-Kil Lim is an assistant professor at Sungkyul University and was previously senior researcher at Samsung Economy Research Institute. He had been employed as a senior consultant at IBM Business Consulting Services and a senior researcher at ETRI, Korea. He received a Ph.D. degree in Industrial Engineering from KAIST, Korea. His research interests are in the area of SCM, e-Business, information technology application, etc.

MyungKeun Shin is received the B.S. and M.S. in Computer Science and the Ph.D. in Information & Communication from Korea Advanced Institute of Science and Technology, Korea. He is currently a researcher in the Department of Telecom Solution Business TFT, SK C&C, Korea. His research interests are Knowledge Management system, Database transaction, and Information Retrieval, etc.

Young-Kuk Kim received the M.S. and Ph.D. degrees in Computer Science and Statistics from Seoul National University, Korea and Computer Science in University of Virginia, USA in 1987 and 1995, respectively. He is currently a professor in Computer Engineering at Changnam National University, Korea. His research interests include Mobile and Ubiquitous Information Systems, Real-Time, Mobile and Main-Memory Databases, etc.