Applying the Site Information to the Information Retrieval from the Web

Yasuhito Asano
Graduate School of Science, the University of Tokyo, Hongo 7-3-1, Bunkyo-ku, Tokyo, Japan

Hiroshi Imai
Graduate School of Information Science and Technology,
the University of Tokyo, Hongo 7-3-1, Bunkyo-ku, Tokyo, Japan

Masashi Toyoda and Masaru Kitsuregawa
Institute of Industrial Science, the Univeristy of Tokyo, Komaba 4-6-1, Meguro-ku, Tokyo, Japan

Abstract

In recent years, several information retrieval methods using information about the Web-links are developed, such as HITS and Trawling. In order to analyze the Web-links dividing into links inside each Web site (local-links) and links between Web sites (global-links) for the information retrieval, it is required that a proper model of the Web site, a phrase used ambiguously in daily life. In the existing researches, a Web server is used as a model of the Web site. This idea works relatively well in case that a Web site corresponds to a server such as public Web sites, but works poorly in case that multiple Web sites correspond to a server such as private Web sites on rental Web servers. In this paper, we propose a new model of the Web site, "directory-based site" to handle typical private sites, and a method to identify them using information about the URL and the Web-links. We verify the method can approximately identify about 66% *of over 110 thousands servers whether each server has multiple directory-based sites or not, and extract over 500 thousands of directory-based sites and 4 million global-links by computational experiments using* `jp`*-domain URLs and Web-links data contains over 23 million URLs and 100 million Web-links, collected from July to August 2000, by Toyoda and Kitsuregawa. We also propose a new framework of the Web-links based information retrieval that uses the directory-based sites and the global-links instead of the Web pages and the whole Web-links respectively, and examine effectiveness of our framework by comparing a result of Trawling on our framework to one on the existing framework.*

1. Introduction

Information retrieval from the World Wide Web becomes an important part of lives of many people today. The most popular methods of searching are the text-based search engines, such as Yahoo!, Google and Altavista. While the text-based search methods are very useful in practice, they have some weaknesses due to ambiguity of language.

On the other hand, several information retrieval methods using the information about the Web-links are developed, such as HITS proposed by Kleinberg [8], PageRank proposed by Brin and Page [6], Cocitation proposed by Bharat et al. [5], and Trawling proposed by Kumar et al. [9] and a method to discover Web communities proposed by Murata [11], as methods to cooperate with the text-based searching methods, not to supersede them. Indeed, some of them are used in the search engines, such as a scoring function of Google.

The fundamental idea of them is as follows: when a Web page v is linked from a Web page u, then we can consider that v can contain valuable information for u. For example, HITS finds *authority* pages with a given topic, which are Web pages linked from many Web pages related to the topic, and *hub* pages, which are Web pages link to many authority pages with the topic.

When we consider the idea, however, we come up against a natural question: can we handle every Web-link equally? The answer is probably no since when we consider a Web-link from a page u to a page v, if u and v belong to different Web sites then v will be valuable for u as described above, but if u and v belong to the same Web site then the link is considered to be made for convenience of the Web browsing. Therefore it will be useful for the information retrieval from the Web to analyze the Web-links,

Proceedings of the 3rd International Conference on Web Information Systems Engineering (WISE'02)
0-7695-1766-8/02 $17.00 © 2002 IEEE

dividing them into links inside each Web site (we call them *local-links*) and links between Web sites (*global-links*), and then now we discuss the "Web site".

The concept of the "Web site" used in daily life without a clear definition, as the vague phrase which indicates a set of Web pages that have related topics and are written by a person, a company, or a group. The existing researches, to our best knowledge, used a Web server as a model of the Web site. In such model, any URLs that have the same name of the Web server (or its part instead of the whole name) are considered to belong to the same site. This model works relatively well in case that a Web site is corresponding to a server such as public Web sites made by companies, governments or other social organisms, but works poorly in case that multiple sites are corresponding to a server such as private Web sites on rental Web servers such as `geocities`, internet service providers such as `so-net`, or universities and so on. It is significant for the information retrieval from the Web that identifying such sites and links between such sites since information obtained from the private Web sites becomes important nowadays, since information about relatively minor topics are often found in the private Web sites rather than the public Web sites.

In this paper, as a new model of the Web site to deal with such private Web sites as well as the public Web sites, we proposed the *directory-based site* defined in the next section. We also proposed an approximation method to identifying the directory-based sites using the knowledge of the URL and the Web-links, and confirmed that the method can determine about $2/3$ of over 110 thousands of Web servers whether they have at least two directory-based sites or not, by computational experiments using URL data of `jp`-domain constructed by Toyoda and Kitsuregawa as [12] and [13].

Once if the method to find the directory-based sites is established, we can analyze the Web-links dividing them into *global-links* (links between such sites) and *local-links* (links inside sites), and then we can make use of an own feature of each type of the Web-links for information retrieval.

The global-links are considered to be more important for several existing information retrieval methods such as HITS or Trawling, since a Web-link to a page in another Web site implies a reference of information on the page, rather than a Web-link inside a Web-site is frequently made for the purpose of navigation. On the other hand, the local-links are important for several existing ones such as [10] since it can find a cluster of Web-pages that have common topics, *information unit*, for pages and links in a given Web site.

In this paper, we proposed a new framework of the information retrieval based on the Web-link analysis. Our framework uses the directory-based sites and the global-links while the existing one uses the Web pages and the whole Web-links. Our framework can avail the merit of the global-links for the information retrieval described above, and some relations between the directory-based sites. For an example, when there are global-links from a directory-based site A to another directory-based site B and from B to A (we say there is a *mutual-link* between A and B), it is considered A and B have some relation. Note that we can not find such a relation between sites without introducing some model of the Web-site and the global-links since it is not necessarily there are both Web-links from a page u in A and v in B and from B to A even if there is a mutual-link between A and B.

We confirmed effectiveness of our framework by comparing a result of Trawling on our framework to one on the existing framework. As a result, two merits of our framework were found as follows: (1) the number of the links required for the information retrieval is extremely reduced (to about $1/20$ times, indeed), (2) the number of relatively small *cores* of the Web communities found by Trawling much increases.

We note that as a first step to find features of the global-links and the local-links, we found interesting results that the degree distributions of them are surprisingly different [3]. The degree distribution of the Web graph plays an important role of the existing information retrieval methods, such as HITS, Trawling, PageRank used in Google [6], Cocitation proposed by Bharat et al. [5] and so on, and therefore [9] and [7] investigated the degree distribution of the Web graph and reported the same result that the Web graph obeys to the *power law* approximately. For a given graph G, if the fraction of nodes with degree d is proportional to $1/d^{\alpha}$ for some constant $\alpha > 0$, G is called a *power law graph* or it is said that G obeys to the power law. Since the fact that the Web graph is a power law graph approximately have been known, researches on the power law graph are developed in recent years, in particular [2] proposed a simple random generation model of the power law graph with the asymmetry of in-degrees and out-degrees. Note that these results treated every Web-link equivalently, and do not consider the global-links and the local-links. Our result that they have the different degree distributions implies that another model that can deal with the two kinds of links and the different distributions by developing the power law graph model will be needed for a model of the Web graph.

We describe here the data and the environment for the computational experiments in this paper. The `jp`-domain URL data we used consists of over 23 million URLs and 100 million links and collected from July to August 2000, by Toyoda and Kitsuregawa similarly to [12], [13]. Note that in the data there are 112,744 `jp`-domain Web servers. The machines we used are SUN Enterprise4500 (UltraSPARCII 400MHz, 8CPUs and 10GB memory) with Solaris 2.7, and a PC (Athlon XP 2100+, 1.5GB Memory) with Windows XP Professional.

The rest of this paper is organized as follows. In section 2, we propose a new model of the Web site, *directory-based site*, and describe the definition of the global-links and the local-links. In section 3, we describe a method to identify the directory-based sites and the result of applying the method to `jp`-domain URL data. In section 4, we propose a new framework of the Web-link based information retrieval that uses the directory-based sites and the global-links, and show a result of Trawling on our framework and one on the existing framework. In section 5, we describe the concluding remarks.

2 The model of the Web site

We propose a model of the Web site, called *directory-based site* as follows.

Definition 1 The model of the Web site,
Directory-based site: *If in a given Web server there is a directory in which an account X, excepting the administrator of the server, is allowed to make or delete files, then a set of Web-pages in the directory is called a site of X, and a set of Web-pages not in such directories is called a site of the administrator of the server. If there is no such a directory, then a set of Web-pages in the server is also called a site of the administrator of the server. The sites in the model is called* **directory-based sites**.

We call the server in the latter case (only the directory-based site of the administrator exists) a *one-site server*, for convenience.

Once when the model of the Web site is given, we can define the global-link and the local-link as follows.

Definition 2 *For a given model of the Web-site, let A and $B \neq A$ denote such sites based on the model (e.g. directory-based sites).* (1) *If there is a link from a page v in A to a page w in B, we say there is a* **global-link** *from A to B.* (2) *A link from a page v to a page w where v and w are in A is called a* **local-link** *inside A.*

The model of the Web sites, the directory-based sites can deal with typical private Web sites in the rental Web servers, such as `geocities`. On the other hand, for example, a more complex Web site in which pages of main contents are placed in a primary server but CGI files (e.g. BBS) are placed in another server can not be handled by the model. It is the open problem whether we can construct a model of the Web site which can deal with such complex Web sites and we can compute proper approximations of the model.

We also propose the definition of the mutual-link since it will have a significant meaning for the information retrieval from the Web, since we can find that many related Web sites are connected by the mutual-links, such as Web communities made by friends or people have common hobbies, in the real Web today.

Definition 3 *For a given model of the Web-site, let A and $B \neq A$ denote such sites based on the model (e.g. directory-based sites). If there are a global-link from A to B and a global-link from B to A, we say there is a* **mutual-link** *between A and B.*

It seems that similar ideas to the global-link, such as *the remote-only link* in [7] or *the non-nepotistic link* in [9] (link from a page u in a server S_1 to a page v in another server S_2) are proposed, but the global-links are different from them since, even if we use the Web servers as a model of the Web sites, the remote-only link and the non-nepotistic link represent a link from a page to a page, but the global-link is a relation from a site to a site.

It is hard to identify the directory-based sites precisely according to the above definition since information about accounts in the Web servers are not available in usual, and therefore we should propose an approximation method to distinguish the directory-based sites using available information.

3 The approximation methods to identify the directory-based sites

The following is the outline of our method to distinguish the Web servers between one-site servers and servers which have at least two directory-based sites. Step by step, we identify and remove Web servers which can be considered as the directory-based sites or the one-site servers from the left Web servers at the step as a filter.

Method to identify the directory-based sites

1. By using the knowledge of a *tilde*-symbol in the URL and the knowledge of the famous rental Web servers or the internet service providers, identify directory-based sites.
2. By using the knowledge of the famous companies and organizations, identify one-site servers.
3. Consider any server which has at most one directory as a one-site server.
4. For a given parameter c, consider any server which has at most c pages as a one-site server.
5. By decomposing a graph, which consists of Web-pages and Web-links in each server, into connected components, consider a server which has components satisfying the condition described in Section 3.3 to be having directory-based sites.

6. By using the statistics (showed in Section 3.4) about the directory-based sites found in the step 1, identify the directory-based sites.

Step (1) and (2) are described in Section 3.1, step (3) and (4) is in Section 3.2, step (5) is in Section 3.3, and step (6) is in Section 3.4.

step	left before	distinguished
1	112,744	18,721
2	94,023	10,049
3	83,974	22,512
4	61,462	16,246
5	45,216	6,746
6	38,470	167

Table 1. The numbers of the distinguished servers at each step

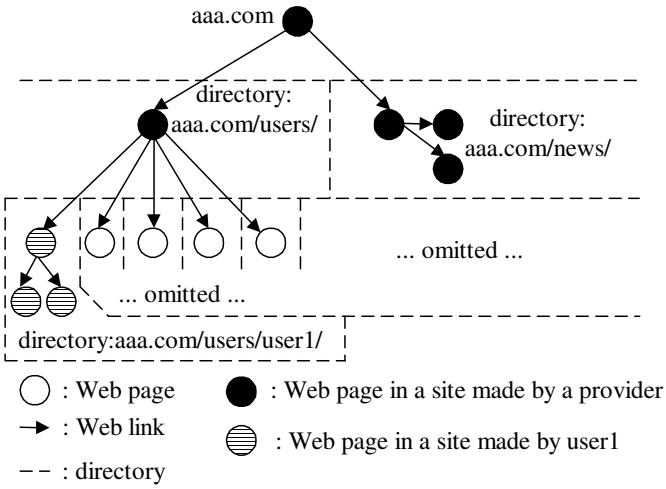

Figure 1. An example of directory-based sites

Table 1 shows the summary of the result of our method. The column `left before` shows the number of the undistinguished servers before the step, and `distinguished` shows the number of the distinguished servers at the step. As a result, we obtained 74,441 distinguished servers (about $2/3$ of the original servers) and 38,303 undistinguished servers, which are considered to be one-site servers in the result. From that, we obtained 573,328 directory-based sites.

While this result contains have some errors as described in the following subsections, by obtaining the result we can analyze the Web-links dividing them into the global-links and the local-links approximately.

3.1 Filters based on Knowledge of the URL

In many Web servers, particularly a number of servers of the universities and the internet service providers, a tilde symbol (~) in the URL is used to represent a name of a user, for example:

```
www.mars.dti.ne.jp/~tk491114/
www.komaba.ecc.u-tokyo.ac.jp/~g440879/
```

Therefore we decided that if we can find a tilde symbol in a URL, we separate the URL at a "slash" symbol just after the tilde symbol, and let the forepart denote a name of the directory-based site and let the back part denote an index of the page in the site. We count the number of servers that have at least one URL contains the tilde symbol among the `jp`-domain URL data, and as a result, 15,044 servers are found among 112,744 `jp`-domain servers.

Furthermore, we utilize well-known information about very famous sites, such as geocities, so-net, and so on. The following URLs are sample sites on the famous Web-servers.

```
www.geocities.co.jp/Hollywood/2762/
www05.u-page.so-net.ne.jp/jd5/niwachan/
www.alpha-net.ne.jp/shunshun/
```

By using such information, we can determine which slash symbol is boundary of a name of sites and an index of the page in the site. In this paper, we use a simple text file that describes in a line a suffix of a name of a server and a number that denotes which slash symbol is boundary, on the basis of our knowledge about the famous servers, as follows.

```
geocities.co.jp 3
u-page.so-net.ne.jp 3
alpha-net.ne.jp 2
```

We use the suffixes instead of the whole name of the servers since servers that have a same suffix usually have a same structure of directories. Note that a URL which have not enough slash symbols in a server whose suffix are found in the text file is regarded to belong to a site whose name is equal to a name of the server. Figure 1 illustrates when we have information "`aaa.com 3`". Since we divide URLs into names of directory-based sites and names of pages in each directory-based site at the 3rd slash symbol, pages represented as black circles belong to a directory-based site named `aaa.com` (site of the administrator of `aaa.com`) and pages represented as shadowed circles belong to a directory-based site named `aaa.com/users/user1` and so on.

We investigated about 40 famous `jp`-domain internet service providers as the famous servers, and obtained such information for 17 servers, and found the rest contains the tilde symbol in their URLs. We can add more information about other famous servers, but the cost to investigate the

servers will increase, especially when we consider apply this method to all servers throughout the world.

We tried to extract sites from the jp-domain servers by using these information, and we obtained the following results.

type	servers	new sites
tilde	15,044	286,962
famous	3,677	71,921
none	94,023	94,023
total	112,744	452,906

Table 2. The numbers of the original servers and the new sites

In Table 2, the column "type" denotes the type of servers, "tilde" means the servers that have the tilde symbol in the URLs of the pages in them, "famous" means the servers that have the suffixes as described above, "none" is the rest. The "servers" column denotes the number of the original servers for each type, and the "new sites" column denotes the number of the new sites extracted by the described method.

From the results, we can see that about 19.1 new sites on average are extracted from a server contains the tilde symbol and about 19.6 on average from a famous server.

We also use the knowledge of the famous Web servers, not of the providers or the rental Web servers, but of the famous companies and organizations in the filter at step (2). We implemented this filter similarly to the filter at step (1), and used information about 735 servers, for example, namco.co.jp and yomiuri.co.jp. As a result, 10,049 one-site servers were found.

Note that the filters at step (1) and (2) are based on the knowledge of the URLs, and therefore if the input of the knowledge has errors then the result contains some errors.

3.2 Filters based on the numbers of the directories and the pages

By the definition of the directory-based sites, if there is only one directory in a given Web server, then the server must be a one-site server, and thus the filter at step (3) remove such Web servers as one-site servers. As a result, we found 22,512 one-site servers at the step.

Then we consider removing Web servers which have relatively small numbers of Web pages as one-site servers, since the smaller the number of the pages is, the smaller possibility there are multiple directory-based sites in the server will be. To implement the filter at step (4), we varied parameter $c \in \{10, 20, 30, 40, 50\}$, and outputted servers which have at most c pages as a one-site servers, and then sampled 50 servers from each output to check errors among them.

c	one-site servers	errors
10	8,441	0
20	16,246	0
30	22,958	1
40	28,105	4
50	32,105	7

Table 3. The numbers of the one-site servers and the errors found

Table 3 shows the result, and the column one-site servers shows the numbers of the servers which have at most c pages, and errors shows the numbers of the errors we could see. Note that invalid URLs (that no longer exists) are not counted as errors. We decided $c = 20$ by considering the result, and therefore we found 16,246 one-site servers at the step.

3.3 Filter using the connected components decomposition

When we investigated the result of step (1), in some Web servers we found that directory-based sites of the administrators and others form independent connected components, and therefore we decide to use the following algorithm as the filter at step (5). The input of the algorithm is a graph G consists of Web pages and Web-links in a given Web server.

Find-sites-by-decomposition

1. Decompose G into connected components $C_1, ..., C_k$.
2. For each C_i $(1 \leq i \leq k)$, compute $d(C_i) = \min\{level(p) \mid p \; is \; a \; page \; in \; C_i\}$, where $level(p)$ denotes the level of the directory in which p is stored, that is, the number of the slash symbols in the full URL of p. For example, if the URL of p is www.geocities.co.jp/Hollywood/5288/index.html , then $level(p) = 3$.
3. Output the most frequent value D in $\{d(C_i) \mid 1 \leq i \leq k\}$.

If $D > 1$, pages have level at most $(D - 1)$ are considered to belong to the site of the administrator, and the other pages are considered to belong to the other sites in the server. When $D = 1$, the server is considered to be a one-site server. As a result, we found D for 6,746 servers. We sampled 50 servers from the result to check errors by our eyes, and found 16 errors, which are occurred in particular when a structure of directories in a given server is far

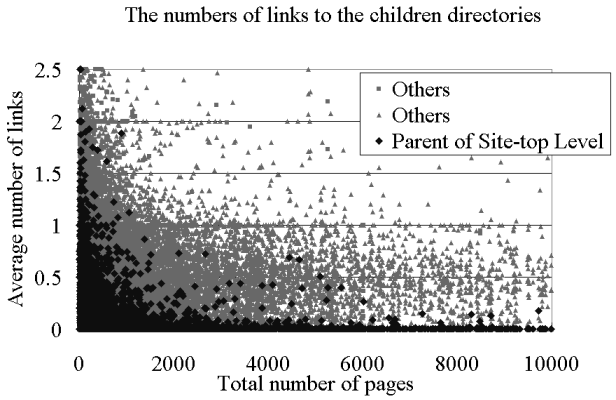

Figure 2. The numbers of the links to the children directories

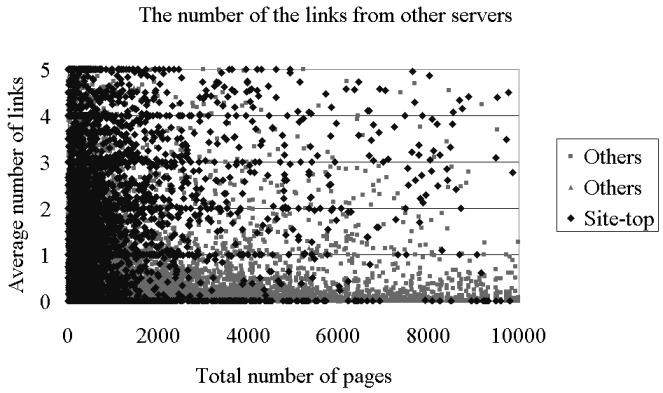

Figure 3. The numbers of the links from other servers

from our assumption (that is, the directory-based sites do not form independent components), or some existing Web-links in actual are not collected in the data.

3.4 Filter based on the statistics about the directory-based sites

As a final step of our method, we consider to utilize statistics about the Web-links around and the number of pages and directories of the directory-based sites found in step (1).

We collected statistics of the information showed in the below figures. Note that `site-top level` means the minimum level of the directory in each directory-based site of the accounts excepting the administrator found in step (1), and `site-top` means the pages named `index.html` and `index.htm` in the site-top level directory.

Figure 2 shows for each page in each server, the average number of the links to the pages in the children directories. "`Site-top Level-1`" means the parent directory of each site-top level directory. Figure 3 shows the average number of the links from the pages in the other servers than a given server. Figure 4 shows for each level $L > 1$ of directories in each server, how many times the number of directories with level L increases compared to the number of directories with level $L-1$. Figure 5 shows for each page in each server, the average number of the *back-links*, links from the pages in the obtained directory-based sites in the server.

From these results, we can see the following features. (1) The numbers of the directories tend to increase drastically at the site-top level rather than others (Figure 4). (2) The numbers of the back-links to the pages in the above directories of the site-top level are approximately equal to zero (Figure 5). (3) The pages in the parent directories of the site-top level directories tend to have fewer links to the pages in the children directories than others, in particular when the total number of the pages in the server is larger (Figure 2). (4) The pages in the site-top level directories tend to have more links from the other severs than others, in particular when the total number of the pages in the server is larger (Figure 3). In particular, we can see that (1) and (2) are outstanding features, and then we construct the following algorithm on the basis of the features. The input of the algorithm is pages, Web-links in and the statistics obtained above about a given Web server.

Find-sites-using-statistics

1. Output a set of integer L satisfying the following condition: the number of the directories of level L is at least r times as many as the number of ones of level $L-1$. Note that where r is determined the number of the pages and the statistics showed in Figure 4, but we omit to mention the actual values of the parameter. This step is called *candidates-selection phase*.
2. For each L, compute a score, and output L with the highest score at least T, a given threshold. The scoring function and T are determined on the basis of the features described above, but we omit the detail. This step is called *scoring-phase*.

We found 167 servers which have multiple directory-based sites among 2,170 candidates of servers obtained in the *candidates-selection phase* in the above algorithm. We sampled 32 servers and checked errors by our eyes, and there are 6 errors and 10 invalid ones (URLs no longer exist) and 16 correct ones. We can increase the number of

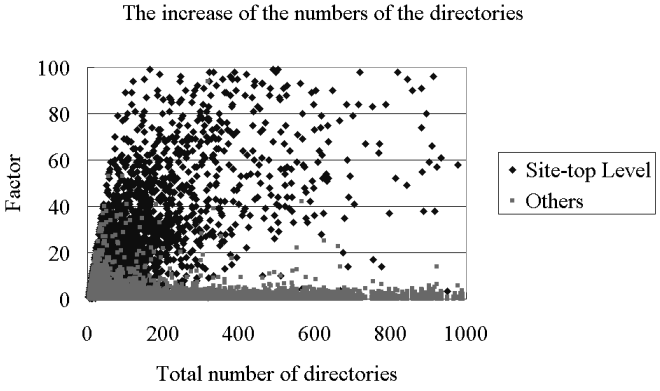

Figure 4. The increase of the numbers of the directories

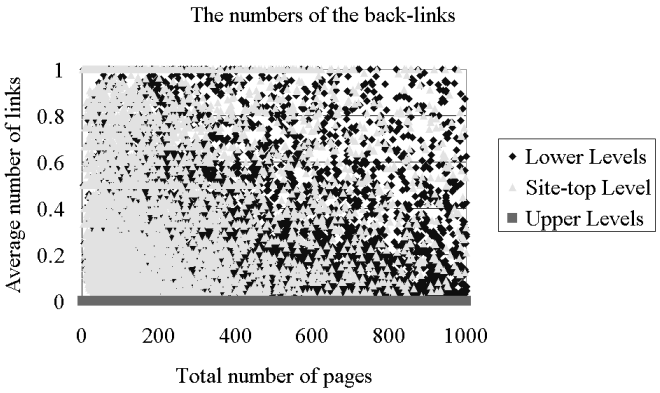

Figure 5. The numbers of the back-links

the candidates in the *candidates-selection phase* and extract more servers than the above one from the candidates in the *scoring-phase* by varying parameters used in the phases, but how to decide them to make the ratio of the correct ones not worse is left as one of the open problems. As a result, we obtained 137,384 directory-based sites at step (5) and (6).

4 Web-link based information retrieval on a new framework

4.1 Outline of this section

Our new framework of the information retrieval based on the Web-link analysis uses the directory-based sites instead of the Web pages, and uses the global-links instead of the whole Web-links. In other words, our framework is based on relations between the directory-based sites while the existing framework was based on relations between the Web pages. In this section, we investigate merits of our framework by comparing a result of Trawling on our framework to one on the existing framework. Section 4.2 summarizes Trawling, Section 4.3 describes the problems of Trawling and how our framework dissolves the problems, and Section 4.4 shows results of the comparison by computational experiments and merits of our framework.

4.2 Review of Trawling

Trawling proposed by Kumar et al. can find the Web communities from the whole Web graph, by scanning the edges of the Web graph and finding small *cores* of the Web communities.

Definition 4 [9] *A* **core** (i, j) *is a small complete bipartite subgraph* (A, B)*, where* $|A| = i$*,* $|B| = j$*, and every vertices (corresponding to a Web page) in* B *is linked from* A*. Note that it may contain edges inside* A *or* B*, though in usual a bipartite subgraph does not allow such edges. Vertices (pages) in* A *(or* B*) are called* **fans** *(or* **centers***, respectively).*

We describe the outline of Trawling as follows. The input of Trawling is the edges, that is, a list of pairs of a source and a destination of each Web-link.

1. By scanning the edges sorted by the source id, prune edges whose sources do not have enough out-degree. In this step, only *non-nepotistic links* are counted for out-degree. Note that this step and the next step are called *pre-pruning phase*.

2. By scanning the edges sorted by the destination id, prune edges whose destinations do not have enough in-degree or do have too much in-degree.

3. To compute cores (i, j) for $\forall i > 0$, and $\exists j > 0$, do the following procedures called *inclusion-exclusion pruning phase*.

   (a) By scanning the edges sorted by the source id, find sources have out-degree exactly j as fans, and for each fan obtain a set of destinations, $C(x) = \{c_1, ..., c_t\}$, linked from the fan, as centers.

   (b) For each fan x and $C(x)$, do the following.

      i. Let $N(c_k)$ denote the neighborhood of c_k, the set of fans that linked to c_k. By scanning the edges sorted by the destination id, find $N(c_k)$ for $1 \leq k \leq t$.

ii. Let $S = N(c_1) \cap N(c_2) \cap ... \cap N(c_t)$. Output $(S, C(x))$ as a core $(|S|, j)$, if $S > 0$ and $(S, C(x))$ is not *nepotistic*. (If $|S|$ has at least two vertices that represent Web pages in the same Web server, the core is called *nepotistic*).

iii. Prune edges whose source is x.

After the above procedures to find cores $(3, 3)$, Trawling indeed applies the *a priori* algorithm described in [1] and [9] to the remaining edges, to enumerate remaining cores. Moreover, Trawling uses several techniques to prune edges and to avoid to repeat sorting, and so on. Refer [9] for details.

They found about 135 thousands cores, for $3 \leq i, j \leq 9$, in *inclusion-exclusion pruning phase* using a link-data generated from over 200 million URLs, and discovered that 96 percents of 400 sample cores (200 $(3, 3)$ cores and 200 $(3, 5)$ cores) correspond to parts of real Web communities. It implies that the cores obtained by Trawling are surprisingly reliable as Web communities.

4.3 Problems of Trawling and our new framework

Trawling seems to have some strange behaviors around typical private Web sites, since it uses the Web servers as a model of the Web sites. Trawling at step (3.b.ii) remove the nepotistic cores, which has at least two fans in the same Web server, but we already saw that such pages can belong to different directory-based sites. Figure 6 and 7 show examples of two different types of the nepotistic core. In Figure 6, three vertices in the left are fans, and two of them placed in `geocities.co.jp` but made by two different users. Three vertices in the right are centers. On the other hand, in Figure 7, two of fans are placed in `sony.co.jp`, which belong to one site made by the corporation. Trawling can not distinguish cores that contain fans in different directory-based sites but in the same server (e.g. Figure 6) between cores that contain fans in the same directory-based site and the same server (e.g. Figure 7), though the former has important for the information retrieval and for the latter case it is natural to regard multiple fans in the same directory-based site as only a fan. Moreover, Trawling does not distinguish a case there are multiple centers in the same server (nor same directory-based site) between a case there are no such centers, however, when there are multiple centers in the same directory-based site it is more natural to contract them and treat it as one center.

Our new framework can dissolve these problems. We use the directory-based sites to represent the fans and the centers instead of the Web pages, then it does not need to delete the nepotistic cores at step (3.b.ii) to find cores that contain fans in different sites but in the same server, and

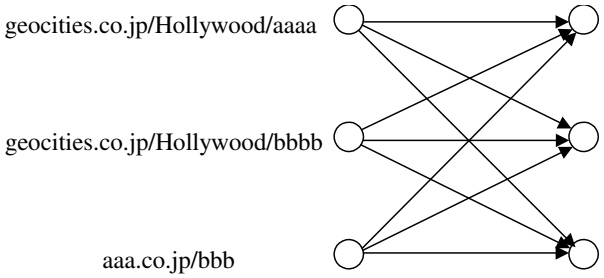

Figure 6. The nepotistic core (1)

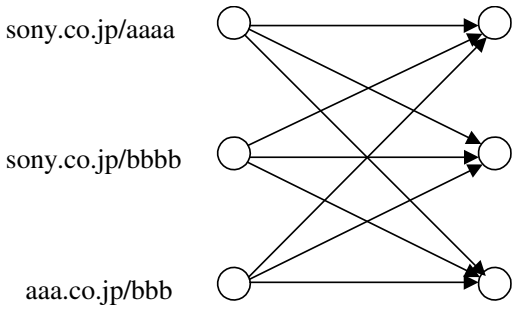

Figure 7. The nepotistic core (2)

it does not need to distinguish the non-nepotistic links at step (1) since only the global-links between the directory-based sites are used on our framework. It can also deal with multiple fan (or centers) pages in the same directory-based site as one fan (or one center, respectively).

Another merit of our framework is to reduce the running time. It takes much time to identify the directory-based sites and construct the global-links from the original data of the Web-links, but this framework can make Trawling quite simple and expected to reduce the time to compute cores since the number of the global-links is quite smaller than the number of the whole Web-links.

4.4 Comparison

We implemented Trawling and execute it on several frameworks by using the `jp`-domain URL data. Besides the existing framework, to emphasize a difference of models of the Web-sites, we implement two variation frameworks: (A) uses the directory-based sites and the global-links, (B) uses the Web servers as a model of the Web sites and links between the servers (note that they correspond to global-links when we use this model and they are different from the non-nepotistic links, as mentioned in Section 2.1). We call Trawling on the existing framework "original Trawling" and call Trawling on the framework (A) (or (B)) "our variation (A)" (or our variation (B), respectively).

Table 4 shows the running time to compute the cores

where j is fixed to 3. `Const.` column shows time to read and construct link-data, `Pre-pr.` column shows time of the pre-pruning phase, and `In-Ex.` column shows time of the inclusion-exclusion phase. Note that while the construction of the link-data and the pre-pruning phase (step (1) and (2)) of original Trawling is computed on Sun Ultra4500 since it requires over 3.4GB, others can be computed on the PC with Athlon XP 2100+. Note that Kumar et al. implemented Trawling as an I/O-algorithm to deal with their huge data on smaller main memory and larger disk, on the other hand, we implemented all our programs including the original Trawling to be run on the main memory. We can see that our variations are extremely faster than the original Trawling since the number of the global-links is relatively smaller than the whole Web-links and the original Trawling must deal with the nepotistic links or cores. Table 5 shows the number of the links used in the experiments. `Whole` column shows the following: `Original` row shows the number of the whole Web-links in the `jp`-domain data, `Our (A)` row shows the number of the global-links, and `Our (B)` row shows the number of the links between the servers. `Pre-pruning` shows the number of the remaining edges after the pre-pruning phase.

Algorithm	Const.	Pre-pr.	In-Ex.
Original	183	448	271
Our (A)	21	1.9	44
Our (B)	21	1.9	1.1

Table 4. The running time to compute the cores (minutes), $j = 3$

Algorithm	Whole	Pre-pruning
Original	92,148,273	14,162,005
Our (A)	5,444,114	4,471,334
Our (B)	3,913,973	3,478,610

Table 5. The number of the links used

Table 6 shows the number of the cores, where j is fixed to 3, obtained from the original one and our variations (A) and (B). We can see that our variation (A) generated much more cores than the original one, on the other hand, our variation (B) found surprisingly very few cores. The reasons for the results are considered as follows: (1) our variation (A) can find the *nepotistic cores* contains fans in the different directory-based sites but in the same server, but the original one can not distinguish such ones. (2) in our variation (B), a server (used as a model of the Web site) tends to have much larger out-degree than a directory-based site has in usual

i	Original	Ours (A)	Ours (B)
3	59	441	7
4	20	236	2
5	19	160	0
6	7	141	0
7	5	92	0
8	1	70	0
9	1	50	1

Table 6. The number of cores, $j = 3$

$i \backslash j$	3	4	5	6	7	8	9
3	59	45	40	28	15	15	7
4	20	13	3	2	6	6	4
5	19	8	0	4	5	3	1
6	7	9	2	1	0	5	0
7	5	0	0	0	0	0	1
8	1	2	1	0	0	0	0
9	1	0	0	0	0	1	0

Table 7. The number of cores (Original, $3 \leq j \leq 9$**)**

and therefore there will be few cores of smaller j. From this result, we found it is not good to use the Web server as a model of the Web site for Trawling. Table 7 and 8 show the number of cores, where $3 \leq j \leq 9$, obtained from the original one and our variation (A), selected as better two methods. We can see for smaller j, our variation (A) generated more cores, but for larger j the original one found more cores, and the total number of cores of our (A) is still larger than the original. The reason is that in the original one does allow centers of a core exist in the same server, but our variation (A) does not allow centers of a core exist in the same directory-based site since it use the center directory-based sites instead of the center pages. Then such cores contain center pages in the same directory-based site are contracted to cores with smaller j, and therefore they are found when j is smaller in our variation (A).

5 Concluding Remarks

In conclusion, we proposed the new model of the Web site, the directory-based site, to deal with typical private Web sites and analyze the Web-links dividing them into the global-links and the local-links. We also extracted over 500 thousands of the directory-based sites from `jp`-domain URL data approximately and using them divided the Web-links into the global-links and the local-links.

One of the future work should be more precise evalua-

$i \backslash j$	3	4	5	6	7	8	9
3	441	137	58	27	11	9	3
4	236	87	18	8	6	1	1
5	160	41	8	3	2	1	0
6	141	27	9	4	2	0	1
7	92	14	5	0	1	0	0
8	70	8	5	1	0	6	1
9	50	6	4	0	0	0	0

Table 8. The number of cores (Our (A), $3 \leq j \leq 9$)

tion of the results obtained by Trawling on our framework, such as an investigation whether the cores obtained using our framework include more interesting information than the existing framework or not.

Another future work derived from our new framework is to utilize some graph structures consists of the mutual-links for the information retrieval. There are many related Web sites such as Web communities made by friends or people have common hobbies connected each other by the *mutual-links*. Now we are trying to find the Web communities by enumerating maximal cliques or similar dense graph structures formed by the mutual-links.

Moreover, we are also developing **Web-Linkage Viewer** [4] as Figure 8, a visualization system drawing the Web-links understandably by dividing the global-links and the local-links, and drawing the global-links on a spherical surface and the local-links in cones emanating from the surface to separate them clearly but draw them in the same space. We expect it will be one of the methods for users to find maximal cliques or cores or several graph structures useful for the information retrieval from the Web.

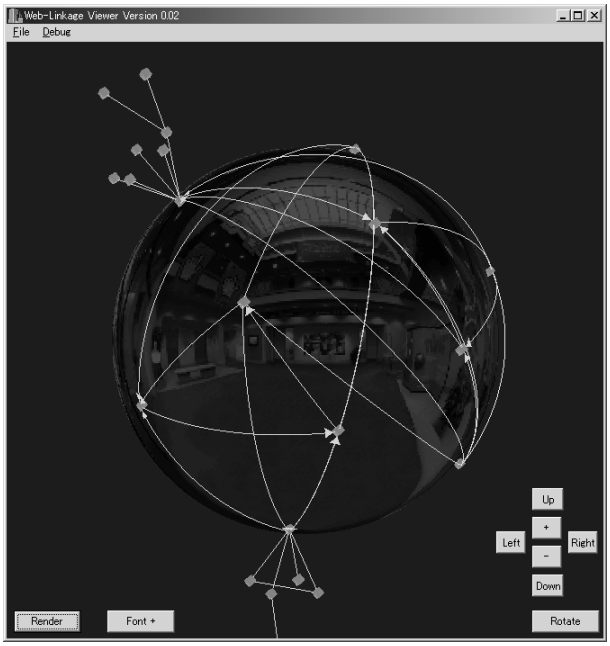

Figure 8. The Web-Linkage Viewer

References

[1] R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In *Proceedings of the 20th International Conference on Very Large Databases*, pages 487–499, 1994.

[2] W. Aiello, F. Chung, and L. Lu. Random evolution in massive graphs. In *Proceedings of the 42nd Annual Symposium on Foundation of Computer Science*, pages 510–519, 2001.

[3] Y. Asano, H. Imai, M. Toyoda, and M. Kitsuregawa. Focusing on Sites in the Web. In *Proceedings of IASTED International Conference Information Systems and Databases 2002*, to appear, 2002.

[4] Y. Asano, H. Imai, M. Toyoda, and M. Kitsuregawa. The Web-Linkage Viewer: Finding graph structures in the Web. In *Proceedings of the 3rd International Conference on Web-Age Information Management*, pages 441–442, 2002.

[5] K. Bharat, A. Broder, M. R. Henzinger, P. Kumar, and S. Venkatasubramanian. The connectivity server: fast access to linkage information on the Web. In *Proceedings of the 7th International World Wide Web Conference*, 1998.

[6] S. Brin and L. Page. The anatomy of a large-scale hypertextual Web search engine. In *Proceedings of the 7th International World Wide Web Conference*, pages 14–18, 1998.

[7] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph structure in the Web. In *Proceedings of the 9th International World Wide Web Conference*, 2000.

[8] J. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 668–677, 1998.

[9] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Trawling the Web for emerging cyber-communities. In *Proceedings of the 8th International World Wide Web Conference*, 1999.

[10] W. Li, K. Candan, Q. Vu, and D. Agrawal. Retrieving and organizing Web pages by information unit. In *Proceedings of the 10th International World Wide Web Conference*, 2001.

[11] T. Murata. Discovery of Web communities based on the co-occurrence of references. In *Proc. of the 3rd International Conf. on Discovery Science, LNCS1967*, pages 65–75, 2000.

[12] M. Toyoda and M. Kitsuregawa. Finding related communities in the Web. In *Poster Proceedings of the 9th International World Wide Web Conference*, pages 70–71, 1999.

[13] M. Toyoda and M. Kitsuregawa. A Web community chart for navigating related communities. In *Poster Proceedings of the 10th International World Wide Web Conference*, pages 62–63, 2000.