IEICE TRANS. INF. & SYST., VOL.E87–D, NO.9 SEPTEMBER 2004

PAPER

Finding Neighbor Communities in the Web Using an Inter-Site Graph

Yasuhito ASANO[†a)], ***Nonmember***, **Hiroshi IMAI**[††], ***Member***, **Masashi TOYODA**[†††], ***Nonmember***, ***and*** **Masaru KITSUREGAWA**[†††], ***Member***

SUMMARY In this paper, we present Neighbor Community Finder (NCF, for short), a tool for finding Web communities related to given URLs. While existing link-based methods of finding communities, such as HITS, trawling, and Companion, use algorithms running on a Web graph whose vertices are pages and edges are links on the Web, NCF uses an algorithm running on an *inter-site graph* whose vertices are sites and edges are *global-links* (links between sites). Since the phrase "Web site" is used ambiguously in our daily life and has no unique definition, NCF uses *directory-based sites* proposed by the authors as a model of Web sites. NCF receives URLs interested in by a user and constructs an inter-site graph containing neighbor sites of the given URLs by using a method of identifying directory-based sites from URL and link data obtained from the actual Web on demand. By computational experiments, we show that NCF achieves higher quality than Google's "Similar Pages" service for finding pages related to given URLs corresponding to various topics selected from among the directories of Yahoo! Japan.

key words: *information retrieval, Web community, Web site*

1. Introduction

In recent years, methods of finding Web communities (set of related Web sites) from the Web are developed, such as "Similar Pages" service in Google. In particular, a special attention has been paid on methods using characteristic graph structures of the links on the Web, since they are not subject to linguistic problems, such as dummy keywords or a word having multiple meanings. HITS proposed by Kleinberg [9] and trawling proposed by Kumar et al. [10] are well-known examples of such methods. HITS finds authority pages and hub pages having the same topic as a given page, and trawling finds communities, regardless of topics, as many as possible in the snapshot of the whole Web. These methods are based on the following idea: if page u has a link to page v, then page v may contain valuable information for the author of u. That is, these methods use algorithms running on a Web graph, whose vertices are pages and edges are links between pages. It can also be said that these methods treat a page as a unit of information.

When we utilize a Web graph for finding communities, the following natural question arises: can we handle every link equally? The answer is probably no, since humans frequently consider a Web site as a unit of information. That is, for a link from a page u to a page v, if u and v are in different Web sites then v will be valuable for u as described above, but otherwise (i.e. if u and v are in the same Web site), the link may be made for convenience of navigation or browsing.

A practical example is a *mutual-link*. It is known that a mutual-link between two sites A and B (i.e. there are a link from a page in A to a page in B and a link from a page B to a page in A) is made when these sites are related and authors of the sites know each other. Note that if we consider a page as a unit, we cannot find a mutual-link between site A and B when no pair of page (u, v) for $u \in A$ and $v \in B$ links each other. Such a case frequently occurs, for example, when a site has a top page and another page for links to other sites. Therefore, it is expected to be more natural and better to use a site as a unit of information for finding communities. However, there have been no method of identifying sites from data of URLs and links, this idea have not been utilized in existing works.

In recent years, the authors have shown that this expectation is true by proposing a new model of Web sites, named *directory-based sites*, and establishing a method of identifying directory-based sites from data of URLs and links [2], [3].

In this paper, we consider a method of finding communities based on this work, named Neighbor Community Finder (NCF, for short). NCF receives at least one URL as an input from a user, and returns sets of sites as communities related to the given URLs. Since the pages with given URLs will have a topic interested in by the user, NCF can also be used for finding sites having the topic.

The outline of NCF is as follows.

1. Obtain data of pages (URLs) and links in neighborhood of the given pages by using a HTML parser and a search engine (for back-links).
2. Identify directory-based sites from the data and construct an *inter-site graph* containing the sites. Note that an inter-site graph is a directed graph whose vertices are directory-based sites and edges are *global-links* between sites. Links inside a site are called *local-links* in the site.
3. Finding communities related to the given pages by enumerating maximal cliques of mutual-links in the inter-site graph.

As for (1), our aim is to find communities of sites in the

Manuscript received September 30, 2003.

[†]The author is with Graduate School of Information Sciences, Tohoku University, Sendai-shi, 980–8579 Japan.
[††]The author is with Graduate School of Information Science and Technology, the Univeristy of Tokyo, Tokyo, 113–0033 Japan.
[†††]The authors are with Institute of Industrial Science, the Univeristy of Tokyo, Tokyo, 153–8505 Japan.
a) E-mail: asano@nishizeki.ecei.tohoku.ac.jp

neighborhood of the sites containing the given pages. This is based on the same idea that related pages are frequently laid in the neighborhood of the given pages, as one used in HITS, Companion and Cocitation [7], and trawling. Note that these methods use pages at most two links distant from the given pages in a Web graph. We claim that this idea will work better in an inter-site graph than a Web graph, since such neighbor pages can be laid in the same site as the given pages in the Web graph. The authors actually have shown that trawling works better in an inter-site graph than a Web graph [2].

For construction of an inter-site graph as described in (2), we use a method established by the authors in [3] and [2]. The method named *filters* consists of filtering and error correction phases. The filtering phase consists of seven filters, and each filter finds some Web servers and determines whether they contain only one directory-based site or multiple directory-based sites (i.e. two or more sites), and transfers the remaining servers to the next filter. The authors have examined that this method can determine whether Web servers contain only one site or multiple sites almost correctly (more than 90%) and extracts about five times as many directory-based sites as Web servers by using data sets of URLs and links in `.jp` domain crawled in 2000 and 2002 by Toyoda and Kitsuregawa as [12]. They have verified the usefulness of the inter-site graph by showing that trawling on an inter-site graph can find communities corresponding to *nepotistic cores* and *hidden cores*, while trawling on a Web graph cannot find such communities.

For (3), we decide to adopt maximal cliques of mutual-links rather than *cores* (directed complete bipartite small subgraph) used in trawling, since the authors have verified that enumerating maximal cliques is more suitable for finding communities, in particular communities of personal sites, than trawling. On the inter-site graph constructed from the data set in 2002, it is shown in [2] that about 45% of communities obtained by the enumerating cliques are communities of personal sites, though trawling finds very few such cliques. Note that these methods enumerate a number of communities on the whole Web regardless of user's interest, by using a huge snapshot of the Web. In this paper, we apply the maximal clique enumeration method to finding communities related to pages given by a user, by using a small graph composed of neighbor sites.

In order to show how effective NCF is, we compare the results of NCF with "Similar Pages" service in Google by computational experiments. As instances for the experiments, we use seven sites provided by voluntary users and 12 sites corresponding to six topics chosen from among Yahoo!(Japan)'s second level directories. For each topic, we select one personal site and one official site from among the sites registered to the directory corresponding to the topic. As a result, our NCF achieves better results than the Google's service for most of the given instances in both quality and quantity.

The rest of this paper is organized as follows. In Sect. 2, we describe preliminaries, that is, the definitions of directory-based sites and an inter-site graph and so on. In Sect. 3, we describe the mechanism of proposed Neighbor Community Finder. In Sect. 4, we compare the results of NCF with the results of "Similar Pages" service in Google. In Sect. 5, we describe concluding remarks.

2. Preliminaries

In this section, we describe the definition of directory-based sites, a new model of Web sites proposed in [2]. Then, we describe the definition of an *inter-site graph*, whose vertices correspond to directory-based sites.

2.1 Directory-Based Sites

The phrase "Web site" is used ambiguously in our daily life, and therefore it is difficult to present a unique definition of a Web site. In this paper, we use the following definition which seems not to be apart from the concept used in our daily life. Note that similar definitions are found in [1] and [6], although they did not find sites on the whole Web according to their definitions.

Definition 1: A Web site is a set of Web pages that are written by a single person, company, or cohesive group.

If every Web page includes Meta information about its authors, this definition will be well-defined and we can compute Web sites easily according to this definition. However, such information does not exist in the real Web unfortunately and therefore it is hard to compute Web sites according to this definition. Thus, we have to consider a method of estimating Web sites in a restricted situation.

We have observed several Web servers containing a number of Web sites of users, such as rental Web servers and ISPs and universities, and found the following facts. In several Web servers, each user is given a directory and a set of Web pages in the directory (and its subdirectories) frequently forms a Web site of the user. For example, `www.geocities.co.jp/Hollywood-Cinema/1737/` is a directory in `www.geocities.co.jp/` and a set of pages forms a Web site of a user. On the basis of such instances, we propose a new model of Web sites, called a *directory-based site* model as is set out below.

Definition 2: For a Web server, let $\{d_1, \ldots, d_k\}$ be a given set of directories in the server such that d_i $(1 \leq i \leq k)$ is neither the root directory of the server nor a subdirectory of any other d_j $(j \neq i)$. Then, for each i, a directory-based site whose top directory is d_i denoted by D_i is defined to be the set of Web pages in the directory d_i and all its subdirectories. That is, D_i consists of pages that are contained in d_i or a subdirectory of d_i. On the other hand, the set of Web pages in the server but not in $\{d_1, \ldots, d_k\}$ (or not in any of their subdirectories) is called a directory-based site of the administrator of the server. For convenience, a directory-based site different from the directory-based site of the administrator is called a user's directory-based site or a directory-based site of a user.

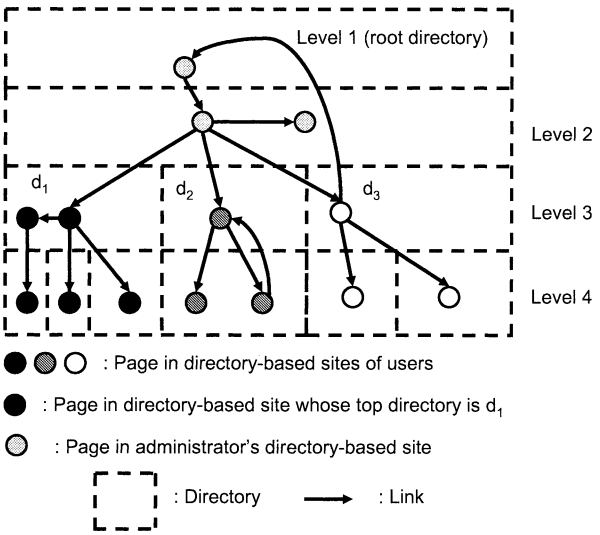

Fig. 1 An example of directory-based sites in a multi-site server.

If all the pages in a given server are in the site of the administrator of the server (i.e. $k = 0$ in Definition 2), the Web server is called a *single-site server*. Otherwise (i.e. $k \geq 1$ and at least one directory is given), the server is called a *multi-site server*.

Figure 1 shows an example of directory-based sites in a Web server when $\{d_1, d_2, d_3\}$ are given. In this figure, black (dark gray, white) circles represent pages in the directory-based site with top directory d_1 (d_2, d_3, respectively), and light gray circles represent pages in the directory-based site of the administrator of this server. The directory-based site model can deal with a typical personal Web site which consists of pages in one directory in a rental Web server, an ISP, or a university and so on.

Since we are concerned with finding communities from the real Web and other related things, throughout this paper, we should find $d_1, \ldots, d_k$ in Definition 1 for each server from a set of Web servers with data of URLs and links in each server. We describe the method of identifying directory-based sites proposed by the authors in the next section.

2.2 Inter-Site Graph

By using the definition of directory-based sites, we can define an inter-site graph as follows. For convenience, we define an inter-server graph and an intra-server graph here.

Definition 3: Let A and B be two distinct 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 with v and w in A is called a **local-link** inside A.

Definition 4: (1) A graph which consists of directory-based sites as vertices and global-links as edges is called an **inter-site graph**. (2) For each site, a graph which consists of pages in the site as vertices and local-links in the site as edges is called an **intra-site graph** for the site. (3) A graph which consists of servers as vertices and links between servers as edges is called an **inter-server graph**. (4) For each server, a graph which consists of pages in the server as vertices, and links in the server as edges is called an **intra-server graph**.

3. The Mechanism of NCF

In this section, we describe the mechanism of NCF, more precisely and technically than Sect. 1.

3.1 Outline

As an input, NCF receives at least one URL from the user. Let these URLs be $\{u_1, \ldots, u_h\} = U$ and S_k be the server containing u_k for $1 \leq k \leq h$. The main routine of NCF is as follows. The detail of each step is described in Sect. 3.3 to 3.5.

1. Construct a *seed graph* G. A seed graph is the inter-site graph which consists of directory-based sites in $\{S_1, \ldots, S_h\}$ and global-links between them.
2. By repeating a *growth step*, grow G. A growth step finds directory-based sites adjacent to sites in G and adds them to G.
3. Enumerate maximal cliques formed by mutual-links in G and output them as neighbor communities.

3.2 Filters: A Method Finding Directory-Based Sites

For constructing an inter-site graph, we have to identify directory-based sites from data of URLs and links. In [3] and [2], the authors have proposed a method named *filters* and have shown that it can identify directory-based sites almost correctly by using .jp domain data sets collected in 2000 and 2002 by Toyoda and Kitsuregawa. In this subsection, we summarize this method and the result of it for the data sets in 2000 and 2002.

Our method consists of a filtering phase and an error correction phase (error correction of filters using clique, ECFC for short). In the filtering phase, there are seven filter steps and we call the i-th filter step is called Filter i ($0 \leq i \leq 6$).

Filter 0: By using our knowledge for a level of directories corresponding to users' Web sites on each famous rental Web server or ISP, find directory-based sites in multi-site servers.

Filter 1: By using a *tilde*-symbol in a URL as a symbol representing directories corresponding users' Web sites, find directory-based sites in multi-site servers.

Filter 2: By using our knowledge of famous companies and organizations, find single-site servers.

Filter 3: Consider any server having at most one directory as a single-site server.

Filter 4: For a given parameter c, consider any server which has at most c pages as a single-site server.

Filter 5: For each server, we consider its associated graph which consists of Web-pages as vertices and links as edges, and decompose it into the connected components. Then,

considering each component as a site, determine whether the server is a multi-site server or a single-site server.

Filter 6: By using information about the numbers of back-links and directories, which is obtained by statistics, find multi-site servers and a level of directories corresponding to directory-based sites of users in each of them.

Since one of the weak points of Filters 5 and 6 is that their ratios of errors are relatively larger than those of Filters 0 to 4 as described below, in ECFC, we find single-site servers which are regarded as multi-site servers incorrectly at Filters 5 and 6. While we compute maximal cliques in the inter-site graph as described in Sect. 3.5, we have found a number of cliques which consist of directory-based sites within one Web server. Investigating such cliques, we have found that they are mainly derived from errors in Filter 5 and 6. If an error that mistakes a single-site server in correct for a multi-site server occurred, cliques of wrong directory-based sites in the server could be found, since links inside a site frequently form a clique. Therefore, we have decided to use this fact to identify errors in Filter 5 and 6. At first, we enumerate maximal cliques in the inter-site graph composed of directory-based sites found in Filters 5 and 6. Then, we enumerate cliques such that every directory-based site in the clique belongs to one Web server.

Note that the remaining Web servers after these filters and ECFC are regarded as single-site servers.

Table 1 and Table 2 show the numbers of the identified servers, errors, and obtained directory-based sites by Filters 0 to 6 and ECFC. "Remains" column shows the number of remaining servers before each filter, and "Identified" column shows the number of identified servers by each filter. "ECFC" row shows the number of identified servers at ECFC ("Identified" column), and identified servers in ECFC are not counted in the total number of identified servers, since they are determined to multi-site servers in Filter 5 and 6, but adjusted to single-site servers in ECFC. The number of errors in 150 sampled servers are shown in "Errors" column. ECFC also removes a part of the directory-based sites obtained Filters 5 and 6, and the number of removed sites is shown at "Sites" column.

Table 1 The numbers of the identified servers (2000).

Filter	Remains	Identified	Errors	Sites
0	112,744	3,677	0	71,921
1	109,067	150,44	0	286,962
2	94,023	10,049	1	10,049
3	83,974	22,512	0	22,512
4	61,462	16,246	0	16,246
5	45,216	6,746	49	119,945
6	38,470	167	31	17,439
ECFC	38,303	(247)	10	-19,766
Remains	38,303	-	20	38,303

Table 2 The numbers of the identified servers (2002).

Filter	Remains	Identified	Errors	Sites
0	373,737	14,642	2	400,123
1	359,095	32,710	1	809,617
2	326,385	17,666	1	17,666
3	308,719	150,764	0	150,764
4	157,955	55,542	0	555,42
5	102,413	27,590	56	344,853
6	74,823	871	43	225,604
ECFC	73,952	(1,006)	4	-103,034
Remains	73,952	-	14	73,952

The results of this method for the data set in 2000 and 2002 are as follows. The filters and ECFC have identified 563,611 directory-based sites from 112,744 servers for the data set in 2000. For the data set in 2002, they have identified 1,975,087 directory-based sites from 373,737 servers. They have also estimated error rate of this method by sampling 150 servers randomly from the identified servers by each filter and ECFC and the remaining serves. As a result, the estimated error rate is about 6.8% for the data set in 2000, and 4.5% for the data set in 2002, and therefore it can be said that this method identifies directory-based sites almost correctly, in practice.

The details of the filters, ECFC, and the estimation of the error rate are described in [2]. The filters are also described in [3], [4].

Note that since NCF uses data of URLs and links in the neighborhood of the given pages rather than huge data of whole URLs and links in .jp domain as described above, we utilize this method with slight modifications. For this purpose, we also prepare a filter database describing our knowledge used Filters 0 and 2. This filter database consists of pairs of a string corresponding to a suffix of the name of a server and integer corresponding to the level of top directories of users' directory-based sites in servers whose names contain the suffix. For given URL u, a function $db(u) \geq 0$ for this database returns an integer. If $db(u) > 1$, the $db(u)$-th slash symbol in the URL represents the top directory of user's directory-based site, otherwise, the server with u is regarded as a single-site server. Otherwise ($db(u) = 0$), it means that the database cannot determine which slash symbol is so. If such a slash symbol is found, we can find a name of the directory-based site induced from the URL. Let *sitename*(u) be the name of the directory-based site, that is, a prefix part of u starts from the first character and ends at the slash symbol. Let *pagename*(u) be a suffix part of u starts from the character just behind the slash symbol. For example, if u is "`www.geocities.co.jp/Playtown-Dice/1722/src/SRC.html`", *sitename*(u) is "`www.geocities.co.jp/Playtown-Dice/1722/`" and *pagename*(u) is "`src/SRC.html`". Furthermore, we modify the order of the filters for NCF as described in the following subsections.

3.3 Constructing a Seed Graph

When NCF receives seed URL set U, NCF begins to construct a *seed graph* and *neighbors set* N_v, that is a set of URLs $\{u\}$ such that $u \notin S_k$ (for $1 \leq k \leq h$) and the page of u is adjacent to a page in the seed graph by a link. Let $G = (V, E)$ be an empty graph, R be an empty set of graphs, N_v be an empty set of URLs, and N_e be an empty set of links. Each vertex $v \in V$ has a label *label*(v) corresponding

to some part of its URL. Fig. 2 to 4 illustrate the outline of the construction of a seed graph. The following description is the detail of this step.

Construct-seedgraph(U, G, R, N_v, N_e)

1. For each URL $u \in U$, do the following procedure:
   a. If $db(u) > 0$, do the following "**create intra-site graph**" procedure:
      i. If there is no vertex in V whose label equals to $sitename(u)$: Create a new intra-site graph $G_i = (V_i, E_i)$, where $i = |V| + 1$ and add a vertex with label $pagename(u)$ to V_i. Then, add a vertex with label $sitename(u)$ to V and add G_i to R.
      ii. Otherwise: Let $v \in V$ with a label $sitename(u)$ and G_i be the corresponding intra-site graph. If there is no vertex in V_i with a label $pagename(u)$, add a vertex with a label $pagename(u)$ to V_i. (Otherwise, do nothing.)
   b. Otherwise: Do a "**create intra-site graph**" procedure, by using $servername(u)$ instead of $sitename(u)$. The graphs created here called *temporary intra-server graphs*.
2. For each graph $G_i \in R$, call **crawling**(G, G_i, N_v, N_e) procedure described below.
3. For each temporary intra-server graph G_t, do the following.
   a. By using Filters 1 and 3 to 6, and ECFC, compute $a > 0$ such that the a-th slash symbol represents the top directory of user's directory-based site in the server and add this result (i.e. the name of the server and the integer a) to the filter database.
   b. Divide G_t into the multiple intra-site graphs correctly by using the above result of the filters.
4. Output G, G_i $(1 \le i \le |V|)$, and N_v.

crawling(G, G_i, N_v, N_e)

1. Set $S = V_i$, and for each $s \in S$, let u_s be the URL corresponding to s.
2. For each u_s, properly add new vertices and edges to G_i by doing the breadth first search. Note the following:
   - When $|V_i| \ge M$, terminate the search. We set $M = 600$ for intra-site graphs and $M = 300$ for temporary intra-server graphs.
   - When the search visits v and if there is a page with URL w in the neighborhood of v such that w does not belong to the directory-based site corresponding to G_i, do the following:
     a. If there is no vertex in V with a label equal to a prefix part of w: Then add w to N_v and a new pair of URLs (v, w) to N_e.
     b. Otherwise: Let G_j be the intra-site graph containing w. If there is no vertex in V_j with a label equal to $pagename(w)$, add a vertex with label $pagename(w)$ to V_j. Moreover, if $(i, j) \notin E$, add a new edge (i, j) (corresponding to a global-link) to E.

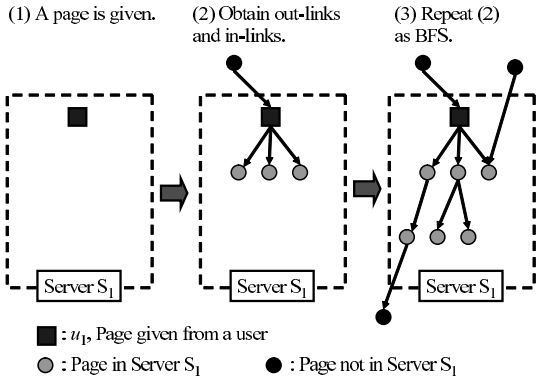

Fig. 2 Crawling pages in a server.

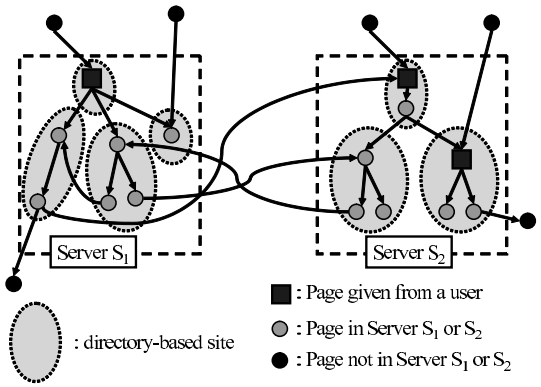

Fig. 3 Identifying directory-based sites.

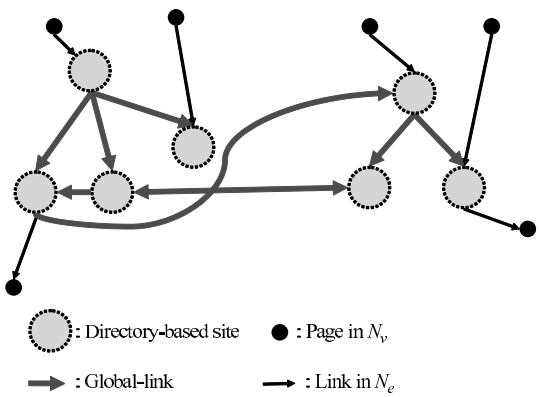

Fig. 4 A seed graph (the inter-site graph and the neighbors set are shown).

Note that we use an existing search engine, such as Google or Altavista, in order to find pages linked to u_s (i.e. in-link) and we use "libwww-perl" presented by W3C as a HTML parser in order to find pages links from u_s.

3.4 Growing the Seed Graph

By using the following *growth* procedure, NCF adds sites containing URLs in the neighbor sets (i.e., sites adjacent to sites in the seed graph) to G in order to grow the seed graph G. The inputs of the growth procedure are G, $R = \{G_i \mid 1 \leq i \leq |V|\}$, N_v and N_e.

Growing the Seed Graph

1. Set N'_v and N'_e to be empty.
2. Set $G' = G$, and $\{G'_i\} = \{G_i\}$.
3. Call **Construct-seedgraph**$(N_v, G', \{G'_i\}, N'_v, N'_e)$.
4. Update G, $\{G_i\}$, N_v and N_e by G', $\{G'_i\}$, N'_v and N'_e, respectively.

Repeating the procedure can grow the seed graph by one hop of global-link, and therefore our system is considered to grow the initial subgraph on the basis of the inter-site graph, in contrast, the previous works (HITS [9], Companion [7], and so on) grow a graph by one hop of a link on the basis of the Web graph. This difference would be significant for information retrieval, because growth by one hop of local-link yields no effect to results of HITS or Companion, but growth of one hop of global-link would affect the results. Note that the two kinds of growth cannot be distinguished unless we identify sites according to some proper model.

3.5 Enumerating Maximal Cliques

After at least one growth procedures, NCF finds neighbor communities in G by enumerating maximal cliques of mutual-links.

By using the `.jp` domain URL data sets, the authors have shown that maximal cliques in the mutual-link graph correspond to communities (even a K_2 corresponds to a community) in [2]. They have also shown an interesting fact that communities of personal sites are commonly found by this method, while such communities are very few in the communities obtained by trawling using the same data.

They have also shown the following reasons why a Web graph and an inter-server graph are not good for this method.

Fact 5: Even if there is a mutual-link between a site A and a site B, there can be no pair of page $a \in A$ and page $b \in B$ which links to each other.

This fact will be a problem when we use a Web graph to enumerate cliques. The upper picture in Fig. 5 shows an example of this fact.

Fact 6: When we use an inter-server graph, a mutual-link between a site A and a site B will be ignored if A and B belongs to the same Web server.

Fact 7: Even if there is a mutual-link between a server S_1 and another server S_2 in an inter-server graph, we cannot distinguish the following three situations. (1) there is a mutual-link between a site $A \in S_1$ and $B \in S_2$. (2) there are multiple mutual-links between sites in S_1 and sites in S_2. For example, even if there are a mutual-link between $A \in S_1$ and $B \in S_2$ and another mutual-link between $C \in S_1$ and $D \in S_2$, only one mutual-link between S_1 and S_2 is found in the inter-server graph. (3) there is no pair of sites $(A \in S_1, B \in S_2)$ linked each other, but there are a link from A to B and a link from a site $C \in S_2$ to a site $D \in S_1$ $((A, B) \neq (C, D))$.

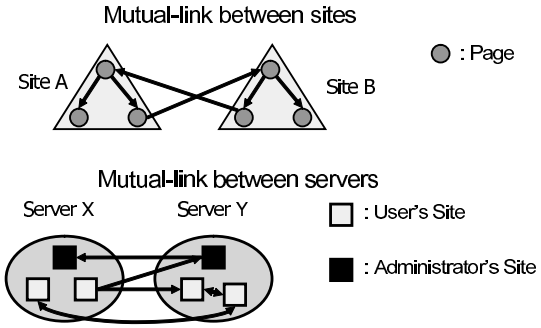

Fig. 5 Examples for Facts 5, 6, and 7.

The lower picture in Fig. 5 shows an example of these facts. These facts will be a problem when S_1 and S_2 are ISPs (or rental servers) and contain a number of sites of users. Thus, if we want to find a valuable relationship between personal sites on such servers, we must use the mutual-link graph.

These facts implies that mutual-links are useful for mining communities only when sites are obtained according to some proper model, such as our directory-based sites.

4. Experiments and Comparison with Google's Similar Pages Service

We compare communities obtained by NCF with pages obtained by Google's "Similar Pages" service. Our NCF can use multiple seed URLs as an input and this fact will be useful for finding communities related to user's interests since multiple seeds are more reliable data than a single seed. However, we use results for sets which consist of only one seed to compare with Google's service in fairness, since Google's service allows only a single URL as an input.

Table 3 shows comparisons of the communities (i.e. maximal cliques) obtained by NCF with the pages obtained by Google's "Similar Pages" service. "Number" column in "Cliques" columns (or "Google" columns) shows the number of cliques (or pages, respectively) obtained. "QoS" (Quality of samples) column in "Cliques" columns (or "Google" columns) shows the number of cliques which consist of related sites (or the number of related pages, respectively) to the seed URL in 20 samples (if obtained cliques or pages are less than 20, we use all of the cliques or pages).

The seeds corresponding to IDs 1 to 7 are personal sites given by voluntary users and the topics of them are mainly specialized hobbies and so on. IDs 1 and 2 (3 and 4) uses the same seed URL, but the number of applied growth procedures is one for ID 1 (3) and two for ID 2 (4, respectively). The details of the results for IDs 1 to 7 (e.g. sizes of graphs)

Table 3 Comparison with Google's "Similar Pages" service.

ID	Cliques		Google	
	Number	QoS	Number	QoS
1	6	6/6	16	0/16
2	83	19/20	16	0/16
3	9	8/9	0	0/0
4	156	17/20	0	0/0
5	15	15/15	15	13/15
6	13	10/13	3	0/3
7	28	15/20	5	3/5
8	5	5/5	25	16/20
9	12	11/12	0	0/0
10	7	7/7	30	13/20
11	24	15/20	0	0/0
12	3	3/3	7	5/7
13	5	5/5	0	0/0
14	14	13/14	0	0/0
15	149	19/20	24	19/20
16	139	20/20	28	18/20
17	8	8/8	0	0/0
18	46	20/20	24	20/20
19	16	15/16	25	10/20

are shown in [2]. The seeds of IDs 8 to 19 are sites registered on Yahoo! Japan for 10 topics. For each topic, we select one public site and one personal site. IDs of even (odd) numbers are corresponding to public (personal) sites. IDs 8 and 9 are sites about cooking, 10 and 11 are sites about news, 12 and 13 are about investment, 14 and 15 are about movies, 16 and 17 are about models, 18 and 19 are about armies.

As a result, in most cases our NCF returns better results than Google's service in both quantity and quality. In particular, when seeds are personal sites, the results of NCF are much better. For IDs 1 and 2, Google's service returns 16 pages, but there are no related pages in them, in contrast to most of the maximal cliques represent communities having the same topic as the seed. For ID 6, a similar result can be seen. For IDs 3, 4, 9, 11, 12, 13, and 17, Google's service returns no pages, while most of the maximal cliques (i.e. results of NCF) have good quality. These bad results of Google's service will be due to that these seed pages are personal sites having relatively specialized topics or they contains many pictures and illustrations instead of poor text information. (Note that contents of these sites have good quality for their topics) However, NCF returns good results by using link information even under such difficult situations.

Google's service returns as good results as NCF in quality for IDs 5, 7, 8, 10, 12, 15, 16, and 18. In particular, for IDs 8, 10, and 12, Google's service returns better results in quantity than NCF. The seeds for these IDs are well-known sites for given topics and contain plenty of text information. Moreover, input sites for IDs 8, 10, and 12 have very few mutual-links. These results have shown that such situations are advantageous to Google's service, and it will be a future work to improve NCF by combining with our ideas using mutual-links and the ideas used by HITS or trawling.

As a result, we conclude that our NCF is useful to find communities in response to a user's query (i.e. seed pages). In particular, it is shown that NCF is suitable for finding communities of personal sites and specialized topics.

5. Concluding Remarks

In conclusion, we have presented a new tool for finding neighbor communities related to given URLs by users, named Neighbor Community Finder, on the basis of an inter-site graph. We have verified that communities obtained by NCF is better than "Similar Pages" service in Google in both quality and quantity for various topics, in particular when a given URL corresponds to a personal site. Note that [4] is a preliminary version of this paper. More experiments compared to other methods of finding related pages (e.g. [8], [11]) will be a future work.

On the other hand, we also try to apply our idea that using an inter-site graph instead of a Web graph to other research fields based on graph structures of links on the Web. We have shown that distinguishing an inter-site graph from intra-site graphs is useful for more reasonable drawing of the Web graph than existing tools. We have presented Web-Linkage Viewer, a visualization system of links on the Web understandably by drawing an inter-site graph on a spherical surface and drawing an intra-site graph for each site in a cone emanating from a point representing the site on the surface. We have also examined that our Web-linkage Viewer produces more understandable drawing of structures in the Web graph than existing tools using several examples. See [2] and [5].

References

[1] B. Amento, L.G. Terveen, and W.C. Hill, "Does "authority" mean quality? Predicting expert quality ratings of Web documents," Proc. SIGIR'00, pp.296–303, 2000.
[2] Y. Asano, A New Framework for Link-based Information Retrieval from the Web, Ph.D. Thesis, The University of Tokyo, March 2003.
[3] Y. Asano, H. Imai, M. Toyoda, and M. Kitsuregawa, "Applying the site information to the information retrieval from the Web," Proc. 3rd International Conference on Web Information Systems Engineering, pp.83–92, 2002.
[4] Y. Asano, H. Imai, M. Toyoda, and M. Kitsuregawa, "Finding neighbor communities in the Web using an inter-site graph," Proc. 14th International Conference on Database and Expert Systems Applications, pp.558–568, 2003.
[5] Y. Asano and T. Nishizeki, "Web-linkage viewer: Drawing links in the Web based on a site-oriented framework," Proc. 11th International Symposium on Graph Drawing (GD 2003), pp.498–499, 2003.
[6] N. Craswell, D. Hawking, and S. Robertson, "Effective site finding using link anchor information," Proc. SIGIR'01, pp.250–257, 2001.
[7] J. Dean and M.R. Henzinger, "Finding related pages in the World Wide Web," Proc. 8th International World Wide Web Conference, pp.389–401, 1999.
[8] G.W. Flake, S. Lawrence, and C.L. Giles, "Efficient identification of Web communities," Proc. 6th International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD-2000), pp.150–160, 2000.
[9] J. Kleinberg, "Authoritative sources in a hyperlinked environment," Proc. 9th Annual ACM-SIAM Symposium on Discrete Algorithms, pp.668–677, 1998.
[10] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins, "Trawling the Web for emerging cyber-communities," Proc. 8th International

World Wide Web Conference, pp.403–416, 1999.
[11] T. Murata, "Finding related Web pages based on connectivity information from a search engine," Poster Proc. 10th International World Wide Web Conference, 2001.
[12] M. Toyoda and M. Kitsuregawa, "Observing evolution of Web communities," Poster Proc. 11th International World Wide Web Conference, 2002.

Yasuhito Asano received B.S., M.S. and D.S. in Information Science, the University of Tokyo in 1998, 2000, and 2003, respectively. Since 2004, he joined Graduate School of Information Sciences, Tohoku University, and he is currently a research associate at Tohoku University. His research interests include web mining, network algorithms. He is a member of IPSJ, OR Soc. Japan.

Hiroshi Imai obtained B.Eng. in Mathematical Engineering, and M.Eng. and D.Eng. in Information Engineering, University of Tokyo in 1981, 1983 and 1986, respectively. In 1986–1990, He was an associate professor of Department of Computer Science and Communication Engineering, Kyushu University. He was also a visiting associate professor at School of Computer Science, McGill University in 1987 and a visiting scientist at IBM T. J. Watson Research Center in 1988. Since 1990, he joined Department of Information Science, University of Tokyo, and he is currently a professor of Department of Computer Science. His research interests include algorithms, computational geometry, optimization, and quantum computation and information. He is also a member of IPSJ, OR Soc. Japan, IEEE and ACM.

Masashi Toyoda received the B.E. in science in 1994, the Master and Doctor of Science from Tokyo Institute of Technology, in 1996 and 1999, respectively. In 1999, he joined Institute of Industrial Science, University of Tokyo as a research fellow. He is currently an associate professor at the University of Tokyo. His research interests include web mining, information visualization, and user interface. He is a member of IPSJ, JSSST, IEEE and ACM.

Masaru Kitsuregawa received the B.E. degree in electronics engineering in 1978, and the PhD degree in information engineering in 1983 from the University of Tokyo. In 1983 joined Institute of Industrial Science, the University of Tokyo as a lecturer. He is currently a professor at the University of Tokyo and is a director of center for information fusion. His research interests include database engineering, data mining, advanced storage system. He is a member of steering committee of IEEE ICDE, PAKDD etc. He served a chair of ACM SIGMOD Japan Chapter and was a trustee member of VLDB endowment and the chairman of the technical group on data engineering in IEICE Japan. He is a director and a fellow of Information Processing Society of Japan. He is an associate editor of IEEE Transaction on Knowledge and Data Engineering. He also serves an advisor of Storage Networking Industry Association Japan Forum.