

Finding Neighbor Communities in the Web using Inter-Site Graph

Yasuhito Asano¹, Hiroshi Imai², Masashi Toyoda³, and Masaru Kitsuregawa³

¹ Graduate School of Information Sciences, Tohoku University

² Graduate School of Information Science and Technology, the University of Tokyo

³ Institute of Industrial Science, the University of Tokyo

Abstract. In recent years, link-based information retrieval methods from the Web are developed. A framework of these methods is a Web graph using pages as vertices and Web-links as edges. In the last year, the authors have claimed that an inter-site graph using sites as vertices and global-links (links between sites) as edges is more natural and useful as a framework for link-based information retrieval than a Web graph. They have proposed *directory-based sites* as a new model of Web sites and established a method of identifying them from URL and Web-link data. They have examined that this method can identify directory-based sites almost correctly by using data of URLs and links in .jp domain. In this paper, we show that this framework is also useful for information retrieval in response to user's query. We develop a system called **Neighbor Community Finder** (NCF, for short). NCF finds Web communities related to given URLs by constructing an inter-site graph with neighborhood sites and links obtained from the real Web on demand. We show that in several cases NCF is a more effective tool for finding related pages than Google's service by computational experiments.

1 Introduction

In recent years, information retrieval methods from the Web using characteristic graph structures of the Web-links are developed. HITS proposed by Kleinberg [9] and Trawling proposed Kumar et al. [10] are examples of well-known such methods. Such information retrieval methods are based on the following idea: if page u has a link to page v , then page v is considered to contain valuable information by the author of u . Thus, these methods are considered to be algorithms running on a Web graph which consists of the pages as the vertices and the links as the edges, and it can be said that they treat a page as a unit of information.

If we consider a Web graph as a framework for link-based information retrieval as above, 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. However, 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 the top page and a page for links to other sites are different.

Therefore, we claim that *inter-site graph*, which consists of sites as vertices and links between sites as edges, is a more natural framework for link-based information retrieval than a Web graph. Since a method of identifying Web sites from URLs or HTML files had not been established, several researches have used a Web server instead of a Web site. Actually, HITS and Trawling use only links to pages in other servers or domains. This idea works relatively well when a Web site corresponds to a server such as official Web sites made by companies, governments or other social organizations, but works poorly when multiple Web sites correspond to a server such as personal Web sites on a server of internet service providers (ISPs) or universities, or rental servers and so on. This seems to be wasting valuable information, since information about relatively minor and specialized topics including important scientific results is frequently laid on such personal Web sites.

In 2002, the authors proposed a new model of Web sites, called a *directory-based site* model to deal with typical personal sites [2], [3]. In the directory-based site model, we regard a set of pages in a directory and all its subdirectories, and therefore if we can find directories corresponding to users' sites correctly from the Web, we can treat personal sites well. They have also proposed a method of identifying directory-based sites. It consists of several procedures called *filters* and an error correction phase. Each filter finds some Web servers and determines whether they contain only one site or multiple sites (i.e. two or more sites), and transfers the remaining servers to the next filter. They 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.

They have shown that an inter-site graph is more suitable for finding communities (i.e. sets of related sites) containing personal sites than a Web graph or an inter-server graph by using Trawling. They have also proposed a new information retrieval method utilizing mutual-links and shown that maximal cliques of mutual-links correspond to communities. These cliques contain a large number of communities of personal sites, although Trawling could find a small number of such communities. See [2].

Since Trawling and enumerating maximal cliques described above are not suitable for information retrieval in response to user's query such as Google's "Similar Pages" service, in this paper we present a new information retrieval tool, called a **Neighbor Community Finder** (NCF, for short), to find related communities in the neighborhoods of given URLs by users. This system first constructs an inter-site graph containing neighbor sites of the given sites,

by crawling required Web pages, and obtaining in-links by search engines, and identifying directory-based sites by the filters.

Then this system enumerates maximal cliques in this inter-site graph to find neighbor communities related to the given sites. We show that NCF is a more effective tool for finding related pages than Google's service in several cases by computational experiments.

The rest of this paper is organized as follows. In Section 2, we describe a new framework of link-based information retrieval using a site as a unit and a method of implementing this framework. In Section 3, we propose NCF and describe how it works. In Section 4, we show several results of NCF and compare them with results of Google's service. In Section 5, we describe concluding remarks.

2 Site-oriented Framework for Information Retrieval

In this section, we describe our site-oriented framework for information retrieval from the Web proposed in [2]. First, we describe a new model of Web sites, called *directory-based sites*, since a phrase "Web site" is used ambiguously in our daily life, and therefore it is hard to present a unique definition of the Web site. For example, the following definition which seems not to be apart from the concept used in our daily life. Note that similar definition is found in [1] and [6], although they did not find sites from the whole Web according to their definition.

Definition 1. *A Web site is a set of Web pages that are written by a single person, company, or 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. Unfortunately, such information does not exist in the real Web and therefore it is hard to compute Web sites according to this definition. Thus, we have to consider a method of estimating Web sites under a restricted situation, such as our directory-based sites described below.

Next, we describe our method of identifying directory-based sites, called *filters*, and summarize the results for the jp-domain data sets collected in 2000 and 2002 by Toyoda and Kitsuregawa. Then, we describe the definition of an *inter-site graph* with directory-based sites as vertices.

2.1 Directory-based Site

Definition 2. [2]: *For a Web server, let $\{d_1, \dots, d_k\}$ 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 such of which is 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 $\{d_1, \dots, d_k\}$ (and 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.*

If all pages in a given server are in the site of the administrator of the server (i.e. $k = 0$ in Definition 3), 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*.

2.2 Filters

We now describe an outline of our method of identifying directory-based sites. It 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 ($0 \leq i \leq 6$). Note that the remaining Web servers after these filters are regarded as single-site servers.

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. For example, it is well-known that in

tt geocities.co.jp the 3rd level directories are the top directories of sites of users.

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. For example, it is well-known that `www.sony.co.jp` is a single-site server.

Filter 3: considers any server having at most one directory as a single-site server. **Filter 4:** considers any server which has at most 20 pages as a single-site server.

Filter 5: for each server, we consider its associated graph with pages in the server as vertices and links between these pages as edges, and decompose it into the connected components. Then, regarding 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, find multi-site servers and a level of directories corresponding to top directories of sites of users. Frequently, these directories have few back-links and a number of these directories are much larger than a number of parent directories of them.

ECFC: it enumerates maximal cliques of the directory-based sites found in Filters 5 and 6, and finds any clique such that every directory-based site in the clique belongs to one server. It removes such servers from the results of Filters 5 and 6, then regards them single-site servers.

The authors have examined this method by using the jp-domain URL data sets. The filters and ECFC have identified 74,441 servers among 112,744 servers and found 563,611 directory-based sites for the data set in 2000. For the data set in 2002, they have identified 299,785 servers among 373,737 servers and found 1,975,087 directory-based sites. They have also estimated error rate of this method by sampling 150 servers randomly from the identified servers by each filter and ECFC. 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 [4], [3].

2.3 Inter-site Graph

Now, as a framework for information retrieval, we can use an inter-site graph or a mutual-link graph defined as follows. For convenience, we also 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 sites as vertices and mutual-links as edges is called a **mutual-link graph**. (4) A graph which consists of servers as vertices and links between servers as edges is called an **inter-server graph**. (5) 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 Neighbor Community Finder

3.1 Outline of the System

We describe the outline of NCF. As an input, receive at least one URL from the user. Let these URLs be $\{u_1, \dots, u_h\} = U$ and S_i be the server containing u_k for $1 \leq k \leq h$. The detail of each step is described in Section 3.2 to 3.4.

1. Construct a *seed graph* G . A seed graph is the inter-site graph which consists of directory-based sites in $\{S_1, \dots, S_h\}$ 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.

We also prepare a filter database describing our knowledge used in Filters 0 and 2 for NCF. 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 “http://www.geocities.co.jp/Playtown-Denei/1722/src/SRC.html”, $sitename(u)$ is “http://www.geocities.co.jp/Playtown-Denei/1722/” and $pagename(u)$ is “src/SRC.html”.

3.2 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 Web-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 Web-links. Each vertex $v \in V$ has a label $label(v)$ corresponding to some part of its URL.

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

1. For each URL $u \in U$, do the following “new URL addition” 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 ECF, 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 ($1 \leq i \leq |V|$), 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 $|V_i| \geq M$, 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 $\text{pagename}(w)$, add a vertex with label $\text{pagename}(w)$ to V_j . Moreover, if $(i, j) \notin E$, add a new edge (global-link) (i, j) to E .

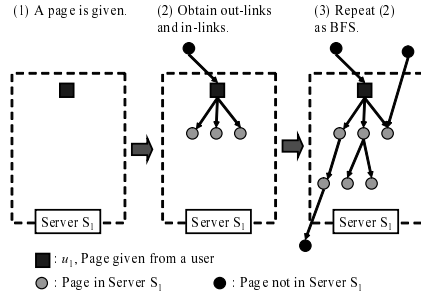


Figure 1. Crawling pages in a server.

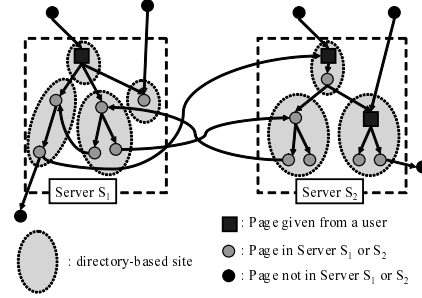


Figure 2. Identifying directory-based sites.

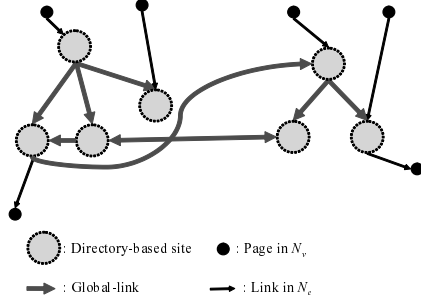


Figure 3. A seed graph (the inter-site graph and the neighbors set are shown).

Figures 1 to 3 illustrate the outline of the construction of a seed graph. 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 .

3.3 Growth of 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 .

Growth

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 Web-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.

3.4 Enumerating Maximal Cliques

After the growth procedures, NCF finds neighbor communities in G by enumerating maximal cliques formed 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) and communities of personal sites occupy relatively large amount. Note that such communities are very few in the results of Trawling using the same data. They have also shown in [2] that a Web graph and an inter-server graph are not good for this method. This fact has also shown that mutual-links are useful for information retrieval only when sites are obtained according to some proper model.

3.5 Experiments and Comparison with Google's Similar Pages Service

We also 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 1 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. "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) 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,

Table 1. Comparison with Google’s “Similar Pages” service.

ID	Cliques		Google	
	Number	Quality of samples	Number	Quality of samples
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

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 several 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.

On the other hand, for IDs 8, 10, and 12, Google’s service returns better results than NCF. Google’s service returns as good results as NCF in quality for IDs 5, 7, 15, 16, and 18. The seeds for these IDs are well-known sites for given topics and contain plenty of text information, but having 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 a useful tool to find communities in response to 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.

4 Concluding Remarks

In conclusion, we have shown that our site-oriented framework is useful for information retrieval in response to user's query by developing **Neighbor Community Finder**, a tool to find communities related to given URLs by users. We have also shown comparison with Google's service. 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 consider other applications of our site-oriented framework to several research fields based on graph structures of Web-links. We have shown that distinguishing global-links from local-links is useful for constructing more reasonable drawing of the Web graph than existing tools. We have presented **Web-Linkage Viewer**, a visualization system drawing Web-links understandably by drawing sites and global-links on a spherical surface and drawing pages and local-links in cones emanating from a point representing a site on the surface. We 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. In *Proceedings of SIGIR'00*, pages 296–303, 2000.
2. Y. Asano. *A New Framework for Link-based Information Retrieval from the Web*. PhD thesis, The University of Tokyo, December 2002.
3. Y. Asano, H. Imai, M. Toyoda, and M. Kitsuregawa. Applying the site information to the information retrieval from the Web. In *Proceedings of the 3rd International Conference on Web Information Systems Engineering*, pages 83–92, 2002.
4. 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*, pages 154–159, 2002.
5. 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.
6. N. Craswell, D. Hawking, and S. Robertson. Effective site finding using link anchor information. In *Proceedings of SIGIR'01*, pages 250–257, 2001.
7. J. Dean and M. R. Henzinger. Finding related pages in the World Wide Web. In *Proceedings of the 8th International World Wide Web Conference*, 1999.
8. G. W. Flake, S. Lawrence, and C. L. Giles. Efficient identification of Web communities. In *Proceedings of the Sixth International Conference on Knowledge Discovery and Data Mining (ACM SIGKDD-2000)*, pages 150–160, 2000.
9. J. Kleinberg. Authoritative sources in a hyperlinked environment. In *Proceedings of the 9th Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 668–677, 1998.
10. 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.
11. T. Murata. Finding related Web pages based on connectivity information from a search engine. In *Poster Proceedings of the 10th International World Wide Web Conference*, 2001.