Creating a Web Community Chart for Navigating Related Communities

Masashi Toyoda and Masaru Kitsuregawa
Institute of Industrial Science, University of Tokyo
4-6-1 Komaba Meguro-ku, Tokyo, JAPAN
toyoda, kitsure@tkl.iis.u-tokyo.ac.jp

ABSTRACT

Recent research on link analysis has shown the existence of numerous web communities on the Web. A web community is a collection of web pages created by individuals or any kind of associations that have a common interest on a specific topic. In this paper, we propose a technique to create a web community chart, that connects related web communities, from thousands of seed pages. This allows the user to navigate through related web communities, and can be used for a 'What's Related Community' service that provides not only the web community including a given page but also related web communities. Our technique is based on a related page algorithm that gives related pages to a given page using only link analysis. We show that the algorithm can be used for creating the chart by applying the algorithm to each seed, then using similarities of the results to classify seeds into clusters and to deduce their relationships. We perform experiments to create a web community chart of companies and organizations from thousands of seed pages. First, we improve the precision of an existing related page algorithm, Companion, and evaluated the improved version, Companion–, by an user study. Then the chart is created using Companion–. The result chart consists of web communities including related pages, and paths between related web communities. From the chart, we can find many web communities of companies classified by their category of business, and relationships between the communities.

KEYWORDS: World Wide Web; Link analysis; Web community; Related web communities

INTRODUCTION

Recent research on link analysis has shown the existence of numerous *web communities* on the Web. A web community is a collection of web pages created by individuals or any kind of associations with a common interest on a specific topic, such as fan pages of a baseball team, and official pages of computer vendors. Some link analysis techniques [8, 5, 12, 10, 7] consider the Web as a graph, which nodes are web pages and edges are hyperlinks, and automatically identify such web communities by extracting distinctive graph structures. Web communities slightly differ from real communities. That is, web communities may consist of competitors or authors who do not know each other, because they have similar graph structure. In the following, we use the term "community" for web community.

Using those techniques, one can know the existence of the community on an interesting topic, and can collect various information on the topic from pages in the community. However, we cannot know the existence of communities on related topics, since those techniques have not concerned relationships between communities.

Our goal is not only to identify communities but also to create a global *web community chart* that connects related communities, so that the user can navigate through related pages and communities. For example, if we want to know about a computer before buying it, we can collect information from the users community of the computer. Then, we can navigate to the community of computer vendors for checking other computers, or to the community of computer shops for buying the computer. The web community chart allows the user to perform new type of navigation through the Web. It provides additional paths not only to related pages, but also to related communities.

The web community chart can be also used for a community version of 'What's Related' service. Originally, the 'What's Related' service provides only related pages from a given page (that is the current browsing page in Netscape, and one of the keyword search results in such as Google, Altavista, etc.). Using the web community chart, we can provide an extended 'What's Related' service, which provides not only a community (a set of related pages to the given page), but also shows other related communities.

As the first step to our goal, we developed a technique that creates a subset of the global web community chart from thousands of seed pages on a broad topic. To identify com-

HT'01 8/01 Aarhus, Denmark
® 2001 ACM ISBN 1-59113-420-7/01/0008…$5.00

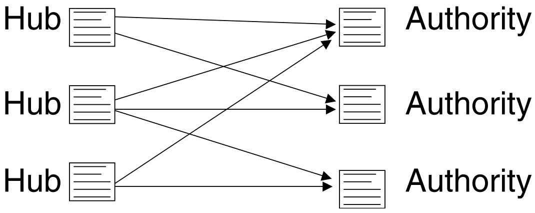

Figure 1: Typical graph structure of hubs and authorities

munities and to deduce relationships between communities, we use a modified version of a related page algorithm, Companion [6], which provides related pages to a given page. First, we extend the seed set by applying the algorithm to each page, and adding related pages into the seed set. Then we again apply the algorithm to each page in the extended seed set, and investigate how each page derives other pages as related pages. Using these derivation relationships between pages, we classify pages into communities, and connect related communities for navigation.

In this paper, we create a web community chart using around 5000 pages of companies and organizations. The result chart consists of communities including related companies, and paths between related communities. From the chart, we can find many communities of companies classified by their category of business, and relationships between the communities. For example, there are paths between a community of computer vendors, one of software companies, and one of computer device companies.

The rest of this paper is organized as follows. We first review related work, and describe our method for creating the web community chart. Then we explain our experiments, that includes the modification of Companion, the user study of the modified version, the details of the chart creation, and results. Finally, we discuss the results, and summarize the paper.

RELATED WORK

Most research on web communities [8, 5, 12, 10] is based on the notion of *authorities* and *hubs* proposed by Kleinberg [9]. An authority is a page with good contents on a topic, and is pointed to by many good hub pages. A hub is a page with a list of hyperlinks to valuable pages on the topic, that is, points to many good authorities. HITS [9] is an algorithm that extracts authorities and hubs from a given subgraph of the Web with efficient iterative calculation. Figure 1 shows a typical graph structure extracted by HITS. As shown in the graph, HITS extracts frequently co-cited pages as authorities.

A set of authorities and hubs was regarded as a community in [8, 5, 12, 10]. Gibson et al. [8] investigated the characteristic of communities derived by HITS. Chakrabarti et al. [5] improved the HITS algorithm by exploiting anchor texts, and evaluated result communities. Kumar et al. [12, 10] performed trawling on a huge snapshot of the Web, and found more than 100,000 communities. The trawling found communities by extracting complete bipartite graphs that consist of authorities and hubs.

HITS can also be used to find pages related to a given seed page. Finding related pages is similar to finding a community including the seed. Our method is based on a related page algorithm, Companion [6] proposed by Dean et al., which takes a seed page as an input, then outputs related pages to the seed. Dean et al. specialized HITS [9] for finding related pages, and improved the precision by exploiting link weighting and the order of links in a page. Companion first builds a subgraph of the Web near the seed, and extracts authorities and hubs in the graph using HITS. Then authorities are returned as related pages.

In addition, Flake et al. [7] redefined a community including given seed pages as a subgraph that is separated from the Web using a maximum flow / minimum cut framework.

These techniques can automatically identify individual communities, however, have not concerned the relationship between communities. To build the web community chart, we use the notion of authorities and hubs not only to identify communities, but also to deduce their relationships.

Recent document clustering approaches on the Web, such as [11, 3], have also exploited link analysis for clustering web pages, and [11] also considered relationships between clusters. Pitkow and Pirolli [11] proposed clustering algorithms using co-citation analysis. They performed hierarchical clustering to show relationships between clusters by a hierarchy. Rather, we create a graph of communities, since, in our experiments, the relationships between communities are too complicated to represent only by a hierarchy.

METHOD FOR CREATING A WEB COMMUNITY CHART

The main idea of our method is applying a related page algorithm to a number of pages, then investigate how each page derives other pages as related pages. To identify web communities and to deduce their relationships, we first put focus on the relationship between a seed page and derived related pages by the algorithm.

Consider that a page s derives a page t as a related page, and t also derives s as a related page. This often means that the both pages s and t are pointed to by similar sets of hubs. For example, a fan page of a baseball team derives other fan pages as related pages. When we apply the related page algorithm to one of the other fans, the page derives the original fan, because those fan pages are mutually linked by each other, that is, pointed to by similar sets of hubs. If each fan derives other fans as related pages, we can consider that these fans form a fan community.

Then, consider that a page s derives a page t as a related page, but t does not derive s as a related page. This means that t is pointed to by many different hubs, so that t derives a different set of related pages excluding s. For example, a fan page of a baseball team often derives an official page of the team as one of related pages. However, when we apply the algorithm to the official page, it derives official pages of other teams as related pages instead of the fan page. This is due to the fact that the official page of the team is often linked together with official pages of other teams in a number of more generic hubs, and the number of such hubs is greater than the number of hubs for the fans. In this case, we can consider that the official page is related to the fan community, but the page itself is a member of the baseball team community. This is the mechanism by which we find related communities.

Under these observations, we put focus on the former *symmetric derivation relationship* for identifying communities. Using this symmetric relationship, we refine the definition of communities and their relationships. We define that a community is a set of pages strongly connected by the symmetric relationships, and that two communities are related when a member of one community derives a member of an another community.

IMPROVING THE PRECISION OF COMPANION

Since the central part of the web community chart includes popular web pages, the result of the related page algorithm should be precise with popular pages. However, HITS [9] and Companion [6] provide insufficient precision.

Therefore, we first improved the precision of Companion, and performed an user study to evaluate the modified version of Companion, that we call Companion– here. In this user study, we compared the precision of Companion– with two related page algorithms, HITS [9] and Companion [6], and found that Companion– provides better precision.

Previous Related Page Algorithms HITS and Companion, and Our Modified Algorithm Companion–

In this section, we describe details of the three algorithms, and mention differences between these algorithms.

Build the Vicinity Graph First, each algorithm builds a vicinity graph, which is a subgraph of the web around the seed. A vicinity graph is a directed graph, (V, E), where nodes in V represent web pages, and edges in E represent links between these pages. Vicinity graphs for three algorithms are shown in Figure 2.

The vicinity graph for HITS (the top of Figure 2) is a collection of nodes that can be reached from the seed page in two steps following incoming and outgoing links. It is the same graph used in [9] to find related pages. If a node has more than Nb incoming links, Nb links are randomly selected. This incoming link selection is also performed in the other two algorithm, Companion and Companion–.

The vicinity graph for Companion (the center of Figure 2)

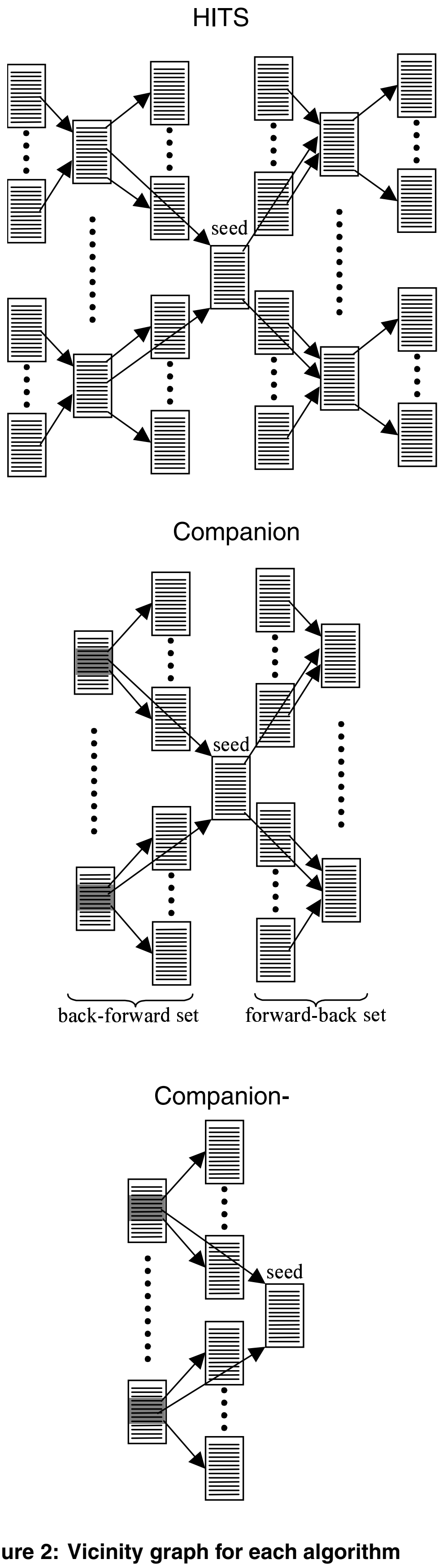

Figure 2: Vicinity graph for each algorithm

includes nodes that can be reached from the seed page by following incoming links then outgoing links (*back-forward set*), and by following outgoing links then incoming links (*forward-back set*). When following outgoing links from each node pointing to the seed in the back-forward set, not all the links are followed but only R links immediately preceding the link pointing to the seed, and R links immediately succeeding the link.

The vicinity graph for Companion– (the bottom of Figure 2) includes only the back-forward set of the vicinity graph for Companion. The only difference between Companion and Companion– is the vicinity graph.

After building the vicinity graph, near-mirror pages are eliminated from the graph. To detect near-mirrors, we use a *shingling* method [4], proposed by Broder et al., in the same way with the trawling paper [12]. This method calculates hash values for each page from (fixed length) sequences of hyperlinks in the page, then compares some of the smallest hash values. Then two pages that include the same sequence of links are detected as near-mirrors. In the original Companion algorithm, near-mirrors are not eliminated but combined into a single node by aggregating edges from the near-mirrors. However, in this experiments, we eliminate near-mirrors.

In addition, in all algorithms, we did not use stop URLs that should be eliminated from the vicinity graph. Originally, Companion eliminated 21 stop URLs that are unrelated to most queries and have numerous incoming links, such as www.yahoo.com.

In the following user test, we chose R to be 10, and Nb to be 2000. It was also reported in [6] that selecting a small value for R and a large value for Nb was better than selecting moderate values such as 50 and 50. Those parameters worked better in our experiments, while the original Companion chose R to be 4, and Nb to be 2000.

As mentioned above, Companion in our experiments is slightly different from the original setting. Therefore, note that the following user test is not the precise comparison with regard to the Companion algorithm.

Assign Weights to Edges To each edge, we assign two kinds of weights, an *authority weight* and a *hub weight* for decreasing the influence of a single server. The authority weight is used for calculating an authority score of each node, and the hub weight is used for calculating a hub score of each node. Companion uses the following weighting method proposed by Bharat and Henzinger [2], and we also use that method in Companion–. In HITS, each edge has the value 1 for both weights.

- If two nodes of an edge have the same server part in their URLs, the edge has the value 0 for both weight.
- If one node has n incoming edges from nodes in the same server, we assign each edge an authority weight of $1/n$.
- If one node has m outgoing edges to nodes in the same server, we assign each edge a hub weight of $1/m$.

Calculate Hub and Authority Scores Then we calculate a hub score, $hub(n)$, and an authority score, $auth(n)$ for each node n in the vicinity graph, (V, E). The following is the process of the calculation, where $auth\_weight(n, m)$ and $hub\_weight(n, m)$ represent the authority weight and the hub weight of the edge from n to m, respectively.

Step 1. Initialize $hub(n)$ and $auth(n)$ of each node n to 1.

Step 2. Repeat the following calculation until $hub(n)$ and $auth(n)$ have converged for each node n.

For all node n in V,

$$hub(n) \leftarrow \sum_{(n,m)\in E} auth(m) \times hub\_weight(n, m)$$

For all node n in V,

$$auth(n) \leftarrow \sum_{(m,n)\in E} hub(m) \times auth\_weight(m, n)$$

Normalize $hub(n)$, so that the sum of squares to be 1.

Normalize $auth(n)$, so that the sum of squares to be 1.

Step 3. Choose nodes with the N highest authority scores as results.

User Study

Using those algorithms, we performed the user study as follows.

Data Set and Experimental Environment Our data set for experiments is an archive of Japanese web pages. The archive includes about 17 million pages (90GB) in the 'jp' domain, or ones in other domains but written in Japanese characters. We collected these pages from July to September 1999 by running a simple web crawler that collects web pages from given seed pages in the breadth-first order.

From the archive, we built a connectivity database that can search outgoing and incoming links of a given page. Basic functions of the database were similar to the connectivity server [1] developed in DIGITAL, Systems Research Center. Our database indexed about 120 million hyperlinks between about 30 million pages (17 million pages of pages in the archive, and 13 million pages pointed to by pages in the archive). We implemented the whole system on Sun Enterprise Server 6500 with 8 CPU and 4GB memory. All the following experiments, including the web community chart creation, were performed on this system.

Subjects Ten volunteers served as subjects in the user test. The subjects consisted of an assistant professor, two assistants, two postdoctoral researchers, and five students in our university. All the subjects usually use the WWW.

URL of seed pages	Short description	# of inlinks	HITS	Companion	Companion–
weather.is.kochi-u.ac.jp/	Kochi Univ., Weather Home	1205	6/10	5/10	9/9
www.watch.impress.co.jp/pc/index...	PC Watch	1056	5/10	6/10	9/10
www.peugeot.co.jp/	Official Peugeot Japan	423	10/10	10/10	10/10
www.mahjong.or.jp/	Mahjong Walker	168	2/9	0/10	9/9
www.maccentral.or.jp/pokemon/	Pokemon site	164	2/10	9/10	10/10
www.ops.dti.ne.jp/~glass/	Stock market information	104	5/8	7/8	10/10
www.red-hell.com/	Urawa Reds (a soccer team) fan page	113	10/10	10/10	10/10
www.i-kochi.or.jp/prv/kochi/	Kochi Prefecture Information	109	1/10	2/10	8/8
www.japan.msf.org/	Medicines Sans Frontiers Japan	85	8/9	8/8	9/9
www2j.biglobe.ne.jp/~tatuta/	Free market information	71	0/10	0/10	9/10
www.panda.org/	WWF International	61	9/10	10/10	9/9
www.tintin.com/	Airline mileage service information	51	9/10	9/9	8/8
lang.nagoya-u.ac.jp/~matsuoka/Japan...	A Guide to Japan	43	0/10	0/10	7/8
www.spice.or.jp/~mt0711/index.html	Overseas travel Information	33	9/9	8/8	10/10
www.mars.dti.ne.jp/~o-shin/	Relational Database Information	26	4/10	8/10	10/10
www.triathlon.or.jp/	Triathlon World	26	9/9	10/10	10/10
www.alc.co.jp/nihongo/nihongo1.html	Japanese Language Center	23	5/9	10/10	8/10
plaza.harmonix.ne.jp/~kamao/	Virtual domain service information	18	0/10	2/10	7/9
www.isp.ne.jp/~nakajima/index.html	Movie information	15	0/10	2/10	8/8
islamcenter.or.jp/	Islamic Center Japan	12	7/10	7/10	7/10
www2e.biglobe.ne.jp/~TKG/	Puzzle information	8	0/9	0/10	8/8
www3.famille.ne.jp/~s370902/camera/...	Camera information	4	8/10	1/9	4/10
archives.math.utk.edu/popmath.html	POP Mathematics	1	8/10	7/9	7/10
home.att.ne.jp/green/asj	The Africa Society of Japan	1	6/9	8/10	7/10
Average precision			0.54	0.61	0.91

Table 1: # of related pages to # of accessible pages

Seed Set We asked each subject to give us some seed pages, such as, one of the pages collected on a topic by the subject, or pages that the subject want to find related pages. Twenty-four pages are collected as the seed set (1 to 4 pages from each subject).

Process To each seed page, we applied three algorithms, HITS, Companion, and Companion–, then made three lists of the top 10 authorities. Each subject were required to evaluate authority lists corresponding to seed pages, which the subject supplied to us. The subjects browsed each authority page with a web browser, check that the page can be accessed, and answered whether the page had contents on a related topic to the seed or not.

Result Table 1 is the result of evaluation by the subjects, which contains the number of related pages to the number of accessible pages for each authority list. Pages are sorted by their popularity indicated by the number of incoming links from other web servers. The average precision is an average of (# of related pages)/(# of accessible pages).

As shown in Table 1, most of the seeds are popular pages that have more than 10 incoming links[1]. Therefore, the precision of this result has much effect on the quality of the web community chart.

In most case, Companion– gave better results than HITS and Companion, and was outstanding at the average precision. This higher precision was obtained by narrowing the vicinity graph. We found that the forward-back set often included famous and unrelated pages, such as www.yahoo.com, and those pages absorbed authority scores. This *topic-drifting* lowered the precision of HITS and Companion. Stop URLs could prevent topic-drifting to stop URLs, however, could not prevent drifting to other famous URLs. Companion– provided good results without stop URLs by cutting the forward-back set, while the coverage of the result decreased.

CREATING A WEB COMMUNITY CHART

In this section, we describe algorithms for building the web community chart from a given seed set, and the result of our experiments. We first extend the seed set by applying Companion– to each seed and gathering results. Then, using the symmetric derivation relationship, seeds in the extended seed set are classified into communities, and related communities are connected by edges. We also perform experiments to create a web community chart of companies and organizations from thousands of seed pages. The result chart is mainly classified by categories of business, and the user can navigate related communities of companies and organizations. The following is the details of our algorithms and the result of our experiments.

[1] In our archive, less than 5% of pages have more than 10 incoming links

Seed Set

As a seed set, we use a manually maintained page list that includes 4,691 unique pages of companies, associations, organizations, and schools. Note that the seed set is a small subset of all the company pages in the archive.

Extending the Seed Set

We first extend the seed set, since the number of the seeds is too small to find sufficient symmetric relationships. The extended seed set is generated by applying Companion– to each seed separately. From each result, we select the top N authorities, and aggregate them into the extended seed set.

Building the Authority Derivation Graph

The second step is to build a directed graph that shows derivation relationships between seeds. Nodes are seeds in the extended seed set, and each directed edge, from a node s to an another node t, represents the fact that s derives t as one of the authorities using the Companion– algorithm. We create directed edges between nodes by applying Companion– to each node in the extended seed set, so that an edge from a node s to an another node t exists when s derives t as one of the top N authorities. This graph is called the authority derivation graph (ADG) in the following.

Then, we filter out nodes that derive unreliable results. A node s is filtered out, if s satisfies the following conditions:

- s does not derive at least N authorities with positive authority scores, or
- s is not included in the top N authorities derived from s itself.

The first case occurs when the neighborhood graph of s does not have enough nodes and links. The second case occurs when s is pointed to by many pages in the same server, and links referring to s are assigned small weights. In our experiments, we have found that Companion– often gives unreliable results in both case.

In our experiments, we chose N to be 10, since Companion– gives enough precision with top 10 authorities in the previous user study. We found that using the value 7 to 10 for N provides better quality and coverage of result communities than using larger or smaller values for N.

ADG built from our seed set includes 13,166 nodes and 70,201 edges. The size of ADG seems small, when considering the number of seeds. This result is due to the fact that only 1,633 of 4,691 seeds derive the reliable results. ADG is disconnected, and consists of one large connected component with 12,836 nodes, and 38 small connected components with less than 20 nodes.

Extracting the Symmetric Derivation Graph

The third step is to extract a symmetric derivation graph (SDG) from ADG. In this step, we put focus on the symmetric derivation relationships between nodes in ADG, that is, two nodes derive each other using Companion–. SDG includes nodes in A, and an edge from s to t exists when s and t are mutually connected in ADG.

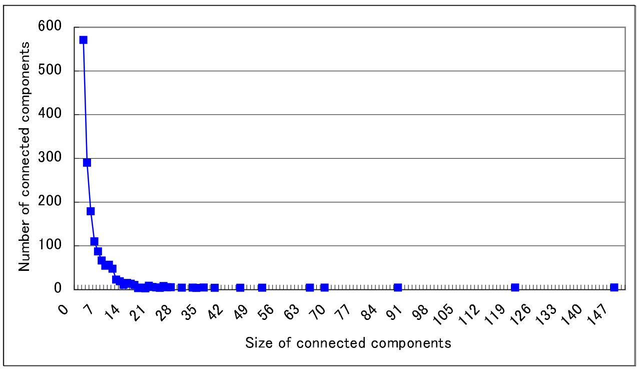

Figure 3: Distribution of connected components in the symmetric derivation graph

In our experiments, SDG is a disconnected graph with 8,362 nodes. Figure 3 shows the size and the number of each connected component in SDG. We found that almost all components consist of 2 to 20 nodes on the same category. However, there are some components including more than 20 nodes, that can be classified into multiple categories. Large components include more than 100 nodes in more than 10 categories, requiring further partitioning to identify more precise communities.

To show the necessity of further partitioning, we depict one of the connected components that includes 29 nodes on multiple categories in Figure 4. In total, this component can be regarded as a community of companies related to computer hardware. However, further observation of this component reveals that it includes three communities. There are computer vendors (NEC, TOSHIBA, SONY, etc.) on the top-left, companies of computer devices (Adaptec, Intel, Logitec, etc.) on the top-right, and companies of digital still camera (OLYMPUS, Minolta, etc.) at the bottom. In this case, we can partition the component into these three communities, by cutting the edge between any two communities.

Web Community Identification

This step identifies web communities by partitioning SDG. We observed the following facts from large connected components in SDG.

- Nodes in a *triangle* with edges share the same topic in most case, because a triangle is a complete graph, and each node derives other two nodes by Companion–. For example, triangles on the top-right of Figure 4 includes companies in the same category of business, such as Intel, Adaptec, and IO-DATA.
- Two triangles that share a edge also share the same topic in most case. In Figure 4, all the three communities include triangles connected by edges.

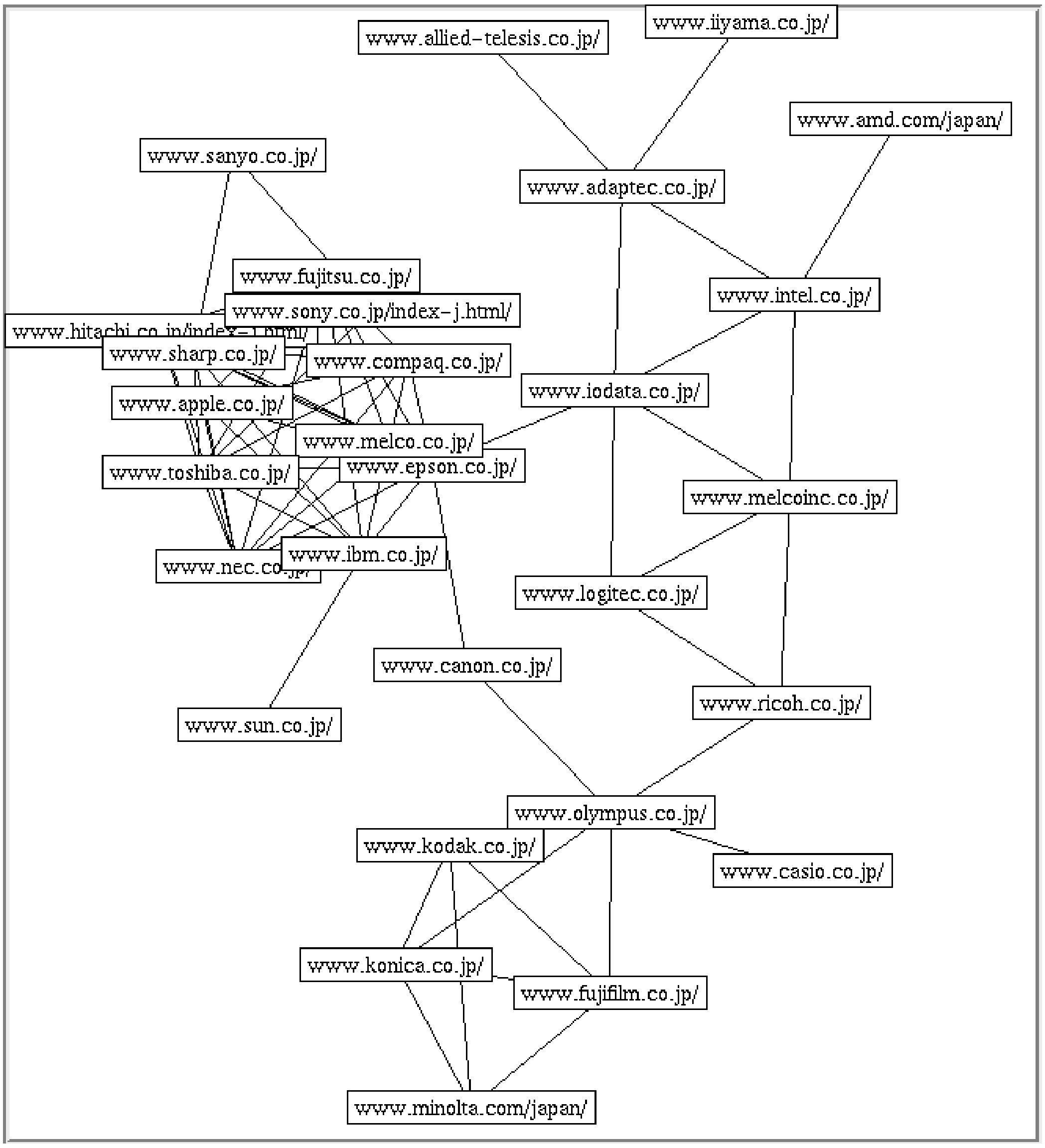

Figure 4: A connected component in the symmetric derivation graph

Under these observations, we use a node triangle as a unit of partitioning. To partition large connected components in SDG, we use a simple algorithm that finds densely connected *cores* of partitions, then adds isolated nodes to these cores. First, cores are made by extracting triangles connected by edges. This means that we extract all the graphs denser than a set of triangles connected by edges. Then we add isolated nodes to these cores. After finishing this process, every node in SDG becomes a member of a partition. The following explains the process in detail:

1. Extract triangles of nodes from SDG. Regard a subgraph with triangles connected by edges as a core.
2. Add each node to a core, if the node has edges connected to the core. Each core then becomes a partition. When there are multiple candidates, take into account of directed edges in ADG, that is, to select a core that has more incoming edges from the node in ADG.
3. Eliminate partitions from SDG.
4. Extract each remaining connected component in SDG as a partition.

Using this method, the connected subgraph in Figure 4 is clearly partitioned into three categories of business. An another large subgraph with 120 companies (of steels, construction machines, etc.) is also clearly partitioned into categories of business. In some subgraphs, we observed that companies

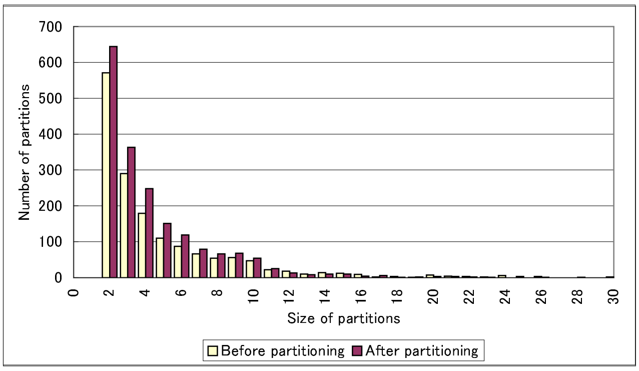

Figure 5: Distribution of partitions

in the same category are partitioned by their locations. There are also some subgraphs that are not clearly partitioned. One example is the largest subgraph with 147 nodes that consists of many hotels and golf courses. Although it is partitioned into some hotel communities and some golf course communities, boundaries between hotels and between golf courses are not clear.

We show the distribution of partitions in Figure 5, which shows the size and the number of partitions, before and after the partitioning process. After the partitioning, the size of the largest partition has decreased to 28. The number of partitions with less than 12 nodes has increased, and the number of partitions with 12 or more nodes has decreased.

Creating the Web Community Chart

Finally, we construct a web community chart that can be used to navigate from a community to other related communities. The chart is a directed graph that includes communities as nodes, and directed edges between related communities. Each edge has a weight that represents the strength of the relationship.

We make edges in the chart using ADG. We create a directed edge from a community c to another community d with a weight w, when there exists w directed edges in ADG from nodes in c to nodes in d.

Our chart includes 1,882 communities. Although we have not checked all the communities, we found many valuable communities. Each of them consists of companies (or organizations) doing the same category of business. In most case, we can also understand the relationships between the communities connected by edges. In the following, we show examples of the valuable communities. Communities in each item are strongly connected in the chart.

- Communities of companies related to computers, such as computer vendors and software companies
- Communities of the Linux operating system, such as Linux users groups and distribution package providers
- Communities of the mass media, such as TV stations, newspapers
- Communities of heavy industry, such as steels and construction machines
- A community of travel agents and related communities, such as hotels and car-rental companies
- Communities of companies related to music, such as online CD shops and music instrument shops
- A community of government agencies, and communities of related organizations

In Figure 6, we show a part of the web community chart that consists of communities connected by highly weighted edges. Each box represents a community that includes list of URLs. Note that the category label on each box is attached manually. In a community, each node is assigned a connectivity score that is a number of directed edges in ADG from the node to other nodes in the community. URLs in the box are sorted by the connectivity score in the descending order. The number attached to each directed edge denotes the weight.

In Figure 6, we chose the ‘Computer’ community as a center, since it has most edges in the chart. We selected only communities that have more than 15 edges between the ‘Computer’. Therefore, there are more communities, that are not shown in Figure 6, connected by lower weighted edges. For example, there are communities of computer shops and audiovisual equipment companies around ‘Computer’.

As shown in Figure 6, these communities are clearly classified and actually related to the ‘Computer’ community. On the top of Figure 6, there are three communities that are also in Figure 4. At the bottom, there are three more communities that have only outgoing edges to the ‘Computer’. The ‘Software’ community includes Lotus, Microsoft, Oracle, etc., and obviously related to the ‘Computer’. The companies in the ‘Cable’ community provides cables and optical fibers. The ‘Hitachi group’ community is slightly different from other communities. Although Hitachi is famous as a computer company, it is also one of the largest conglomerate in Japan. Since, all the companies in the ‘Hitachi group’ derive ‘www.hitachi.co.jp’ as one of authorities, the community has a highly weighted edge to the ‘Computer’.

We found that directed edges in the chart have a tendency to connect minor or specific communities to popular or general ones. For example, in Figure 6, there are more edges from ‘Computer device’ to ‘Computer’ than ones with the opposite direction. In this case, it shows that computer vendors are more general than computer device vendors.

A community may include incorrect companies, but such companies are assigned low connectivity scores in most case. For example, the last four companies in the ‘Hitachi group’ are not in the Hitachi group. Since, each company is connected by single edge in SDG, they are of the bottom of the

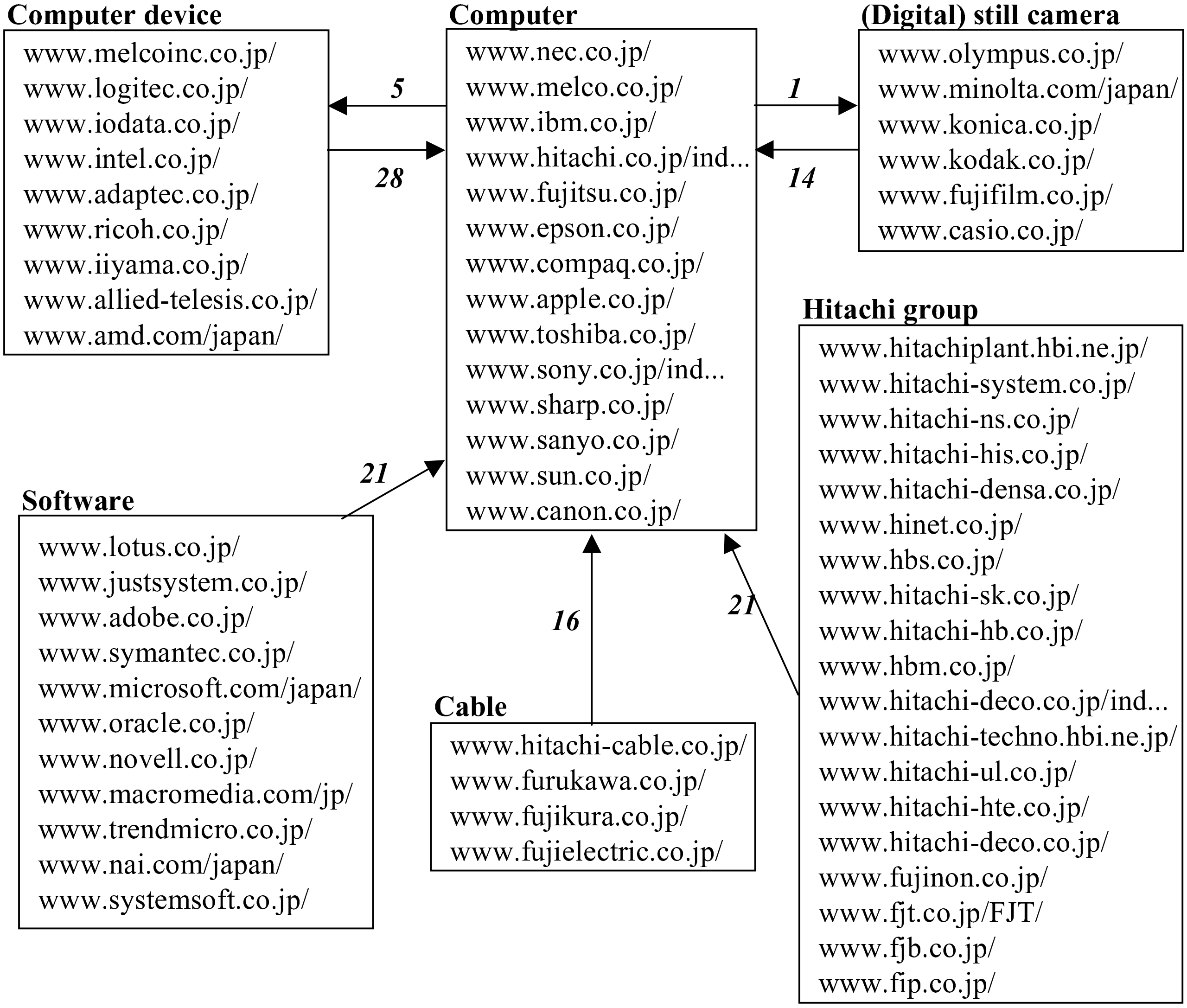

Figure 6: A part of the web community chart

list.

DISCUSSION

Using the web community chart, the user can navigate from a community to other related communities. The weights and directions of edges can be used as guideposts for deciding the next visiting community. The user can find closely related communities by following highly weighted edges. However, these keys are not sufficient for end users. Therefore, it is still required to attach appropriate labels to communities, and to integrate keyword search function. It is also difficult problem to describe differences between related communities.

The web community chart can be also used for 'What's Related Communities' service, which provides not only a community (a set of related pages to the given page), but also shows other related communities. To realize this service, we first show a community including the given page, then also show neighboring communities. Sorting these communities by edge weights makes it easier to select the next visiting community.

The results of our technique depend on the web archive, the seed set, and the parameter N for building ADG. The web archive used in our experiments is a small subset of the entire Web, and the seed set is also a small subset of all the company pages in our archive. We are interested in applying our technique to a larger archive and a larger seed set, and now constructing new data sets. We are also planning to investigate how the results will be influenced by changing the archive and the seed set.

It is important to select an appropriate value for the parameter N for building ADG. Using a large value increases noise edges in ADG, and using a small value decrease coverage of ADG. In our experiments, using the value 7 to 10 for N provides better quality and coverage of result communities. When we use a smaller value than 7 for N, the coverage become less than half of the one using the value 10 for N. Using a larger value than 10 for N increases noise edges, and combine indirectly related communities into a single community. For example, when we use 12 for N, 'Computer' and 'Computer device' communities in Figure 4 are combined into a single community. The appropriate value for N may also depends on the size of the seed set. It is also future work to investigate the relationship between N and the size of seed set.

SUMMARY

We have proposed the technique to create the web community chart, which the user can navigate from one community to other related communities. Our technique is based on a related page algorithm, which provides related pages to a given page. First, we apply the algorithm to given seed pages, then investigate how each page derives other pages as related pages. To identify communities and their relationships, we introduce the notion of the symmetric derivation relationship, which two seed derives each other by the algorithm. We defined that a community is a set of pages strongly connected by the symmetric relationships, and that two communities are related when a member of one community derives a member of an another community.

Since existing related page algorithm, HITS and Companion provide insufficient precision, we first developed an improved algorithm, Companion–, and evaluate precision by an user study. Then, using Companion–, we created the web community chart from the seed set of companies and organizations. We have shown that the result chart consisted of clearly classified communities by their categories of business, and navigation paths between related communities.

The community chart allows the user to perform new type of navigation through the web. It provides additional paths not only to related pages, but also to related communities. The chart can also be used for the advanced version of the 'What's Related' service, which provides not only related pages to a given page, but also other related communities.

REFERENCES

1. Krishna Bharat, Andrei Broder, Monika Henzinger, Puneet Kumar, and Suresh Venkatasubramanian. The Connectivity Server: fast access to linkage information on the Web. In *Proceedings of the 7th International World Wide Web Conference*, 1998.

2. Krishna Bharat and Monika Henzinger. Improved Algorithms for Topic Distillation in a Hyperlinked Environment. In *Proceedings of ACM SIGIR '98*, 1998.

3. D. Boley, M. Gini, R. Gross, S. Han, K. Hastings, G. Karypis, V. Kumar, B. Mobasher, and J. Moore. Partitioning-Based Clustering for Web Document Categorization. *Desision Support Systems*, 27(3):329–341, 1999.

4. A. Broder, S. Glassman, M. Manasse, and G. Zweig. Syntactic clustering of the web. In *Proceedings of the 6th International World Wide Web Conference*, 1997.

5. S. Chakrabarti, B. Dom, P. Raghavan, S. Rajagopalan, D. Gibson, and Jon Kleinberg. Automatic resource compilation by analyzing hyperlink structure and associated text. In *Proceedings of the 7th International World Wide Web Conference*, 1998.

6. Jeffrey Dean and Monika R. Henzinger. Finding related pages in the World Wide Web. In *Proceedings of the 8th World-Wide Web Conference*, 1999.

7. Gary W. Flake, Steve Lawrence, and C. Lee Giles. Efficient Identification of Web Communities. In *Proceedings of KDD 2000*, 2000.

8. David Gibson, Jon Kleinberg, and Prabhakar Raghavan. Inferring Web Communities from Link Topology. In *Proceedings of HyperText98*, 1998.

9. Jon M. Kleinberg. Authoritative Sources in a Hyperlinked Environment. In *Proceedings of the ACM-SIAM Symposium on Discrete Algorithms*, 1998.

10. Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, and Andrew Tomkins. Extracting large-scale knowledge bases from the web. In *Proceedings of the 25th VLDB Conference*, 1999.

11. J. Pitkow and P. Pirolli. Life, death, and lawfulness on the electronic frontier. In *Proceedings of International Conference on Computer and Human Interaction*, 1997.

12. Sridhar Rajagopalan Ravi Kumar, Prabhakar Raghavan and Andrew Tomkins. Trawling the Web for emerging cyber-communities. In *Proceedings of the 8th World-Wide Web Conference*, 1999.