# Extracting Evolution of Web Communities from a Series of Web Archives

Masashi Toyoda
toyoda@tkl.iis.u-tokyo.ac.jp

Masaru Kitsuregawa
kitsure@tkl.iis.u-tokyo.ac.jp

Institute of Industrial Science, University of Tokyo
4-6-1 Komaba Meguro-ku, Tokyo, JAPAN

## ABSTRACT

Recent advances in storage technology make it possible to store a series of large Web archives. It is now an exciting challenge for us to observe evolution of the Web. In this paper, we propose a method for observing evolution of web communities. A web community is a set of web pages created by individuals or associations with a common interest on a topic. So far, various link analysis techniques have been developed to extract web communities. We analyze evolution of web communities by comparing four Japanese web archives crawled from 1999 to 2002. Statistics of these archives and community evolution are examined, and the global behavior of evolution is described. Several metrics are introduced to measure the degree of web community evolution, such as growth rate, novelty, and stability. We developed a system for extracting detailed evolution of communities using these metrics. It allows us to understand when and how communities emerged and evolved. Some evolution examples are shown using our system.

## Categories and Subject Descriptors

H.5.4 [**Information Interfaces and Presentation**]: Hypertext/Hypermedia; H.3.m [**Information Storage and Retrieval**]: Miscellaneous

## General Terms

Experimentation, Measurement, Algorithms

## Keywords

Web, Link analysis, web community, evolution

## 1. INTRODUCTION

Recent advances in storage technology make it possible to store and keep a series of large Web archives. It is now an exciting challenge for us to observe evolution of the web, since it has experienced dramatic growth and dynamic changes in its structure. We could see a lot of phenomena in the Web that correspond to social activities in the real world. For example, if some topic becomes popular in the real world, many pages about the topic are created, then good quality pages are pointed to by public bookmarks or link lists for that topic, and these pages become densely connected.

In this paper, we propose a method for observing the evolution of *web communities*. A web community is a collection of web pages created by individuals or associations with a common interest on a topic, such as fan pages of a baseball team, and official pages of computer vendors. Recent research on *link analysis* [7, 8, 10, 11, 12, 13, 15] shows that we can identify a web community on a topic by extracting densely connected structure in *the web graph*, in which nodes are web pages and edges are hyperlinks. The web community slightly differs from a community of people, for example, a web community may include competing companies.

Since a web community represents a certain topic, we can understand when and how the topic emerged and evolved in the Web. Such information is important in the following situations:

- Answering historical queries about topics on the Web. For example, how many and what kinds of web pages have been created related to the terrorist attack on America on September 11, 2001?
- Tracking and analyzing user communities of consumer products. For example, there are several groups of user sites on major PCs. Statistics, such as how many members (URLs) have appeared and disappeared, will be useful information. Marketing people can further examine individual pages of interesting communities to understand reputation of their products.
- Observing and tracking social and cultural trends over time would be interesting topics for sociological research. Actually, we are currently examining the gender movement in the Web with professors of a women's university in Japan.
- Observing the emergence of quality web communities on a specific topic. Finding emergent web communities according to users' interest is challenging task for new search services.

For extracting such information, we analyze evolution of web communities using four Japanese web archives crawled from 1999 to 2002 with 119 million pages in total. Statistics


*HT'03,* August 26–30, 2003, Nottingham, United Kingdom.

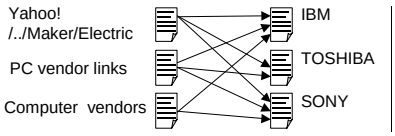


**Figure 1: Typical graph of hubs and authorities**

of these archives and community evolution are examined. From these results, we describe the global behavior of web community evolution. For describing evolution behaviors, we introduce several evolution metrics for communities that measure the degree of evolution, such as growth rate, novelty, and stability.

By using such metrics, users can extract detailed evolution of target communities. To examine their feasibility, we developed a system for extracting communities based on the evolution, which is composed of two parts. The first component extracts whole communities and their relevance from each web archive. It is based on our previous work of *web community chart* [17] that is a graph of communities, in which related communities are connected by weighted edges. The main advantage of our web community chart is existence of relevance between communities. We can navigate through related communities, and locate evolution around a particular community. The second component is a web community evolution viewer for browsing how communities evolved through three years. It provides various ways for locating the evolution of communities such as emerged and growing. With some examples, we demonstrate that we can easily locate interestingly evolving communities by combining evolution metrics and relevance.

## 1.1 Prior Work

Most research on web communities is based on the notion of *authorities* and *hubs* proposed by Kleinberg [13]. 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 [13] is an algorithm that extracts authorities and hubs from a given subgraph of the Web with efficient iterative calculation. Figure 1 shows an example graph structure extracted by HITS. Authorities are major computer companies such as IBM, TOSHIBA, and SONY. They are densely connected by hubs (link lists of computer companies). As shown in this graph, HITS extracts frequently co-cited pages as authorities. HITS has been improved and refined [4, 7, 8] by exploiting anchor texts, edge weighting, document similarity, and Document Object Models. Dean and Henzinger tailored HITS for finding related pages. They proposed a related page algorithm, Companion [10], which takes a seed page as an input, and outputs related pages to the seed, and improved the precision by considering the order of links in a page.

A set of authorities and hubs was regarded as a community in [12, 14, 16]. Gibson et al. [12] investigated the characteristic of communities derived by HITS. Kumar et al. [14, 16] extracted more than 100,000 cores of communities from a huge web snapshot based on their graph evolution model. A core was a small complete bipartite graphs that consist of authorities and hubs. Lempel and Moran [15] proposed another approach based on a random walk model for calculating authorities. Flake et al. [11] redefined a community including given seed pages as a subgraph that is separated from the Web using a maximum flow/minimum cut framework. Although these techniques can automatically identify individual communities, they have not considered relationships between communities. The evolution of communities has also not been examined using a series of web archives. In this paper, we examine how these identified communities evolve over time, and use relevance for locating evolutions around a given community.

The change frequency and lifetime of web pages has been studied in [5, 9]. They estimate frequency of web page modifications, and use the results for web crawlers to determine timing for re-crawl. They are based on the page level analysis. The site (or server) level analysis of web graph evolution is also studied in [3]. Rather, we analyze the evolution of communities in the Web. Recently, the Internet Archive began the Wayback Machine service [1] that allows us to see past web pages stored in the Internet Archive's web archive. Its capability is still limited. That is, the user can only specify a single URL, and see the past pages of that URL. It is impossible to understand what topics are popular in the past, which we are targeting in this paper.

## 1.2 Organization of the Paper

The rest of this paper is organized as follows. In Section 2, we briefly describe the web community chart and its browser. Section 3 introduces evolution of web communities, and evolution metrics that measure various changes in communities. In Section 4, we describe the details of our web archives, and the global behavior of community evolution. Section 5 demonstrates our web community evolution viewer with examples of evolution. In Section 6, we discussed detailed issues, and future work. Finally, we conclude in Section 7.

# 2. WEB COMMUNITY CHART

The web community chart is a graph that consists of web communities as nodes, and weighted edges between related communities. The weight of each edge represents the relevance of communities at both ends. In this section, we first explain intuition for underlying techniques, then briefly describe the algorithm for building the chart. Refer to [17], for more detailed descriptions.

## 2.1 Intuition for Underlying Techniques

Our algorithm builds the web community chart from a given set of seed pages. The main idea is applying a HITS [13] based related page algorithm (RPA) to each seed, and then investigate how each seed derives other seeds as related pages. RPA first builds a subgraph of the Web around the seed, and extracts authorities and hubs in the graph using HITS. Then authorities are returned as related pages. Since existing RPAs, such as HITS and Companion [10], provide insufficient precision, we use an improved algorithm, Companion– [17]. We have gained a better precision by pruning error prone parts of the subgraph used in HITS and Companion. The algorithm of Companion– is given in Appendix.

To identify web communities and to deduce their relationships, we put focus on relationships between a seed page and related pages derived by Companion–. Figure 2 depicts an example of these derivation relationships. In this example, we use five seed pages, IBM, TOSHIBA, SONY, and two

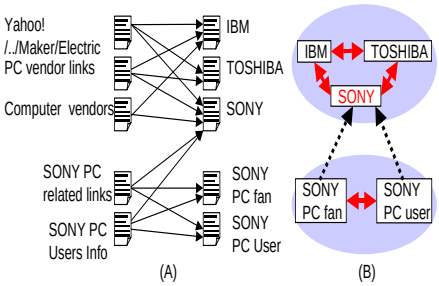


**Figure 2: An example of derivation relationships**

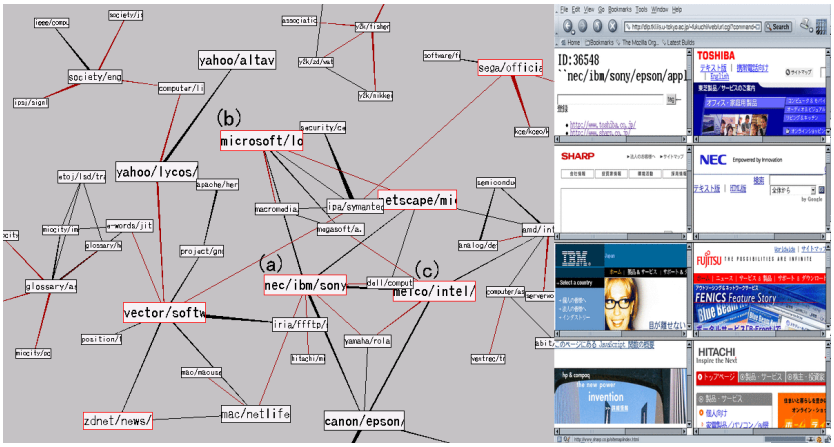


**Figure 3: A part of the web community chart**

fan pages of SONY PC. In Figure 2, the graph (A) shows how each seed is pointed to by hub pages, and the directed graph (B) shows how each seed derives each other as related pages.

In this situation, IBM, TOSHIBA, and SONY derive each other as related pages by Companion–, since they are mainly pointed to by link lists of electric companies. They are symmetrically connected to each other in the graph (B). In the case of two SONY PC fans, they derive each other and SONY as related pages, because they are mainly pointed to by pages of SONY PC related links. As a result, these fans derive SONY, but SONY does not derive these fans, since the number of electric company lists is greater than link lists of SONY PC. From the graph (B) in Figure 2, we can see that symmetric derivation is a strong relationship, and asymmetric derivation is a weak relationship. Under these observations, we define that a community is a set of pages densely connected by symmetric derivation relationships, and two communities are related if there is an asymmetric derivation relationship between members of them.

## 2.2 Algorithm for Building Charts

Here we describe our algorithm for building charts. The first step is selecting a seed set from a web archive. As seeds, we select web pages that have inlinks from $IN$ or more different servers. Here, $IN$ is a parameter to determine the seed set. Only the number of different servers is counted, and intra-server links are not considered, because links from the same server are often made by the same author.

The second step is to build a directed graph that shows how each seed derives other seeds by Companion–. Nodes represent seeds in the seed set. Each directed edge, from a node $s$ to another node $t$, represents the fact that $s$ derives $t$ as one of the related pages by Companion–. We create directed edges between nodes by applying Companion– to each seed, so that an edge from a node $s$ to another node $t$ exists when $s$ derives $t$ as one of the top $N$ authorities, where $N$ is a parameter. We call this graph the authority derivation graph (ADG) in the following.

The third step is to extract a symmetric derivation graph (SDG) from ADG, and to extract web communities from SDG. In this step, we put focus on the symmetric derivation relationship, in which two nodes derive each other by Companion–. SDG includes nodes in the seed set, and an edge from $s$ to $t$ exists when $s$ and $t$ point to each other in ADG. Then we extract densely connected subgraphs in SDG as *cores* of web communities, and form cores into communities by adding remaining nodes to these cores. Although cliques in SDG seem to be a good definition of cores, we have found many communities sparser than cliques in SDG. Therefore, we use a heuristic definition of the core. We use a node triangle as a unit of extraction, and a core is defined as a set of triangles that share edges in SDG. Note that a core becomes a 1-connected subgraph of SDG, and may be a complete graph with four or more nodes. In the following, we explain the detailed extraction process. After finishing this process, every connected node in SDG becomes a member of a community. Note that communities become disjunctive sets of nodes.

1. Extract all triangles of nodes from SDG. Then make every core that is a subgraph consists of triangles sharing edges. When two cores share some nodes, we temporarily isolate them, and pass them to the next step.

2. Add each remaining node in SDG to a neighboring core, if the node has edges connected to the core. When there are multiple candidates, select one core taking into account of directed edges in ADG. That is, to select a core that has the most incoming edges from the node in ADG. Each core then becomes a community.

3. There remain connected components that do not form triangles, such as lines of nodes. We also extract such components as communities.

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 relationships. 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 $d$. In the following, we use the simplified weight that is the sum of weights of directed edges between $c$ and $d$ ignoring directions. This is because the semantics of the direction is not yet clear. We call this simplified weight as the *relevance* between communities at both ends.

## 2.3 An Example of a Web Community Chart

Figure 3 shows a part of the web community chart built from our web archive in 2002 (see Section 4 for details), using our chart browser. Communities including keyword "Computer" are displayed in Figure 3. Each box with a label represents a web community, and edges represent relationships between communities. The size of each node is determined by the number of URLs and edges connected to the node. The thickness of each edge represents the relevance value of edges. Labels on communities are automatically attached by selecting frequent keywords from anchor texts that point to URLs in the community. When the user clicks one of a community, its pages are displayed with a web browser as shown in Figure 3.

Using this chart, the user can overview and navigate communities related to computers. The community (a) includes major computer companies in Japan, such as SONY and NEC, and its contents are shown in the web browser. Around the community (a), there are communities of related companies. The community (b) includes Japanese pages of software makers, such as Microsoft and Lotus. The community (c) includes computer device and peripheral companies such as Intel and Adaptec.

# 3. EVOLUTION OF WEB COMMUNITIES

This section explains how web communities evolve, and what kinds of metrics can measure degree of the evolution, such as growth rate and novelty. We first explain the details of changes of web communities, and then introduce evolution metrics that can be used for finding patterns of evolution. Here we summarize the notations used in this section.

$t_1, t_2, ..., t_n$**:** Time when each archive crawled. Currently, we use a month as the unit time.

$W(t_k)$**:** The Web archive at time $t_k$.

$C(t_k)$**:** The web community chart at time $t_k$.

$c(t_k), d(t_k), e(t_k), ...$**:** Communities in $C(t_k)$.

## 3.1 Types of Changes

We observe the evolution of communities from a series of web archives by (1) building web community charts ($C(t_1)$, $C(t_2)$, ..., $C(t_n)$) from all web archives, and (2) investigating differences between neighboring charts.

There are two ways to see the evolution of communities, backward and forward. For simplicity, here we explain backward examination. That is, first we select a web community chart $C(t_k)$, and see how communities had been evolved until time $t_k$ by comparing $C(t_{k-1})$ and $C(t_k)$. We can do the same thing for forward examination by comparing $C(t_k)$ and $C(t_{k+1})$. Here we show how communities change from $C(t_{k-1})$ to $C(t_k)$, such as growing and shrinking.

**Emerge:** A community $c(t_k)$ emerges in $C(t_k)$, when $c(t_k)$ shares no URLs with any community in $C(t_{k-1})$. Note that not all URLs in $c(t_k)$ are newly appeared in $W(t_k)$. Some URLs in $c(t_k)$ may be included in $W(t_{k-1})$, and do not have enough connectivity to form a community.

**Dissolve:** A community $c(t_{k-1})$ in $C(t_{k-1})$ has dissolved, when $c(t_{k-1})$ shares no URLs with any community in $C(t_k)$. Note that not all URLs in $c(t_{k-1})$ disappeared from $W(t_{k-1})$. Some URLs in $c(t_{k-1})$ may still be included in $W(t_k)$ losing connectivity to any community.

**Grow and shrink:** When $c(t_{k-1})$ in $C(t_{k-1})$ shares URLs with only $c(t_k)$ in $C(t_k)$, and vice versa, only two changes can occur to $c(t_{k-1})$. The community grows when new URLs appear in $c(t_k)$, and shrinks when URLs disappeared from $c(t_{k-1})$. When the number of appeared URLs is greater than the number of disappeared URLs, we consider that it grows. In the reverse case, we consider that it shrinks.

**Split:** A community $c(t_{k-1})$ may splits into smaller communities. In this case, $c(t_{k-1})$ shares URLs with multiple communities in $C(t_k)$. A split is caused by disconnections of URLs in SDG (see Section 2.2). Split communities may grow and shrink. They may also merge (see the next item) with other communities.

**Merge:** When multiple communities $(c(t_{k-1}), d(t_{k-1}), ...)$ share URLs with a single community $e(t_k)$, these communities are merged into $e(t_k)$ by connections of their URLs in SDG. Merged communities may grow and shrink. They may also split before merging.

## 3.2 Evolution Metrics

Our evolution metrics measure how a particular community $c(t_k)$ has evolved. For example, we can know how much $c(t_k)$ has grown, and how many URLs newly appeared in $c(t_k)$. Our metrics can be used for finding various patterns of evolution described in Section 3.1. To measure changes of $c(t_k)$, we need to identify the community at time $t_{k-1}$ corresponding to $c(t_k)$. We define this *corresponding community*, $c(t_{k-1})$, as the community that shares the most URLs with $c(t_k)$. If there were multiple communities that share the same number of URLs, we select a community that has the largest number of URLs.

We can reversely identify the community at time $t_k$ corresponding to $c(t_{k-1})$. When this corresponding community is just $c(t_k)$, we call the pair $(c(t_{k-1}), c(t_k))$ a *main line*. Otherwise, the pair is called a *branch line*. A main line can be extended to a sequence by tracking such symmetrically corresponding communities over time. A community in a main line is considered to keep its identity, and can be used for a good starting point for finding changes around its topic.

The metrics are defined by differences between $c(t_k)$ and its corresponding community $c(t_{k-1})$. To define metrics, we use following attributes representing how many URLs the focused community obtains or loses.

$N(c(t_k))$**:** the number of URLs in the $c(t_k)$.

$N_{sh}(c(t_{k-1}), c(t_k))$**:** the number of URLs shared by $c(t_{k-1})$ and $c(t_k)$.

$N_{dis}(c(t_{k-1}))$**:** the number of disappeared URLs from $c(t_{k-1})$ that exist in $c(t_{k-1})$ but do not exist in any community in $C(t_k)$.

$N_{sp}(c(t_{k-1}), c(t_k))$**:** the number of URLs split from $c(t_{k-1})$ to communities at $t_k$ other than $c(t_k)$.

$N_{ap}(c(t_k))$**:** the number of newly appeared URLs in $c(t_k)$ that exist in $c(t_k)$ but do not exist in any community in $C(t_{k-1})$.

$N_{mg}(c(t_{k-1}), c(t_k))$**:** the number of URLs merged into $c(t_k)$ from communities at $t_{k-1}$ other than $c(t_{k-1})$.

Then our evolution metrics are defined as follows.

The *growth rate*, $R_{grow}(c(t_{k-1}), c(t_k))$, represents the increase of URLs per unit time. It allows us to find most growing or shrinking communities. The growth rate is defined as follows. Note that when $c(t_{k-1})$ does not exist, we use zero as $N(c(t_{k-1}))$.

$$R_{grow}(c(t_{k-1}), c(t_k)) = \frac{N(c(t_k)) - N(c(t_{k-1}))}{t_k - t_{k-1}} \quad (1)$$

*stability*, $R_{stability}(c(t_{k-1}), c(t))$, represents the amount of disappeared, appeared, merged and split URLs per unit time. When there is no change of URLs, the stability becomes zero. Note that $c(t_k)$ may not be stable even if the growth rate of $c(t_k)$ is zero, because $c(t_k)$ may lose and obtain the same number of URLs. A stable community on a topic is the best starting point for finding interesting changes around the topic. The stability is defined as follows.

$$R_{stability}(c(t_{k-1}), c(t_k)) =$$

$$\frac{N(c(t_{k-1})) + N(c(t_k)) - 2N_{sh}(c(t_{k-1}), c(t_k))}{t_k - t_{k-1}} \quad (2)$$

The *novelty*, $R_{novelty}(c(t_{k-1}), c(t_k))$, represents the number of newly appeared URLs per unit time. When the novelty becomes high, we can say that $c(t_k)$ has grown mainly by newly appeared URLs. We can find most emerged communities at time $t_k$ by sorting communities by the novelty. The following is the definition of the novelty.

$$R_{novelty}(c(t_{k-1}), c(t_k)) = \frac{N_{ap}(c(t_k))}{t_k - t_{k-1}} \quad (3)$$

The *disappearance rate*, $R_{disappear}(c(t_{k-1}), c(t_k))$, is the number of disappeared URLs from $c(t_{k-1})$ per unit time. Higher disappear rate means that the community has lost URLs mainly by disappearance. The disappear rate is defined as

$$R_{disappear}(c(t_{k-1}), c(t_k)) = \frac{N_{dis}(c(t_{k-1}))}{t_k - t_{k-1}} \quad (4)$$

The *merge rate*, $R_{merge}(c(t_{k-1}), c(t_k))$, is the number of absorbed URLs from other communities by merging per unit time. Higher merge rate means that the community has obtained URLs mainly by merging. The merge rate is defined as follows.

$$R_{merge}(c(t_{k-1}), c(t_k)) = \frac{N_{mg}(c(t_{k-1}), c(t_k))}{t_k - t_{k-1}} \quad (5)$$

The *split rate*, $R_{split}(c(t_{k-1}, c(t_k))$, is the number of split URLs from $c(t_{k-1})$ per unit time. When the split rate is low, we can know that $c(t_k)$ is larger than other split communities. Otherwise, $c(t_k)$ is smaller than other split communities. The split rate is defined as follows.

$$R_{split}(c(t_{k-1}), c(t_k)) = \frac{N_{sp}(c(t_{k-1}), c(t_k))}{t_k - t_{k-1}} \quad (6)$$

By combining these metrics, we can represent some complex evolution patterns. For example, a community has stably grown when its growth rate is positive, and its disappearance and split rates are low. Similar evolution patterns can be defined for shrinkage.

Longer range metrics (more than one unit time) can be calculated for main lines. For example, the novelty metrics of a main line ($c(t_i)$, $c(t_{i+1})$, ..., $c(t_j)$) is calculated as follows.

$$R_{novelty}(c(t_i), c(t_j)) = \frac{\sum_{k=i}^{j} N_{ap}(c(t_k))}{t_j - t_i} \quad (7)$$

Other metrics can be calculated similarly.

| Year | Period | Crawled pages | Total URLs | Links |
|---|---|---|---|---|
| 1999 | Jul. to Aug. | 17M | 34M | 120M |
| 2000 | Jun. to Aug. | 17M | 32M | 112M |
| 2001 | Early Oct. | 40M | 76M | 331M |
| 2002 | Early Feb. | 45M | 84M | 375M |

**Table 1: Details of our web archives**

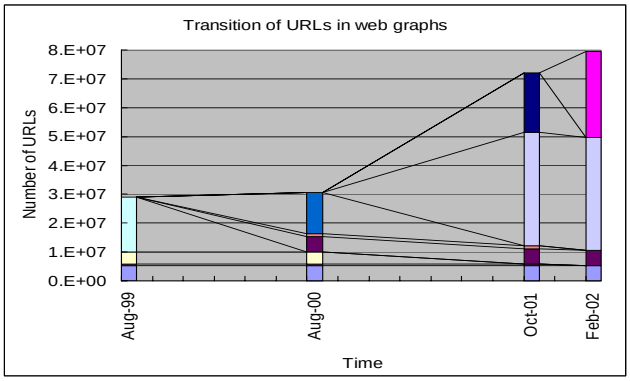


**Figure 4: Transition of URLs in web graphs**

# 4. ANALYSIS OF WEB ARCHIVES AND EVOLUTION OF WEB COMMUNITIES

## 4.1 Web Archives and Graphs

For experiments, we used four web archives of Japanese web pages (in .jp domain) crawled in 1999, 2000, 2001, and 2002 (See Table 1). We used the same web crawler in 1999 and 2000, and collected about 17 million pages in each year. In 2001, the number of pages became more than twice of the 2000 archive, since we improved the crawling rate. Our crawlers collected pages in the breadth-first order.

From each archive, we built a web graph with URLs and links by extracting anchors from all pages in the archive. Our graph included not only URLs inside the archive, but also URLs outside pointed to by inside URLs. As a result, the graph included URLs outside .jp domain, such as .com and .edu. Table 1 also shows the number of links and the total URLs. For efficient link analysis, each web graph was stored in a main-memory database that provided out-links and in-links of a given URL. Its implementation was similar to the connectivity server [2]. We implemented the whole system on Sun Enterprise Server 6500 with 8 CPU and 4GB memory. Building our connectivity database of 2002 took about one day.

By comparing these graphs, we found that the Web was extremely dynamic. More than half of the URLs disappeared or changed its location in one year. We first examined how many URLs in our web graphs were changed over time, by counting the number of URLs shared between these graphs. Figure 4 depicts the transition of URLs in our web

| Year | Period | Seeds | Communities |
|---|---|---|---|
| 1999 | Jul. to Aug. | 657K | 79K |
| 2000 | Jun. to Aug. | 737K | 88K |
| 2001 | Early Oct. | 1404K | 156K |
| 2002 | Early Feb. | 1511K | 170K |

**Table 2: The number of seeds and communities**

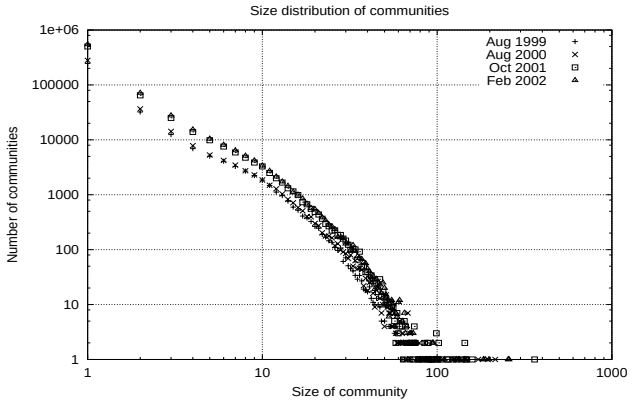


**Figure 5: Size distribution of communities**

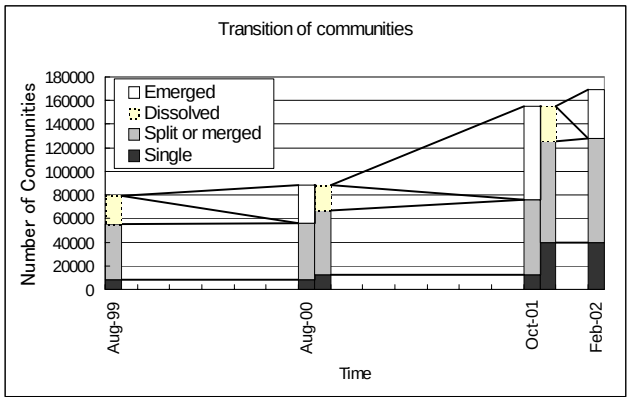


**Figure 6: Transition of communities**

| | $t_{k-1}$: Aug. 99<br>$t_k$: Aug. 00 | Aug. 00<br>Oct. 01 | Oct. 01<br>Feb. 02 |
|---|---|---|---|
| # Branch lines at $t_{k-1}$ | 28,467 | 32,490 | 41,501 |
| # Main lines | 26,723 | 34,396 | 83,771 |
| # Branch lines at $t_k$ | 29,722 | 41,752 | 44,305 |

**Table 3: Number of main lines in split or merged communities**

graphs. Each bar shows the number of URLs, and is separated into blocks according to life spans of URLs[1]. Blocks with the same set of URLs have the same color, and lines are drawn between these blocks, so that you can see when these URLs appeared and how long they survived. As shown in Figure 4, about 60% of URLs disappeared from both 1999 and 2000 graphs. From 2001 to 2002, about 30% of URLs disappeared in four months. The number of URLs surviving through four archives was only about 5 million. In [9], Cho reported that more than 70% of pages survived more than one month in their four month observation. This result is close to our result from 2001 to 2002 (30% disappearance in four months). Although it is not easy to estimate the rate for one year from [9], we can say that our results does not deviate so much.

We also examined the indegree distributions of our web graph. Indegree of a URL stands for the number of URLs pointing to the URL. Previous work, such as [6, 14, 16], shows indegree distributions fit to a *power law*, in which the probability of the positive integer value $i$ is proportional to $1/i^k$ for a small positive number $k$. The reported number of $k$ is 2.1. Indegree distributions of our graphs also exhibit a power law. The exponent of the power law is around 2.2 for all graphs. It means that our graphs are slightly sparser than ones of previous work.

## 4.2 Evolution of Web Community Charts

In this section, we describe the global behavior of community evolution. From the above four web graphs, we built four community charts using the technique described in Section 2.2. To compare web community charts in the same condition, we fixed values of parameters, $IN$ and $N$, for the chart building algorithm in Section 2.2. We used the value 3 as $IN$, that is, we selected seeds that have in-links from three or more different servers. Using a larger value than 3 drastically decreased seeds, because of its power-law distribution. We used the value 10 as $N$, that is, the top 10 results by Companion– were used for calculating relevance between seeds. We selected the value 10, because Companion– provided enough precision with top 10 authorities in our previous work [17], and members of major communities did not change much from the value 9 to 11[2]. It took about half a day to build the chart for 2002. Most of the time was spent on calculating related pages of seeds. Table 2 shows the number of seeds and communities in the chart created for each year. In the following, we show evolution statistics of web community charts.

### 4.2.1 Size distribution

The size distribution of communities also follows the power law and its exponent did not change so much over time. Figure 5 shows log-log plots of the communities size distributions for all charts. All four curves roughly fit to a power law distribution with exponent 2.9 to 3.0. This result is similar to distributions of connected components in the Web graph reported in [6].

### 4.2.2 Types of Changes

Although the size distribution of communities is stable, the structure of communities changes dynamically. Figure 6 shows how many communities are involved in each type of change from $t_{k-1}$ to $t_k$. Each bar represents the number of communities in charts at the time. Bars at 2000 and 2001 are split vertically, since they have the previous and the next charts to be compared. Each block represents the number of

[1]In Figure 4, we ignored a few million URLs that appeared intermittently in our web graphs, e.g. URLs that appeared only in 1999 and 2001

[2]Ideally, these parameters should be determined locally in a chart depending on topics and density of the web graph. It is future work to develop such adaptive algorithms.

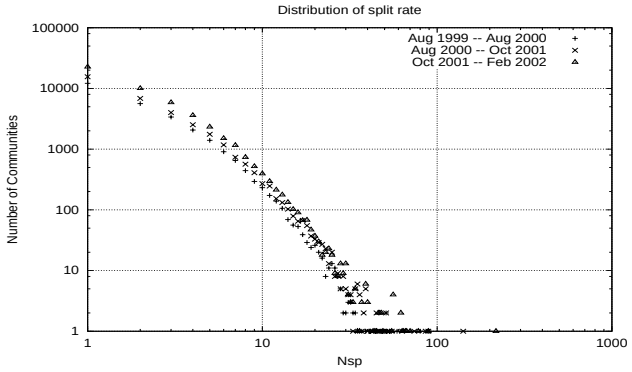


**Figure 7: Distribution of split rate**

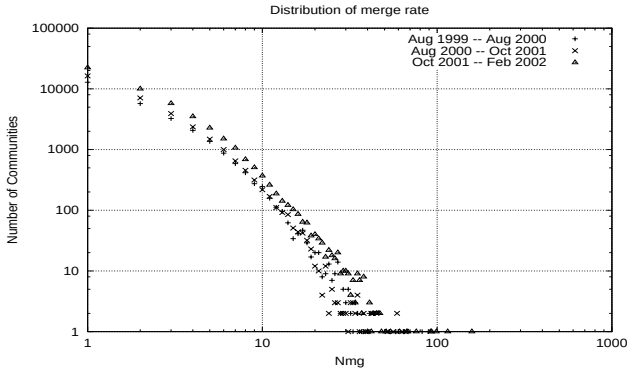


**Figure 8: Distribution of merge rate**

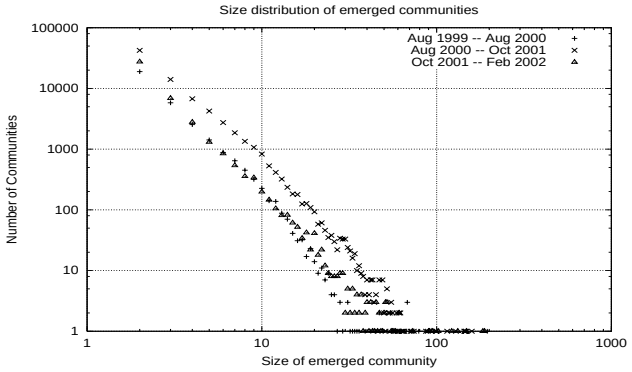


**Figure 9: Size distribution of emerged communities**

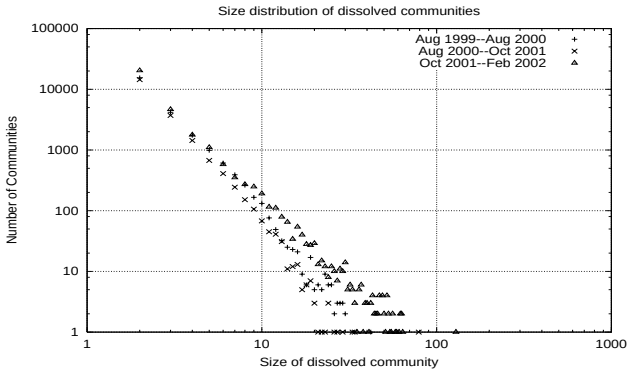


**Figure 10: Size distribution of dissolved communities**

communities involved in a particular change. Dotted blocks represent dissolved communities from $t_{k-1}$, and white blocks represent emerged communities. Gray blocks represent communities that are involved in split or merge. Finally, black blocks represent single communities that are not involved in these changes, but may grow or shrink.

We can see that the structure of our chart changes mainly by split and merge, in which more than half of communities are involved. The number of single communities is small (10% in 1999, 14% in 2000, and 25% in 2001). Since the seed sets of the charts are stable parts in our archives, the number of communities dissolved from each chart is rather small. About 24% to 30% of communities are dissolved in one year from 1999 to 2001, while 20% of communities are dissolved in four months from 2001.

Changes by split and merge are complicated, since split communities may be merged with other communities in the next time. However, it is not totally chaotic. We can see rather stable communities by extracting main lines (See Section 3.2). Table 3 shows the number of main lines and branch lines in each intervals. About half of survived (not dissolved) communities in 1999 and 2000 are included in main lines for one year, and about 66% of survived communities in 2001 are included in main lines for four months. Note that the main lines include single communities. We can use those main lines as a good starting point for finding changes around the topic. In the following section, we show the detailed behavior of each changes.

### 4.2.3 *Split and Merged Communities*

We first show the distributions of the split rate and merge rate in Figure 7 and 8. We plot the number of split or merged communities as a function of the number of split and merged URLs ($N_{sp}$ and $N_{mg}$ in Section 3.2) in the log-log scale. Both distributions roughly follow the power law, and show that split or merge rate is small in most cases. Their shapes and scales are also similar. That is, when communities at $t_{k-1}$ split with a split rate, almost the same number of communities are merged at $t_k$ with the same rate as the split rate. This symmetry is part of the reason why the size distribution of communities does not change so much.

### 4.2.4 *Emerged and Dissolved Communities*

The size distributions of emerged and dissolved communities also follow the power law, and contribute to preserve the size distribution of communities. Figure 9 and 10 show these distributions for all periods in the log-log scale. In most cases, the exponent of the power law is greater than 3.2, while the exponent of the whole chart is around 3.0. This means that small communities are easy to emerge and dissolve.

### 4.2.5 *Growth Rate*

Finally, we examine the distribution of the growth rate. Figure 11 shows the number of communities as a function of the growth rate. We use a log scale on the y-axis. For simplicity, we only plot the growth rate of main lines in this

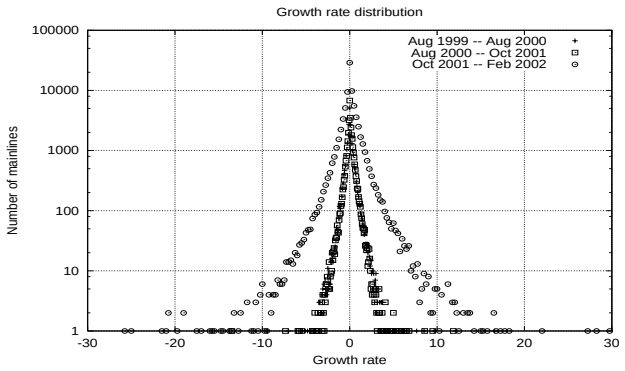


**Figure 11: Distribution of growth rate**

graph. The growth rate is small for most of communities, and the graph has clear y-axis symmetry. This is also part of the reason why the size distribution of communities is preserved over time.

# 5. EVOLUTION VIEWER AND EXAMPLES

So far, we have seen the global behavior of community evolution. In this section, we demonstrate our evolution viewer that displays the details of community evolution on a specific topic. Our viewer provides various means to extract evolving communities as follows: (1) searching communities by keywords or a URL; (2) Sorting and filtering communities by the evolution metrics defined in Section 3.2. Using our viewer, the user can extract various kinds of evolving communities, such as the most growing communities, and the most emerging communities related to a specific topic. Some evolution examples are shown with our viewer.

## 5.1 Visualizing Evolution of a Community

Our evolution viewer displays evolution of communities as shown in Figure 12 and 14. Each community is represented as a rectangle including the list of its URLs. URLs are drawn by different styles according to their lifetime. URLs written in italic style are newly appeared one, and others appears in the previous chart. Each URL can be browsed with a web browser by clicking it with the mouse. Labels on communities are automatically attached by selecting frequent keywords from anchor texts that point to URLs in the community. Each column represents a time (the month when each archive crawled).

Basically, the viewer displays main lines of communities. Communities in a main line are arranged horizontally, and lines are drawn between them, so that the user can easily compare their differences. The thickness of each line represents the number of shared URLs between communities at both ends. The viewer shows only a main line as default. If the user needs more detailed view, the viewer shows a main line and branch lines around it (See Figure 14).

## 5.2 Searching, Sorting, and Filtering

The user can select communities at a time by providing a URL or keywords. If the user provides a URL, the viewer shows the community including that URL. If the user provides keywords, the viewer shows communities including pages with the keywords. In addition, the user can select communities related to a specified community in the chart.

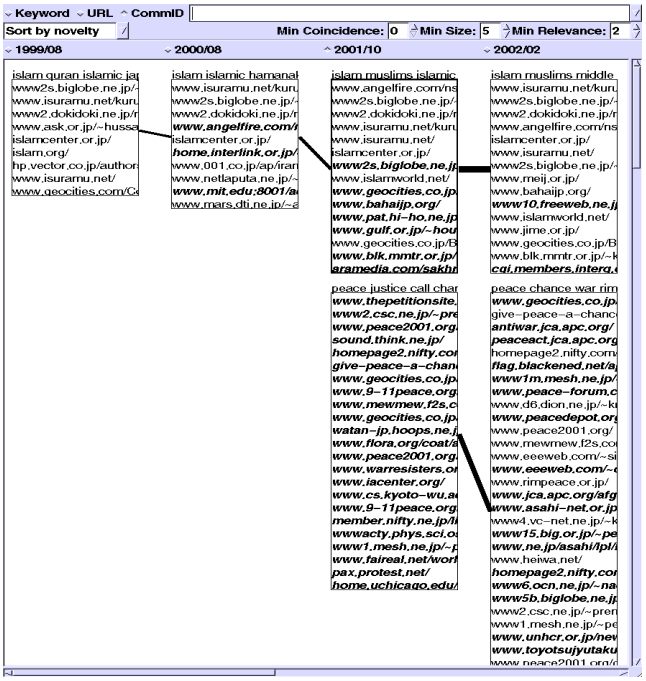


**Figure 12: An emerged community around an Islam information community**

The system automatically extract main lines for selected communities over the given time range. The time range can be specified by the user. In our experiments, since we have only four web archives, time range specification looks meaningless, but when we have more series of web archives, it will be important.

The number of selected communities might become large. In such a case, the user can sort the communities and browse the communities in a sorted order. The user can also specify filtering condition and reduce the number of target communities. The metrics that is introduced in Section 3.2 is used for both sorting and filtering. When the user does not select communities, all communities at the time are sorted. Actually, sorting and filtering communities by the evolution metrics are very powerful means to find the communities with various evolving patters. In addition to the evolution metrics, we can use the following metrics for sorting and filtering: (1) Relevance between the given community and neighboring communities; (2) Size of a community (the number of URLs in the community); (3) The number of shared URLs between communities.

To realize those functions, the browser uses several indices. For locating communities, a keyword index, a URL index, and a relation index are prepared for each chart. The keyword index allows us to retrieve communities including pages with specified keywords. The URL index is used for finding the community including the specified URL. The relation index provides communities neighboring a given community in the chart.

We use an evolution index for each chart, for tracking changes of communities and calculating metrics. This index at time $t_k$ provides a lists of past communities at $t_{k-1}$ and a list of future communities at $t_{k+1}$, which share URLs with a given community at $t_k$. Using this index, we can track main

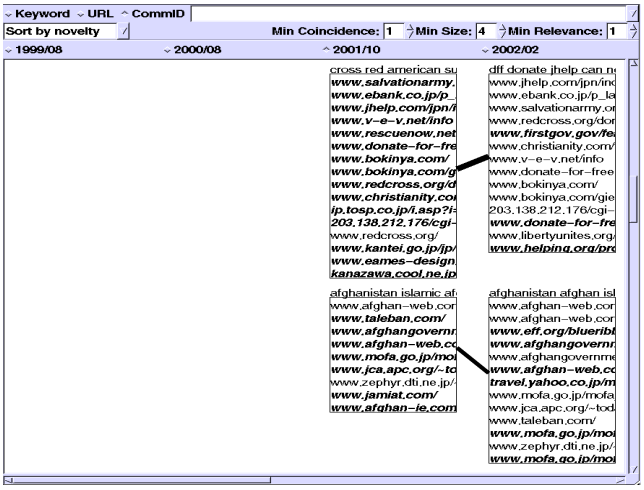


**Figure 13: Emerged communities around a pacifist community**

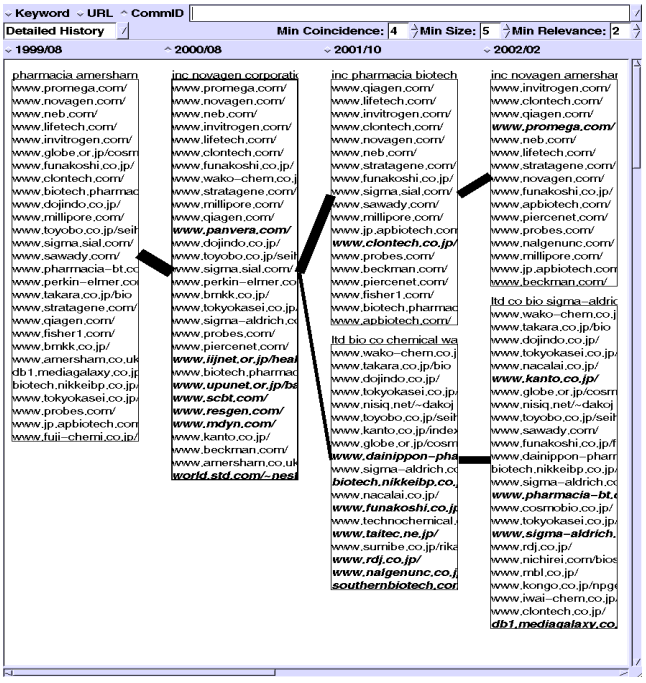


**Figure 14: Communities of biotechnology companies**

and branch lines to the past and the future. To these lists, we also attached evolution attributes such as the number of shared URLs, split URLs, and merged URLs, which are used for calculating the evolution metrics.

## 5.3 Examples

Firstly, we show a typical example to find emerged communities around a topic using Figure 12. In this example, we want to find emerged communities related to Islam after the terrorist attack on America on September 11, 2001. Using our viewer, we first find a community of Islam in 2001 by the keyword "Islam". The main line of this community is displayed on the top of Figure 12, which includes URLs such as "islamcenter.or.jp" and "www.islamworld.net". Next, we list up communities related to that community. At this time, we mainly obtain communities about other religions (not shown in Figure 12). Then for finding emerged communities around Islam, we sort these related communities by their novelty metrics from 2000 to 2001. Finally, we obtained Figure 12. There emerged a community of pacifist movements against the retaliatory bombing, which includes URLs such as "www.peace2001.org" and "www.9-11peace.org".

We can recursively find emerging communities around one of those related communities. Figure 13 depicts emerged communities around the pacifist community in Figure 12. The first one includes pages inviting donations for Afghanistan, such as "www.donate-for-free.com" and "www.redcross.com". The second one includes pages about Afghanistan and Taleban (most of URLs include the string "afghan"). It means that such hubs were rapidly created after the terrorist attack. (Note that our archive in 2001 was crawled in early October.) This example shows that communities grow very quickly when their topics has a great impact to the real society.

Figure 14 shows a typical evolution pattern, in which a web community gradually grows by absorbing various types of related pages, and then it is refined by splitting into more detailed communities. We can find such communities using our evolution viewer by searching communities that stably grew then largely split at the next time. In Figure 14, Communities of biotechnology and biochemistry companies are gradually growing, as biotechnology become an important business, and awareness of biotechnology is growing. Until 2000, it grows absorbing both Japanese and ".com" companies. From 2001, it splits into these two types of companies. It means that many unsophisticated hubs are created when the topic attracted popularity, then these hubs are gradually refined, or a lot of more sophisticated hubs are created.

# 6. DISCUSSION AND FUTURE WORK

Our system takes about two days for building web graph databases, web community charts, and an evolution index, from a web archive (45M pages). Since it is significantly shorter than the crawling time of the web archive, our system can analyze evolution at the same interval of crawling. We have shown a web community browser in Section 2, and a web community evolution viewer in Section 5. Since, these tools are separately developed, we plan to integrate these tools, and realize seamless navigation through related communities, and through past communities. Web archives used in our experiments are small subsets of the entire Web, and the crawling interval is still long. We are interested in applying our technique to larger archives, and investigating how the results will be influenced by changing the archive. We are also planning to crawl web pages more frequently, and observe more fine-grained evolution.

# 7. CONCLUSIONS

We have proposed a method for observing the evolution of web communities. Evolution metrics are defined to measure the degree of community evolution. Our method is based on our previous work of web community chart that is a graph of communities, in which related communities are connected by edges. The main advantage of our web community chart is the existence of relevance between communities. Combin-

ing evolution metrics and relevance, we can locate evolution around a particular community. The experiments have been performed against four Japanese web archives (in jp domain) crawled from 1999 to 2002 with 119M pages in total, and statistics of web graphs and community evolution are examined. We have found that the size distribution of communities followed the power-law, and its exponent did not change so much over time. This is because most of changes in communities follow the power-law, and the changes are symmetric between split and merge, and between growth and shrinkage. Finally, we have shown a web community evolution viewer for detailed examination of evolution on a specific topic. Various means are available to find evolution patterns using the evolution metrics. Using our viewer, we demonstrated some example evolutions.

# 8. REFERENCES


[1] Wayback Machine, The Internet Archive. http://www.archive.org/.

[2] K. Bharat, A. Broder, M. Henzinger, P. Kumar, and S. Venkatasubramanian. The Connectivity Server: fast access to linkage information on the Web. In *Proc. 7th International WWW Conference*, pages 14–18, 1998.

[3] K. Bharat, B.-W. Chang, M. Henzinger, and M. Ruhl. Who Links to Whom: Mining Linkage between Web Sites. In *Proc. IEEE International Conference on Data Mining*, pages 51–58, 2001.

[4] K. Bharat and M. Henzinger. Improved Algorithms for Topic Distillation in a Hyperlinked Environment. In *Proc. ACM SIGIR '98*, pages 104–111, 1998.

[5] B. E. Brewington and G. Cybenko. How dynamic is the web? In *Proc. 9th WWW Conference*, pages 257–276, 2000.

[6] A. Broder, R. Kumar, F. Maghoul, P. Raghavan, S. Rajagopalan, R. Stata, A. Tomkins, and J. Wiener. Graph Structure in the Web. In *Proc. 9th WWW Conference*, pages 309–320, 2000.

[7] S. Chakrabarti. Integrating the Document Object Model with Hyperlinks for Enhanced Topic Distillation. In *Proc. 10th WWW Conference*, pages 211–220, 2001.

[8] S. Chakrabarti, B. Dom, P. Raghavan, S. Rajagopalan, D. Gibson, and J. Kleinberg. Automatic resource compilation by analyzing hyperlink structure and associated text. In *Proc. 7th International WWW Conference*, pages 65–74, 1998.

[9] J. Cho and H. Garcia-Molina. The Evolution of the Web and Implications for an Incremental Crawler. In *Proc. 26th International Conference on Very Large Databases (VLDB)*, pages 200–209, 2000.

[10] J. Dean and M. R. Henzinger. Finding related pages in the World Wide Web. In *Proc. 8th WWW Conference*, pages 389–401, 1999.

[11] G. W. Flake, S. Lawrence, and C. L. Giles. Efficient Identification of Web Communities. In *Proc. KDD 2000*, pages 150–160, 2000.

[12] D. Gibson, J. Kleinberg, and P. Raghavan. Inferring Web Communities from Link Topology. In *Proc. HyperText98*, pages 225–234, 1998.

[13] J. Kleinberg. Authoritative Sources in a Hyperlinked Environment. In *Proc. ACM-SIAM Symposium on Discrete Algorithms*, pages 668–677, 1998.

[14] R. Kumar, P. Raghavan, S. Rajagopalan, and A. Tomkins. Extracting large-scale knowledge bases from the web. In *Proc. 25th VLDB Conference*, pages 639–650, 1999.

[15] R. Lempel and S. Moran. The Stochastic Approach for Link-Structure Analysis (SALSA) and the TKC Effect. In *Proc. 9th WWW Conference*, pages 387–401, 2000.

[16] S. R. Ravi Kumar, Prabhakar Raghavan and A. Tomkins. Trawling the Web for emerging cyber-communities. In *Proc. 8th WWW Conference*, pages 403–415, 1999.

[17] M. Toyoda and M. Kitsuregawa. Creating a Web Community Chart for Navigating Related Communities. In *Proc. Hypertext 2001*, pages 103–112, 2001.


# APPENDIX

## Algorithm: Companion–

Companion– takes a seed page as an input, then outputs related pages to the seed. It first builds a subgraph of the Web around the seed, and extracts authorities from the subgraph as related pages.

First, it builds a vicinity graph of a given seed, 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. $V$ consists of the seed, a set of nodes pointing to the seed (B), and an another set of nodes pointed to by nodes in B (BF). When following outgoing links from each node in B, the order of links in the node is considered. 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. This is based on an observation that links to related pages are gathered in a small portion of a page. In our experiments, we use the value 10 as $R$ where result authorities are stable around the value.

To each edge, it assigns 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 [4]: (1) If two nodes of an edge are in the same server, the edge has the value 0 for both weights; (2) If a node has $n$ incoming edges from the same server, the authority weight of each edge is $1/n$; and (3) If a node has $m$ outgoing edges to the same server, the hub weight of each edge is $1/m$.

Then it calculates a hub score, $h(n)$ and an authority score, $a(n)$ for each node $n$ in $V$. The following is the process of the calculation, where $aw(n, m)$ and $hw(n, m)$ represent the authority weight and the hub weight of the edge from $n$ to $m$, respectively.

**Step 1.** Initialize $h(n)$ and $a(n)$ of each node $n$ to 1.

**Step 2.** Repeat the following calculation until $h(n)$ and $a(n)$ have converged for each node $n$.

For all nodes $n$ in $V$, $h(n) \leftarrow \sum_{(n,m)\in E} a(m) \times hw(n, m)$

For all nodes $n$ in $V$, $a(n) \leftarrow \sum_{(m,n)\in E} h(m) \times aw(m, n)$

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

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

**Step 3.** Return the $N$ highest authority nodes.