PAPER

Web Community Chart: a Tool for Navigating the Web and Observing its Evolution

Masashi TOYODA[†], *Nonmember and* Masaru KITSUREGAWA[†], *Member*

SUMMARY We propose a web community chart that is a tool for navigating the Web and for observing its evolution through web communities. A web community is a set of web pages created by individuals or associations with a common interest in a topic. Recent research shows that such communities can be extracted by link analysis. Our web community chart is a graph of whole communities, in which relevant communities are connected by edges. Using this chart, we can navigate through related communities. Moreover we can answer historical queries about topics on the Web and understand sociology of web community creation, by observing when and how communities emerged and evolved. We observe the evolution of communities by comparing three charts built from Japanese web archives crawled in 1999, 2000, and 2001. Several metrics are introduced for measuring the degree of community evolution, such as growth rate, novelty. Finally, we develop a web community evolution viewer that allows us to extract evolving communities using the relevance and metrics. Several evolution examples are shown using this viewer.

key words: *link analysis, web community, web community chart, evolution*

1. Introduction

Navigating the Web and observing its evolution are important issues, since the Web has experienced dramatic growth and dynamic changes in its structure. We could see a lot of phenomena in the Web, which 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 the web.

In this paper, we propose a web community chart [16] that is a tool for navigating the Web and for observing its evolution through *web communities*. A web community is a collection of web pages created by individuals or associations with a common interest in a topic, such as fan pages of a baseball team, and official pages of computer vendors. Recent research on *link analysis* [3], [5], [7], [8], [12]–[15] shows that we can identify a web community on a topic by finding densely connected structure in *the web graph*, in which nodes are web pages and edges are hyperlinks. In [16], we proposed the web community chart as a navigation tool that is a graph of communities, in which related communities are connected by edges. The main advantage of our chart is existence of relevance between communities. We show that the chart can be used not only for navigating through related communities but also for locating evolution around a particular community.

Manuscript received 0, 2002.
Manuscript revised 0, 2002.

[†]Institute of Industrial Science, University of Tokyo

Since a web community represents a certain topic, we can understand when and how new topics emerged and evolved in the Web. For example, when mad-cow disease became a serious problem, and what kinds of web pages have been created about the disease. Such information is important in the following situations: (1) answering historical queries about topics in the Web; (2) observing the emergence of quality web communities on a specific topic; (3) understanding sociology of web community creation related to real social activities; and (4) improving the efficiency of web crawlers by giving priority to getting emerging and growing parts in the web at the next crawl.

For extracting such information, we observe the evolution of web communities by comparing three charts built from three Japanese web archives (in jp domain) periodically crawled in 1999, 2000, and 2001 with 70 million pages in total. Several evolution metrics are introduced for measuring the degree of evolution. We describe changes of the community, such as growing and shrinking. Based on these changes, we define our evolution metrics, such as growth rate, novelty, and stability.

By using such metrics, user can extract target communities based on evolving patterns. To examine their feasibility, we developed a web community evolution viewer for browsing how communities evolved through two years. It provides various ways for locating the evolution of communities such as emerged and growing. With several examples, we demonstrate that we can easily locate interestingly evolving web communities by combining evolution metrics and relevance. For example, we can find emerged or growing communities related to a particular community.

1.1 Prior Work

Most research on web communities is based on the notion of *authorities* and *hubs* proposed by Kleinberg [14]. 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

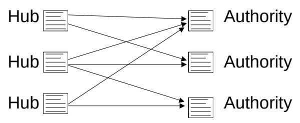


Fig. 1 Graph structure of hubs and authorities

on the topic, that is, points to many good authorities. HITS [14] 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. HITS has been improved and refined [2], [3], [5], [12] by exploiting anchor texts, edge weighting, document similarity, and Document Object Models.

A set of authorities and hubs was regarded as a community in [8], [10], [11]. Gibson et al. [8] investigated the characteristic of communities derived by HITS. Kumar et al. [10], [11] 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. Dean and Henzinger proposed a related page algorithm, Companion [7]. They tailored HITS [14] for finding related pages, and improved the precision by exploiting link weighting and by considering the order of links in a page. Companion extracts authorities in a subgraph of the Web near the seed, and returns authorities as related pages.

Some other approaches have been proposed. Lempel and Moran [15] adopted a random walk model for calculating authorities. Flake et al. [13] 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 communities, they have not considered relevance between communities, and evolution of communities. In this paper, we examined how these identified communities evolve with passing time, and use relevance for locating evolutions around a given community. As far as we know, there is still no published research on examining evolution of web communities.

The change frequency and lifetime of web pages has been studied in [4], [6]. 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. Rather, we analyze the modification of graph structure in the Web.

Recently, the Internet Archive begins the Wayback Machine service [1] that allows us to see past web pages

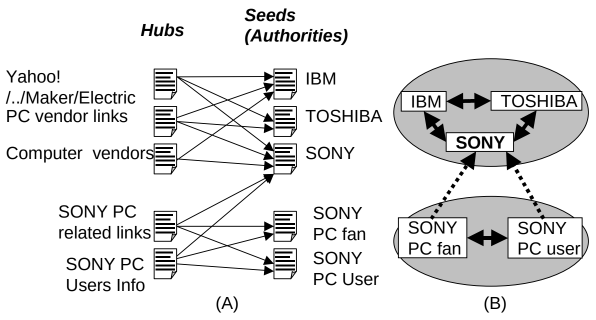


Fig. 2 An example of derivation relationships

stored in the Internet Archive's web archive. Its capability is still very limited. That is, we 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 describe our technique to build the web community chart and its browser. Section 3 introduces evolution of web communities, and evolution metrics. In Section 4, we show our web archives, and build charts from them. Section 5 demonstrates our web community evolution viewer with some examples. In Section 6, we discussed more detailed issues, and future work.

2. Web Community Chart

In this section, we briefly describe our technique to build the web community chart that is a graph including communities as nodes, and weighted edges between relevant communities. Refer to [16], for more detailed descriptions.

2.1 Intuition for Underlying Techniques

Our algorithm builds the web community chart from a given seed set. The main idea is applying a related page algorithm (RPA), such as Companion[7], to each seed, and investigating how each seed derives other seeds as related pages.

Since existing RPAs, such as HITS [14] and Companion [7], provide insufficient precision, we use an improved algorithm, Companion– [16]. We have gained a better precision by ignoring error prone parts in the subgraph. The algorithm is described in Appendix A.

To identify web communities and to deduce their relationships, we focus on relationships between a seed and derived related pages by Companion–. Figure 2 depicts an example of derivation relationships. We use five seed pages, IBM, TOSHIBA, SONY, and two SONY PC fans. In Figure 2, the graph (A) shows how each seed is pointed to by hub pages, and the directed

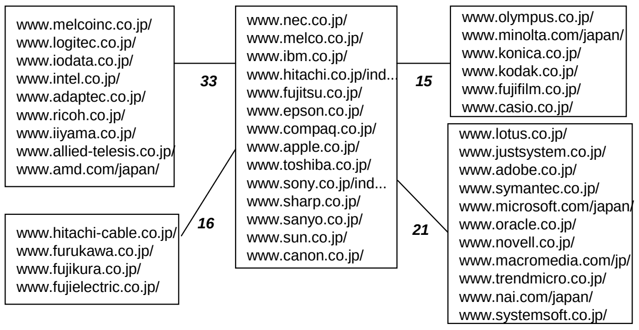


Fig. 3 A part of the web community chart

graph (B) shows how each seed derives each other as related pages by Companion–.

First, consider about IBM, TOSHIBA, and SONY. They derive each other as related pages, since they are mainly pointed to by link lists of electric companies, such as the electric maker directory in Yahoo!. Then, consider about two SONY PC fans. They derive each other and SONY as related pages, because they are mainly pointed to by such as SONY PC related links. In this case, SONY PC fans derive SONY, but SONY does not derive SONY PC fans. It is because major hubs of SONY differ from major hubs of SONY PC fans. That is, SONY is pointed to by not only electric company lists but also SONY PC related links, however, the number of electric company lists is more than one of other link lists. 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 observation, we define that a community is a set of pages densely connected by symmetric derivation relationships (SDR), and two communities are related if there is an asymmetric derivation relationship between members of them.

To extract densely connected seeds, we use a triangle of SDR as a unit (like IBM, SONY, and TOSHIBA in Figure 2), since the triangle is a complete graph and most seeds in the triangle share the same topic. We also found that (1) two triangles, which share an edge, often share the same topic, and (2) two triangles, which share only one seed, often have related but slightly different topic. Based on these observations, we consider a set of triangles sharing edges as the core of a community.

2.2 Algorithm for Building Chart

Here we describe our algorithm for building a chart. The first step is selecting a seed set from a web archive. As seeds, we select web pages that have in-links 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 creator.

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 also extract web communities. In this step, we focus on SDRs, in which two nodes at both ends 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 seeds in SDG as web communities. We use a node triangle as a unit of extraction. We use a simple algorithm that finds densely connected *cores*, then adds isolated nodes to these cores. First, cores are made by extracting triangles that share edges. Note that these cores include complete graphs with four or more nodes. Then we add each isolated node to a neighbor core. After finishing this process, every connected node in SDG becomes a member of a community. Note that communities become disjunctive sets of nodes. The following explains the process in detail:

1. Extract all triangles of nodes from SDG. Then each subgraph, consists of triangles that share edges, is extracted as a core. 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 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 1999. Each box with a label represents a web community, and edges represent relationships between communities. We put labels to these communities for readers convenience. The number attached to each edge denotes the weight. We select the 'Computer' community as a center, since it has most edges in the chart. Each community, around the 'Computer', has edges to the 'Computer', and these weights are more than 15. Actually, there are more communities around the 'Computer' connected by lower weighted edges, that are not shown in Figure 3.

As shown in Figure 3, these communities are clearly classified and actually related to the 'Computer' community. The 'Software' community includes Lotus, Microsoft, etc., and obviously related to the 'Computer'. The companies in the 'Cable' community provides cables and optical fibers.

3. Evolution of Web Communities

In this section, we explain how web communities evolve, and what kinds of metrics can measure degree of the evolution, such as growth rate, and novelty[†].

Here we summarize the notations we use.

W_t: the web archive crawled at time t.
C_t: the web community chart derived from W_t.
$c_t, d_t, e_t, ...$: communities in C_t.

The subscript t denotes time. For simplicity here t takes an integer value between 1 and n instead of the date of crawling. The first time of crawling is denoted by 1, and the last time is denoted by n.

We observe the evolution from a series of periodically crawled web archives $(W_1, W_2, ..., W_n)$ by (1) building web community charts $(C_1, C_2, ..., C_n)$ for all web archives, and (2) investigating differences between neighboring charts. We first explain changes of communities, and then introduce evolution metrics.

3.1 Changes of Communities to Derive Evolution

There are two ways to see the evolution of a community, backward and forward. For simplicity, here we explain backward examination. That is, first we select a web community chart at time t, C_t, and see how communities had been evolved until time t by examining C_{t-1}. We can do the same thing for forward examination of evolution by fixing C_t and then examining C_{t+1}.

Here we show how communities changes from C_{t-1} to C_t, such as growing and shrinking. Changes of a community are complex, when the community splits or merges, since they may involve multiple communities.

Emerge: A community emerges in C_t, when the community shares no URLs with any community in C_{t-1}.

Dissolve: A community in C_{t-1} is dissolved, when the community shares no URLs with any community in C_t.

Grow and shrink: When c_{t-1} in C_{t-1} shares URLs with only c_t in C_t, and vice versa, only two changes can occur to c_{t-1}. The community grows when new URLs are appeared in c_t, and shrinks when URLs are disappeared from c_{t-1}. When the number of appeared URLs is greater than the number of disappeared URLs, we consider it a growing community. In the reverse case, we consider it a shrinking community.

Split: When a community c_{t-1} in C_{t-1} shares URLs with multiple communities in C_t, c_{t-1} splits into some smaller communities by disconnections of URLs. The community may grow and shrink before splitting, and split communities may also grow and shrink. In addition, split ones may merge with other communities.

Merge: When multiple communities in C_{t-1} share URLs with a community c_t in C_t, these communities in C_{t-1} are merged into c_t by connections of their URLs. Each community being merged may grow, shrink, and split before merging, and the result community may grow and shrink.

3.2 Evolution Metrics

Our evolution metrics measure how community c_t evolved using macroscopic changes. For example, we can know how c_t is emerged, or how rapidly c_t grew. Our metrics can be used for finding such as emerged and rapidly growing communities.

To measure changes of c_t, we need to know the *corresponding community* c_{t-1} in C_{t-1}. First we define c_{t-1} be the corresponding community that shares the most URLs with c_t. If there were multiple communities that share the same number of URLs, we select a community that has the largest number of URLs.

The metrics are defined by differences between the community c_t and c_{t-1}. We use the following attributes to define evolution metrics.

$N(c_t)$: the number of URLs in the c_t.

$N_{share}(c_{t-1}, c_t)$: the number of URLs shared between c_t and c_{t+1}.

$N_{disappear}(c_{t-1})$: the number of URLs disappeared from c_{t-1} and not appeared in any community at t.

$N_{split}(c_{t-1}, c_t)$: the number of URLs split from c_{t-1} to communities other than c_t.

[†]Currently, we still have not exploited metrics that measure the evolution of relationships between communities in web community charts. Instead, in Section 5, we show that the relationships are useful for locating evolution patterns.

$N_{appear}(c_t)$: the number of URLs appeared in c_t and not appeared in any community at $t-1$.

$N_{merge}(c_{t-1}, c_t)$: the number of URLs merged in c_t from communities other than c_{t-1}.

Then our evolution metrics are defined as follows.

The *growth rate*, $R_{grow}(c_{t-1}, c_t)$, is the most simple evolution metric defined as

$$R_{grow}(c_{t-1}, c_t) = \frac{N(c_t) - N(c_{t-1})}{N(c_t)}.$$

When c_t grows, the growth rate becomes between 0 and 1. When c_t shrinks, the growth rate fells into a negative value. When c_{t-1} does not exist, c_t is totally new and the growth rate becomes 1. The growth rate allows us to find growing and shrinking communities.

The *stability*, $R_{stability}(c_{t-1}, c_t)$, measures how much URLs are preserved in c_t, and is defined as

$$R_{stability}(c_{t-1}, c_t) = \frac{N_{share}(c_{t-1}, c_t)}{2 \cdot N(c_{t-1})} + \frac{N_{share}(c_{t-1}, c_t)}{2 \cdot N(c_t)}.$$

The stability takes a value from 0 to 1. When there is no change of URLs, the stability is 1. Note that c_t may not be stable even if the growth rate of c_t is 0, because c_t may lose and obtain the same number of URLs. A stable community on a topic can be used as a good starting point for finding interesting changes around the topic.

A community obtains and loses URLs in some ways. For example, the community obtains URLs in two ways. One is appearance of URLs, and the other is merging of URLs. The following metrics measure how much the community obtains or loses URLs.

The *novelty*, $R_{novelty}(c_{t-1}, c_t)$, is the ratio of newly appeared URLs in c_t, and defined as

$$R_{novelty}(c_{t-1}, c_t) = N_{appear}(c_t)/N(c_t).$$

The novelty takes a value from 0 to 1. When the novelty is high, we can say that it mainly obtains newly appeared URLs. We can find emerged communities using the novelty metric. If the novelty is 1, we can say that c_t is a newly born community.

The *merge rate*, $R_{merge}(c_{t-1}, c_t)$, is the ratio of absorbed URLs from other communities by merging, and takes a value from 0 to 1. Higher merge rate means that the community mainly obtains URLs by merging. The merge rate is defined as

$$R_{merge}(c_{t-1}, c_t) = N_{merge}(c_{t-1}, c_t)/N(c_t).$$

The *disappearance rate*, $R_{disappear}(c_{t-1}, c_t)$, is the ratio of disappeared URLs from c_{t-1}, and takes a value from 0 to 1. Higher disappear rate means that the community mainly loses URLs by disappearance. The disappear rate is defined as

$$R_{disappear}(c_{t-1}, c_t) = N_{disappear}(c_{t-1})/N(c_{t-1}).$$

The *split rate*, $R_{split}(c_{t-1}, c_t)$, is the ratio of split URLs from c_{t-1}, and takes a value from 0 to 1. When the split rate is low, we can know that c_t is larger than other split communities. Otherwise, c_t is smaller than other split communities. The split rate is defined as

$$R_{split}(c_{t-1}, c_t) = N_{split}(c_{t-1}, c_t)/N(c_{t-1}).$$

By combining these metrics, we can represent some complex evolution patterns as follows. Note that similar evolution patterns can be defined for shrinkage.

Stable growth: A community stably grows when its growth rate is positive, and its disappearance and split rates are low.

Stable growth by appearance: When a community stably grows, and its novelty is high, the community grows mainly by newly appeared URLs.

Stable growth by merge: When a community stably grows and its merge rate is high, the community is grows mainly by merging.

Year	#Pages	Total URLs	#links	#seeds	#communities
1999	16.8M	29.6M	126M	627K	70K
2000	14.1M	23.5M	100M	600K	68K
2001	40.5M	63.3M	343M	1135K	131K

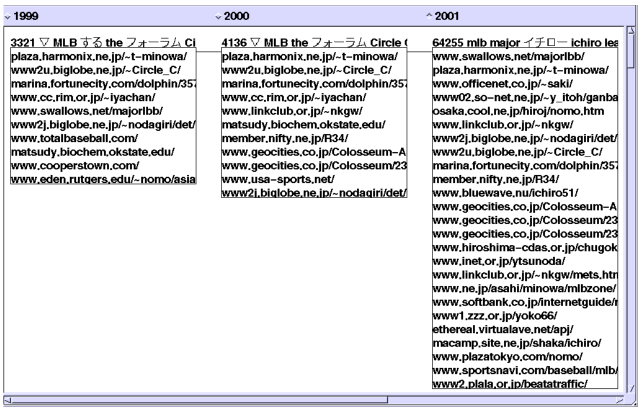
Table 1 Details of our web archives

4. Building Web Community Charts from Series of Web Archives

For experiments, we used three web archives of Japanese web pages (in jp domain) periodically crawled in 1999, 2000, and 2001 (See Table 1). We used the same web crawler in 1999 and 2000, and collected about 17 million pages in each year. The size of the 2001 archive is more than twice of other archives. This is because we improved the crawling rate significantly in 2001. Our crawlers collect pages in the breadth-first order. As you can see in Table 1, the 2000 archive is smaller than one in 1999, because we lost randomly about 3 million pages due to disk crash.

From each archive, we extracted a web graph with URLs and links. Our graph includes not only URLs inside the archive but also those outside the archive that is pointed to by inside URLs. Namely, the graph includes URLs outside jp domain, such as com and edu. Table 1 also shows the number of links and the total URLs. For link analysis, each web graph is stored in a main-memory database that provides out-links and in-links of a given URL. Its implementation details are similar to the connectivity server [9]. We implemented the whole system on Sun Enterprise Server 6500 with 8 CPU and 4GB memory. Building our connectivity database of 2001 takes about one day.

From the above three web graphs, we built three community charts using the technique described in Section 2. To compare web community charts in the same



1999

3321 ▽ MLB する the フォーラム Ci
plaza.harmonix.ne.jp/~t-minowa/
www2u.biglobe.ne.jp/~Circle_C/
marina.fortunecity.com/dolphin/357
www.cc.rim.or.jp/~iyachan/
www.swallows.net/majorlbb/
www2j.biglobe.ne.jp/~nodagiri/det/
www.totalbaseball.com/
matsudy.biochem.okstate.edu/
www.cooperstown.com/
www.eden.rutgers.edu/~nomo/asia

2000

4136 ▽ MLB the フォーラム Circle (
plaza.harmonix.ne.jp/~t-minowa/
www2u.biglobe.ne.jp/~Circle_C/
marina.fortunecity.com/dolphin/357
www.cc.rim.or.jp/~iyachan/
www.linkclub.or.jp/~nkgw/
matsudy.biochem.okstate.edu/
member.nifty.ne.jp/R34/
www.geocities.co.jp/Colosseum-A
www.geocities.co.jp/Colosseum/23
www.usa-sports.net/
www2j.biglobe.ne.jp/~nodagiri/det/

2001

64255 mlb major イチロー ichiro lea
www.swallows.net/majorlbb/
plaza.harmonix.ne.jp/~t-minowa/
www.officenet.co.jp/~saki/
www02.so-net.ne.jp/~y_itoh/ganba
osaka.cool.ne.jp/hiroj/nomo.htm
www.linkclub.or.jp/~nkgw/
www2j.biglobe.ne.jp/~nodagiri/det/
www2u.biglobe.ne.jp/~Circle_C/
marina.fortunecity.com/dolphin/357
member.nifty.ne.jp/R34/
www.bluewave.nu/ichiro51/
www.geocities.co.jp/Colosseum-A
www.geocities.co.jp/Colosseum/23
www.geocities.co.jp/Colosseum/23
www.hiroshima-cdas.or.jp/chugok
www.inet.or.jp/ytsunoda/
www.linkclub.or.jp/~nkgw/mets.htm
www.ne.jp/asahi/minowa/mlbzone/
www.softbank.co.jp/internetguide/
www1.zzz.or.jp/yoko66/
ethereal.virtualave.net/apj/
macamp.site.ne.jp/shaka/ichiro/
www.plazatokyo.com/nomo/
www.sportsnavi.com/baseball/mlb/
www2.plala.or.jp/beatatraffic/

Fig. 4 Evolution of a MLB fan community

condition, we fixed values of parameters, IN and N, for the chart building algorithm in Section 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 decrease seeds. We used the value 10 as N. We selected the value 10, because Companion– provides enough precision with top 10 authorities in our previous work [16], and members of major communities do not change much from the value 9 to 11[†]. It took about one day to build the chart for 2001. Most of the time was spent on calculating related pages of seeds by Companion–. Table 1 shows the number of seeds and communities in the chart of each year.

5. Web Community Evolution Viewer and Evolution Examples

We developed a web community evolution viewer for examining how communities evolved through two years. In this section, we describe the viewer with some simple examples of evolution. Our viewer visualizes the evolution of communities, so that the user can easily compare communities in different time. The user can extract communities evolving in various patterns, by filtering and sorting communities using their relevance and the evolution metrics.

5.1 Visualizing Evolution of a Community

The most basic function of our viewer is showing evolution of a given community in a specified period. Figure 4 depicts how the viewer shows evolution of a community, when the user specifies the community in 2001. Each column represents a time (currently each year). Each community is represented as a rectangle including the list of its URLs. Labels on communities are automatically attached by selecting frequent keywords from anchor texts that point to URLs in the community. Each URL in communities can be browsed with a web browser by clicking it with the mouse.

Once the user specifies a certain community, the viewer displays the corresponding communities for each year which are determined as we discussed in the Section 3.2. We call the sequence of these communities as a *history* of the community. Then communities in the history are arranged horizontally, and lines are drawn between them, so that the user can easily compare their differences. The viewer shows only a history as default. If the user needs more detailed changes (merge and split), the viewer shows multiple communities that share URLs with the given community (See Figure 6).

5.2 Selecting, Sorting, and Filtering

Our viewer provides flexible means to find various kind of evolution. We first describe how to select communities to be displayed, then describe how to sort and filter communities using evolution metrics.

The user can select communities by providing a time, then 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 that have URLs of pages including keywords. In addition, the user can select communities related to a specified community in the chart.

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

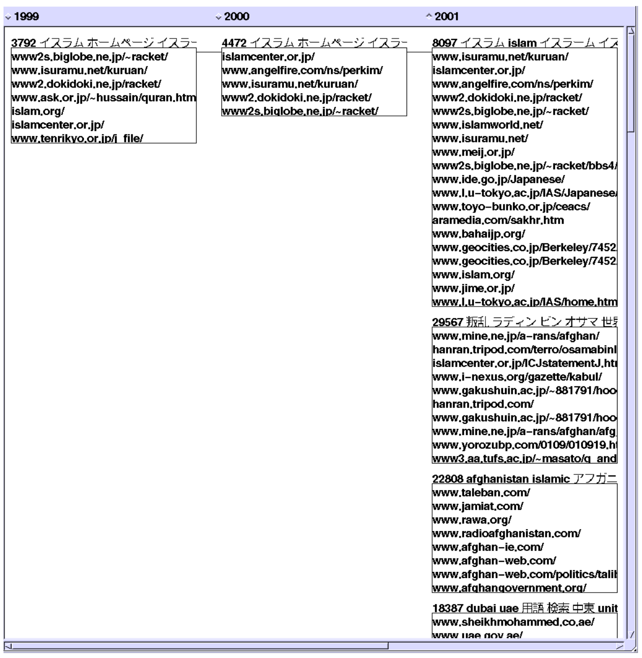
The number of selected communities might become large. So the user can sort the communities and browse the communities in a sorted order. The user can also specify filtering condition and reduce the number of target communities. When the history includes multiple intervals, we use the average of the metric for all intervals. The metrics which 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 patterns.

In addition to the evolution metrics, we can use the following metrics for sorting and filtering.

- Relevance with the given community.
- The number of URLs in the community

To implement these functions, we use chart and evolution databases. A chart database stores URL lists of communities, and their relevance values for each chart. The evolution databases provide how a given community share URLs with communities in neighboring charts. That is, which communities share URLs,

[†]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.



1999

3792 イスラム ホームページ イスラー
www2s.biglobe.ne.jp/~racket/
www.isuramu.net/kuruan/
www2.dokidoki.ne.jp/racket/
www.ask.or.jp/~hussain/quran.htm
islam.org/
islamcenter.or.jp/
www.tenrikyo.or.jp/j_file/

2000

4472 イスラム ホームページ イスラー
islamcenter.or.jp/
www.angelfire.com/ns/perkim/
www.isuramu.net/kuruan/
www2.dokidoki.ne.jp/racket/
www2s.biglobe.ne.jp/~racket/

2001

8097 イスラム islam イスラーム イス
www.isuramu.net/kuruan/
islamcenter.or.jp/
www.angelfire.com/ns/perkim/
www2.dokidoki.ne.jp/racket/
www2s.biglobe.ne.jp/~racket/
www.islamworld.net/
www.isuramu.net/
www.meij.or.jp/
www2s.biglobe.ne.jp/~racket/bbs4/
www.ide.go.jp/Japanese/
www.l.u-tokyo.ac.jp/IAS/Japanese/
www.toyo-bunko.or.jp/ceacs/
aramedia.com/sakhr.htm
www.bahaijp.org/
www.geocities.co.jp/Berkeley/7452
www.geocities.co.jp/Berkeley/7452
www.islam.org/
www.jime.or.jp/
www.l.u-tokyo.ac.jp/IAS/home.htm

29567 叛乱 ラディン ビン オサマ 世
www.mine.ne.jp/a-rans/afghan/
hanran.tripod.com/terro/osamabinl
islamcenter.or.jp/ICJstatementJ.htm
www.i-nexus.org/gazette/kabul/
www.gakushuin.ac.jp/~881791/hoo
hanran.tripod.com/
www.gakushuin.ac.jp/~881791/hoo
www.mine.ne.jp/a-rans/afghan/afg
www.yorozubp.com/0109/010919.h
www3.aa.tufs.ac.jp/~masato/q_and

22808 afghanistan islamic アフガニ
www.taleban.com/
www.jamiat.com/
www.rawa.org/
www.radioafghanistan.com/
www.afghan-ie.com/
www.afghan-web.com/
www.afghan-web.com/politics/talil
www.afghangovernment.org/

18387 dubai uae 用語 検索 中東 unit
www.sheikhmohammed.co.ae/
www.uae.gov.ae/

Fig. 5 Communities about Afghanistan emerged around an Islam information community

and how many URLs are shared. The evolution metrics can be easily calculated by those databases.

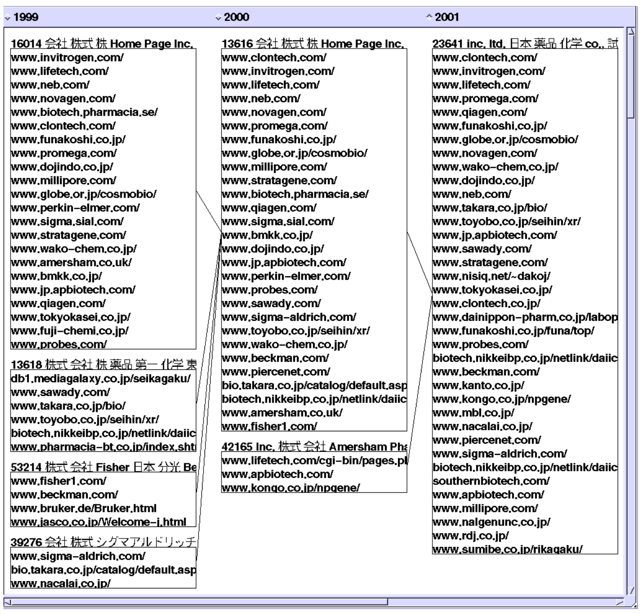
5.3 Examples

Here we show how the relevance and evolution metrics can be used for extracting evolving communities.

Firstly, Figure 4 shows a fan community of Major League Baseball stably grew in Japan. Its growth rate keeps positive, and its disappear and split rate are low for two years. Especially, it rapidly grew from 2000 to 2001, since a Japanese star player Ichiro Suzuki is transferred to Seattle Mariners, and he became an outstanding player in Major League (You can see his name, "ichiro," in some URLs in the community in 2001).

We found the MLB (Figure 4) fan communities as stably growing communities around the community of Japanese baseball teams. We first select communities related to the community of Japanese baseball teams at 2001. Then we sorted these related communities by their growth rate, and filtered communities with high disappear and split rates.

Secondly, we can find emerged communities related to a particular community. In Figure 5, we first select related communities to a community of Islam and Muslim information in 2001. Then we sort these related communities by their novelty metric from 2000 to 2001. There emerged two communities about Afghanistan. We guess that such hubs are rapidly created after the attack on America by terrorists on 11 September, 2001. (Note that our web archive in 2001 was crawled in early October.) From this example, we can see that communities grow very quickly when their topic has a great impact to the real society.



1999

16014 会社 株式 株 Home Page Inc.
www.invitrogen.com/
www.lifetech.com/
www.neb.com/
www.novagen.com/
www.biotech.pharmacia.se/
www.clontech.com/
www.funakoshi.co.jp/
www.promega.com/
www.dojindo.co.jp/
www.millipore.com/
www.globe.or.jp/cosmobio/
www.perkin-elmer.com/
www.sigma.sial.com/
www.stratagene.com/
www.wako-chem.co.jp/
www.amersham.co.uk/
www.bmkk.co.jp/
www.jp.apbiotech.com/
www.qiagen.com/
www.tokyokasei.co.jp/
www.fuji-chemi.co.jp/
www.probes.com/

13618 株式 会社 株 薬品 第一 化学 東
db1.mediagalaxy.co.jp/seikagaku/
www.sawady.com/
www.takara.co.jp/bio/
www.toyobo.co.jp/seihin/xr/
biotech.nikkeibp.co.jp/netlink/daiic
www.pharmacia-bt.co.jp/index.sht

53214 株式 会社 Fisher 日本 分光 Be
www.fisher1.com/
www.beckman.com/
www.bruker.de/Bruker.html
www.jasco.co.jp/Welcome-j.html

39276 会社 株式 シグマアルドリッチ
www.sigma-aldrich.com/
bio.takara.co.jp/catalog/default.asp
www.nacalai.co.jp/

2000

13616 会社 株式 株 Home Page Inc.
www.clontech.com/
www.invitrogen.com/
www.lifetech.com/
www.neb.com/
www.novagen.com/
www.promega.com/
www.funakoshi.co.jp/
www.globe.or.jp/cosmobio/
www.millipore.com/
www.stratagene.com/
www.biotech.pharmacia.se/
www.qiagen.com/
www.sigma.sial.com/
www.bmkk.co.jp/
www.dojindo.co.jp/
www.jp.apbiotech.com/
www.perkin-elmer.com/
www.probes.com/
www.sawady.com/
www.sigma-aldrich.com/
www.toyobo.co.jp/seihin/xr/
www.wako-chem.co.jp/
www.beckman.com/
www.piercenet.com/
bio.takara.co.jp/catalog/default.asp
biotech.nikkeibp.co.jp/netlink/daiic
www.amersham.co.uk/
www.fisher1.com/

42165 Inc. 株式 会社 Amersham Pha
www.lifetech.com/cgi-bin/pages.pl
www.apbiotech.com/
www.kongo.co.jp/npgene/

2001

23641 inc. ltd. 日本 薬品 化学 co., 試
www.clontech.com/
www.invitrogen.com/
www.lifetech.com/
www.promega.com/
www.qiagen.com/
www.funakoshi.co.jp/
www.globe.or.jp/cosmobio/
www.novagen.com/
www.wako-chem.co.jp/
www.dojindo.co.jp/
www.neb.com/
www.takara.co.jp/bio/
www.toyobo.co.jp/seihin/xr/
www.jp.apbiotech.com/
www.sawady.com/
www.stratagene.com/
www.nisiq.net/~dakoj/
www.tokyokasei.co.jp/
www.clontech.co.jp/
www.dainippon-pharm.co.jp/labop
www.funakoshi.co.jp/funa/top/
www.probes.com/
biotech.nikkeibp.co.jp/netlink/daiic
www.beckman.com/
www.kanto.co.jp/
www.kongo.co.jp/npgene/
www.mbl.co.jp/
www.nacalai.co.jp/
www.piercenet.com/
www.sigma-aldrich.com/
biotech.nikkeibp.co.jp/netlink/daiic
southernbiotech.com/
www.apbiotech.com/
www.millipore.com/
www.nalgenunc.co.jp/
www.rdj.co.jp/
www.sumibe.co.jp/rikagaku/

Fig. 6 Communities of biotechnology companies

Finally, Figure 6 shows that a web community gradually grows by absorbing other small communities on the same topic. Communities of biotechnology and biochemistry companies are gradually growing by merging other small communities, as biotechnology become a great business, and awareness of biotechnology is growing. We can find such communities using our evolution viewer by searching stably growing communities with high merge rates. This evolution pattern occurs for the following two reasons: (1) existing hubs are gradually added new links, and become larger and sophisticated; and (2) Larger and sophisticated hubs are newly created over time, and become major hubs.

6. Discussion and Future Work

Our system takes about two days for building a web graph database, a community chart, and databases of evolution from a web archive with 40 million pages. Since it is significantly shorter than the time for crawling (some weeks), our system can analyze evolution at the same interval of crawling.

We have separately developed viewers for web community chart and community evolution. We plan to integrate them, 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 (one year). 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.

References

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

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

[3] K. Bharat and G. A. Mihaila. When Experts Agree: Using Non-Affiliated Experts to Rank Popular Topics. In *Proc. 10th WWW Conference*, 2001.

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

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

[6] J. Cho and H. Garcia-Molina. The Evolution of the Web and Implications for an Incremental Crawler. In *Proc. 26th VLDB Conference*, 2000.

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

[8] D. Gibson et al. Inferring Web Communities from Link Topology. In *Proc. HyperText98*, 1998.

[9] K. Bharat et al. The Connectivity Server: fast access to linkage information on the Web. In *Proc. 7th International WWW Conference*, 1998.

[10] R. Kumar et al. Extracting large-scale knowledge bases from the web. In *Proc. 25th VLDB Conference*, 1999.

[11] R. Kumar et al. Trawling the Web for emerging cyber-communities. In *Proc. 8th WWW Conference*, 1999.

[12] S. Chakrabarti et al. Automatic resource compilation by analyzing hyperlink structure and associated text. In *Proc. 7th International WWW Conference*, 1998.

[13] G. W. Flake, S. Lawrence, and C. L. Giles. Efficient Identification of Web Communities. In *Proc. KDD 2000*, 2000.

[14] J. M. Kleinberg. Authoritative Sources in a Hyperlinked Environment. In *Proc. ACM-SIAM Symposium on Discrete Algorithms*, 1998.

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

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

Appendix A: Algorithm of 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 and hubs in the graph. Then authorities are returned 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.

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, 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]: (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, $hub(n)$, and an authority score, $auth(n)$ for each node n in (V, E). The following is the calculation process, 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. Assign 1 to $hub(n)$ and $auth(n)$ of each node.

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

$hub(n) \leftarrow \sum_{(n,m)\in E} auth(m) \times hub\_weight(n, m)$
$auth(n) \leftarrow \sum_{(m,n)\in E} hub(m) \times auth\_weight(m, n)$
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. Return N nodes with highest authority scores.

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

Masaru Kitsuregawa received the B.E. in electronic engineering in 1978, the Master and Doctor of Engineering degree in information engineering from the University of Tokyo, in 1980 and 1983 respectively. In 1983 he joined the Institute of Industrial Science, The Univ. of Tokyo as a lecturer. He is currently a professor. His research for the last 10 years has been directed toward the design of high performance relational database systems such as parallel hash join algorithm, its skew handler, an intelligent disk system named functional disk system, highly parallel architecture for relational SQL server SDC (Super Database Computer), high speed hardware sorter, disk array, persistent programming system, KD join algorithm, and digital library for earth engineering. He is a member of VLDB endowment, a chairman of Data Engineering Technical Group, IEICE Information and Systems Society, and an Asian coordinator of IEEE TCDE.