Evolution of Web Communities

Introduction

- Web community - set of web pages created by individuals or associations with a common interest on a topic
- Link analysis techniques - to extract web communities, but not for web community evolution
- Degree of web community evolution metrics - growth rate, novelty, and stability
- Web graph – helps to identify web community on a topic by extracting densely connected structure
- In a web graph – *nodes* are web pages and edges are hyperlinks
- web community differs from a community of people – Ex. web community may include competing companies

Application of Web Community Evolution Analysis

- Answering historical queries about topics on the Web
- Statistics on members (URLs) have appeared and disappeared
- Marketing Products
- Tracking social and cultural trends over time
- Emergence of quality web communities on a specific topic

Extraction of Web Community Evolution

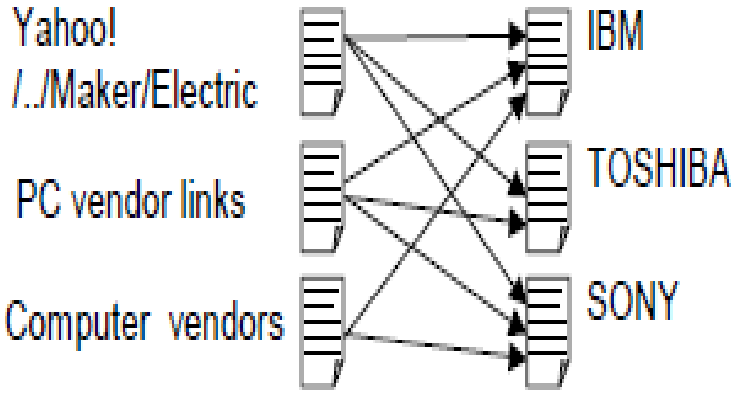
- Consists of two parts
- First part - extracts communities and their relevance from web archive using *web community chart*
- *Web community chart - r*elated communities are connected by weighted edges
- Allows Navigate through related communities, locate evolution around a particular community

Contd…

- Second Part:
- Web community evolution viewer - browsing communities evolution
- Allows to locate evolution of communities - emergence and growing

Authorities and Hub

- Concept proposed by Kleinberg
- An authority - page with good contents on a topic
- A hub - page with a list of hyperlinks to valuable pages on the topic (points to many good authorities)
- HITS – algorithm that extracts authorities and hubs from a given sub-graph



Improvement in HITS

- HITS - improved exploiting anchor texts, edge weighting, document similarity, and Document Object Models
- Dean and Henzinger tailored HITS for finding related pages
- They proposed Companion, a related page algorithm
- Companion - takes seed page as input, outputs related pages to the seed
- Its precision improved by considering the order of links in a page

Work done in Authorities and Hubs

- Kleinberg - A set of authorities and hubs was regarded as a community
- Gibson et al. investigated the characteristic of communities derived by HITS
- Kumar et al. extracted > 100,000 cores of communities from web snapshot based on their graph evolution model
  - A core represents small complete bipartite graphs that consist of authorities and hubs
- Lempel and Moran proposed a random walk model for calculating authorities
- Flake et al. redefined a community - given seed pages as a subgraph separated from the Web using a maximum flow/minimum cut framework

Contd…

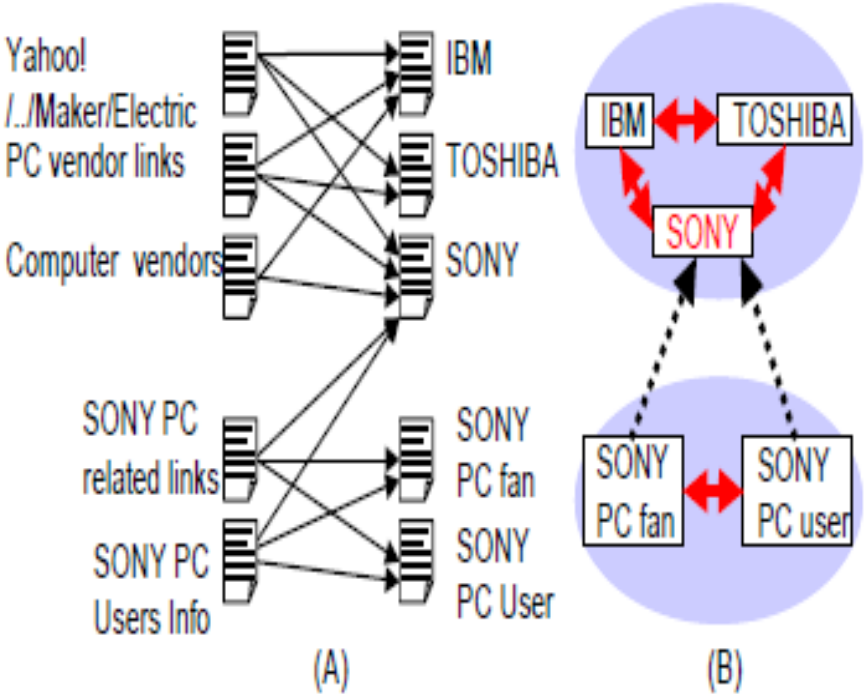
- Change frequency and lifetime of web pages has been studied
- Use this for web crawlers to determine timing for re-crawl
- Based on the page level analysis
- Site (or server) level analysis of web graph evolution is also studied
- Internet Archive to see past web pages stored in the Internet Archive's web archive
- User specify a single URL, see the past pages of that URL
- It is impossible to understand what topics are popular in the past

Overall Working of Web Community Chart Algorithm

- Algorithm builds web community chart from given set of seed pages
- Each seed page applied HITS and related page algorithm (RPA)
- Then, investigate how each seed derives other seeds as related pages
- RPA builds subgraph around the seed, extracts authorities and hubs from the graph using HITS
- Companion– improves the precision of HITS and Companion
- It prunes error parts of subgraph used in HITS and Companion

Example

- In graph (B) two SONY PC fans, they derive each other and SONY as related pages
- SONY does not derive these fans, (no. of electric company lists > link lists of SONY PC)
- Symmetric derivation is a strong relationship – Ex. Community
- Asymmetric derivation is a weak relationship – Ex. between 2 communities



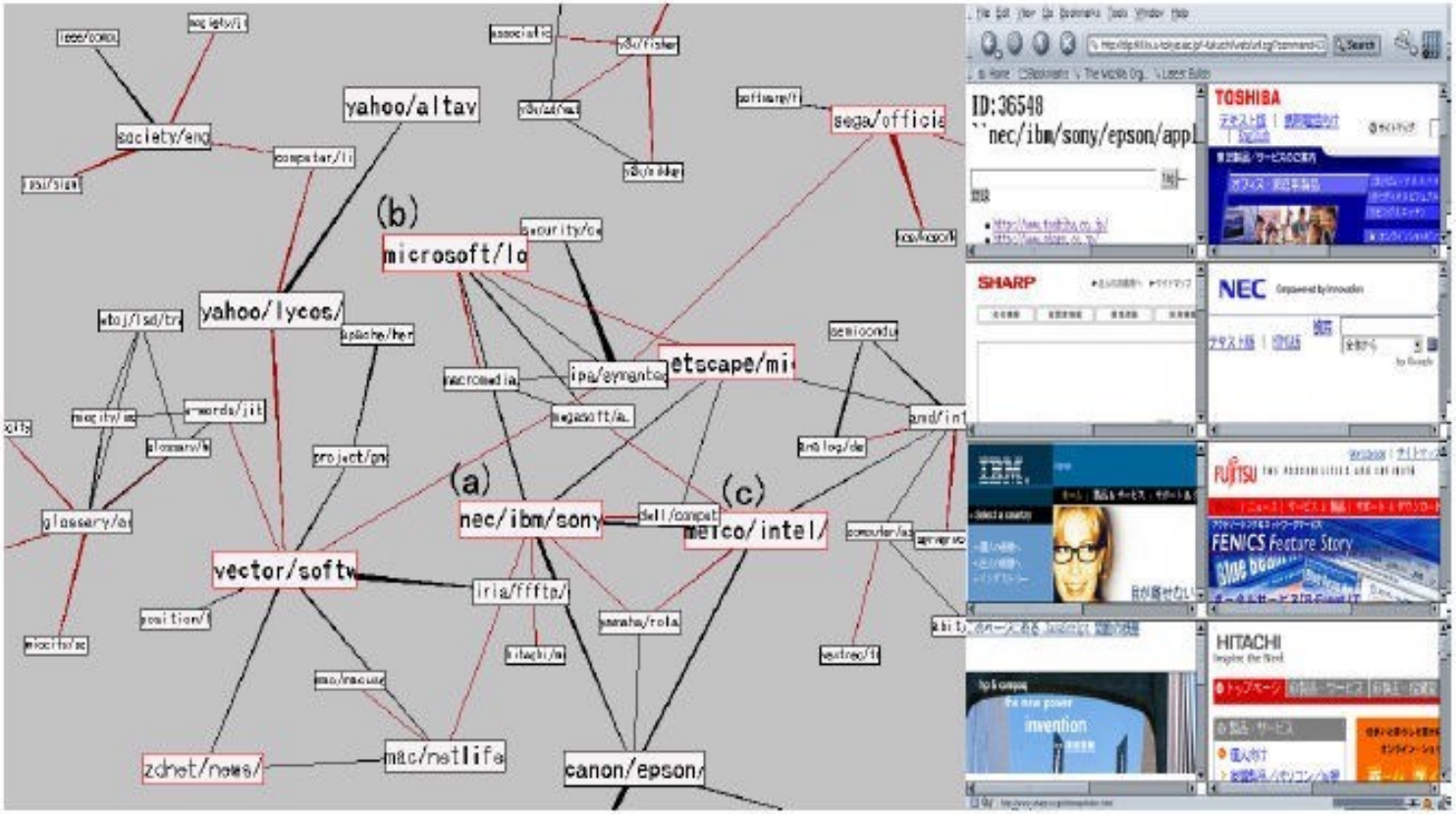
Algorithm for Constructing Web Community Charts

- Requisites:
- Web pages that have inlinks from *more* different servers (parameter to determine seed set)
- Build a directed graph that shows how each seed derives other seeds by Companion– (top *N authorities,* where *N is a parameter) – called* authority derivation graph (ADG)
- extract a symmetric derivation graph (SDG) from ADG, to extract web communities from SDG
- extract densely connected subgraphs in SDG as *cores of web communities*
- form cores into communities by adding remaining nodes to these cores

Contd...

- Core - a core is defined as a set of triangles that share edges in SDG
- Extract all triangles of nodes from SDG. Two cores share some nodes, isolate them, and pass them to the next step
- Add remaining nodes in SDG to a neighboring core, if it has edges
  - If multiple candidates, select a core with most incoming edges.
  - Each core then becomes a community
- Remaining connected components that do not form triangles, extract such components as communities

Example of a Web Community Chart



EVOLUTION OF WEB COMMUNITIES

- Types of Changes – **Emerge, Dissolve, Grow and shrink, Split, Merge**

- Notations used:

- *t1, t2, ..., tn*: ***Time when each archive crawled***
- *W(tk)*: ***The Web archive at time t_k.***
- *C(tk)*: ***The web community chart at time t_k.***
- *c(tk), d(tk), e(tk), ...*: ***Communities in C(t_k).***

Types of Change

- Emerge: A community *c(tk) emerges in C(tk), when c(tk)* shares no URLs with any community in *C(tk−1)*

- Dissolve: A community *c(tk−1) in C(tk−1) has dissolved,* when *c(tk−1) shares no URLs with any community in C(tk)*

- Grow or Shrink: The community grows when new URLs appear in *c(tk), and shrinks when* URLs disappeared from *c(tk−1)*

- Split: A community *c(tk−1) may splits into smaller communities, then c(tk−1) shares URLs with multiple* communities in *C(tk)*

- Merge: When multiple communities (*c(tk−1), d(tk−1), ...)* share URLs with a single community *e(tk), these communities* are merged into *e(tk)*

Evolution Metrics

- *N(c(tk)): the number of URLs in the c(tk), similarly for others...*
- *Growth Rate, Rgrow(c(tk−1), c(tk)), represents the increase* of URLs per unit time
  - allows us to find most growing or shrinking communities
  - when *c(tk−1) does not exist,* use zero as *N(c(tk−1))*

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

Contd...

- *Rstability(c(tk−1), c(t)), - amount of* disappeared, appeared, merged and split URLs per unit time
- No change of URLs, the stability becomes zero
- A stable community is desirable

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

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

Contd...

- *Rnovelty(c(tk−1), c(tk)), represents the number* of newly appeared URLs per unit time
- *c(tk) has grown mainly by* newly appeared URLs
- Most emerged communities at time *tk can be found by sorting communities by the novelty*

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

Contd...

- *disappearance rate, Rdisappear(c(tk−1), c(tk)), is the number* of disappeared URLs from *c(tk−1) per unit time*

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

- *merge rate, Rmerge(c(tk−1), c(tk)), is the number of absorbed* URLs from other communities by merging per unit time

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

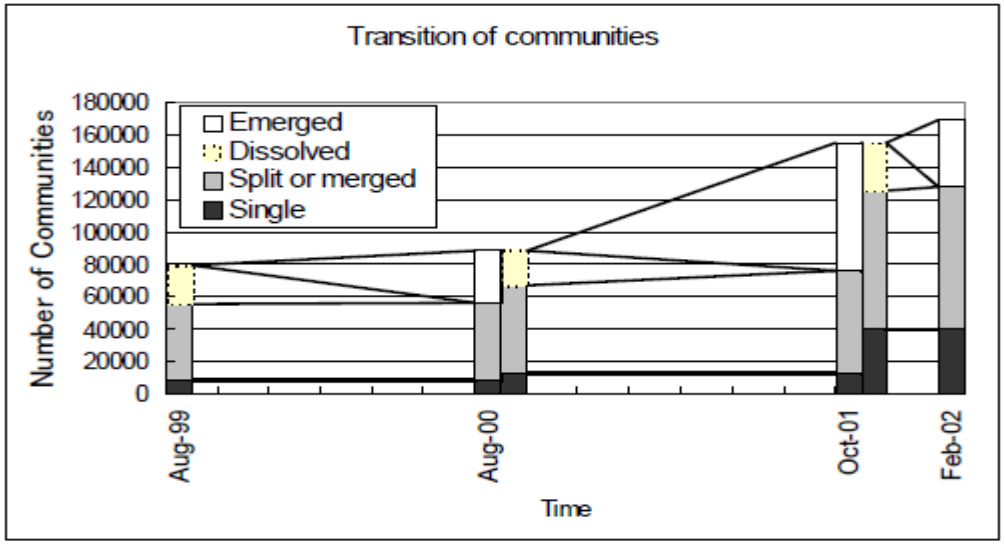
Contd...

- *split rate, Rsplit(c(tk−1, c(tk)), is the number of split* URLs from *c(tk−1) per unit time*

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

ANALYSIS

- For experiments, 4 web archives of Japanese web pages (in .jp domain) crawled in 1999, 2000, 2001, and 2002 used
- From 1999 and 2000, 17 million pages in each year collected
- In 2001, it doubled due to improved crawling rate
- web graph build with URLs and links by extracting anchors from all pages in the archive



Contd...

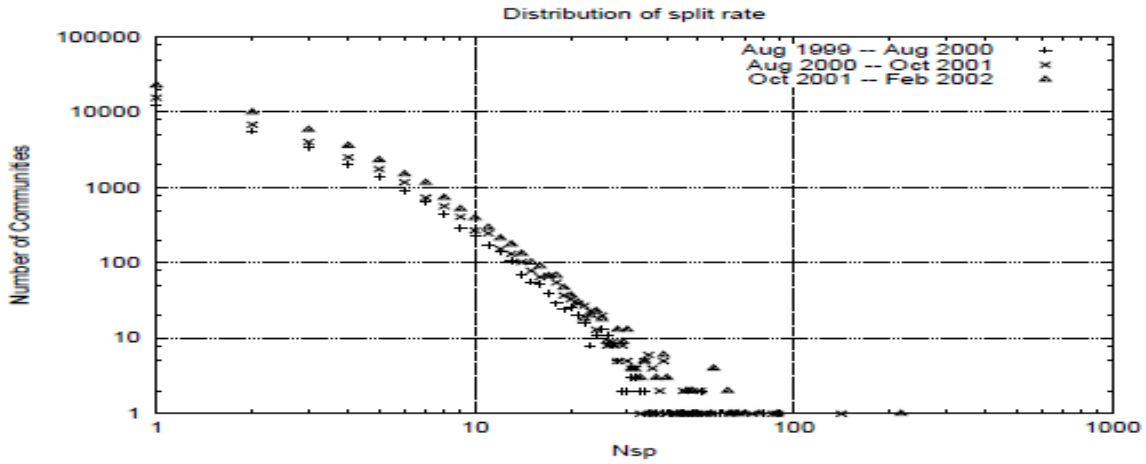


Figure 7: Distribution of split rate

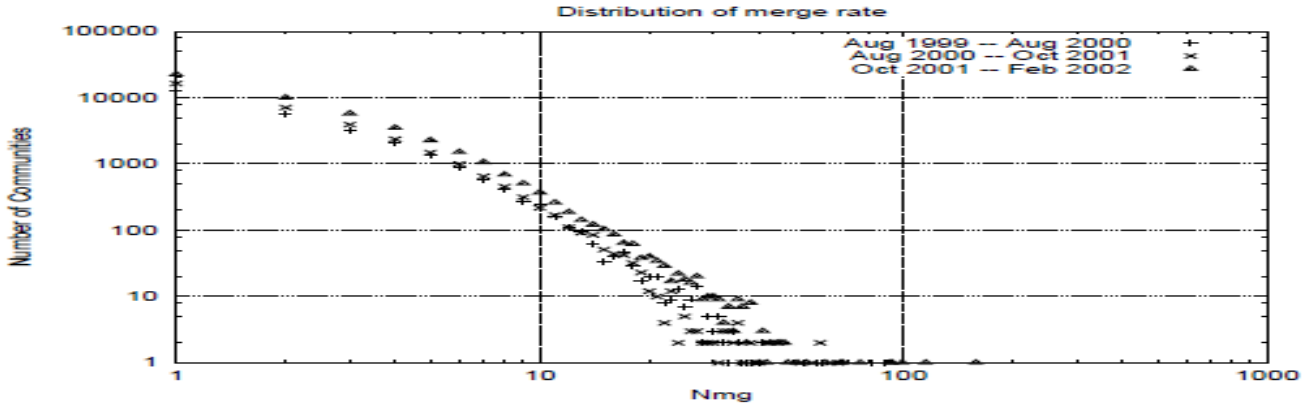


Figure 8: Distribution of merge rate

Contd...

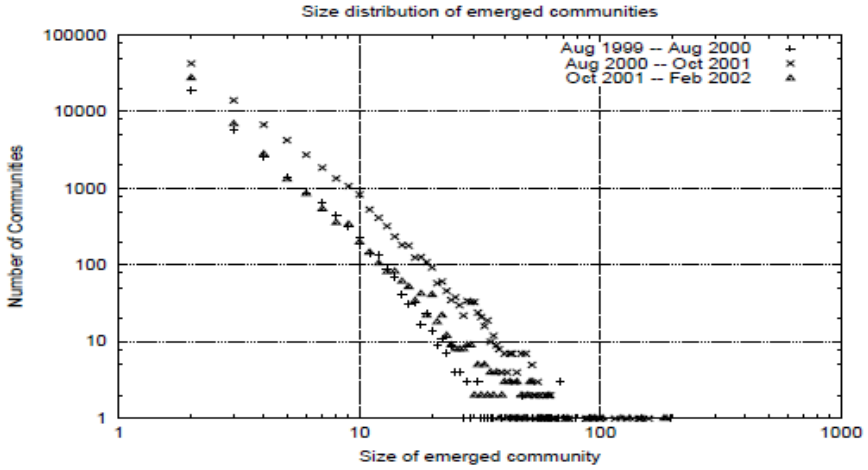


Figure 9: Size distribution of emerged communities

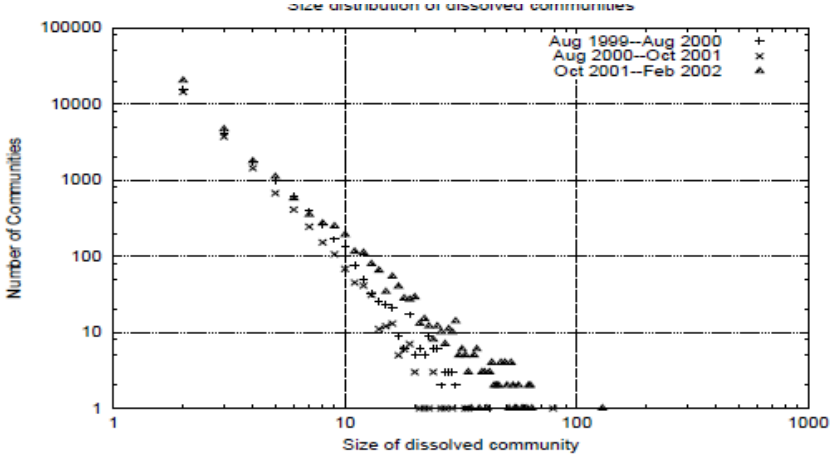


Figure 10: Size distribution of dissolved communities

Findings

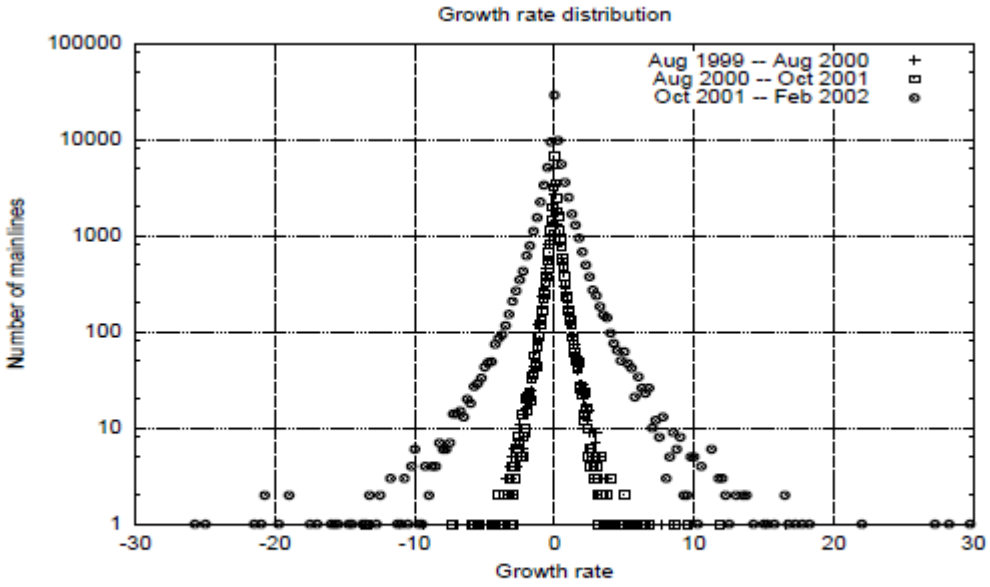


Figure 11: Distribution of growth rate

Findings

- All 4 curves follow power law distribution
- size distribution of communities - follows the power law and its exponent (2.9 to 3.0)did not change so much over time
- split or merge rate is small in most cases (almost the same number of communities are merged at *tk with the same rate as the* split rate)
- *Emerged and Dissolved Communities* - small communities are easy to emerge and dissolve
- growth rate is small for most of communities, and the graph has clear y-axis symmetry

Contd...

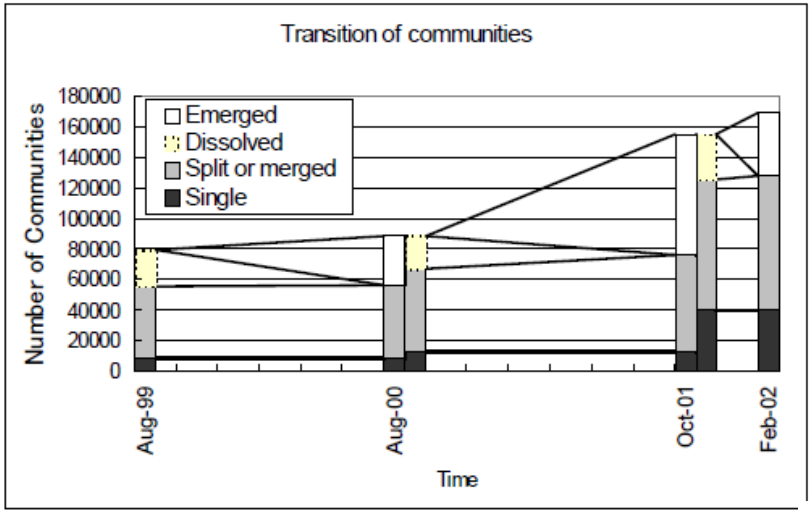


Figure 6: Transition of communities

	t_{k-1}: Aug. 99 t_k: Aug. 00	Aug. 00 Oct. 01	Oct. 01 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

Findings

- *Types of Changes: from tk−1 to tk :*
- Each bar represents the number of communities in charts at the time
- Bars with split vertically, shows it has previous and the next charts to be compared
- more than half of communities are involved in split and merge
- number of single communities is small
- About half of survived communities in 1999 and 2000 are included in main lines for one year
- About 66% of survived communities in 2001 are included in main lines for four months

EVOLUTION VIEWER

- Viewer provides various means to extract evolving communities as follows:
- Searching communities by keywords or a URL
- Sorting and filtering communities by the evolution metrics defined in the viewer
- User can extract various kinds of evolving communities, - most growing communities, the most emerging communities related to a specific topic

Example - find a community of Islam in 2001 by the keyword ”Islam”

Figure 13: Emerged communities around a pacifist community