# Introduction to Information Retrieval

## The PageRank Citation Ranking: Bring Order to the web

# Motivation and Introduction

- Search results (similarity + importance)

- Why is Page Importance Rating important?
  - Huge number of web pages:
  - Diversity of web pages: different topics, different quality, etc.

- What is PageRank?
  - A method for rating the importance of web pages objectively and mechanically using the link structure of the web.

# The History of PageRank

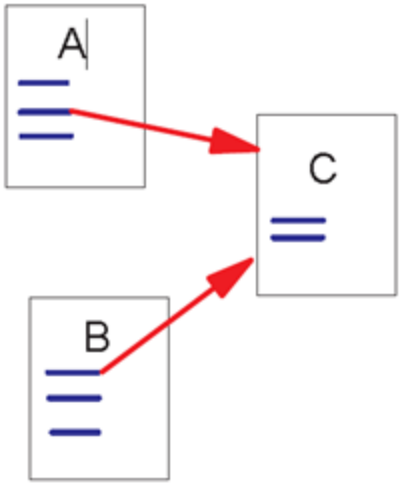
- PageRank was developed by Larry Page (hence the name *Page*-Rank) and Sergey Brin.

- It is first as part of a research project about a new kind of search engine.  That project started in 1995 and led to a functional prototype in 1998.

- Shortly after, Page and Brin founded Google.
- 16 billion...

# Link Structure of the Web

- 150 million web pages  1.7 billion links




Backlinks and Forward links:

➢A and B are C's backlinks

➢C is A and B's forward link

Intuitively, a webpage is important if it has a lot of backlinks.

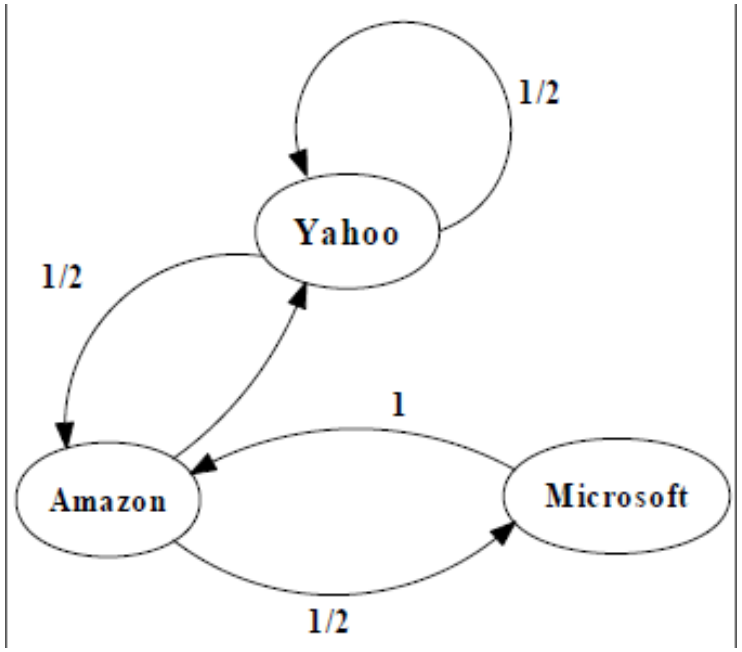
What if a webpage has only one link off www.yahoo.com?

# A Simple Version of PageRank

$$R(u) = c \sum_{v \in B_u} \frac{R(v)}{N_v}$$

- u: a web page
- $B_u$: the set of u's backlinks
- $N_v$: the number of forward links of page v
- c: the normalization factor to make $||R||_{L1} = 1$ ($||R||_{L1} = |R_1 + \ldots + R_n|$)

# An example of Simplified PageRank



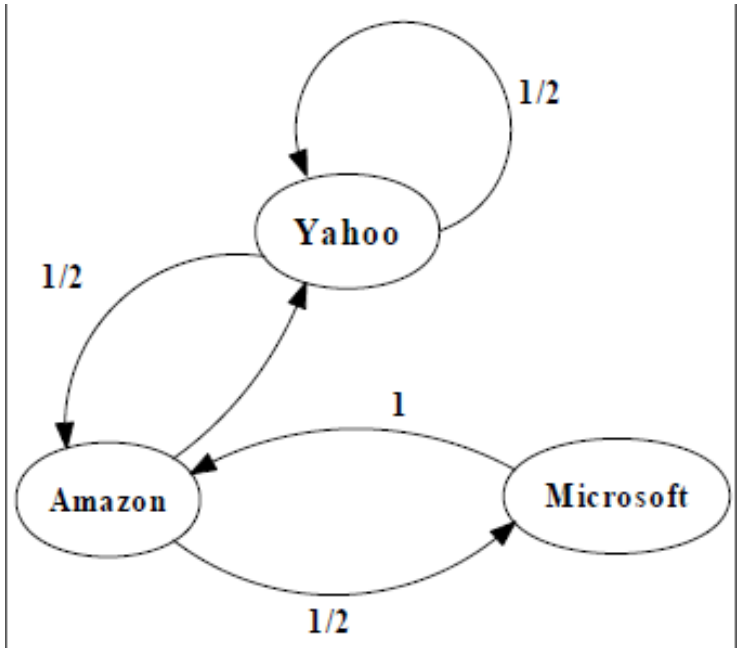

$$M = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix}$$

$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

$$\begin{bmatrix} 1/3 \\ 1/2 \\ 1/6 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix} \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

PageRank Calculation: first iteration

# An example of Simplified PageRank



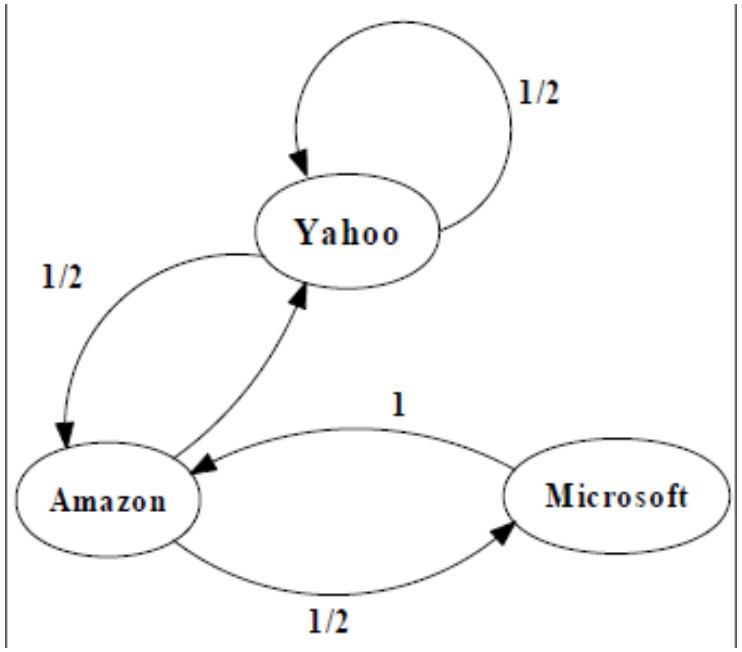

$$M = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix}$$

$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

$$\begin{bmatrix} 5/12 \\ 1/3 \\ 1/4 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix} \begin{bmatrix} 1/3 \\ 1/2 \\ 1/6 \end{bmatrix}$$

PageRank Calculation: second iteration

# An example of Simplified PageRank




$$M = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 1 \\ 0 & 1/2 & 0 \end{bmatrix}$$

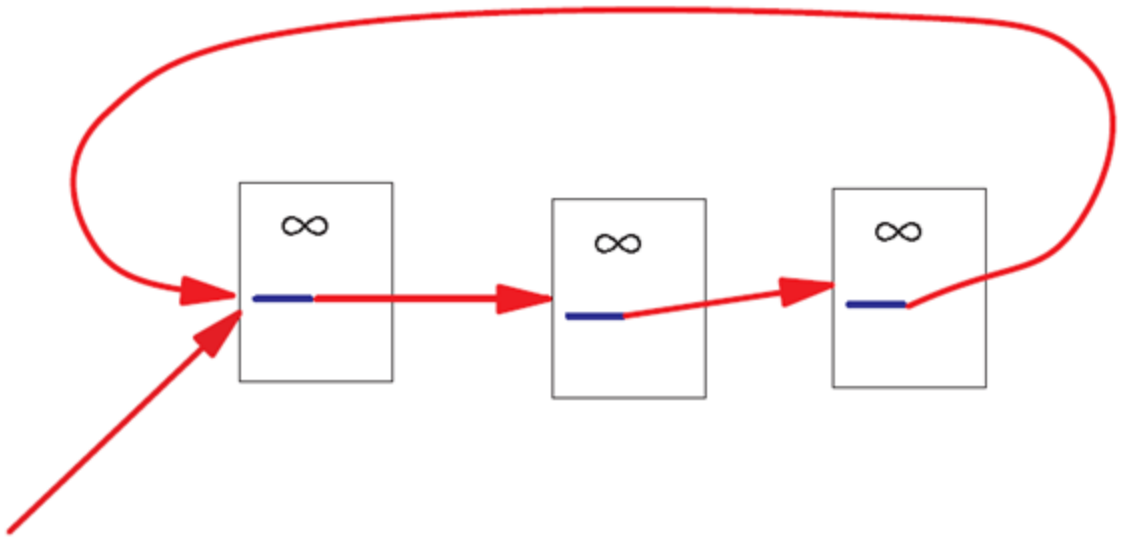
$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

$$\begin{bmatrix} 3/8 \\ 11/24 \\ 1/6 \end{bmatrix} \begin{bmatrix} 5/12 \\ 17/48 \\ 11/48 \end{bmatrix} \ldots \begin{bmatrix} 2/5 \\ 2/5 \\ 1/5 \end{bmatrix}$$

Convergence after some iterations

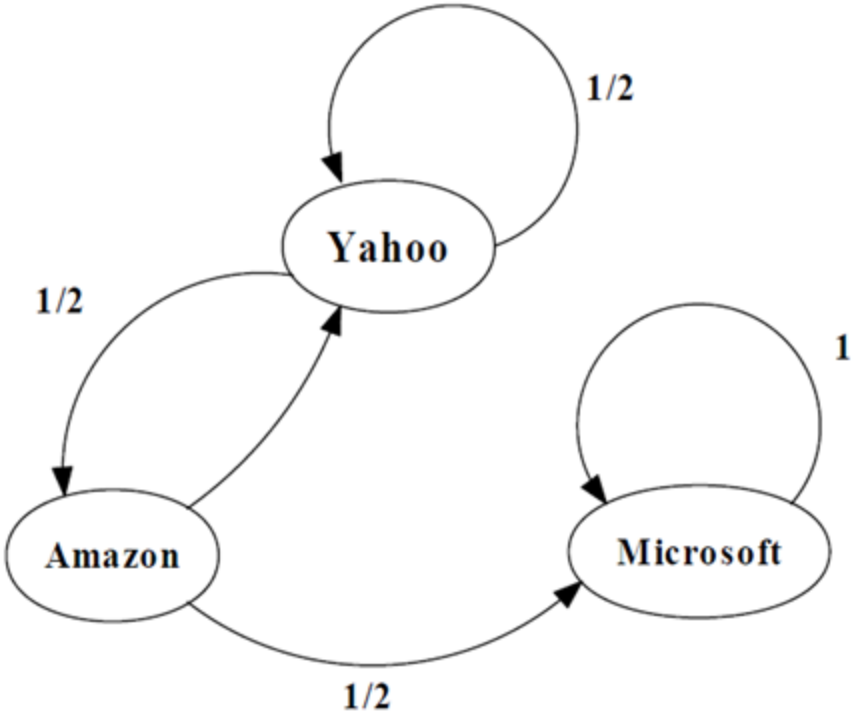
# A Problem with Simplified PageRank

A loop:




During each iteration, the loop accumulates rank but never distributes rank to other pages!

# An example of the Problem

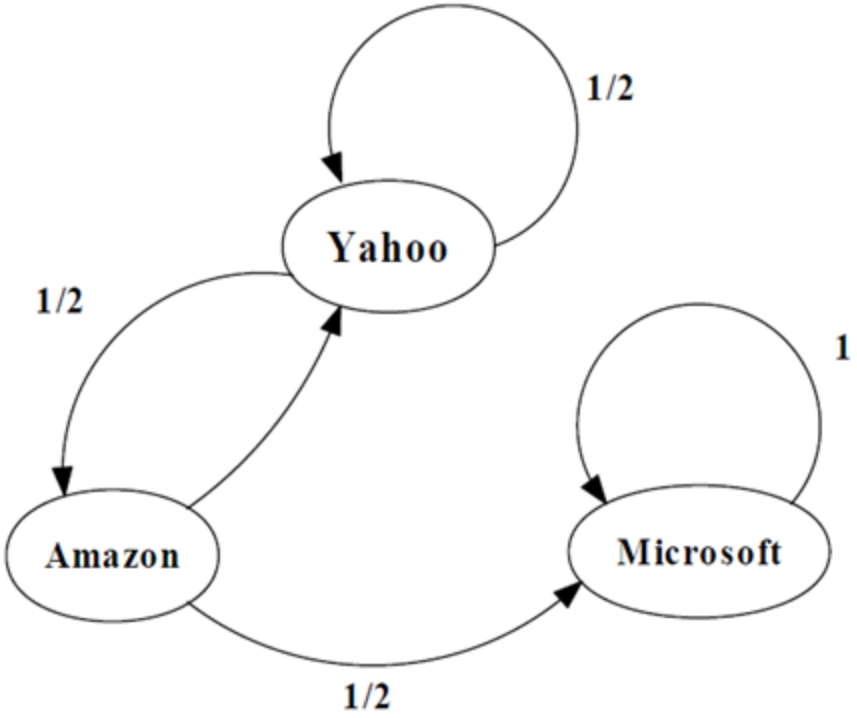



$$M = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

$$\begin{bmatrix} 1/3 \\ 1/6 \\ 1/2 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

# An example of the Problem

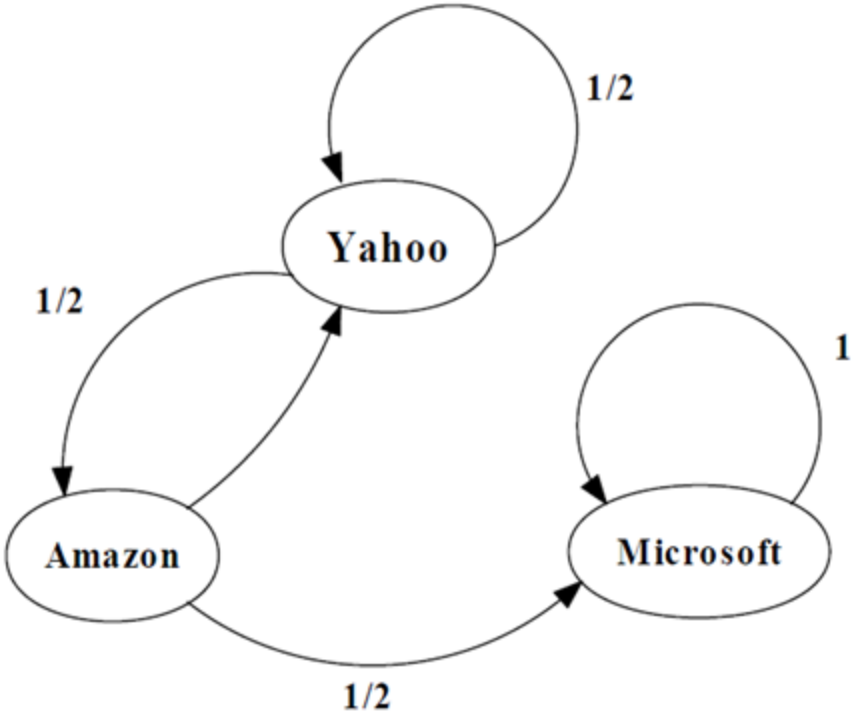



$$M = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

$$\begin{bmatrix} 1/4 \\ 1/6 \\ 7/12 \end{bmatrix} = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix} \begin{bmatrix} 1/3 \\ 1/6 \\ 1/2 \end{bmatrix}$$

# An example of the Problem




$$M = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

$$\begin{bmatrix} 5/24 \\ 1/8 \\ 2/3 \end{bmatrix} \begin{bmatrix} 1/6 \\ 5/48 \\ 35/48 \end{bmatrix} \quad .... \quad \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

# Random Walks in Graphs

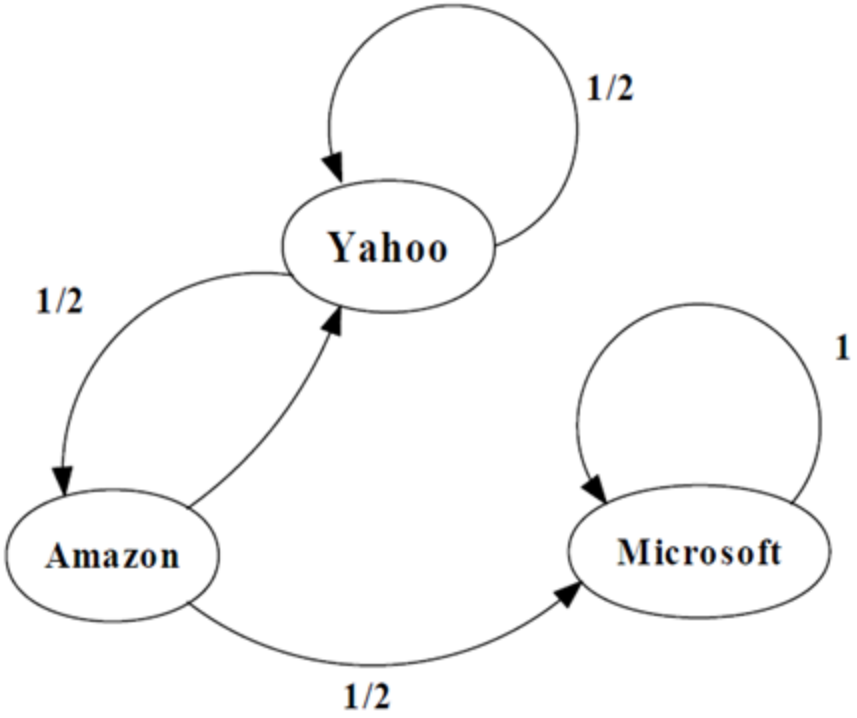
- The Random Surfer Model
  - The simplified model: the standing probability distribution of a random walk on the graph of the web. simply keeps clicking successive links at random
- The Modified Model
  - The modified model: the “random surfer” simply keeps clicking successive links at random, but periodically “gets bored” and jumps to a random page based on the distribution of E

# Modified Version of PageRank

$$R'(u) = c_1 \sum_{v \in B_u} \frac{R'(v)}{N_v} + c_2 E(u)$$

E(u): a distribution of ranks of web pages that “users” jump to when they “gets bored” after successive links at random.

# An example of Modified PageRank

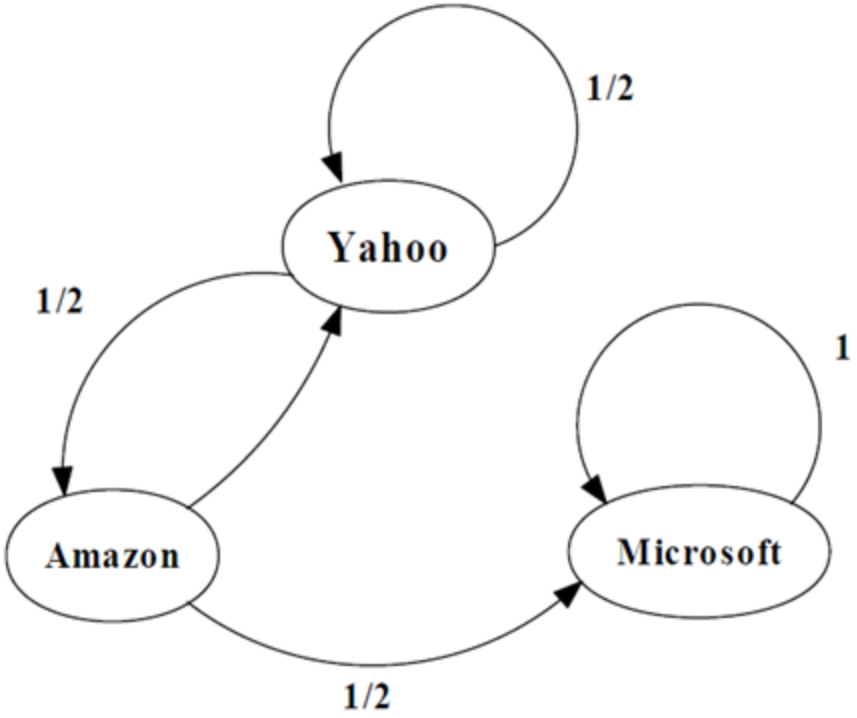



$$M = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

$C_1 = 0.8$ $C_2 = 0.2$

# An example of Modified PageRank




$$M = \begin{bmatrix} 1/2 & 1/2 & 0 \\ 1/2 & 0 & 0 \\ 0 & 1/2 & 1 \end{bmatrix}$$

$$\begin{bmatrix} \text{yahoo} \\ \text{Amazon} \\ \text{Microsoft} \end{bmatrix} = \begin{bmatrix} 1/3 \\ 1/3 \\ 1/3 \end{bmatrix}$$

$C_1 = 0.8 \qquad C_2 = 0.2$

$$\begin{bmatrix} 0.333 \\ 0.333 \\ 0.333 \end{bmatrix} \begin{bmatrix} 0.333 \\ 0.200 \\ 0.467 \end{bmatrix} \begin{bmatrix} 0.280 \\ 0.200 \\ 0.520 \end{bmatrix} \begin{bmatrix} 0.259 \\ 0.179 \\ 0.563 \end{bmatrix} \ldots \begin{bmatrix} 7/33 \\ 5/33 \\ 21/33 \end{bmatrix}$$

# Dangling Links

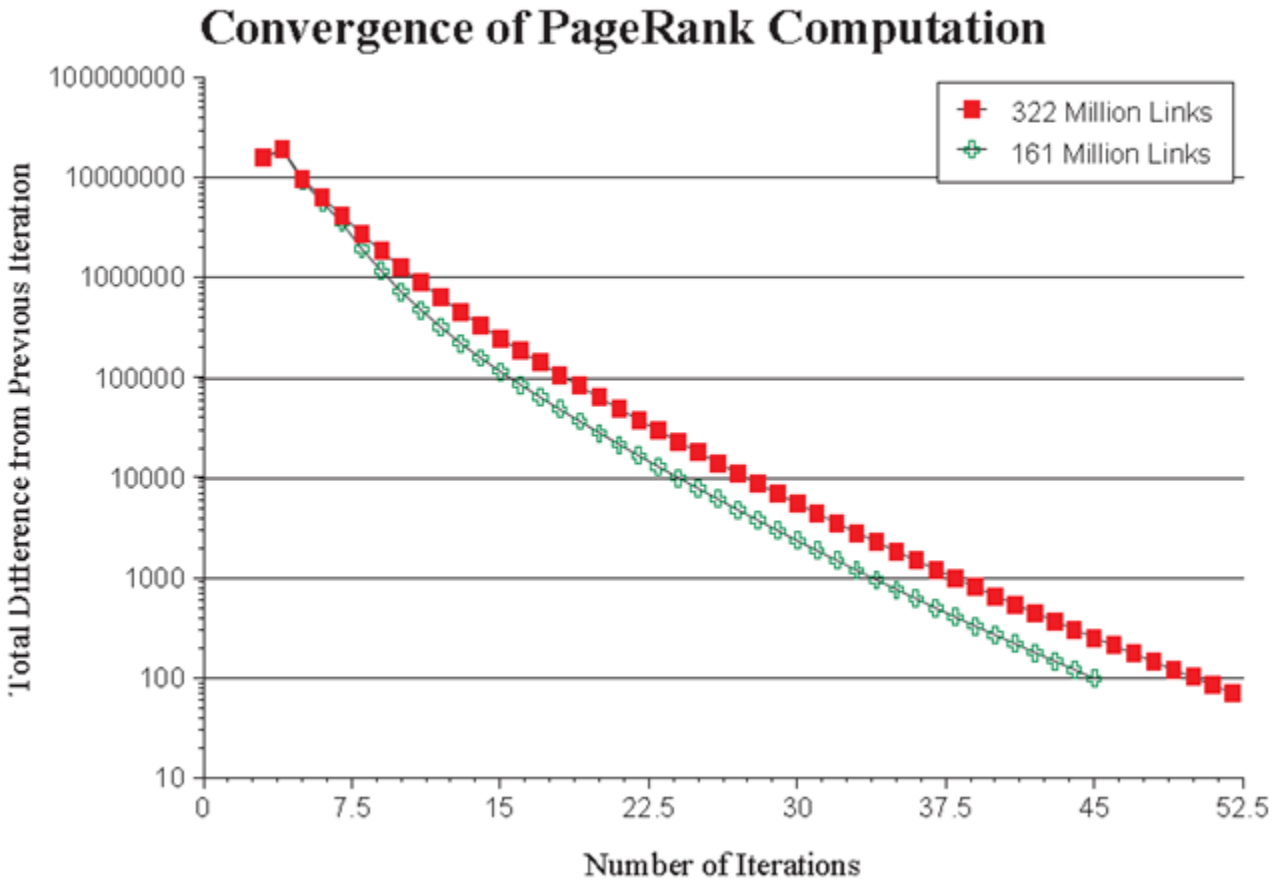
- Links that point to any page with no outgoing links
- Most are pages that have not been downloaded yet
- Affect the model since it is not clear where their weight should be distributed
- Do not affect the ranking of any other page directly
- Can be simply removed before pagerank calculation and added back afterwards

# PageRank Implementation

- Convert each URL into a unique integer and store each hyperlink in a database using the integer IDs to identify pages
- Sort the link structure by ID
- Remove all the dangling links from the database
- Make an initial assignment of ranks and start iteration
  - Choosing a good initial assignment can speed up the pagerank
- Adding the dangling links back.

# Convergence Property

- PR (322 Million Links): 52 iterations
- PR (161 Million Links): 45 iterations
- Scaling factor is roughly linear in *logn*



# Convergence Property

- Theory of random walk:
  - A random walk on a graph is said to be rapidly-mixing if it quickly converges to a limiting distribution on the set of nodes in the graph.
  - A random walk is rapidly-mixing on a graph if and only if the graph is an expander graph.

# Convergence Property

- Expander graph
  - Every subset of nodes S has a neighborhood (set of vertices accessible via outedges emanating from nodes in S) that is larger than some factor α times of |S|.
  - A graph has a good expansion factor if and only if the largest eigenvalue is sufficiently larger than the second-largest eigenvalue.

**The Web is an expander-like graph**

# Searching with PageRank

- Two search engines:
  - Title-based search engine
  - Full text search engine

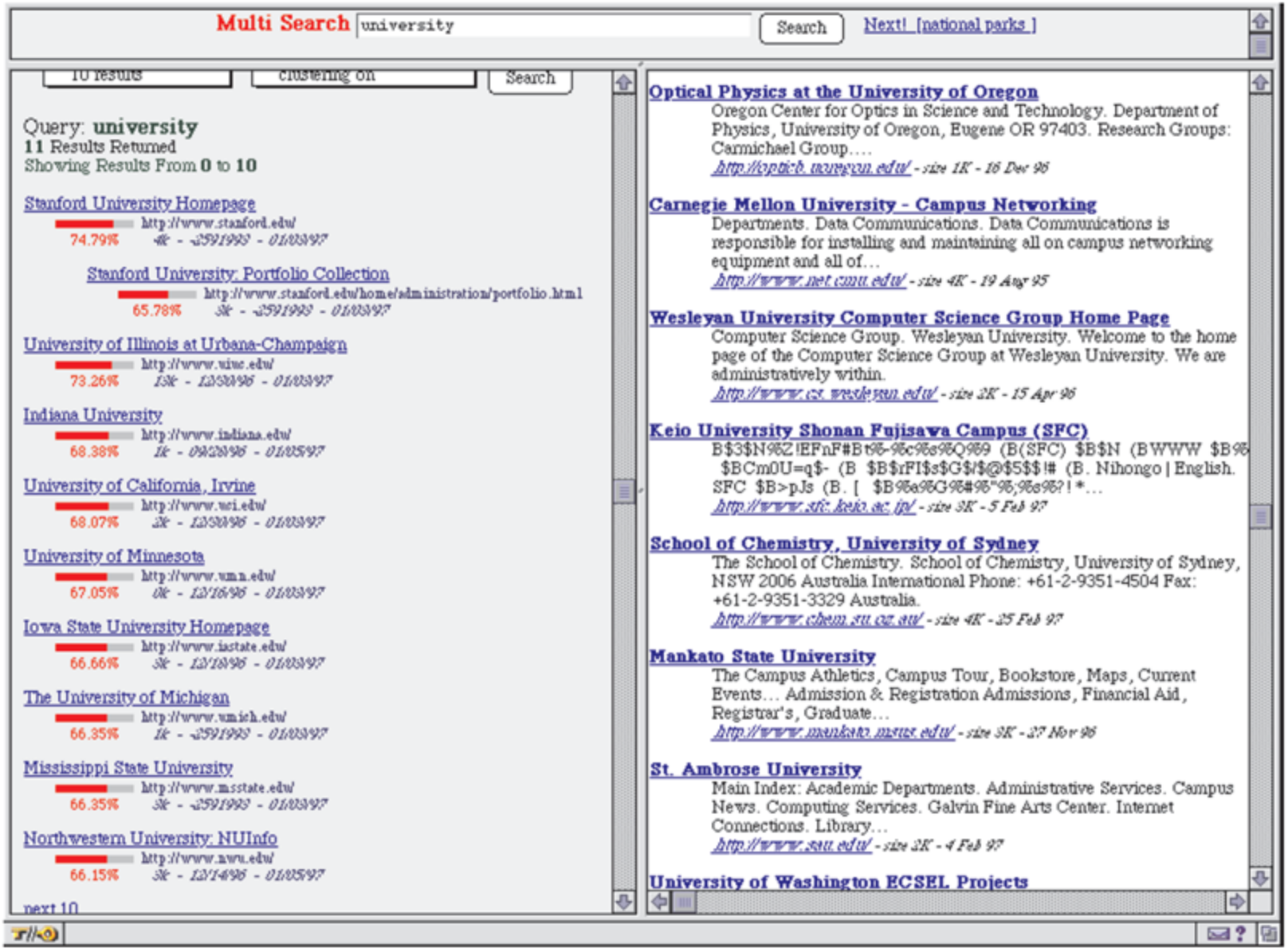
# Searching with PageRank

- Title-based search engine
  - Searches only the “Titles”
  - Finds all the web pages whose titles contain all the query words
  - Sorts the results by PageRank
  - Very simple and cheap to implement
  - Title match ensures high precision, and PageRank ensures high quality

# Searching with PageRank

- Full text search engine
  - Called Google
  - Examines all the words in every stored document and also performs PageRank (Rank Merging)
  - More precise but more complicated

# Searching with PageRank



# Searching with PageRank

| Web Page | PageRank (average is 1.0) |
|---|---|
| Download Netscape Software | 11589.00 |
| http://www.w3.org/ | 10717.70 |
| Welcome to Netscape | 8673.51 |
| Point: It's What You're Searching For | 7930.92 |
| Web-Counter Home Page | 7254.97 |
| The Blue Ribbon Campaign for Online Free Speech | 7010.39 |
| CERN Welcome | 6562.49 |
| Yahoo! | 6561.80 |
| Welcome to Netscape | 6203.47 |
| Wusage 4.1: A Usage Statistics System For Web Servers | 5963.27 |
| The World Wide Web Consortium (W3C) | 5672.21 |
| Lycos, Inc. Home Page | 4683.31 |
| Starting Point | 4501.98 |
| Welcome to Magellan! | 3866.82 |
| Oracle Corporation | 3587.63 |

Top 15 Page Ranks: July 1996

# Personalized PageRank

- Important component of PageRank calculation is *E*
  - A vector over the web pages (used as source of rank)
  - Powerful parameter to adjust the page ranks
- E vector corresponds to the distribution of web pages that a random surfer periodically jumps to

- Instead in Personalized PageRank *E* consists of a single web page

# PageRank vs. Web Traffic

- Some highly accessed web pages have low page rank possibly because
  - People do not want to link to these pages from their own web pages (the example in their paper is pornographic sites…)
  - Some important backlinks are omitted

**use usage data as a start vector for PageRank.**

# The PageRank Proxy

# Conclusion

- PageRank is a global ranking of all web pages based on their locations in the web graph structure
- PageRank uses information which is external to the web pages – backlinks
- Backlinks from important pages are more significant than backlinks from average pages
- The structure of the web graph is very useful for information retrieval tasks.