

# MULTIMEDIA RETRIEVAL



THE UNIVERSITY OF  
SYDNEY

Week03

Semester 1, 2025

# Web Search

- Web Information Retrieval
  - The Web
  - Crawling
  - Ranking: PageRank & HITS

# Web Search – Everyone is doing it!

- First, they do an on-line search



*"First, they do an on-line search."*

© The New Yorker Collection 1998 Amie Levin from cartoonbank.com. All Rights Reserved.

- Police in woods with bloodhounds looking at laptop computer on ground.

# Web Search

- Historically, IR was mainly motivated by text search (libraries, etc.).
- Today: Various other areas and data, e.g. multi media (images, video, etc.), WWW, etc.
- Web search: perfect example for an IR system
  - Goal: Find best possible results (web pages) based on
    - a) Unstructured, heterogeneous, semistructured data
    - b) Imprecise, ambiguous, short queries
  - Note: ‘Best possible results’ is also a very vague specification of the ultimate goal
  - But: Very different from traditional IR tasks!

# Web Search

- Web search is an active research area with high economical impact
- Many open questions & challenges for research
  - Improving existing systems
  - adapting to new scenarios (more data, spam, ...)
  - new challenges (diverse data formats, multimedia, ...),
  - new tasks (desktop search, personalization, ...), etc.
- Many other approaches & techniques exist, e.g.
  - Clustering
  - Specialized search engines, meta search engines, etc.

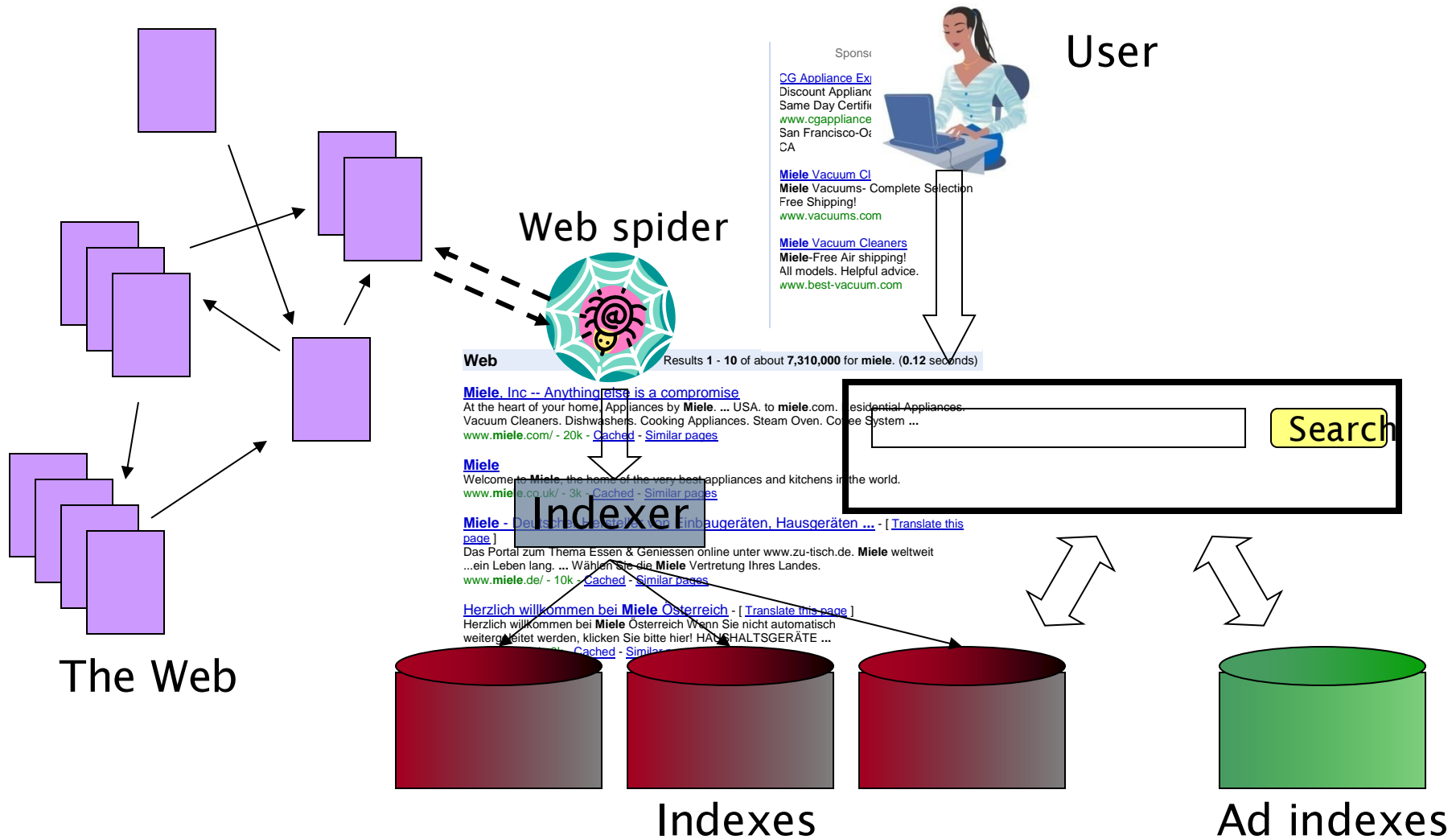
# Brief (non-technical) History

- Early keyword-based engines
  - Altavista, Excite, Infoseek, Inktomi, 1995-1997
  - Extension of traditional IR
- Sponsored search ranking: Goto.com (morphed into Overture.com → Yahoo!)
  - Your search ranking depended on how much you paid
  - Auction for keywords: **casino** was expensive!

# Brief (non-technical) History

- 1998+: Link-based ranking pioneered by Google
  - Blew away all early engines
  - Great user experience in search of a business model
  - Meanwhile Goto/Overture's annual revenues were nearing \$1 billion
- Result: Google added paid-placement “ads” to the side, independent of search results
  - Yahoo followed suit, acquiring Overture (for paid placement) and Inktomi (for search)
- HITS introduced by Kleinberg was used by Teoma

# Web search basics







Web Images Groups News Froogle Local more »

nigritude ultramarine

Search

[Advanced Search](#)  
[Preferences](#)

Web

Results 1 - 10 of about 185,000 for **nigritude ultramarine**. (0.35 seconds)

### [Anil Dash: Nigritude Ultramarine](#)

Do me a favor: Link to this post with the phrase **Nigritude Ultramarine**. ... Just placed a link to your **Nigritude Ultramarine** article on my weblog. Cheers! ...

[www.dashes.com/anil/2004/06/04/nigritude\\_ultra](http://www.dashes.com/anil/2004/06/04/nigritude_ultra) - 101k - Mar 1, 2006 -

[Cached](#) - [Similar pages](#)

### [Nigritude Ultramarine FAQ](#)

**Nigritude Ultramarine** FAQ - frequently asked questions about **nigritude ultramarine** and the realted SEO contest.

[www.nigritudeultramarines.com/](http://www.nigritudeultramarines.com/) - 59k - [Cached](#) - [Similar pages](#)

### [SEO contest - Wikipedia, the free encyclopedia](#)

The **nigritude ultramarine** competition by SearchGuild is widely acclaimed as ...

Comparison of search results for **nigritude ultramarine** during and after the ...

[en.wikipedia.org/wiki/Nigritude\\_ultramarine](http://en.wikipedia.org/wiki/Nigritude_ultramarine) - 37k - [Cached](#) - [Similar pages](#)

### [Slashdot | How To Get Googled, By Hook Or By Crook](#)

The current 3rd result showcases the "**Nigritude Ultramarine** Fighting Force" who ... When discussing **nigritude ultramarine** [slashdot.org] it is important to ...

[slashdot.org/article.pl?sid=04/05/09/1840217](http://slashdot.org/article.pl?sid=04/05/09/1840217) - 110k - [Cached](#) - [Similar pages](#)

### [The Nigritude Ultramarine Search Engine Optimization Contest](#)

It's sweeping the web -- or at least search engine optimizers -- a new contest to rank tops for the term **nigritude ultramarine** on Google.

[searchenginewatch.com/sereport/article.php/3360231](http://searchenginewatch.com/sereport/article.php/3360231) - 57k - [Cached](#) - [Similar pages](#)

Sponsored Links

#### [Business Blogging Seminar](#)

Coming to L.A. March 16

Top bloggers reveal key techniques

[www.blogbusinesssummit.com](http://www.blogbusinesssummit.com)

Los Angeles, CA

#### [Full-Time SEO & SEM Jobs](#)

Find companies big & small hiring full-time SEO & SEM pros right now

[CareerBuilder.com](http://CareerBuilder.com)

#### [SEO Contests](#)

Information on SEO Contests like the **Nigritude Ultramarine** contest.

[www.seo-contests.com/](http://www.seo-contests.com/)

#### [The SEO Book](#)

**Nigritude Ultramarine** & SEO secrets

Fun, free, raw, & different.

[www.seobook.com](http://www.seobook.com)

Ads

Algorithmic results.

Overstock.com

# Characteristics of the Web

- Size: The web is huge! And there are lots of users!
- Documents
  - Extreme variety regarding formats, structure, quality, content (duplicate/near-duplicate) etc.
- Users: Very different skills & intentions, e.g.
  - *Find all information about related patents*
  - *Find some good tourist inform. about Paris*
  - *Find the phone no. of the tourist office*
- Space: The web is a distributed system
- Spam: Expect manipulation instead of cooperation from the document providers
- Dynamic: The web keeps growing & changing

# User Needs

## □ Need

- **Informational** – want **to learn** about something (~40% / 65%)

Low hemoglobin

- **Navigational** – want **to go** to that page (~25% / 15%)

United Airlines

- **Transactional** – want **to do something** (web-mediated) (~35% / 20%)

- Access a service

Seattle weather

- Downloads

Mars surface images

- Shop

Canon S410

- **Gray areas**

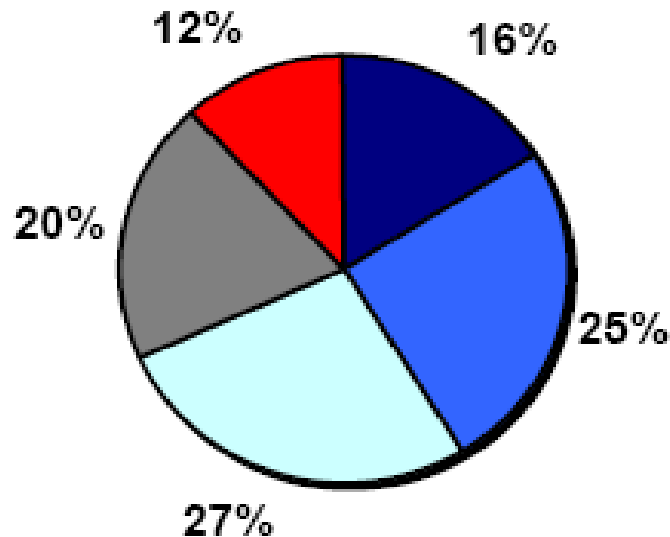
Car rental Brasil

- Find a good hub

- Exploratory search “see what’s there”

# How far do people look for results?

“When you perform a search on a search engine and don't find what you are looking for, at what point do you typically either revise your search, or move on to another search engine? (Select one)”



- After reviewing the first few entries
- After reviewing the first page
- After reviewing the first 2 pages
- After reviewing the first 3 pages
- After reviewing more than 3 pages

(Source: [iprospect.com](http://iprospect.com) WhitePaper\_2006\_SearchEngineUserBehavior.pdf)

# Users' empirical evaluation of results

- Quality of pages varies widely
  - Relevance is not enough
  - Other desirable qualities (non IR!!)
    - Content: Trustworthy, diverse, non-duplicated, well maintained
    - Web readability: display correctly & fast
    - No annoyances: pop-ups, etc
- Precision vs. recall
  - On the web, recall seldom matters
- What matters
  - Precision at 1? Precision above the fold?
  - Comprehensiveness – must be able to deal with obscure queries
    - Recall matters when the number of matches is very small
- User perceptions may be unscientific, but are significant over a large aggregate

# Search engine optimization (Spam)

## □ Motives

- Commercial, political, religious, lobbies
- Promotion funded by advertising budget

## □ Operators

- Contractors (Search Engine Optimizers) for lobbies, companies
- Web masters
- Hosting services

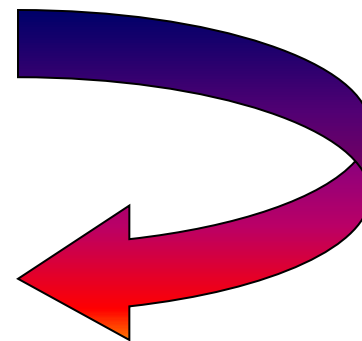
## □ Forums

- E.g., Web master world ( [www.webmasterworld.com](http://www.webmasterworld.com) )
  - Search engine specific tricks
  - Discussions about academic papers ☺

# Simplest forms

- First generation engines relied heavily on *tf/idf*
  - The top-ranked pages for the query **maui resort** were the ones containing the most **maui**'s and **resort**'s
- SEOs responded with dense repetitions of chosen terms
  - e.g., **maui resort maui resort maui resort**
  - Often, the repetitions would be in the same color as the background of the web page
    - Repeated terms got indexed by crawlers
    - But not visible to humans on browsers

Pure word density cannot  
be trusted as an IR signal



# Variants of keyword stuffing

- Misleading meta-tags, excessive repetition
- Hidden text with colors, style sheet tricks, etc.

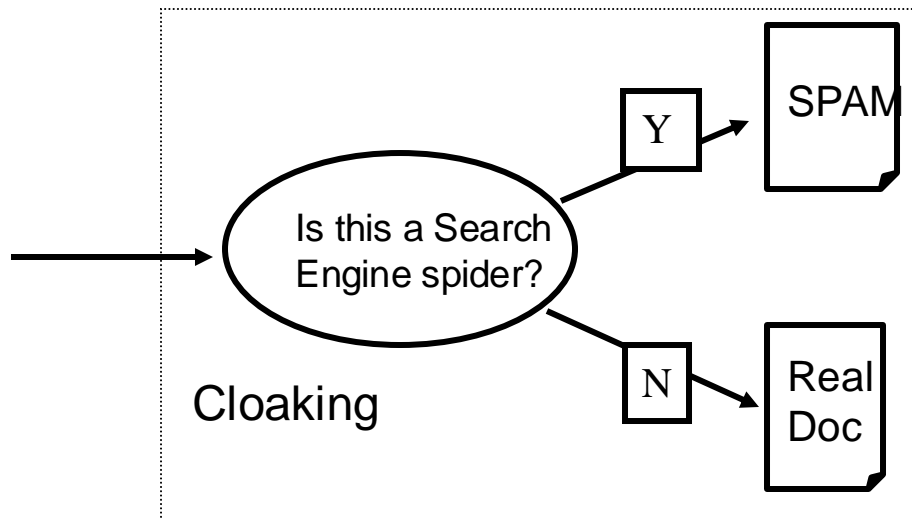
## Meta-Tags =

"... London hotels, hotel, holiday inn, hilton, discount, booking, reservation, sex, mp3, britney spears, viagra, ..."



# Cloaking

- Serve fake content to search engine spider
- DNS cloaking: Switch IP address. Impersonate



# More spam techniques

## □ Doorway pages

- Pages optimized for a single keyword that re-direct to the real target page

## □ Link spamming

- Mutual admiration societies, hidden links, awards – more on these later
- *Domain flooding*: numerous domains that point or re-direct to a target page

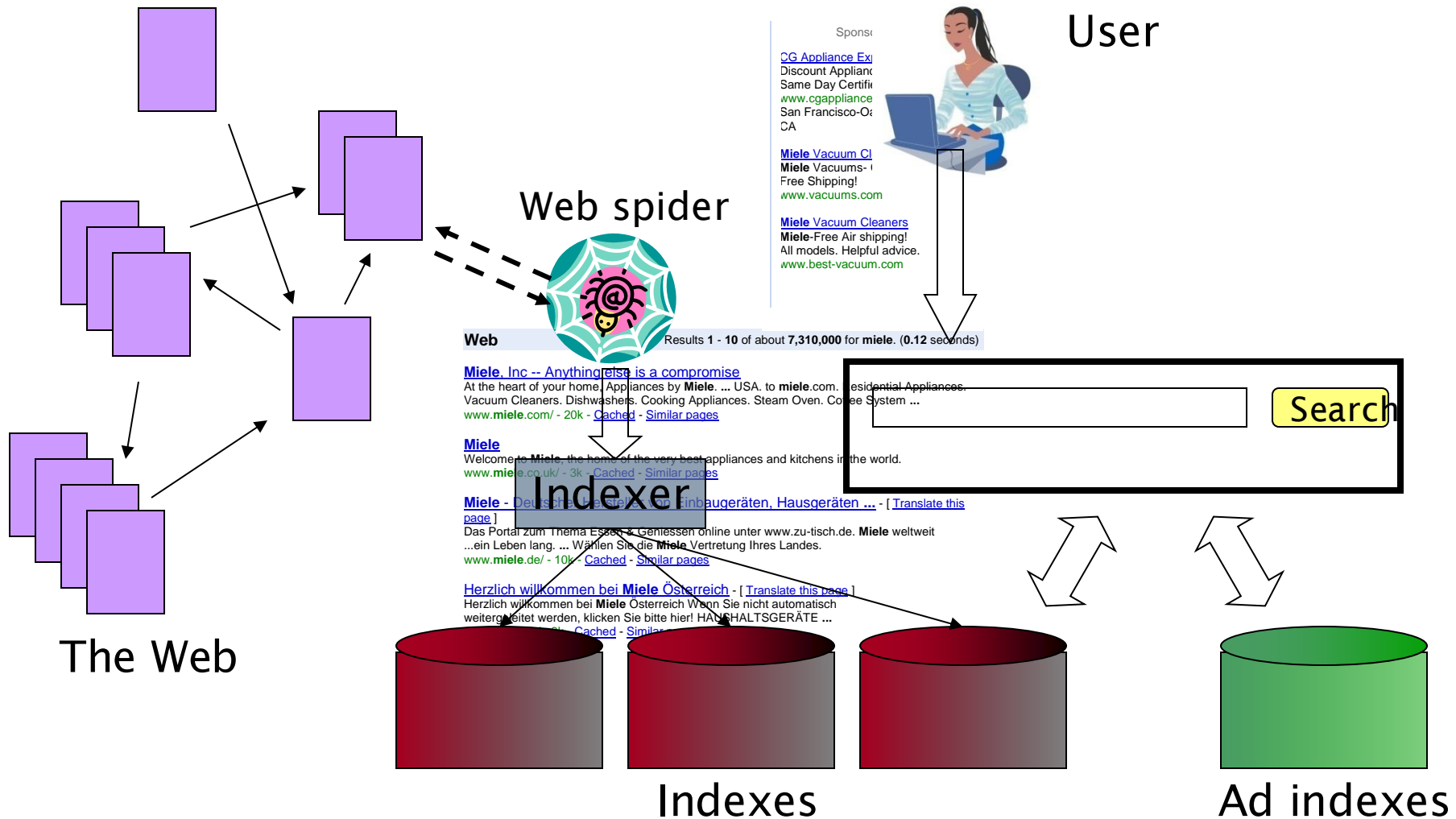
## □ Robots

- Fake query stream – rank checking programs
  - “Curve-fit” ranking programs of search engines
- Millions of submissions via Add-Url

# The war against spam

- Quality signals - Prefer authoritative pages based on:
  - Votes from authors (linkage signals)
  - Votes from users (usage signals)
- Policing of URL submissions
  - Anti robot test
- Limits on meta-keywords
- Robust link analysis
  - Ignore statistically implausible linkage (or text)
  - Use link analysis to detect spammers (guilt by association)
- Spam recognition by machine learning
  - Training set based on known spam
- Family friendly filters
  - Linguistic analysis, general classification techniques, etc.
  - For images: flesh tone detectors, source text analysis, etc.
- Editorial intervention
  - Blacklists
  - Top queries audited
  - Complaints addressed
  - Suspect pattern detection

# Web search basics



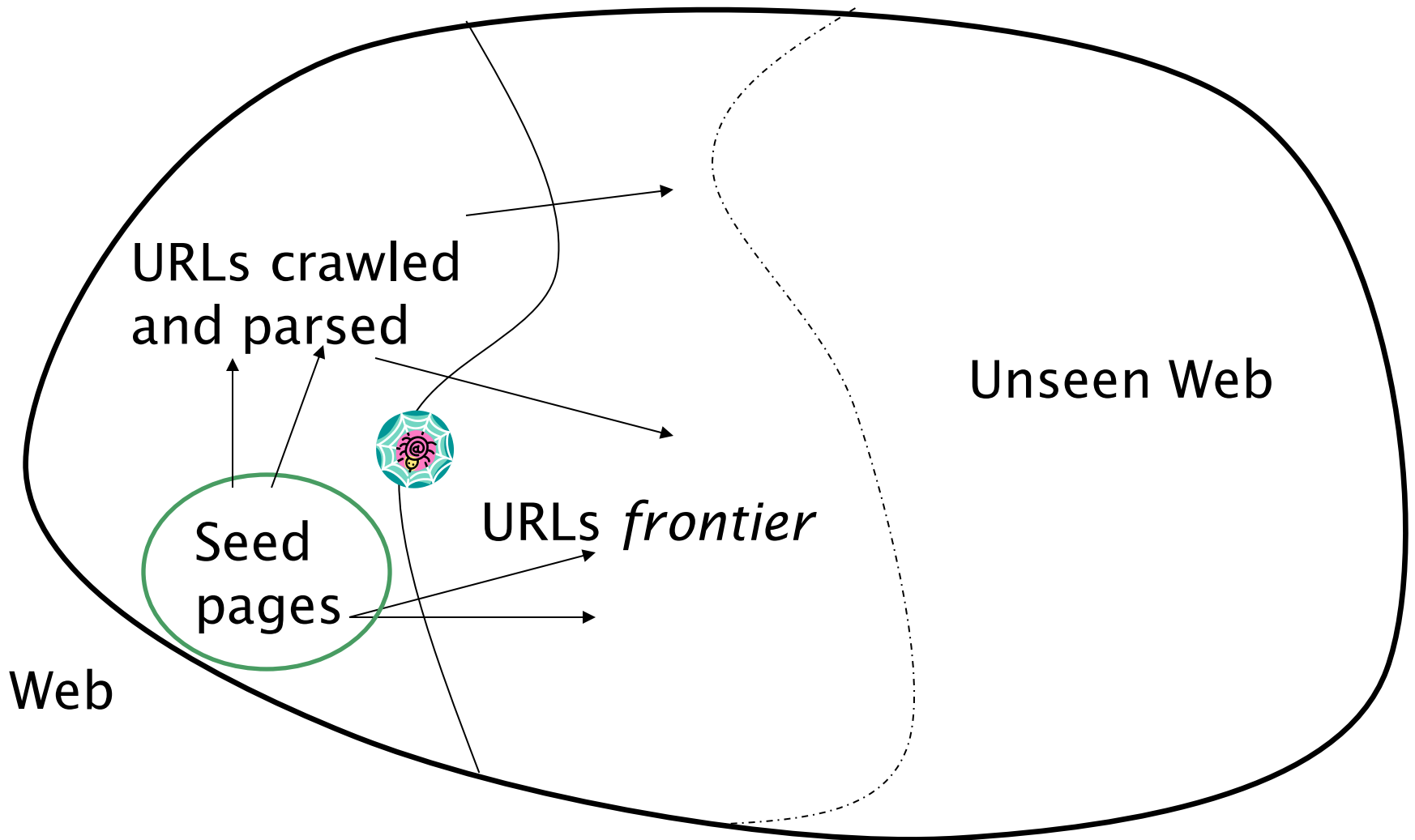
# Basic crawler operation

- Begin with known “seed” pages
- Fetch and parse them
  - ▣ Extract URLs they point to
  - ▣ Place the extracted URLs on a queue
- Fetch each URL on the queue and repeat

Web spider



# Crawling picture



# Simple picture – complications

- Web crawling isn't feasible with one machine
  - All of the above steps distributed
- Even non-malicious pages pose challenges
  - Latency/bandwidth to remote servers vary
  - Webmasters' stipulations
    - How “deep” should you crawl a site's URL hierarchy?
  - Site mirrors and duplicate pages
- Malicious pages
  - Spam pages
  - Spider traps – incl dynamically generated
- Politeness – don't hit a server too often

# What any crawler *must* do

- Be Polite: Respect implicit and explicit politeness considerations
  - ▣ Only crawl allowed pages
  - ▣ Respect *robots.txt* (more on this shortly)
- Be Robust: Be immune to spider traps and other malicious behavior from web servers



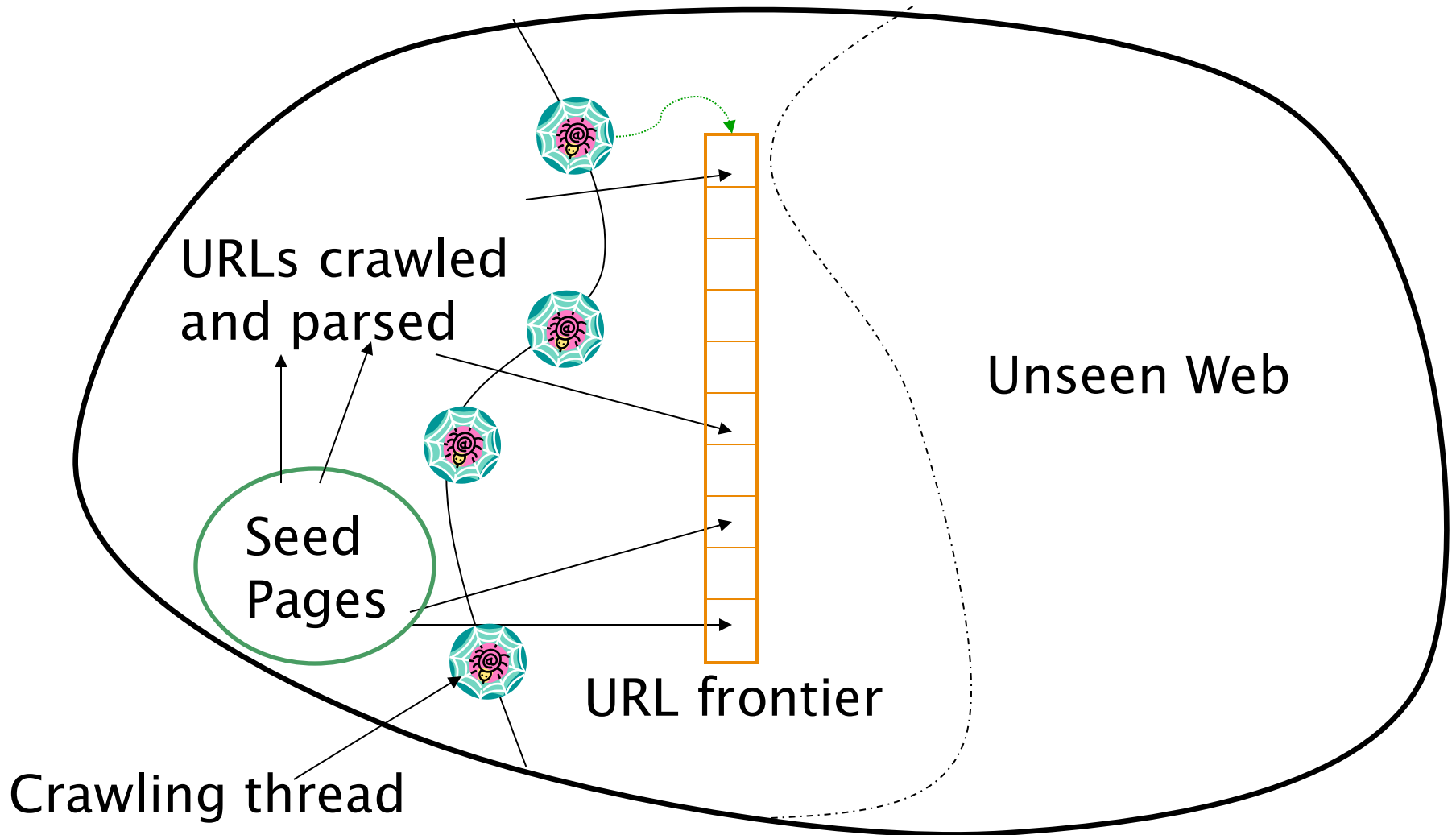
# What any crawler *should* do

- Be capable of distributed operation: designed to run on multiple distributed machines
- Be scalable: designed to increase the crawl rate by adding more machines
- Performance/efficiency: permit full use of available processing and network resources

# What any crawler *should* do

- Fetch pages of “higher quality” first
- Continuous operation: Continue fetching fresh copies of a previously fetched page
- Extensible: Adapt to new data formats, protocols

# Updated crawling picture



# URL frontier: two main considerations

- Politeness: do not hit a web server too frequently
- Freshness: crawl some pages more often than others
  - E.g., pages (such as News sites) whose content changes often

These goals may conflict each other.

(E.g., simple priority queue fails – many links out of a page go to its own site, creating a burst of accesses to that site.)

# URL frontier

- Can include multiple pages from the same host
- Must avoid trying to fetch them all at the same time
- Must try to keep all crawling threads busy

# Explicit and implicit politeness

- Explicit politeness: specifications from webmasters on what portions of site can be crawled
  - robots.txt
- Implicit politeness: even with no specification, avoid hitting any site too often

# Robots.txt

- Protocol for giving spiders (“robots”) limited access to a website, originally from 1994
  - [www.robotstxt.org/wc/norobots.html](http://www.robotstxt.org/wc/norobots.html)
- Website announces its request on what can(not) be crawled
  - For a URL, create a file `URL/robots.txt`
  - This file specifies access restrictions

# Robots.txt example

- No robot should visit any URL starting with `"/yoursite/temp/"`, except the robot called `"searchengine"`:

```
User-agent: *
```

```
Disallow: /yoursite/temp/
```

```
User-agent: searchengine
```

```
Disallow:
```



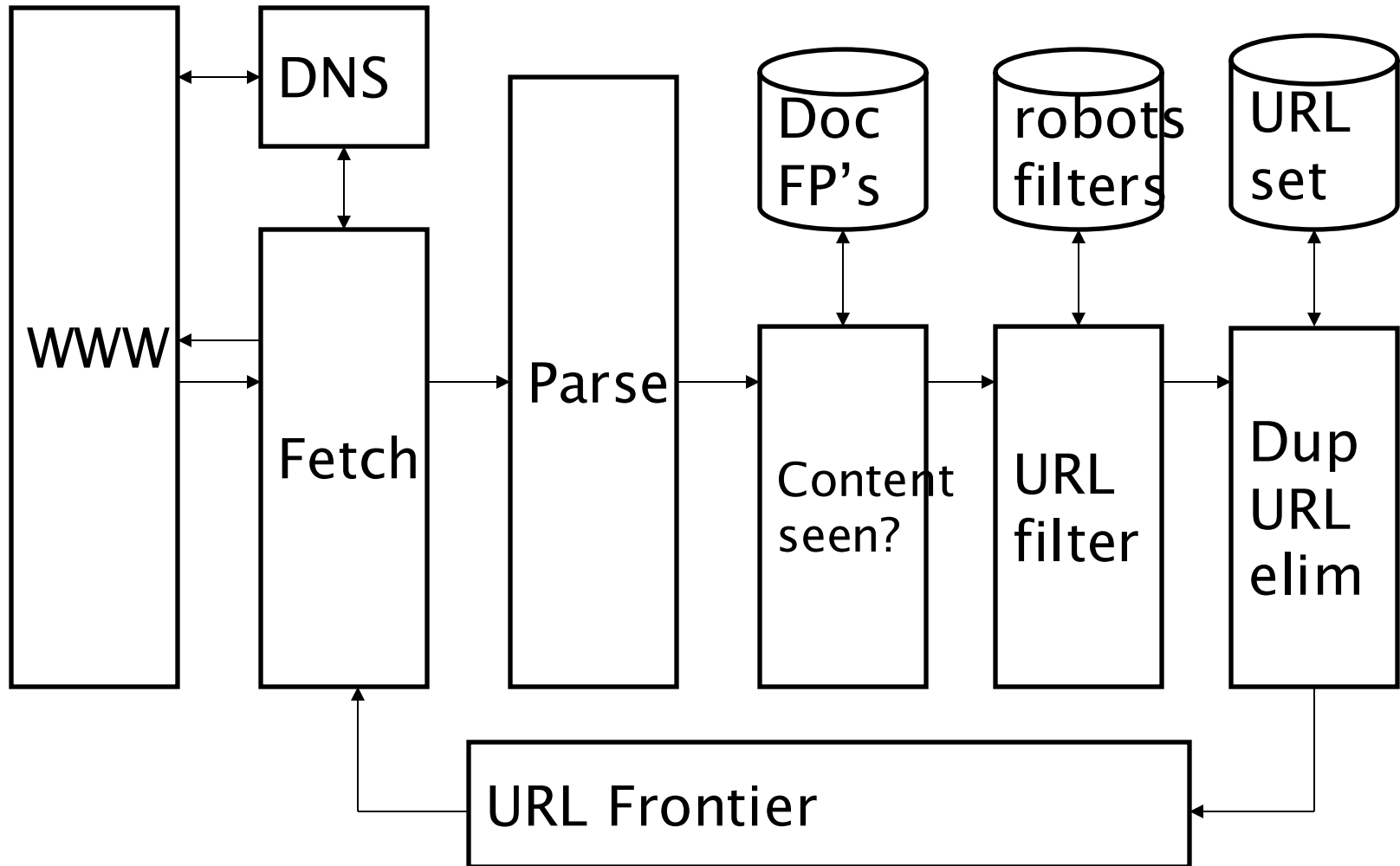
# Processing steps in crawling

- Pick a URL from the frontier
- Fetch the document at the URL
- Parse the URL
  - Extract links from it to other docs (URLs)
- Check if URL has content already seen
  - If not, add to indexes
- For each extracted URL
  - Ensure it passes certain URL filter tests
  - Check if it is already in the frontier (duplicate URL elimination)

Which one?

E.g., only crawl .edu,  
obey robots.txt, etc.

# Basic crawl architecture



# DNS (Domain Name Server)

- A lookup service on the internet
  - ▣ Given a URL, retrieve its IP address
  - ▣ Service provided by a distributed set of servers – thus, lookup latencies can be high (even seconds)
- Common OS implementations of DNS lookup are *blocking*: only one outstanding request at a time
- Solutions
  - ▣ DNS caching
  - ▣ Batch DNS resolver – collects requests and sends them out together

# Parsing: URL normalization

- When a fetched document is parsed, some of the extracted links are *relative* URLs
- E.g., at [http://en.wikipedia.org/wiki/Main\\_Page](http://en.wikipedia.org/wiki/Main_Page) we have a relative link to `/wiki/Wikipedia:General_disclaimer` which is the same as the absolute URL [http://en.wikipedia.org/wiki/Wikipedia:General\\_disclaimer](http://en.wikipedia.org/wiki/Wikipedia:General_disclaimer)
- During parsing, must normalize (expand) such relative URLs

# Content seen?

- Duplication is widespread on the web
- If the page just fetched is already in the index, do not further process it
- This is verified using document fingerprints or shingles

# Filters and robots.txt

- Filters – regular expressions for URL's to be crawled/not
- Once a robots.txt file is fetched from a site, need not fetch it repeatedly
  - ▣ Doing so burns bandwidth, hits web server
- Cache robots.txt files

# Duplicate URL elimination

- For a non-continuous (one-shot) crawl, test to see if an extracted+filtered URL has already been passed to the frontier
- For a continuous crawl – see details of frontier implementation

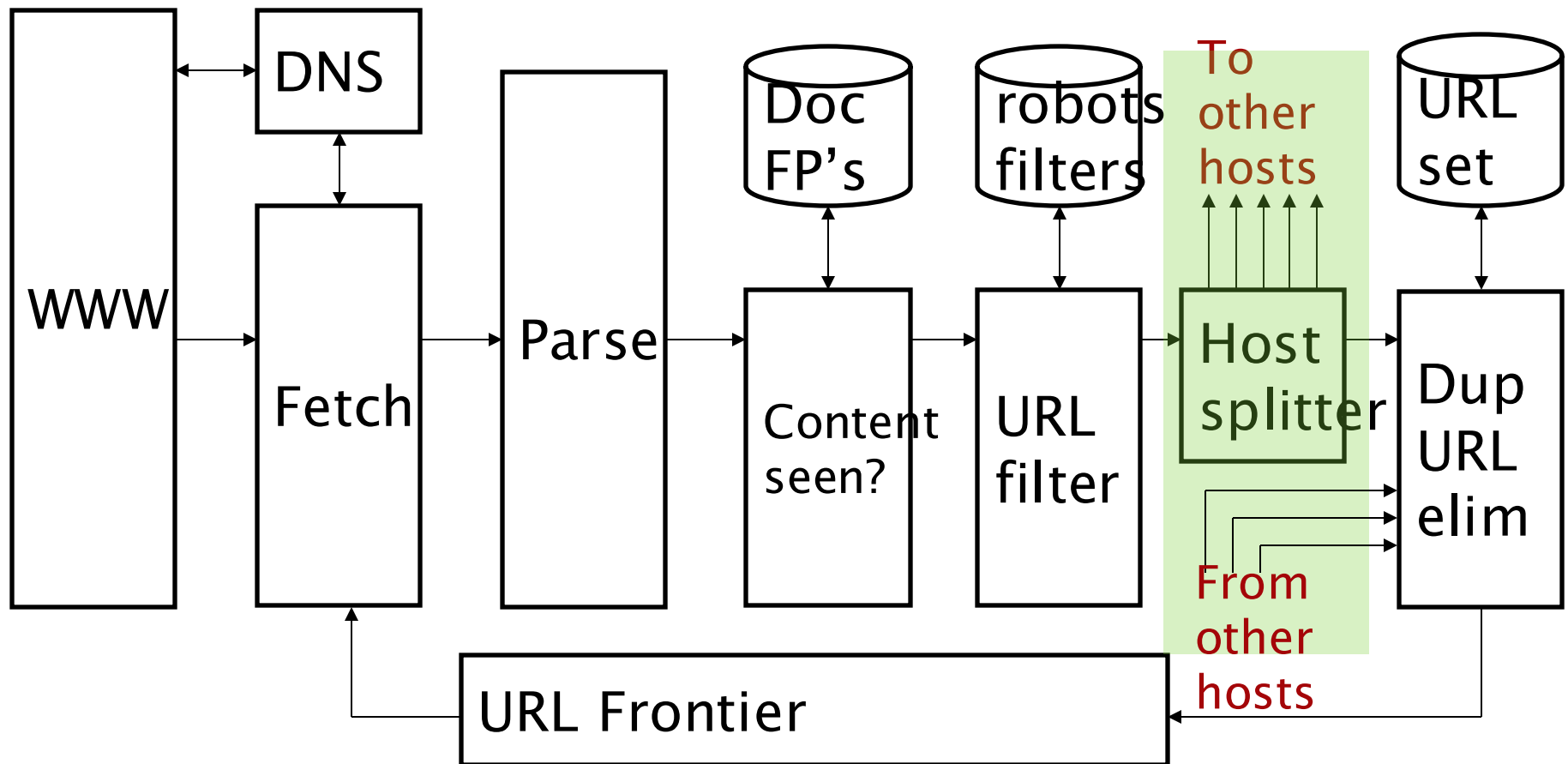
# Distributing the crawler

- Run multiple crawl threads, under different processes – potentially at different nodes
  - ▣ Geographically distributed nodes
- Partition hosts being crawled into nodes
  - ▣ Hash used for partition
- How do these nodes communicate?



# Communication between nodes

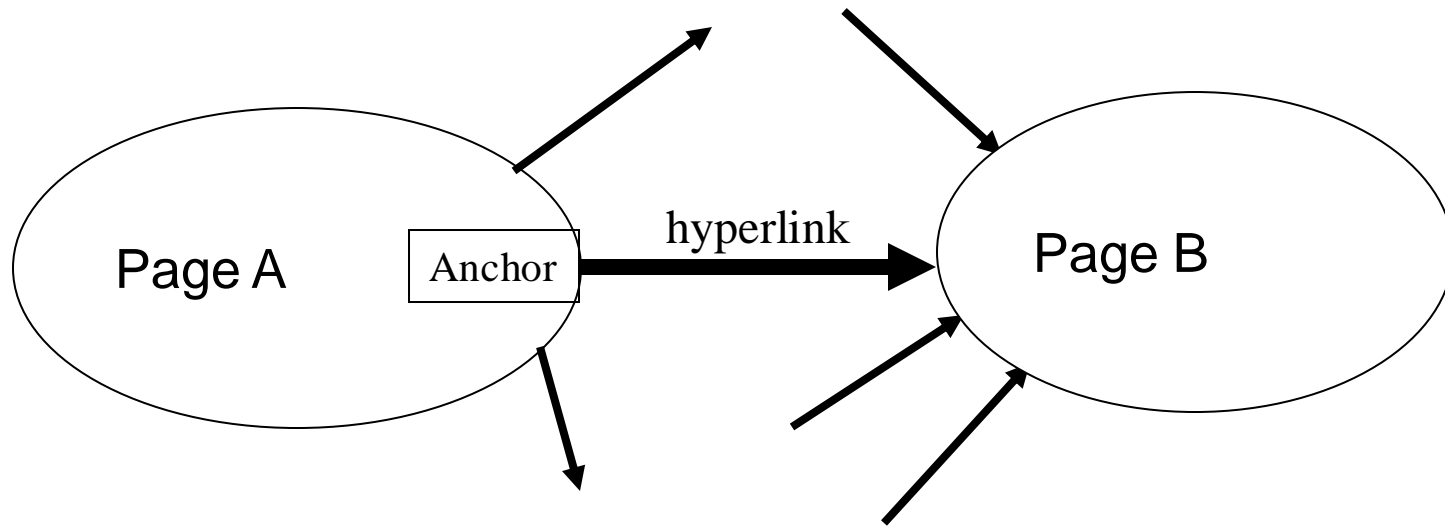
- The output of the URL filter at each node is sent to the Duplicate URL Eliminator at all nodes



# Ranking

- Beyond traditional IR ranking
  - ▣ Purely focus on content
- Must incorporating the following formation as well
  - ▣ Anchor text (peer review)
  - ▣ Link analysis (social relationship)
    - PageRank and variants
    - HITS

# The Web as a Directed Graph

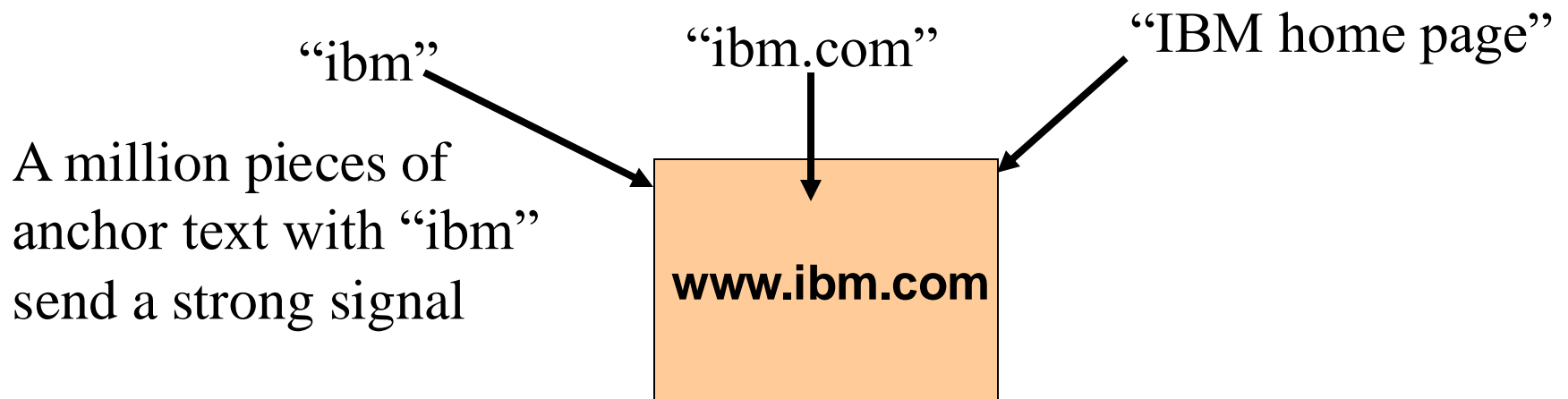


**Assumption 1:** A hyperlink between pages denotes author perceived relevance (quality signal)

**Assumption 2:** The anchor of the hyperlink describes the target page (textual context)

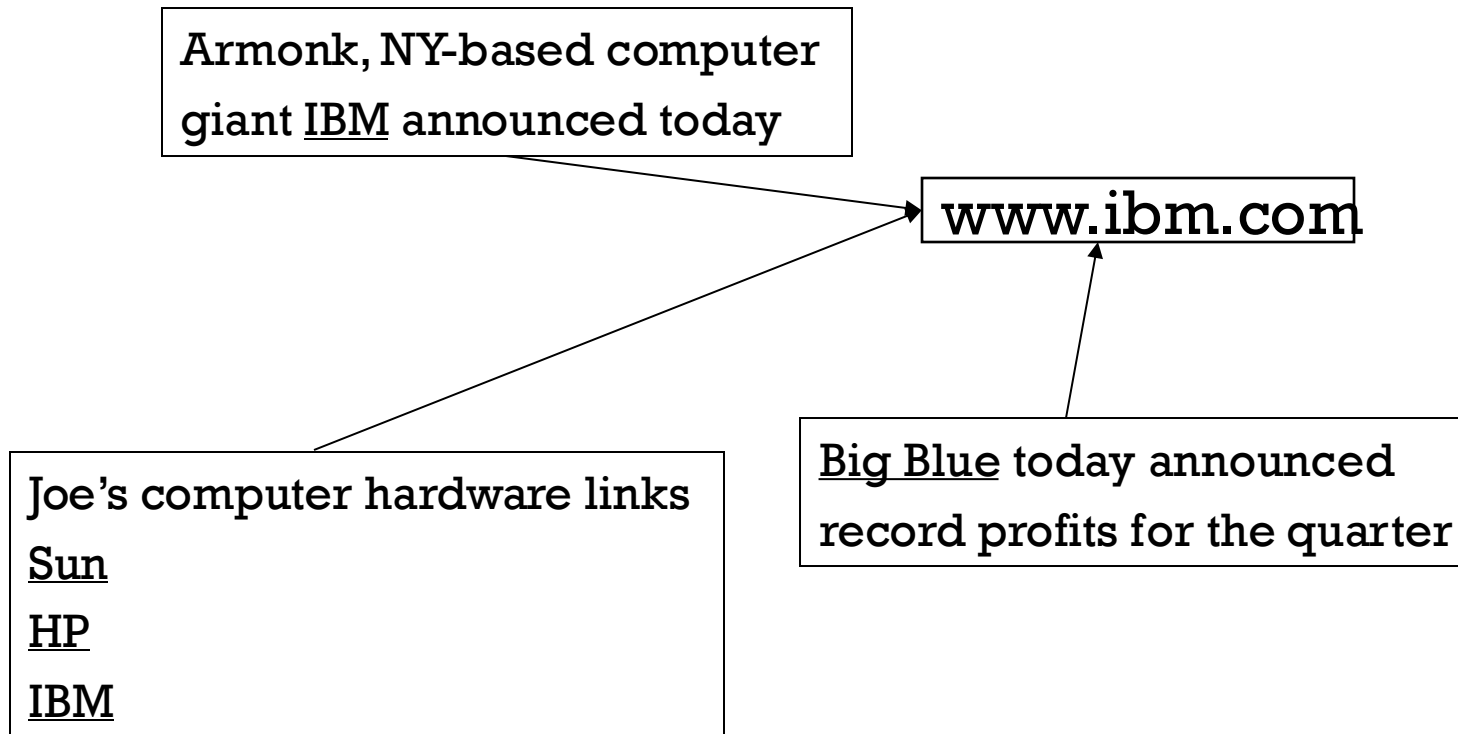
# Anchor Text

- For **ibm** how to distinguish between:
  - ▣ IBM's home page (mostly graphical)
  - ▣ IBM's copyright page (high term freq. for 'ibm')
  - ▣ Rival's spam page (arbitrarily high term freq.)



# Indexing anchor text

- When indexing a document  $D$ , include anchor text from links pointing to  $D$ .



# Indexing anchor text

- Can sometimes have unexpected side effects - e.g., **evil empire**.
  - ▣ Google Bomb
- Can score anchor text with weight depending on the authority of the anchor page's website
  - ▣ E.g., if we were to assume that content from cnn.com or yahoo.com is authoritative, then trust their anchor text



## 众口铄金

To describe the force of public opinion, and even be able to melt metal. Unanimously to confuse right and wrong analogy.

<http://www.searchenginedictionary.com/terms-google-bomb.shtml>

# Citation Analysis

- Citation frequency
- Co-citation coupling frequency
  - Cocitations with a given author measures “impact”
  - Cocitation analysis
- Bibliographic coupling frequency
  - Articles that co-cite the same articles are related
- Citation indexing
  - Who is author cited by? (Garfield 1972)
- PageRank preview: Pinski and Narin '60s
  - which journals are authoritative?

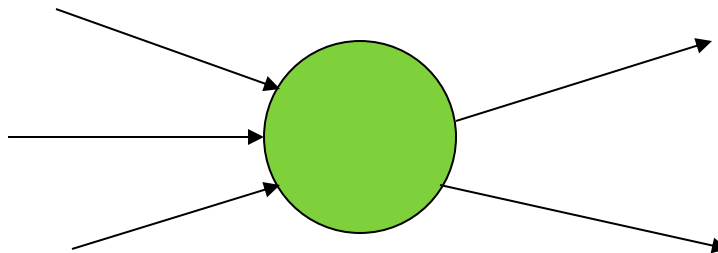
# Query processing

- First retrieve all pages meeting the text query (say ***venture capital***).
- Order these by their link popularity (either variant on the previous page).



# Query-independent ordering

- First generation: using link counts as simple measures of popularity.
- Two basic suggestions:
  - Undirected popularity:
    - Each page gets a score = the number of in-links plus the number of out-links ( $3+2=5$ ).
  - Directed popularity:
    - Score of a page = number of its in-links (3).



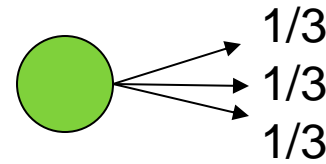
# Spamming simple popularity

- *Exercise:* How do you spam each of the following heuristics so your page gets a high score?
- Each page gets a static score = the number of in-links plus the number of out-links.
- Static score of a page = number of its in-links.

# PageRank Scoring

- Imagine a browser doing a random walk on web pages:

- Start at a random page



- At each step, go out of the current page along one of the links on that page, equiprobably

- “In the steady state” each page has a long-term visit rate - use this as the page’s score.

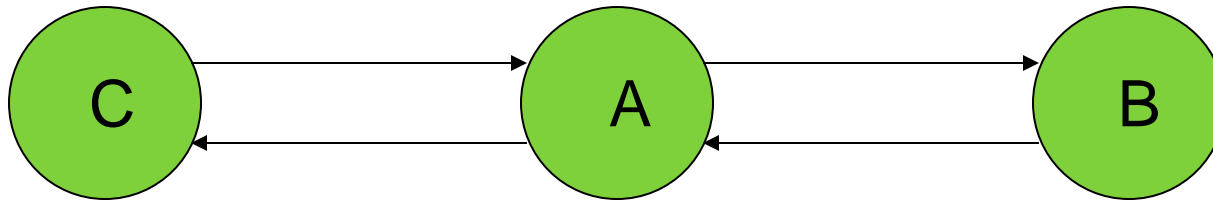
# Mathematically ... ..

- Assume that there are  $n$  web pages and the  $i$ -th page is simply noted with  $i$ .
- The web pages form a directed graph and the graph can be represented with adjacent matrix  $P$ 
  - Element  $p_{i,j}$  represents the connection between the  $i$ -th web page and the  $j$ -th web page
    - 1: if page  $i$  refers to page  $j$ ; 0, otherwise.
- PageRank of the  $i$ -th web page is denoted with  $a_i$

$$a_j = \sum_{i=1}^n \frac{a_i}{OutLinks(i)} \times p_{i,j} = \sum_{i=1}^n a_i \times \frac{p_{i,j}}{OutLinks(i)}$$

$$\sum_{i=1}^n a_i = 1.$$

# A sample



$$\begin{bmatrix} 0 & 1 & 1 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$$p_{i,j}$$

$$\begin{bmatrix} 0 & 0.5 & 0.5 \\ 1 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$$\frac{p_{i,j}}{OutLinks(i)}$$

$$\sum_{j=1}^n p_{ij} = 1.$$

# More Mathematically ... ..

$$a_j = \sum_{i=1}^n a_i \times p_{i,j}$$

$$a = aP$$

$$a(t) = a(t-1)P$$

# How do we compute this vector?

- Let  $\mathbf{a} = [a_1, \dots, a_n]$  denote the row vector of steady-state probabilities.
- If we our current position is described by  $\mathbf{a}$ , then the next step is distributed as  $\mathbf{aP}$ .
- But  $\mathbf{a}$  is the steady state, so  $\mathbf{a}=\mathbf{aP}$ .
- Solving this matrix equation gives us  $\mathbf{a}$ .
  - So  $\mathbf{a}$  is the (left) eigenvector for  $\mathbf{P}$ .
  - (Corresponds to the “principal” eigenvector of  $\mathbf{P}$  with the largest eigenvalue.)
  - Transition probability matrices always have larges eigenvalue 1.

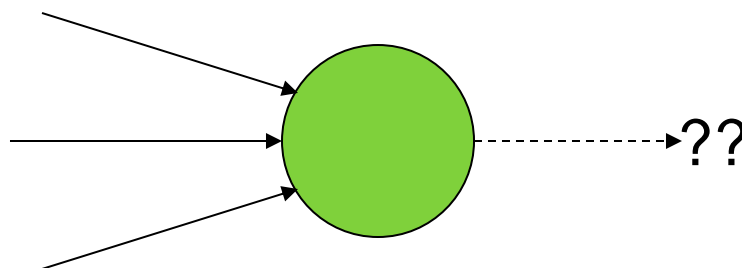
# One way of computing $\mathbf{a}$

- Recall, regardless of where we start, we eventually reach the steady state  $\mathbf{a}$ .
- Start with any distribution (say  $\mathbf{x}=(10\dots0)$ ).
- After one step, we're at  $\mathbf{xP}$ ;
- after two steps at  $\mathbf{xP}^2$ , then  $\mathbf{xP}^3$  and so on.
- “Eventually” means for “large”  $k$ ,  $\mathbf{xP}^k = \mathbf{a}$ .
- Algorithm: multiply  $\mathbf{x}$  by increasing powers of  $\mathbf{P}$  until the product looks stable.



# Not quite enough

- The web is full of dead-ends.
  - ▣ Random walk can get stuck in dead-ends.
    - No through road
  - ▣ Makes no sense to talk about long-term visit rates.



# Random Surfer / Random Walker

- At a dead end, jump to a random web page.
- At any non-dead end, with probability  $(1-\gamma)$ , jump to a random web page.
  - With 90% (*i.e.*  $\gamma=90\%$ ) to follow  $P$  if there are outbound links.
  - Different names for this parameter, such as damping factor.
  - Teleporter probability is  $(1-\gamma)$ .

# Result of Random Surfer / Walker

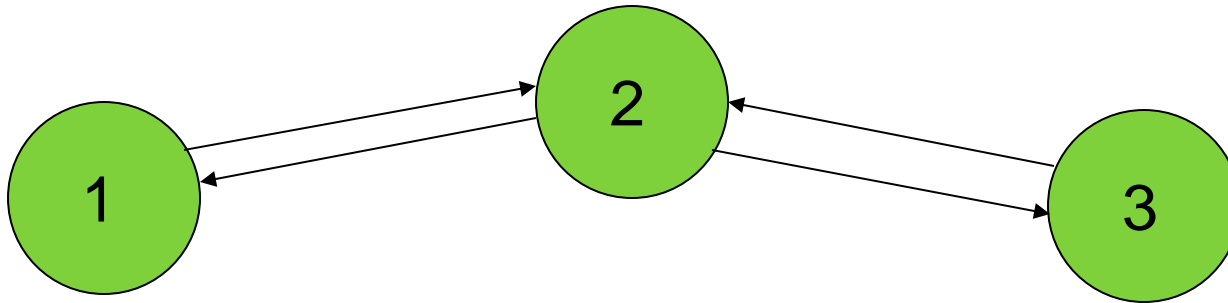
- Now cannot get stuck locally.
- There is a long-term rate at which any page is visited (not obvious, will show this).
- How do we compute this visit rate?
  - How to revise matrix  $P$ ?

# Revise P

$$a_j = \gamma \times \sum_{i=1}^n a_i \times p_{i,j} + (1 - \gamma) / n$$

- In each row
  - Replace the zero elements with  $(1-\gamma)/n$ ;
  - Replace the non-zero elements with  $p_{ij} \times \gamma + (1-\gamma)/n$

# Case study



$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

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

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

$$\gamma = 1/2$$

# Case Study

$$P = \begin{pmatrix} 1/6 & 2/3 & 1/6 \\ 5/12 & 1/6 & 5/12 \\ 1/6 & 2/3 & 1/6 \end{pmatrix}$$

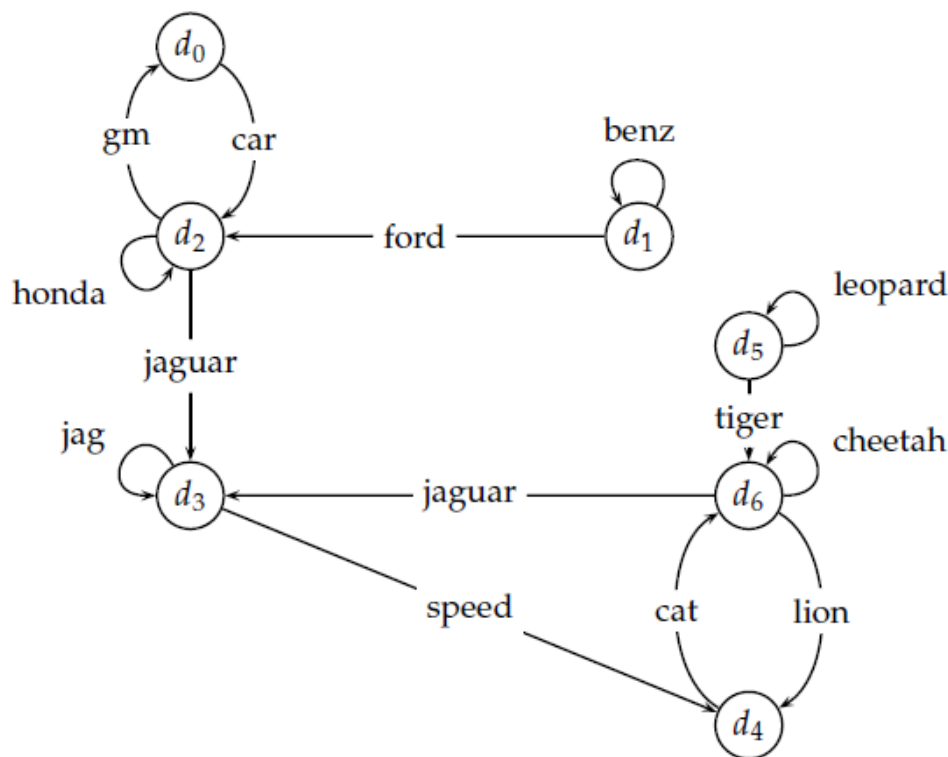
$$\vec{x}_0 P = (1/6 \quad 2/3 \quad 1/6) = \vec{x}_1$$

$$\vec{x}_1 P = (1/6 \quad 2/3 \quad 1/6) \begin{pmatrix} 1/6 & 2/3 & 1/6 \\ 5/12 & 1/6 & 5/12 \\ 1/6 & 2/3 & 1/6 \end{pmatrix} = (1/3 \quad 1/3 \quad 1/3) = \vec{x}_2$$

$\vec{x}_0$	1	0	0
$\vec{x}_1$	1/6	2/3	1/6
$\vec{x}_2$	1/3	1/3	1/3
$\vec{x}_3$	1/4	1/2	1/4
$\vec{x}_4$	7/24	5/12	7/24
...	...	...	...
$\vec{x}$	5/18	4/9	5/18

# How about this one?

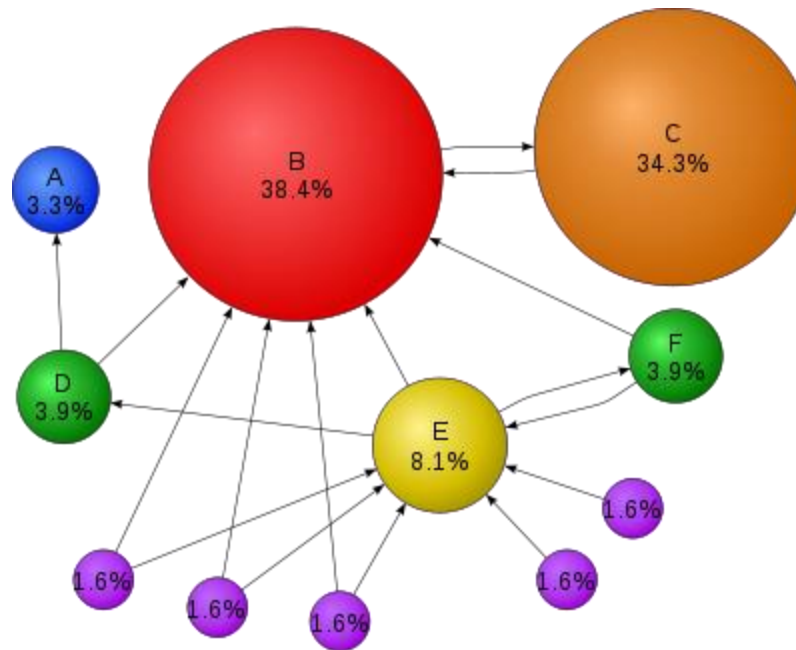
□  $\gamma=0.86$



0.02	0.02	0.88	0.02	0.02	0.02	0.02
0.02	0.45	0.45	0.02	0.02	0.02	0.02
0.31	0.02	0.31	0.31	0.02	0.02	0.02
0.02	0.02	0.02	0.45	0.45	0.02	0.02
0.02	0.02	0.02	0.02	0.02	0.02	0.88
0.02	0.02	0.02	0.02	0.02	0.45	0.45
0.02	0.02	0.02	0.31	0.31	0.02	0.31

$\vec{x} = (0.05 \quad 0.04 \quad 0.11 \quad 0.25 \quad 0.21 \quad 0.04 \quad 0.31)$

# Are the figures correct ?

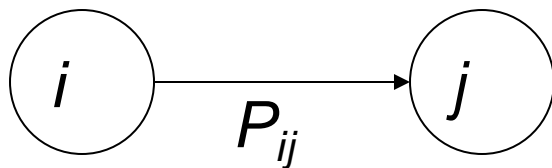


<http://en.wikipedia.org/wiki/PageRank>



# Markov chains (too mathematical)

- A Markov chain consists of  $n$  states, plus an  $n \times n$  transition probability matrix  $\mathbf{P}$ .
- At each step, we are in exactly one of the states.
- For  $1 \leq i, j \leq n$ , the matrix element  $p_{ij}$  tells us the probability of  $j$  being the next state, given we are currently in state  $i$ .



$$\sum_{j=1}^n p_{ij} = 1, p_{ij} > 0.$$

# PageRank summary

- Preprocessing:
  - ▣ Given graph of links, build matrix  $\mathbf{P}$ .
  - ▣ From it compute  $\mathbf{a}$ .
  - ▣ The entry  $a_i$  is a number between 0 and 1: the PageRank of page  $i$ .
- Query processing:
  - ▣ Retrieve pages meeting query.
  - ▣ Rank them by their PageRank.
  - ▣ Order is query-*independent*.
- The reality
  - ▣ PageRank is used in Google, but so are many other clever heuristics.

# PageRank: Issues and Variants

- How realistic is the random surfer model?
  - What if we modeled the back button?
  - Surfer behavior sharply skewed towards short paths
  - Search engines, bookmarks & directories make jumps non-random.
- Biased Surfer Models
  - Topic sensitive (Personalized) PageRank
  - Weight edge traversal probabilities based on match with topic/query (non-uniform edge selection)
  - Bias jumps to pages on topic (e.g., based on personal bookmarks & categories of interest)

# Influencing PageRank: *Personalization*

- Input:
  - Web graph  $P$
  - influence vector  $\mathbf{v}$   
 $\mathbf{v} : (\text{page} \rightarrow \text{degree of influence})$
- Output:
  - Rank vector  $\mathbf{a}$ : (page  $\rightarrow$  page importance wrt  $\mathbf{v}$ )
- $\mathbf{a} = \text{PR}(W, \mathbf{v})$

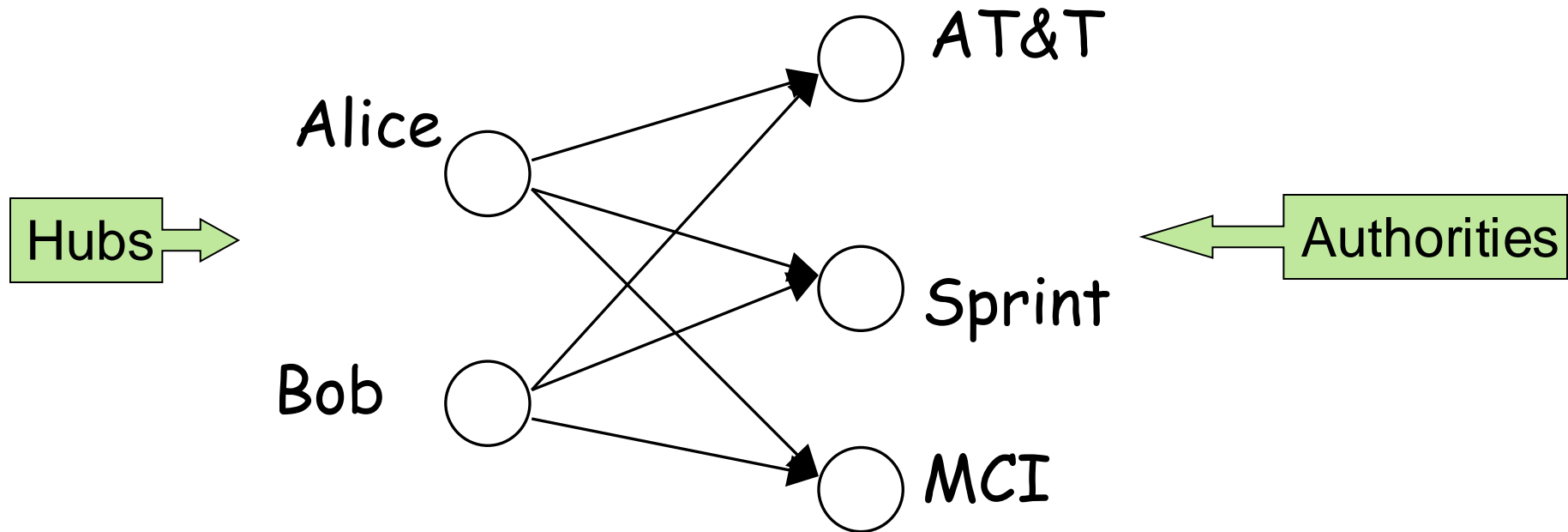
# Hyperlink-Induced Topic Search (HITS)

- In response to a query, instead of an ordered list of pages each meeting the query, find two sets of inter-related pages:
  - *Hub pages* are good lists of links on a subject.
    - e.g., “Bob’s list of cancer-related links.”
  - *Authority pages* occur recurrently on good hubs for the subject.
- Best suited for “broad topic” queries rather than for page-finding queries.
- Gets at a broader slice of common *opinion*.

# Hubs and Authorities

- Thus, a good hub page for a topic *points* to many authoritative pages for that topic.
- A good authority page for a topic is *pointed* to by many good hubs for that topic.
- Circular definition - will turn this into an iterative computation.

# The hope



***Long distance telephone companies***

# High-level scheme

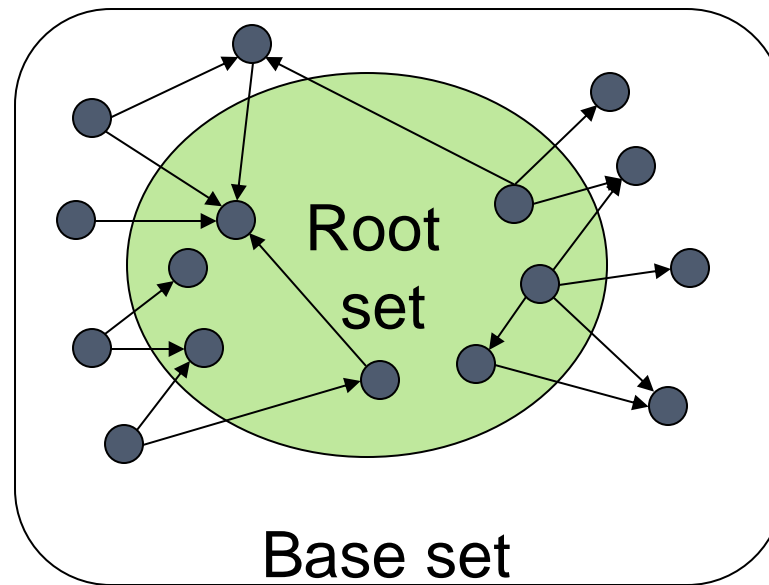
- Extract from the web a base set of pages that *could* be good hubs or authorities.
- From these, identify a small set of top hub and authority pages;  
→ iterative algorithm.



# Base set

- Given text query (say **browser**), use a text index to get all pages containing **browser**.
  - Call this the root set of pages.
- Add in any page that either
  - points to a page in the root set, or
  - is pointed to by a page in the root set.
- Call this the base set.


# Visualization



# Assembling the base set

- Root set typically 200-1000 nodes.
- Base set may have up to 5000 nodes.
- How do you find the base set nodes?
  - Follow out-links by parsing root set pages.
  - Get in-links (and out-links) from a *connectivity server*.
  - (Actually, suffices to text-index strings of the form *href=“URL”* to get in-links to URL.)

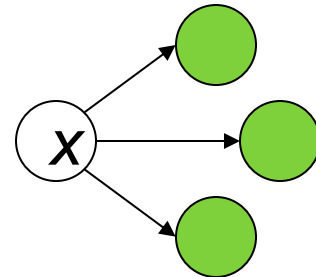
# Distilling hubs and authorities

- Compute, for each page  $x$  in the base set, a hub score  $h(x)$  and an authority score  $a(x)$ .
- Initialize: for all  $x$ ,  $h(x) \leftarrow 1$ ;  $a(x) \leftarrow 1$ ;
- Iteratively update all  $h(x)$ ,  $a(x)$ ; 
- After iterations
  - ▣ output pages with highest  $h()$  scores as top hubs
  - ▣ highest  $a()$  scores as top authorities.

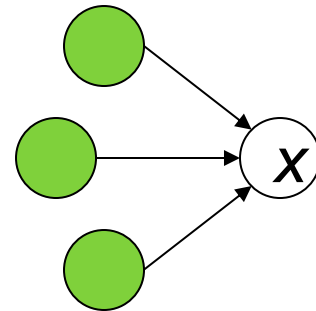
# Iterative update

- Repeat the following updates, for all  $x$ :

$$h(x) \leftarrow \sum_{x \mapsto y} a(y)$$



$$a(x) \leftarrow \sum_{y \mapsto x} h(y)$$



# Scaling

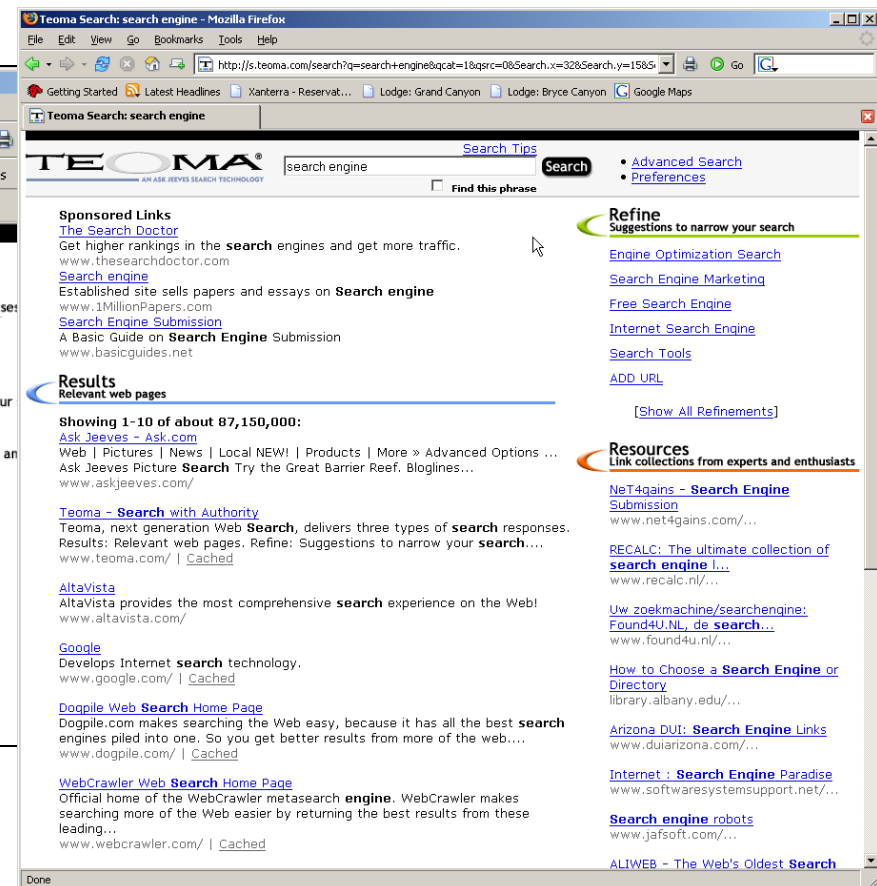
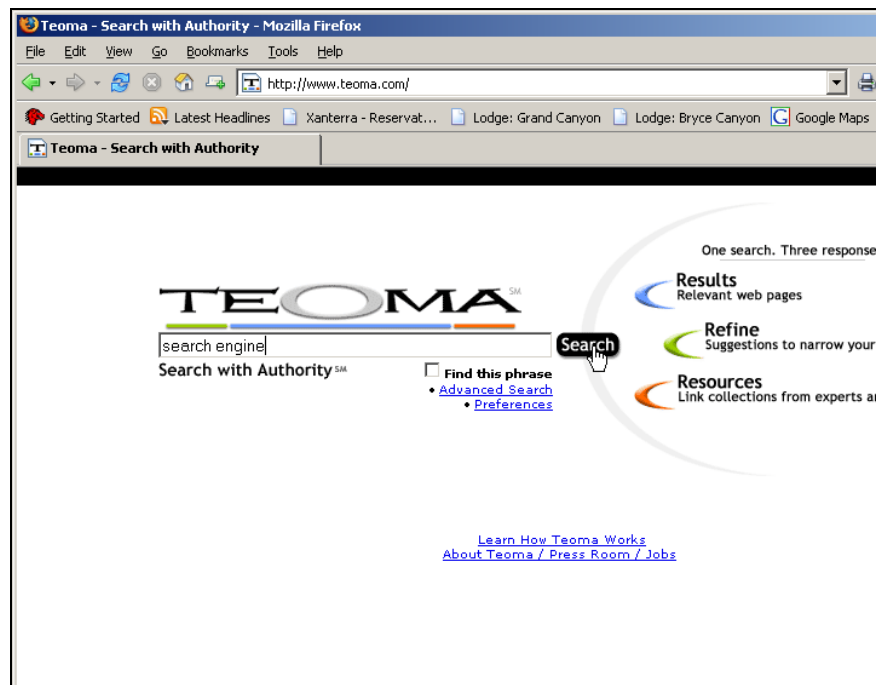
- To prevent the  $h()$  and  $a()$  values from getting too big, can scale down after each iteration.
- Scaling factor doesn't really matter:
  - we only care about the *relative* values of the scores.

# How many iterations?

- Claim: relative values of scores will converge after a few iterations:
  - in fact, suitably scaled,  $h()$  and  $a()$  scores settle into a steady state!
  - proof of this comes later.
- We only require the relative orders of the  $h()$  and  $a()$  scores - not their absolute values.
- In practice, ~5 iterations get you close to stability.

# HITS: Teoma, now search.ask.com

"Teomas underlying technology is an extension of the HITS algorithm ...", C. Sherman, April 2002, <http://dc.internet.com/news/article.php/1002061> (Not online anymore)





<b>PageRank</b>	<b>HITS</b>
<div data-bbox="855 211 929 275" data-label="Image"></div> <ul style="list-style-type: none"> <li>- Hard to spam</li> <li>- Computes quality signal for all pages</li> </ul>	<div data-bbox="1792 211 1866 275" data-label="Image"></div> <ul style="list-style-type: none"> <li>- Easy to compute, real-time execution is hard</li> <li>- Query specific</li> <li>- Works on small graphs</li> </ul>
<div data-bbox="855 589 929 654" data-label="Image"></div> <ul style="list-style-type: none"> <li>- Non-trivial to compute</li> <li>- Not query specific</li> <li>- Does not work on small graphs</li> </ul>	<div data-bbox="1792 589 1866 654" data-label="Image"></div> <ul style="list-style-type: none"> <li>- Local graph structure can be manufactured</li> <li>- Provides a signal only when there is direct connectivity (e.g. home pages)</li> </ul>
Proven to be effective for general purpose ranking	Well suited for supervised directory construction

# More sophisticated *information* retrieval

- Cross-language information retrieval
- Question answering
- Text mining/Knowledge discovery
  - Clustering
- Spamming in Web Search
- More efficient calculation
- Storage, speed, freshness
- Fighting spam
- Make better usage of the structure of the web
- Consider the continuous change of the web's structure
- Better study of its characteristics (practical + theor.):
  - Stability and sensitivity
  - Influence of the linkage
  - Influence of different parameters
- Personalization and specialization
- ...

# Need To Know

- Characteristics of the Web
- Architecture of web information retrieval
- Crawling issues
- Spamming
- Principles of PageRank
- Principles of HITS

# References

- Chapter 14: Multimedia information retrieval and management: technological fundamentals and applications, Springer, 2003.
- Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze, Introduction to information retrieval, Cambridge University Press, 2008. [Library]
  - <http://www-csli.stanford.edu/~hinrich/information-retrieval-book.html>
- Ricardo Baeza-Yates, Berthier Ribeiro-Neto, Modern Information Retrieval, Addison Wesley, 1999. [Library]
  - <http://people.ischool.berkeley.edu/~hearst/irbook/>
- I. Witten, A. Moffat, T. Bell, Managing Gigabytes (2nd edition), Morgan Kaufmann, 1999. [Library]
- C. J. Van Rijsbergen, Information Retrieval, 1979.
  - Access online <http://www.dcs.gla.ac.uk/Keith/Preface.html>
- Wolfgang Hürst, Web Search, Summer Term 2006, Albert-Ludwigs-University

