CS3245

Information Retrieval

12

Lecture 12: Crawling and Link Analysis

Live Q&A
https://pollev.com/jin

Last Time

Chapter 11

1. Classic Probabilistic Approaches for IR

Chapter 12

1. Language Models for IR

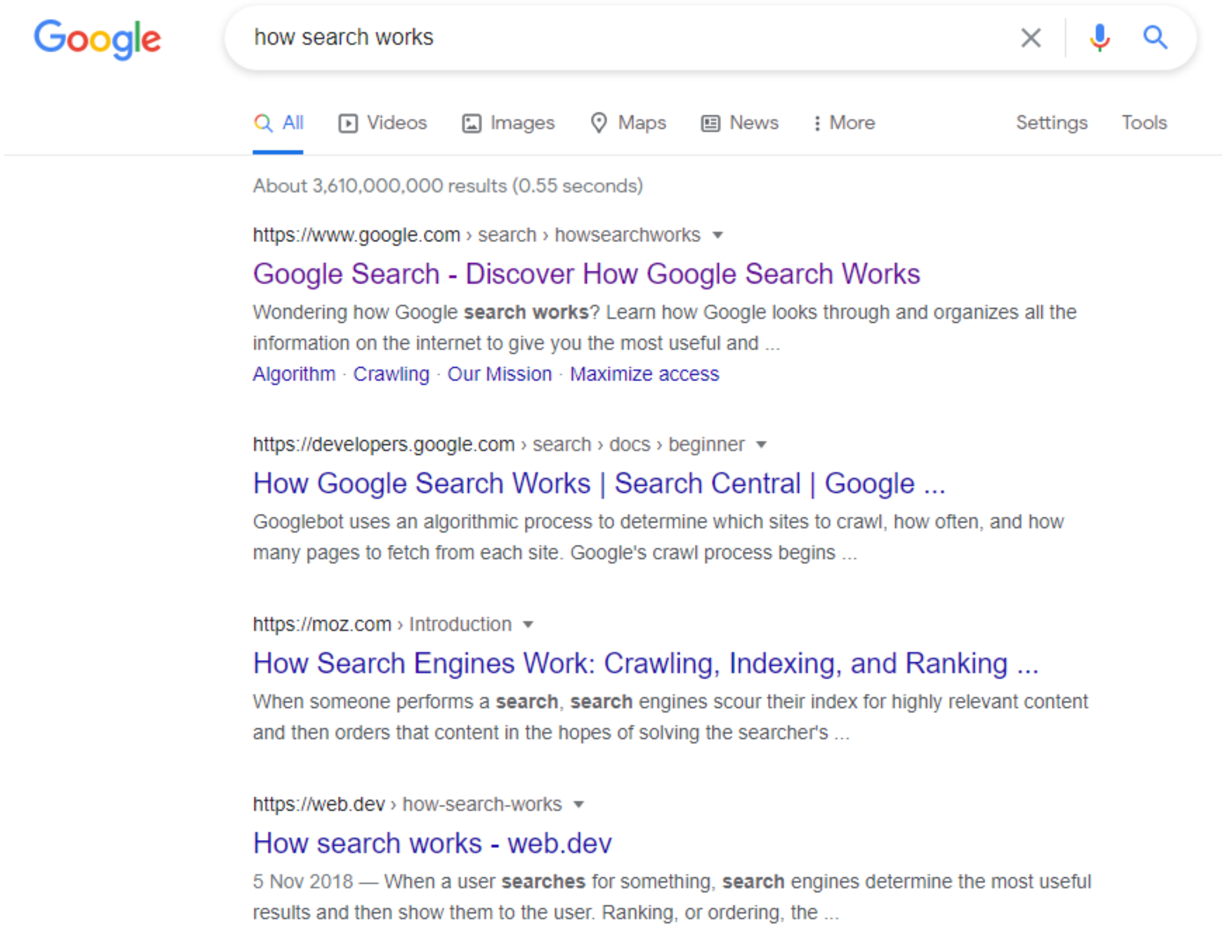
Today – The Web

Chapter 20

- Crawling

Chapter 21

- Link Analysis



About 3,610,000,000 results (0.55 seconds)

https://www.google.com › search › howsearchworks
Google Search - Discover How Google Search Works
Wondering how Google **search works**? Learn how Google looks through and organizes all the information on the internet to give you the most useful and ...
Algorithm · Crawling · Our Mission · Maximize access

https://developers.google.com › search › docs › beginner
How Google Search Works | Search Central | Google ...
Googlebot uses an algorithmic process to determine which sites to crawl, how often, and how many pages to fetch from each site. Google's crawl process begins ...

https://moz.com › Introduction
How Search Engines Work: Crawling, Indexing, and Ranking ...
When someone performs a **search**, **search** engines scour their index for highly relevant content and then orders that content in the hopes of solving the searcher's ...

https://web.dev › how-search-works
How search works - web.dev
5 Nov 2018 — When a user **searches** for something, **search** engines determine the most useful results and then show them to the user. Ranking, or ordering, the ...

CRAWLING

How hard can crawling be?

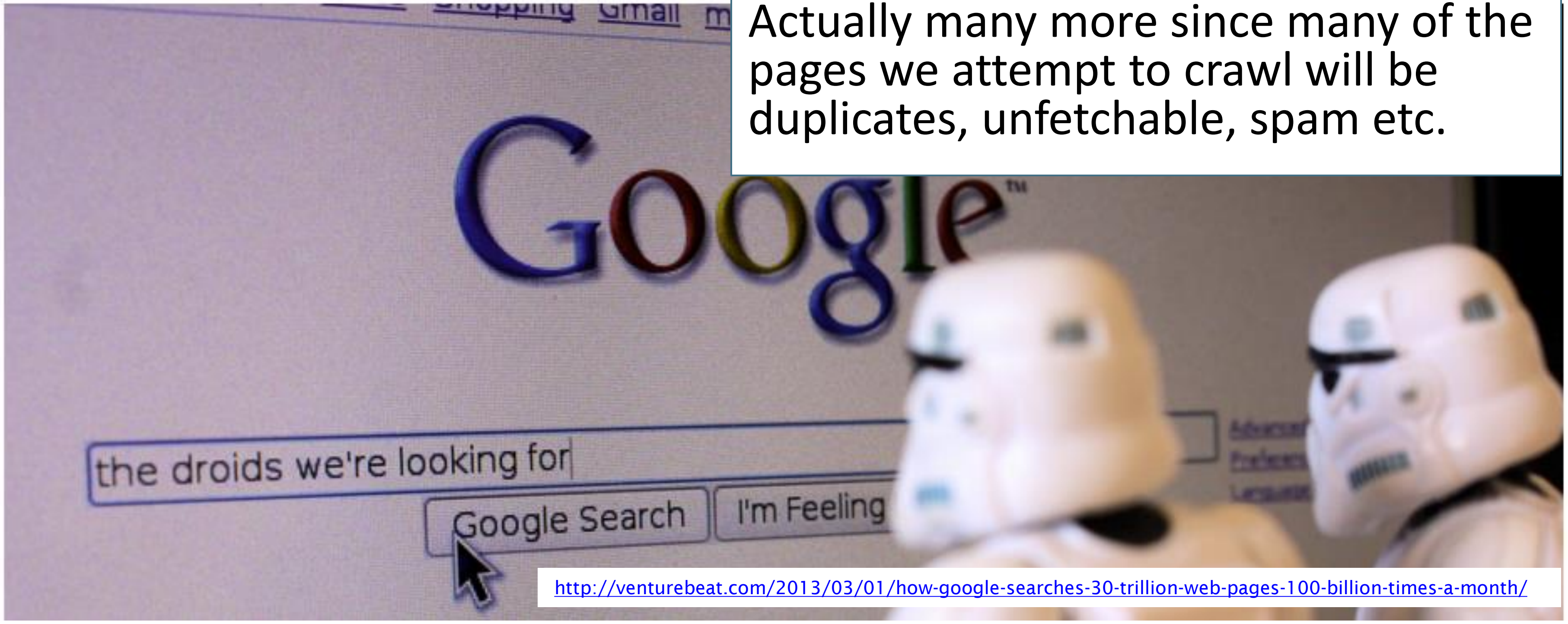
- Web search engines must crawl their documents.
- Getting the content of the documents is easier for many other IR systems.
  - E.g., indexing all files on your hard disk: just do a recursive descent on your file system
- Bandwidth, latency…
  - Not just on crawler side, **but also on the web server side**.

How hard can crawling be?

How Google searches 30 trillion web pages, 100 billion times a month

JOHN KOETSIER, TUNE MARCH 1, 2013 12:43 PM

TAGS: FEATURED, GOOGLE, INDEX, SEARCH ENGINE, WEB SEARCH



To fetch 30T pages in one month, we need to fetch almost 11M pages per second!

Actually many more since many of the pages we attempt to crawl will be duplicates, unfetchable, spam etc.

http://venturebeat.com/2013/03/01/how-google-searches-30-trillion-web-pages-100-billion-times-a-month/

Basic crawler operation

- Initialize queue with URLs of known seed pages
- Repeat
  - **Take URL from queue**
  - **Fetch and parse page**
  - **Extract URLs from page**
  - **Add URLs to queue**
  - Call the indexer to index the page

{https://en.wikipedia.org/,
https://nlp.stanford.edu/IR-book/, …}

https://en.wikipedia.org/

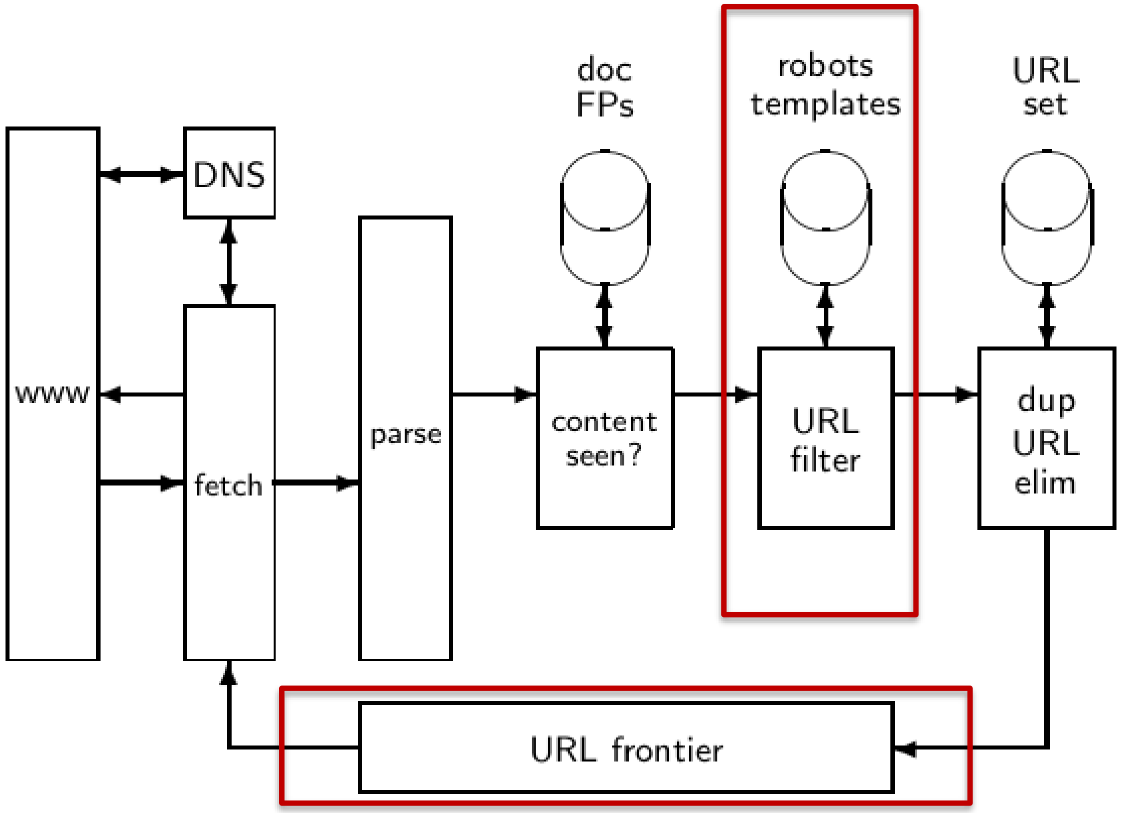
https://en.wikipedia.org/wiki/Main_Page, …

{https://nlp.stanford.edu/IR-book/, …,
https://en.wikipedia.org/wiki/Main_Page, …}

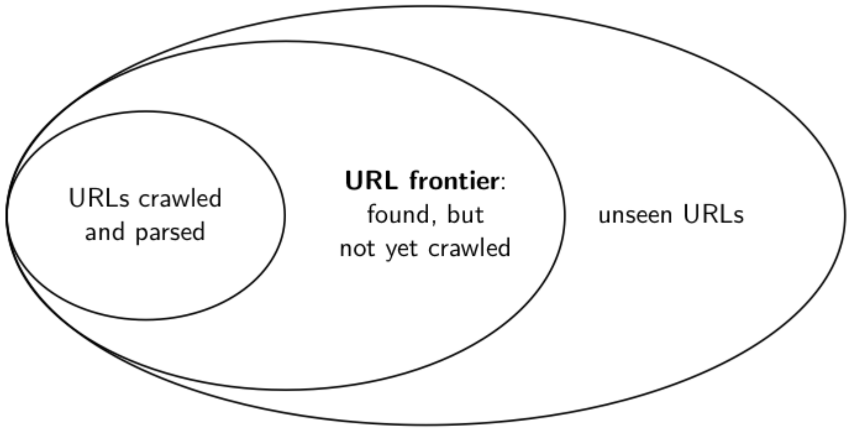
- What's wrong with this Crawler?

Politeness

- Space out the requests for a site
- Crawl only the allowed pages



URL Frontier



- The URL frontier is the data structure that holds and manages URLs we've seen, but not crawled yet.
- May include multiple pages from the same host but must **avoid trying to fetch them all at the same time**

Robots.txt

- Protocol for giving crawlers ("robots") limited access to a website, originally from 1994
- Example:

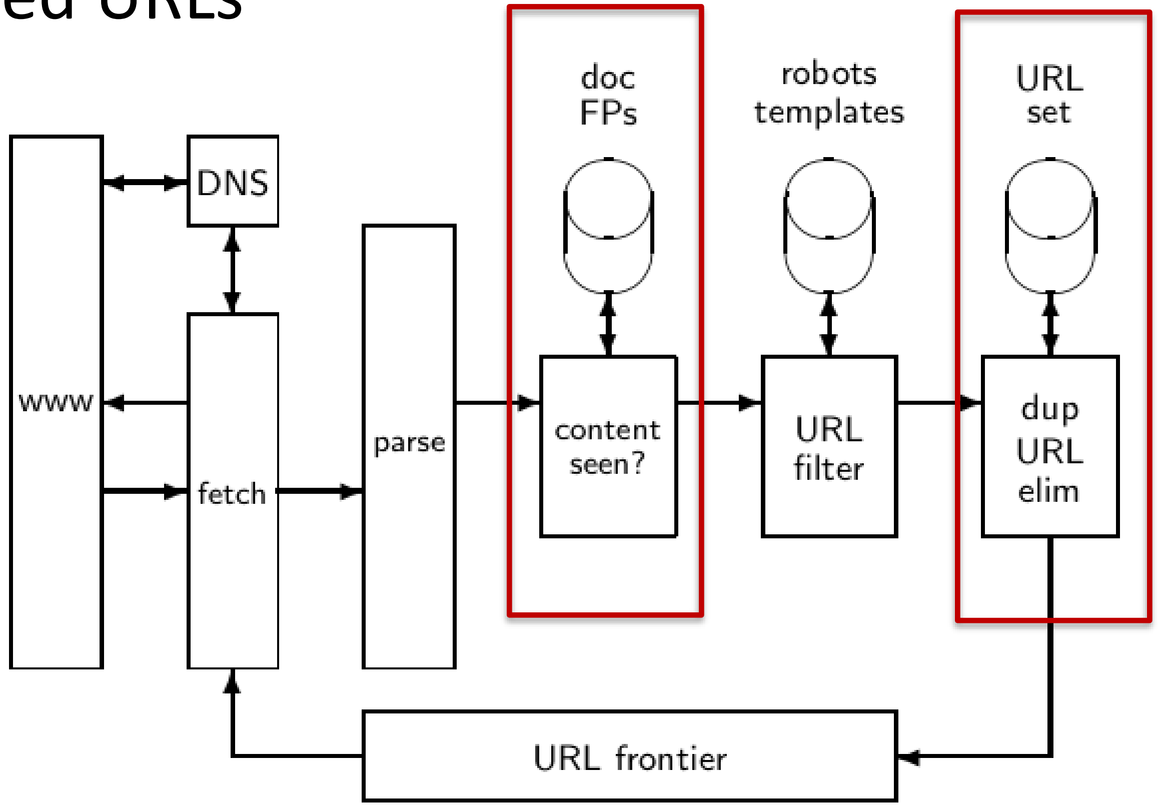
```
# Observed spamming large amounts of
    https://en.wikipedia.org/?curid=NNNNNN
User-agent: MJ12bot
Disallow: /

# Wikipedia work bots:
User-agent: IsraBot
Disallow:
```

Important: cache the robots.txt file of each site we are crawling

Duplicate detection

- Duplicated contents
- Duplicated URLs



Content seen

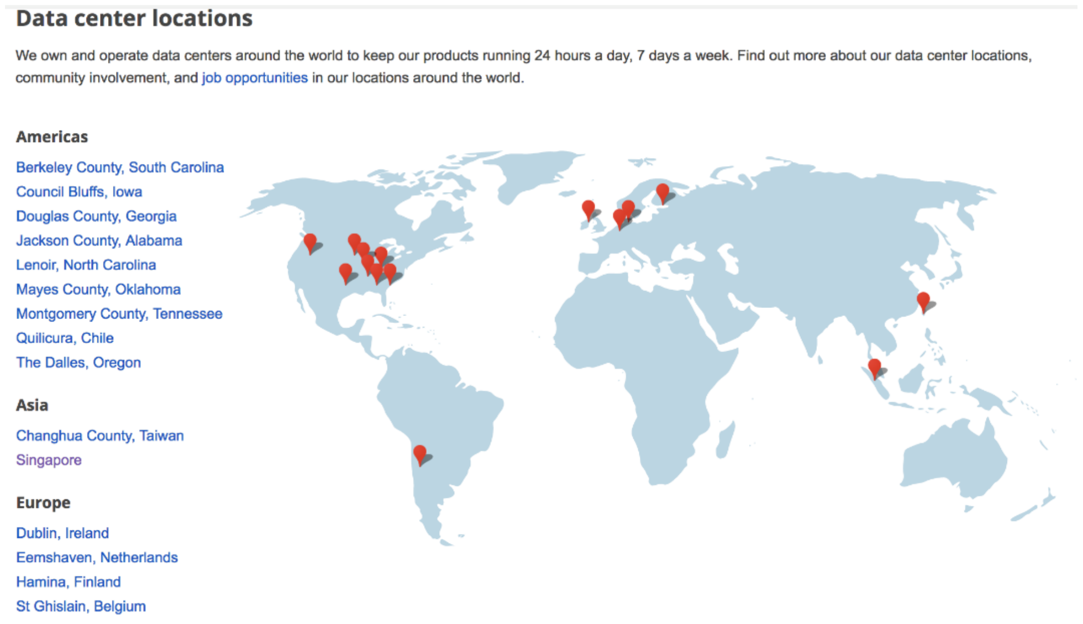
- For each page fetched: check if the content is already in the index
- Check this using document fingerprints or shingles
- Skip documents whose content has already been indexed

URL seen

- Duplicate elimination
  - Ignore the URLs which have been seen before.
- Normalization
  - We need to normalize (expand) all relative URLs.
    - E.g., at http://mit.edu, we may have aboutsite.html which is in fact http://mit.edu/aboutsite.html
- Freshness
  - Crawl some pages (e.g., news sites) more often than others

Scalability

- Run multiple crawl threads
- Use different (geographically distributed) nodes



Data center locations

We own and operate data centers around the world to keep our products running 24 hours a day, 7 days a week. Find out more about our data center locations, community involvement, and job opportunities in our locations around the world.

Americas

Berkeley County, South Carolina
Council Bluffs, Iowa
Douglas County, Georgia
Jackson County, Alabama
Lenoir, North Carolina
Mayes County, Oklahoma
Montgomery County, Tennessee
Quilicura, Chile
The Dalles, Oregon

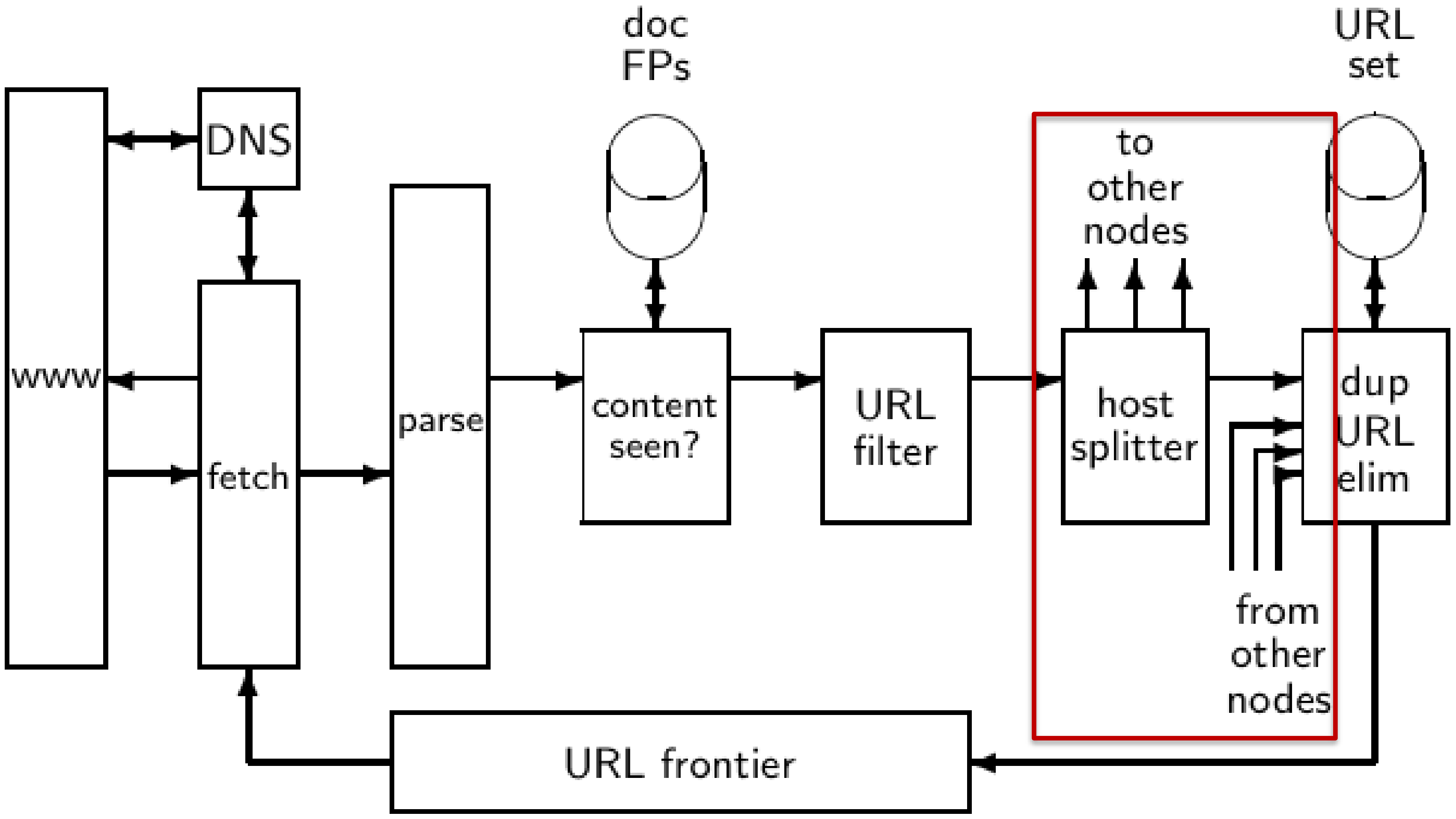
Asia

Changhua County, Taiwan
Singapore

Europe

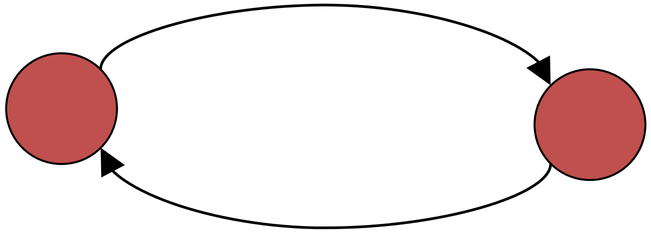
Dublin, Ireland
Eemshaven, Netherlands
Hamina, Finland
St Ghislain, Belgium

Distributed crawling architecture

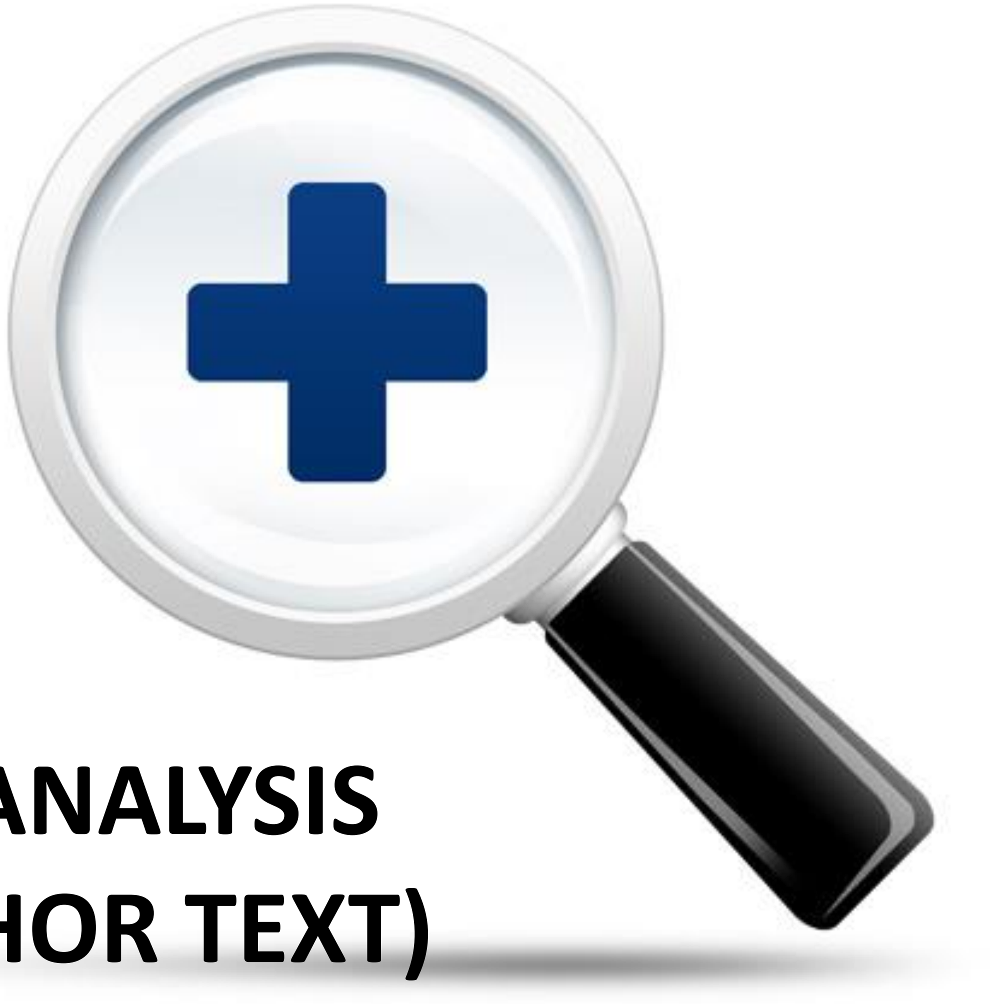


Spider traps

- A set of webpages that cause the crawler to go into an infinite loop or crash
  - A simple loop back

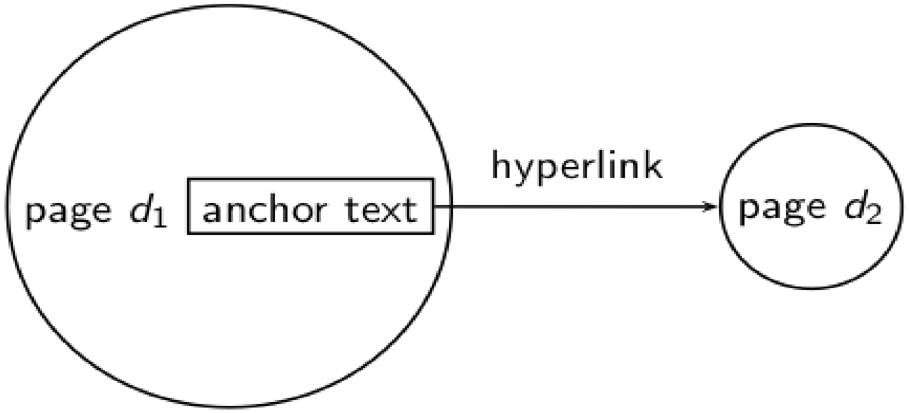


  - Calender page with a dynamic link to the next month
- May be created intentionally or unintentionally



LINK ANALYSIS (ANCHOR TEXT)

Anchor Text



Example: "You can find cheap cars <a href="http://...">here</a>."
Anchor text: "You can find cheap cars here."

- Assumption: The anchor text describes the content of d_2.
  - Anchor text is loosely defined as the text **surrounding** the hyperlink.

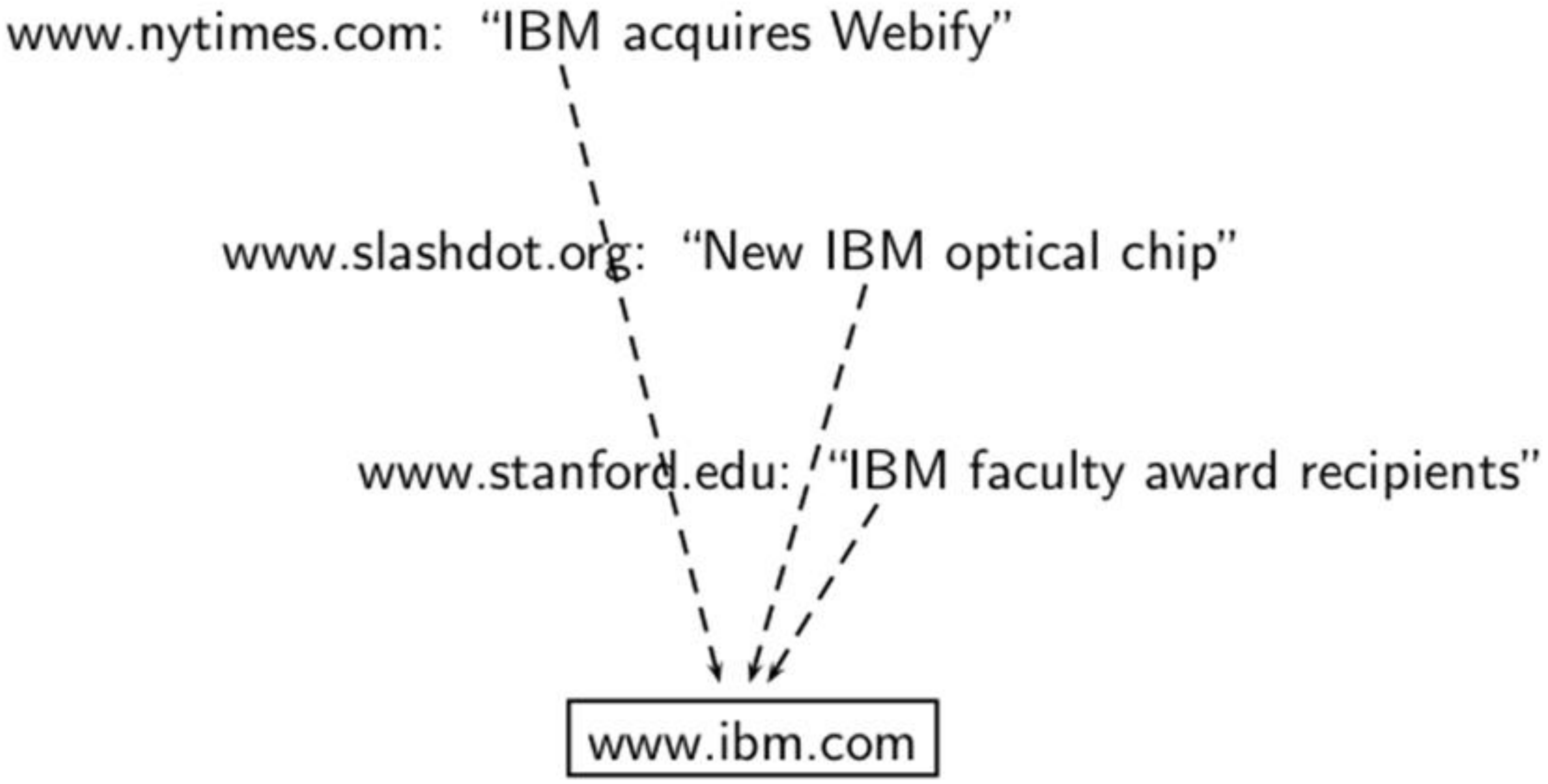
[text of d_2] only vs. [text of d_2] + [anchor text → d_2]

- Searching on [text of d_2] + [anchor text → d_2] is often more effective than searching on [text of d_2] only.

- Example: Query *IBM*
  - IBM Wikipedia article (398 mentions)
  - Many spam pages that include hundreds of mentions of IBM intentionally
  - IBM home page (only 22 mentions with many graphics / video)

- Searching on [anchor text → d_2] is better for the query IBM
  - In this representation, the page with most occurrences of IBM is www.ibm.com.

Anchor text containing *IBM* pointing to www.ibm.com


www.nytimes.com: "IBM acquires Webify"
www.slashdot.org: "New IBM optical chip"
www.stanford.edu: "IBM faculty award recipients"
www.ibm.com

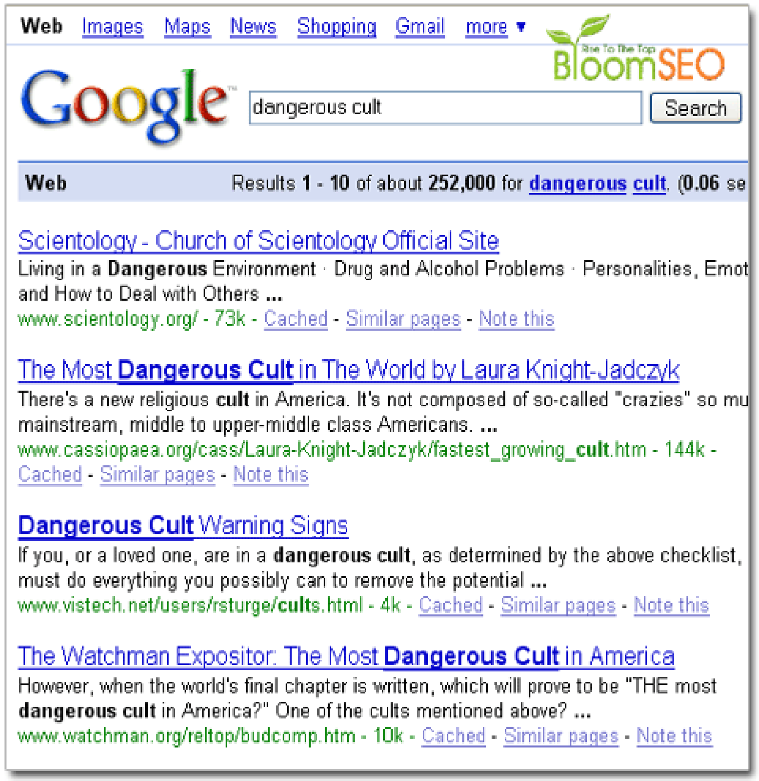
Indexing anchor text

- Thus: Anchor text is often a better description of a page's content than the page itself.

- Anchor text can be weighted more highly than document text.

Google bombs

- Is a search with "bad" results due to maliciously manipulated anchor text.
- E.g., [dangerous cult] on Google, Bing, Yahoo
  - Coordinated link creation by those who dislike the Church of Scientology
- Google introduced a new weighting function in January 2007 that fixed many Google bombs.
- Defused Google bombs: [who is a failure?], [evil empire]



Web Results 1 - 10 of about 252,000 for dangerous cult. (0.06 se

[Scientology - Church of Scientology Official Site]
Living in a **Dangerous** Environment · Drug and Alcohol Problems · Personalities, Emot
and How to Deal with Others ...
www.scientology.org/ - 73k - Cached - Similar pages - Note this

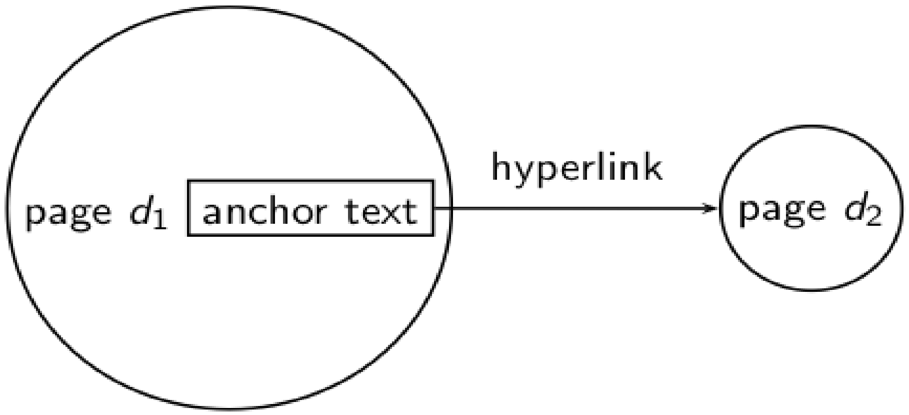
The Most **Dangerous Cult** in The World by Laura Knight-Jadczyk
There's a new religious **cult** in America. It's not composed of so-called "crazies" so mu
mainstream, middle to upper-middle class Americans. ...
www.cassiopaea.org/cass/Laura-Knight-Jadczyk/fastest_growing_**cult**.htm - 144k -
Cached - Similar pages - Note this

Dangerous Cult Warning Signs
If you, or a loved one, are in a **dangerous cult**, as determined by the above checklist,
must do everything you possibly can to remove the potential ...
www.vistech.net/users/rsturge/**cults**.html - 4k - Cached - Similar pages - Note this

The Watchman Expositor: The Most **Dangerous Cult** in America
However, when the world's final chapter is written, which will prove to be "THE most
dangerous cult in America?" One of the cults mentioned above? ...
www.watchman.org/reltop/budcomp.htm - 10k - Cached - Similar pages - Note this

LINK ANALYSIS (PAGERANK)

Quality Signal

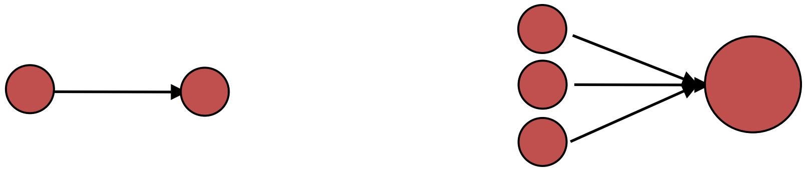


Example: "You can find cheap cars <a href="http://...">here</a>."
Anchor text: "You can find cheap cars here."

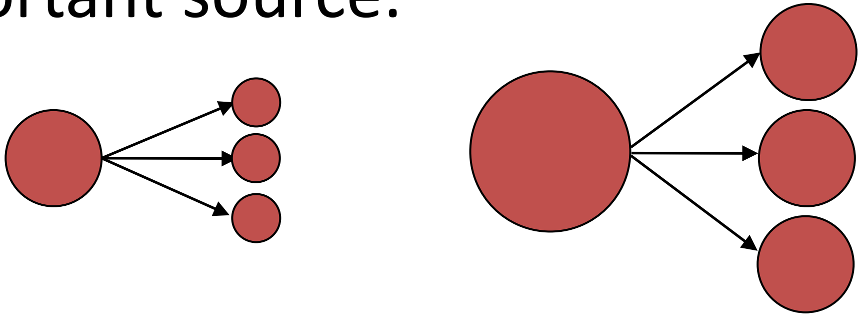
- Assumption: A hyperlink is a quality signal.
  - The hyperlink $d_1 \rightarrow d_2$ indicates that d_1's author deems d_2 high-quality and relevant.

Quality Signal

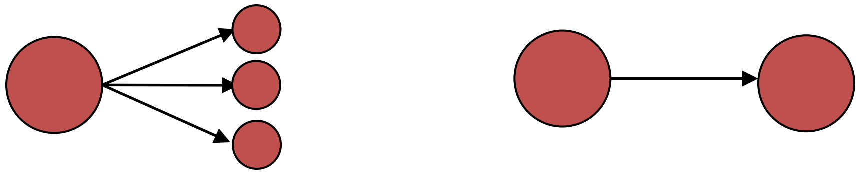
- It is good to have more endorsements.



- It is good for the endorsement to come from an important source.

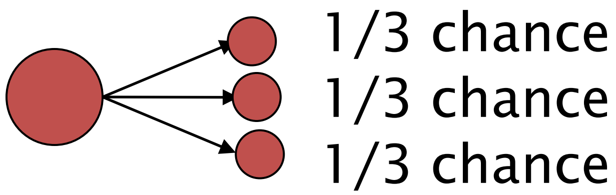


- It is good for the endorsement to be exclusive.



PageRank

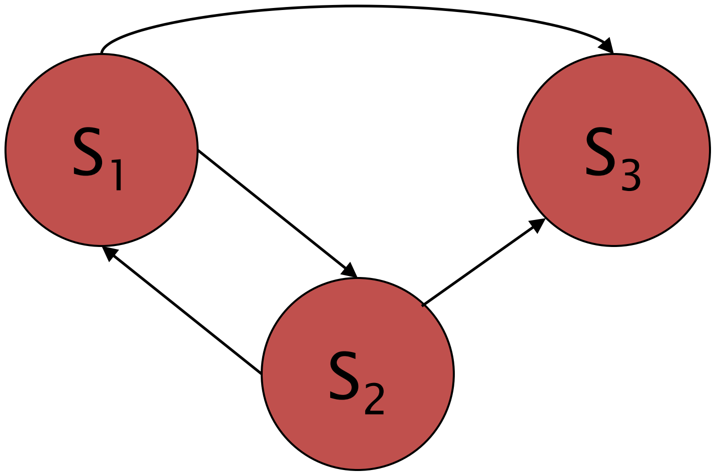
- Imagine a browser doing a random walk on web pages:
  - Start at a random page
  - At each step, follow one of the *n* links on that page, each with *1/n* probability



- Do this repeatedly. Use the "long-term visit rate" as the page's score
  - This is a global score for the page, based on the topology of the network
  - Think of it as *g(d)* from Chapter 7

Random walks

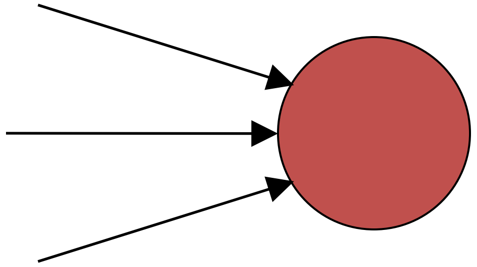
- A first order Markov chain consisting of *n* states and an *nxn* transition probability matrix *A*.
  - Each state correspond to a web page.
  - A_{ik} is the probability of going from state *i* to state *k.*
  - The next state depends only on the current state.



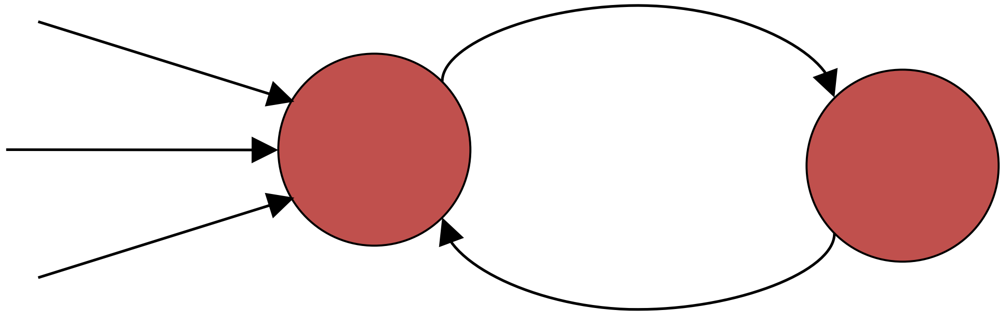
$$\begin{array}{cccc} & S_1 & S_2 & S_3 \\ S_1 & 0 & 1/2 & 1/2 \\ S_2 & 1/2 & 0 & 1/2 \\ S_3 & ?? & ?? & ?? \end{array}$$

Not quite enough

- The web is full of dead ends.
  - What sites have dead ends?
  - Our random walk can get stuck.



Dead End



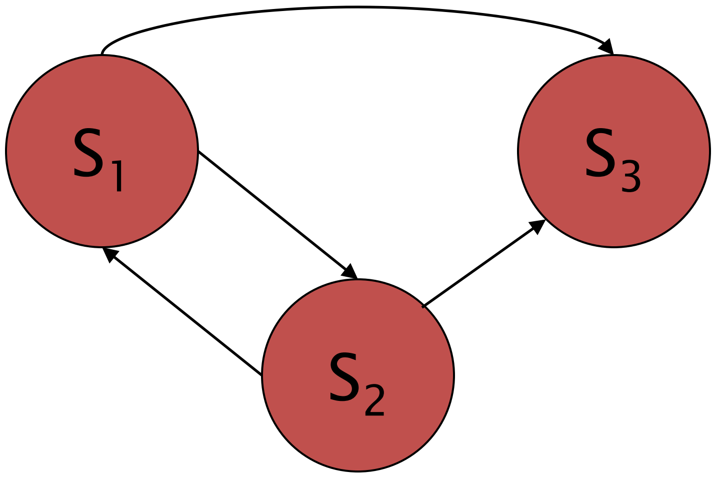
Spider Trap

Teleporting

- When a node has no outlinks
  - Teleport to a random web page
- Otherwise, at each step
  - With probability α (e.g., 10%), teleport to a random web page
  - With remaining probability (e.g., 90%), follow a random link on the page with equal probability

Random walks with teleportation

$$\begin{tikzcd}
S_1 \arrow[dr] \arrow[rr, bend left] & & S_3 \\
& S_2 \arrow[ul] \arrow[ur] &
\end{tikzcd}$$

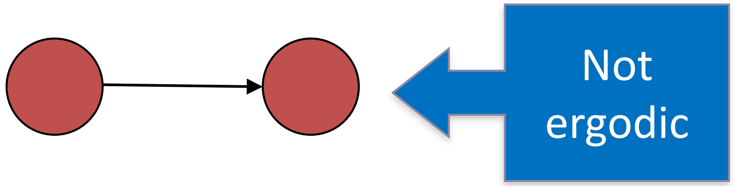


Matrix A with 10% chance of teleportation.

$$\begin{array}{cccc}
 & S_1 & S_2 & S_3 \\
S_1 & 1/30 & 29/60 & 29/60 \\
S_2 & 29/60 & 1/30 & 29/60 \\
S_3 & 1/3 & 1/3 & 1/3
\end{array}$$

Ergodic Markov chains

- A Markov chain is **ergodic** if
  - you have a path from any state to any other
  - you can be in any state at every time step, with non-zero probability



  - With teleportation, our Markov chain is ergodic

- Theorem: With an ergodic Markov chain, there is a **stable** long term visit rate.

Probability vectors

- A probability (row) vector $x = (x_1, \dots, x_n)$ tells us where the surfer is at a particular point of time
- E.g., $(0_1 00 \dots 1_i \dots, 000_n)$ → We're in state *i* with 100% probability

- More generally, the vector $x = (x_1, \dots, x_n)$ means the walk is in state *i* with probability x_i.

$$\sum_{i=1}^{n} x_i = 1$$

Change in probability vector

- If the probability vector is $x = (x_1, \ldots, x_n)$ at this step, what is it at the next step?
  - Recall at row *i* of the transition prob. Matrix *A* tells us where we go next from state *i*.
  - So from *x*, our next state is distributed as *xA*.

S_1 to S_1 *S_2 to S_1* *S_3 to S_1*

1*1/6 + 0*5/12 + 0*1/6

$$\vec{x_0} = (1\ 0\ 0)$$
$$\vec{x_1} = \vec{x_0}\boldsymbol{A} = \begin{pmatrix} 1/6 & 2/3 & 1/6 \end{pmatrix}$$
$$\vec{x_2} = \vec{x_1}\boldsymbol{A} = \begin{pmatrix} 1/3 & 1/3 & 1/3 \end{pmatrix}$$

$\boldsymbol{A} =$

	S1	S2	S3
S1	1/6	2/3	1/6
S2	5/12	1/6	5/12
S3	1/6	2/3	1/6

Steady State

- For any ergodic Markov chain, there is a unique long-term visit rate for each state
  - Over a long period, we'll visit each state in proportion to this rate
  - It doesn't matter where we start

PageRank algorithm

- Regardless of where we start, we eventually reach the steady state *a*
  1. Start with any distribution (say $x = (1\ 0\ \ldots\ 0)$).
  2. After one step, we're at xA.
  3. After two steps at xA^2, then xA^3 and so on.
  4. "Eventually" means for "large" k, $xA^k = a$.

- Algorithm: multiply x by increasing powers of A until the product looks stable.

At the steady state *a*…

$$aA = a$$

- So the rank vector is an eigenvector of the adjacency matrix
  - In fact, it's the first or principal eigenvector, with corresponding eigenvalue 1.

PageRank summary

- Pre-processing:
  - Given a graph of links, build matrix *A*
  - From it compute *a*
  - The page rank a_i is a scaled number between 0 and 1.
- Query processing
  - Retrieve pages meeting query
  - Rank them by their PageRank
  - Order is *query-independent*

How important is PageRank?

- Frequent claim: PageRank is the most important component of web ranking.
- The reality:
  - There are several components that are at least as important: e.g., anchor text, phrases, proximity, tiered indexes ...
  - PageRank in its original form (as presented here) has a negligible impact on ranking.
  - However, variants of a page’s PageRank are still an essential part of ranking.
  - Addressing link spam is difficult and crucial.

Summary

- Crawling – Obtaining documents for indexing
  - Need to be polite
- PageRank – A *g(d)* for asymmetrically linked documents

- Chapters 20 and 21 of IIR
- Resources
  - Paper on Mercator Crawler by Heydon et al.
  - Robot Exclusion Standard

PageRank reflects our view of the importance of web pages by considering more than 500 million variables and 2 billion terms. Pages that believe are important pages receive a higher PageRank and are more likely to appear at the top of the search results"

EXAM MATTERS

General Information

- Date/Time: 25 Apr (Tue), 9-11am
- Venue: MPSH5
- Format
  - Pen-and-paper
  - Open book (only printed / written materials allowed)
  - Calculators allowed

Exam Scope

- All lectures (Weeks 1-12) and tutorials (T1-6) + corresponding sections in the textbook
- For sample questions, refer to tutorials, midterm, and past year exam papers.
- If in doubt, ask on the forum

Types of Questions

- Q1 is true/false questions on various topics (10 marks). No justifications required.
- Q2 is short questions on various topics (36 marks). No need to show work for calculations.
- Q3-6 are long questions, topic-specific (44 marks in total). Need to show work / give justifications as required.

Help Session

- Tentatively on 21 Apr (Fri), 3-5pm.
- Via Zoom
  - Recording will be available afterwards.
- Agenda
  - Exam Paper 21/22 AY Semester 2
  - Q&A

WHERE TO GO FROM HERE

Learning Objectives

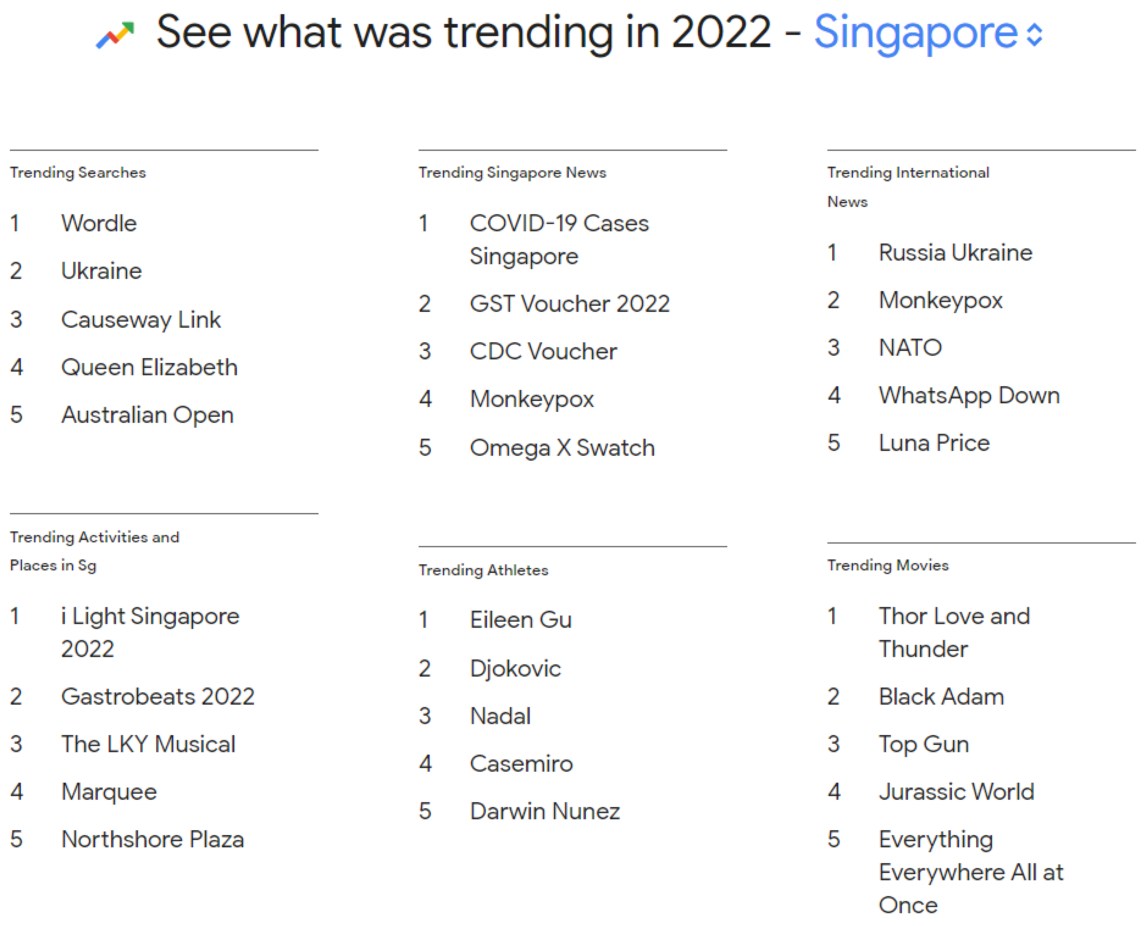
- You have learnt how to build your own search engine!

- In addition, you have picked up skills that are useful for your future
  - **Python** – one of the easiest and more straightforward programming languages to use.
  - **NLTK** – A good set of routines and data that are useful in dealing with NLP and IR.

Photo credits:
http://www.flickr.com/photos/aperezdc/

Opportunities in IR

Keep Searching



See what was trending in 2022 - Singapore

Trending Searches

1. Wordle
2. Ukraine
3. Causeway Link
4. Queen Elizabeth
5. Australian Open

Trending Singapore News

1. COVID-19 Cases Singapore
2. GST Voucher 2022
3. CDC Voucher
4. Monkeypox
5. Omega X Swatch

Trending International News

1. Russia Ukraine
2. Monkeypox
3. NATO
4. WhatsApp Down
5. Luna Price

Trending Activities and Places in Sg

1. i Light Singapore 2022
2. Gastrobeats 2022
3. The LKY Musical
4. Marquee
5. Northshore Plaza

Trending Athletes

1. Eileen Gu
2. Djokovic
3. Nadal
4. Casemiro
5. Darwin Nunez

Trending Movies

1. Thor Love and Thunder
2. Black Adam
3. Top Gun
4. Jurassic World
5. Everything Everywhere All at Once

https://about.google/stories/year-in-search/

Thanks for joining us for the journey!