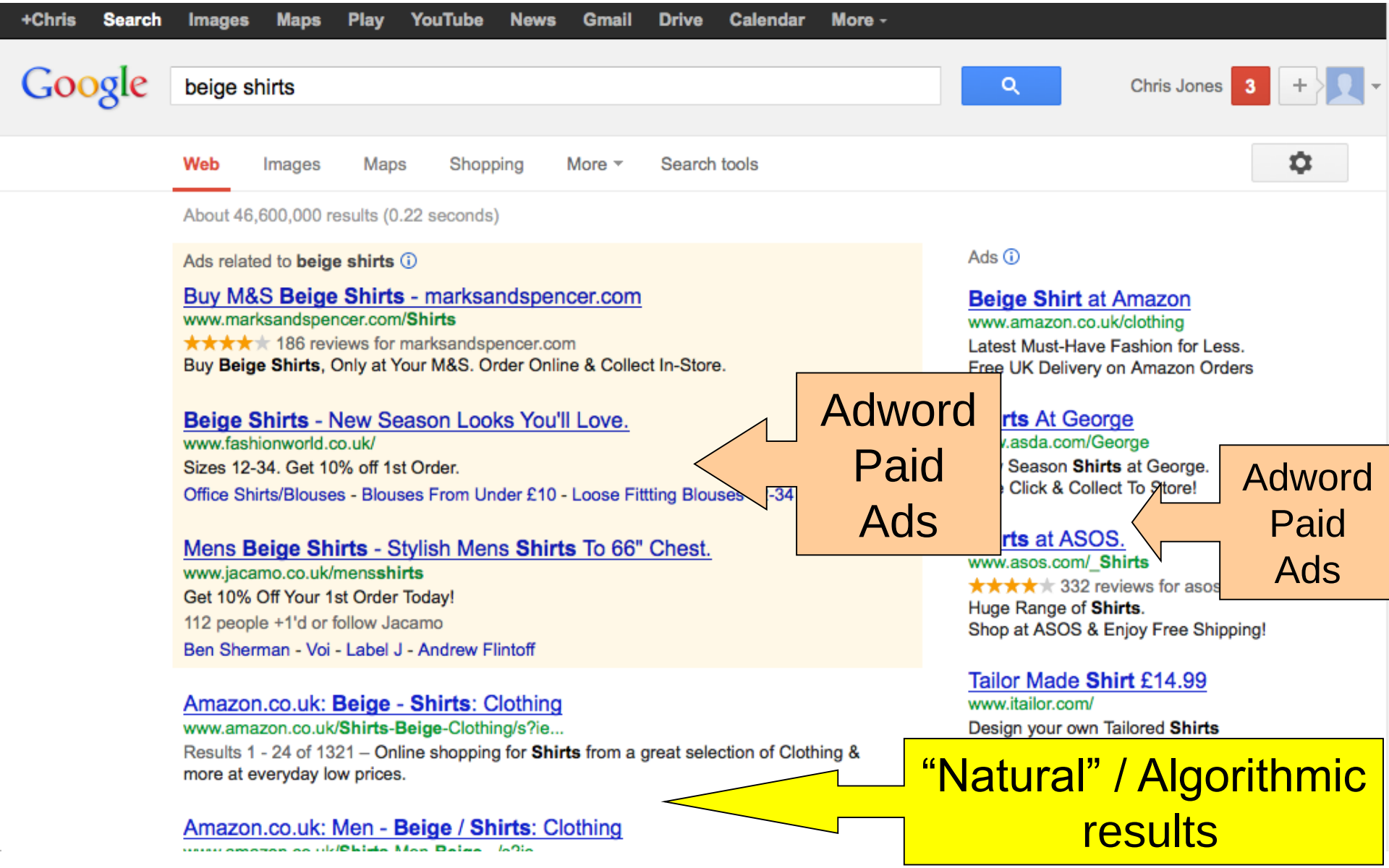
Web Applications

Web Search Engines

Martin Caminada
Chris Jones


+Chris Search Images Maps Play YouTube News Gmail Drive Calendar More
Google
beige shirts
Chris Jones 3
Web Images Maps Shopping More Search tools
About 46,600,000 results (0.22 seconds)
Ads related to beige shirts
Buy M&S Beige Shirts - marksandspencer.com
www.marksandspencer.com/Shirts
186 reviews for marksandspencer.com
Buy Beige Shirts, Only at Your M&S. Order Online & Collect In-Store.
Beige Shirts - New Season Looks You'll Love.
www.fashionworld.co.uk/
Sizes 12-34. Get 10% off 1st Order.
Office Shirts/Blouses - Blouses From Under £10 - Loose Fittting Blouses -34
Mens Beige Shirts - Stylish Mens Shirts To 66" Chest.
www.jacamo.co.uk/mensshirts
Get 10% Off Your 1st Order Today!
112 people +1'd or follow Jacamo
Ben Sherman - Voi - Label J - Andrew Flintoff
Amazon.co.uk: Beige - Shirts: Clothing
www.amazon.co.uk/Shirts-Beige-Clothing/s?ie...
Results 1 - 24 of 1321 – Online shopping for Shirts from a great selection of Clothing & more at everyday low prices.
Amazon.co.uk: Men - Beige / Shirts: Clothing
Ads
Beige Shirt at Amazon
www.amazon.co.uk/clothing
Latest Must-Have Fashion for Less.
Free UK Delivery on Amazon Orders
rts At George
.asda.com/George
Season Shirts at George.
Click & Collect To Store!
rts at ASOS.
www.asos.com/_Shirts
332 reviews for asos
Huge Range of Shirts.
Shop at ASOS & Enjoy Free Shipping!
Tailor Made Shirt £14.99
www.itailor.com/
Design your own Tailored Shirts
Adword Paid Ads
Adword Paid Ads
"Natural" / Algorithmic results

What do web search engines do?

Relevance-ranked documents

"Natural / Algorithmic results"

vs

"pay by click" or "paid for inclusion"

or "sponsored" results

- what else might be returned?

What is relationship between query terms and the documents returned?

Web Search Overview

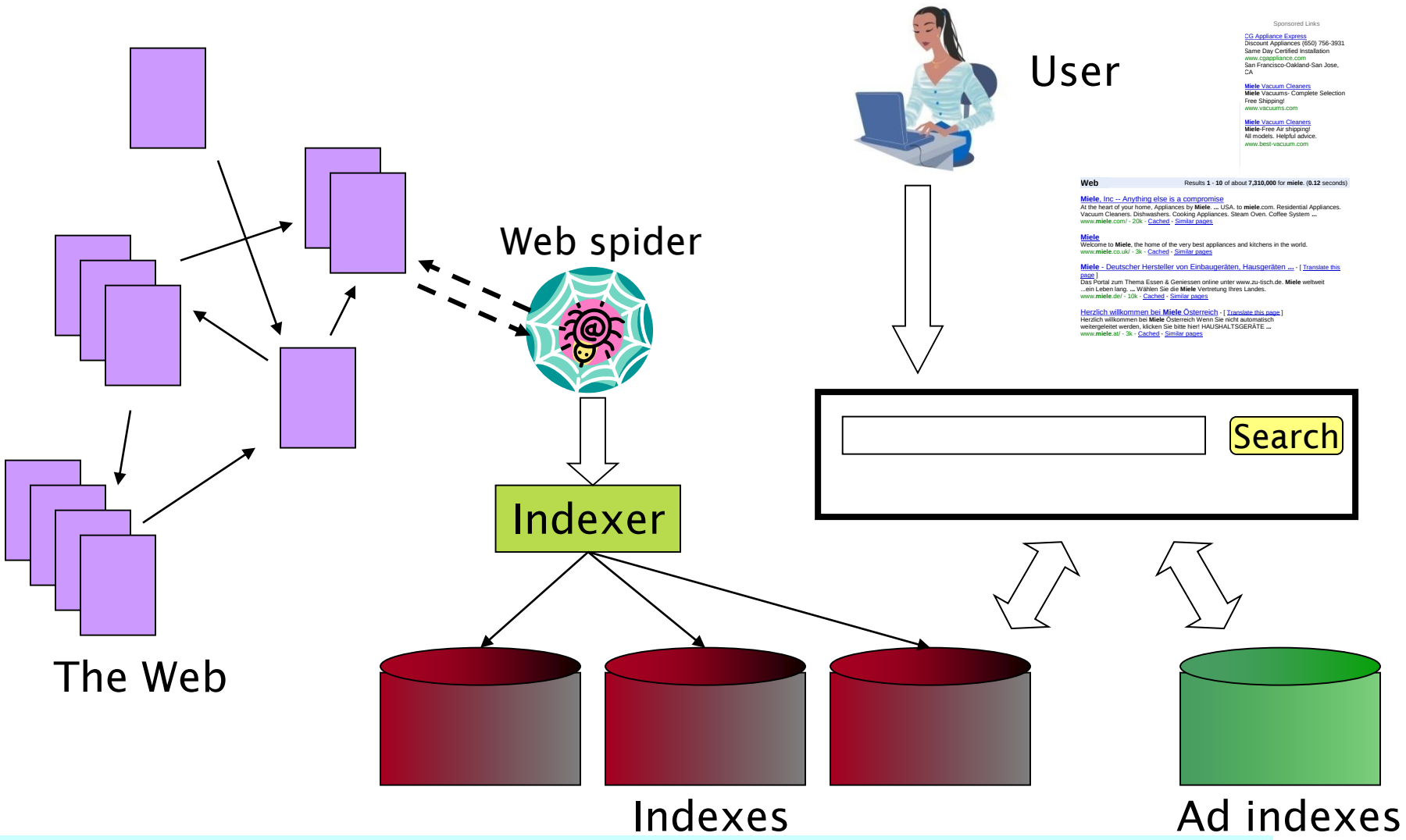


Figure from Manning et al 2008

How Does the Search Engine Find These Documents?

- Uses crawler / spider "robots" to find web pages across the Internet
  - *How many web servers must it visit?*
  - *How many web pages?*
- Creates indexes of the found web documents (and of the advertised web sites)
- Given a query, use the index to find docs that match the query terms (subject to constraints)
- Rank the found documents
- Select the advertised sites

Web Crawling: Finding the Documents

Start with a list of “seed” URLs (initial queue)

Repeat:

Visit a URL

- Parse text / find hyperlinks
- Place the found URLs on the queue (waiting to be visited)

= the ***URL Frontier***

Until no more unseen URLs

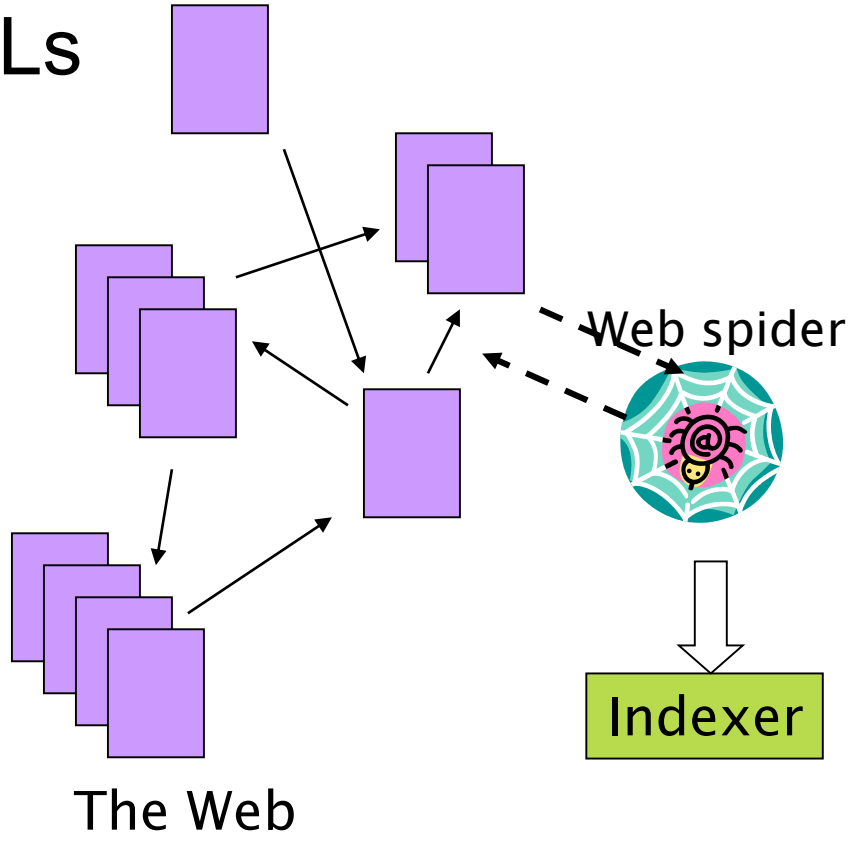


Figure from Manning et al 2008

The URL frontier

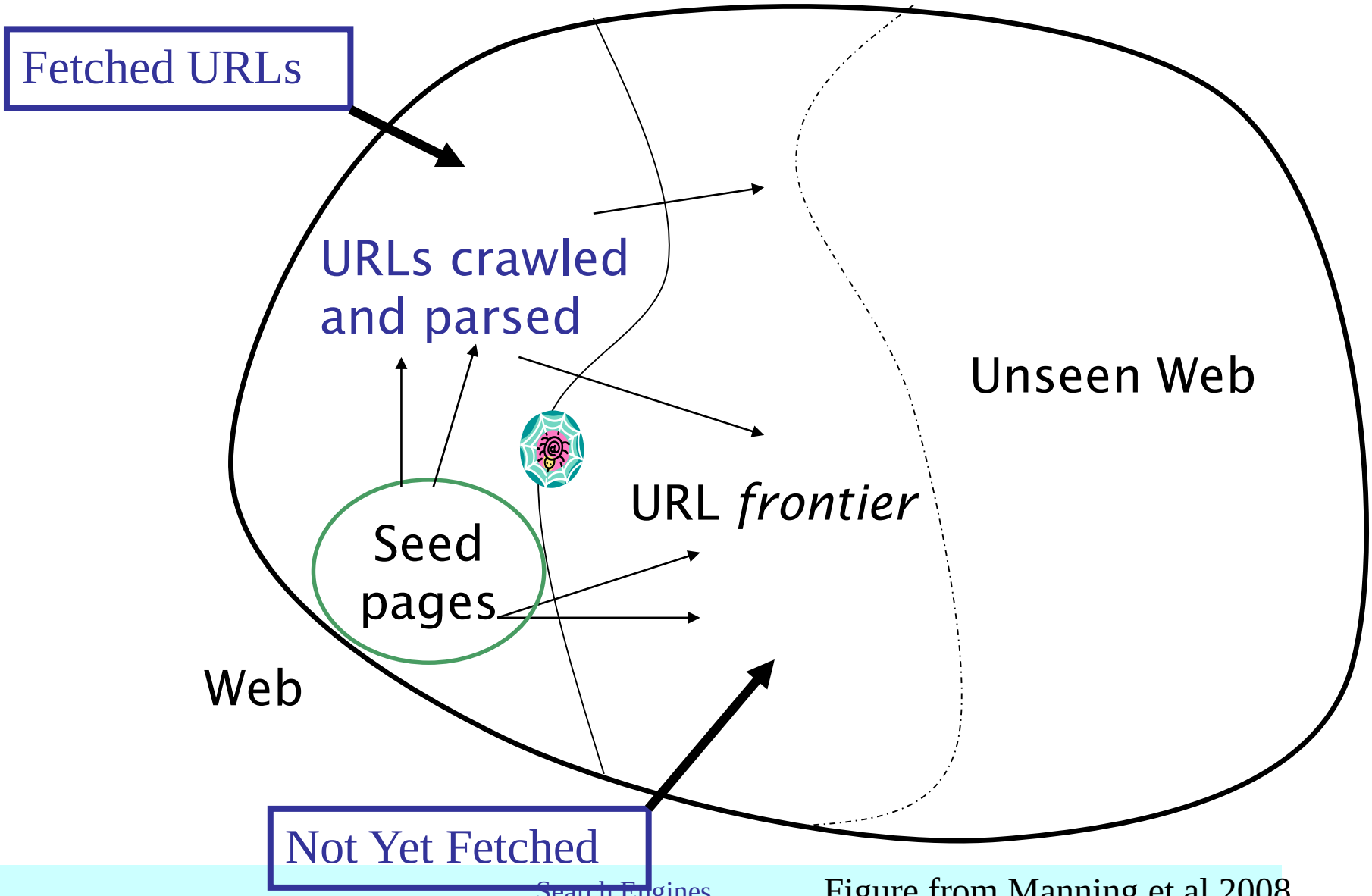


Figure from Manning et al 2008

Some issues with crawling

- **A massive task** – needs multiple computers
- Spiders need to **be robust**
  - Avoid getting stuck in cycles
  - – - accidental or malicious
- Some websites have **junk** in the web pages
- **Duplicate pages** / Mirror sites
- **How deeply a site is accessed**, what types of content are extracted / parsed

Many pages have **dynamic content**
→ the "**Hidden Web**"

– Some of this does get indexed

Crawler Politeness

- Obey the robots exclusion standard
  - http://www.robotstxt.org
    http://en.wikipedia.org/wiki/Robots_exclusion_standard
  - Web master specifies what parts of the site are to be indexed
- Do not hog a web site's resources (limit rate of access to a site – don't try to fetch all pages from a site at once)

“Freshness”

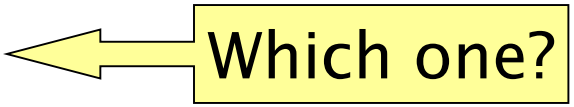
- Need to keep checking pages
  - Pages change

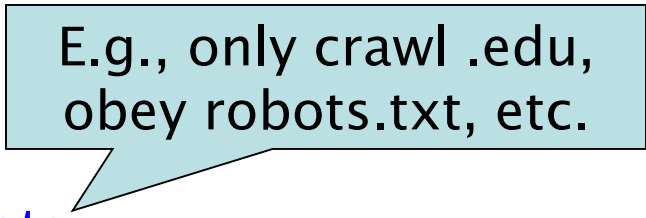
    at different frequencies

    (which ones are the fastest changing?)
  - Pages are removed
- Crawling can be a continuous process
  - When you get to the end of the queue, start again

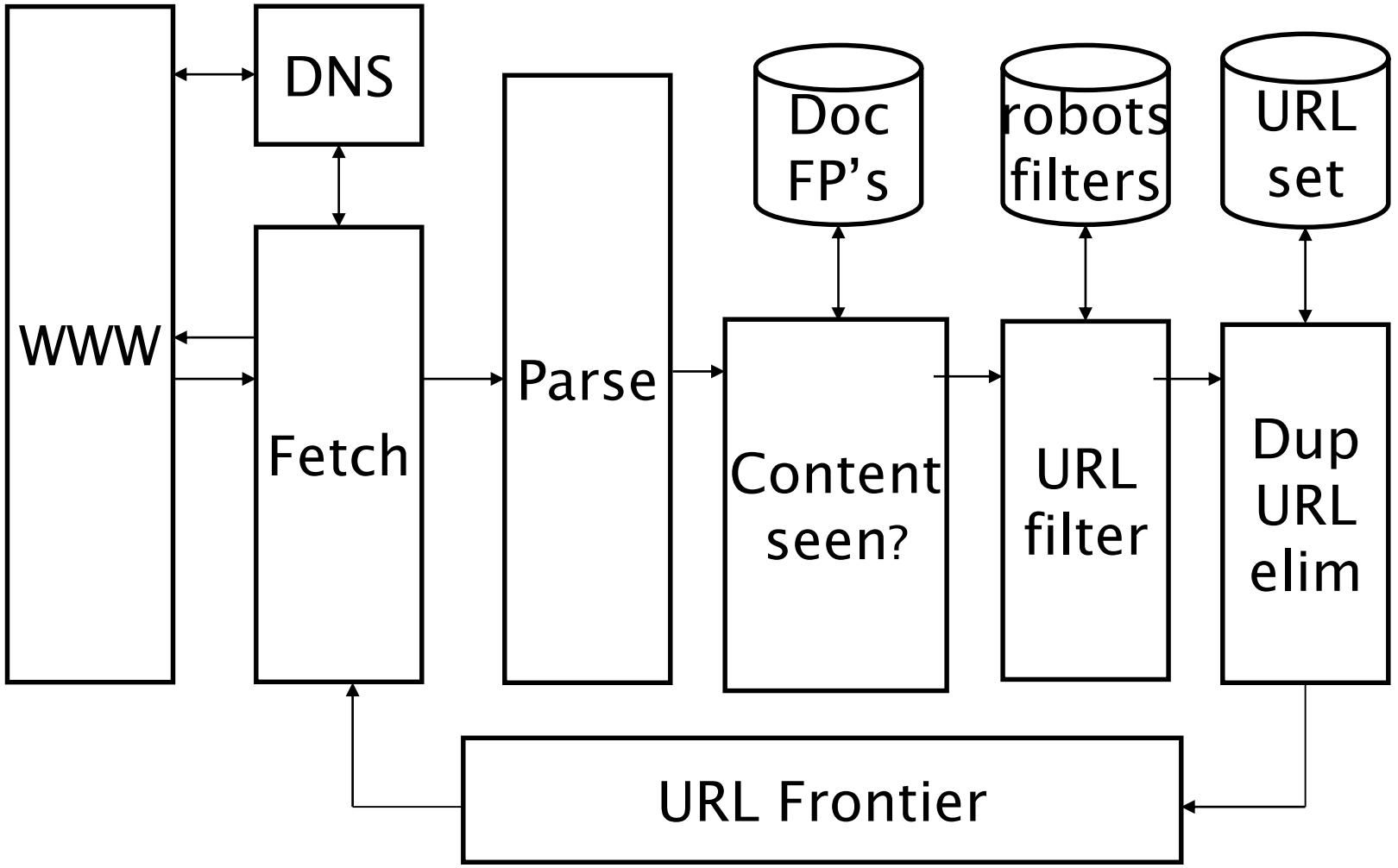
Summary of Crawling Process

- Pick a URL from the frontier
- Fetch the document at the URL
- Check if content already seen
- If not then
  - Extract links from it to other docs (URLs)
  - Save document for indexing
- For each extracted URL
  - Ensure it passes certain URL filter tests
  - Check if it is already in the frontier (duplicate URL elimination) before adding to frontier





Crawl architecture

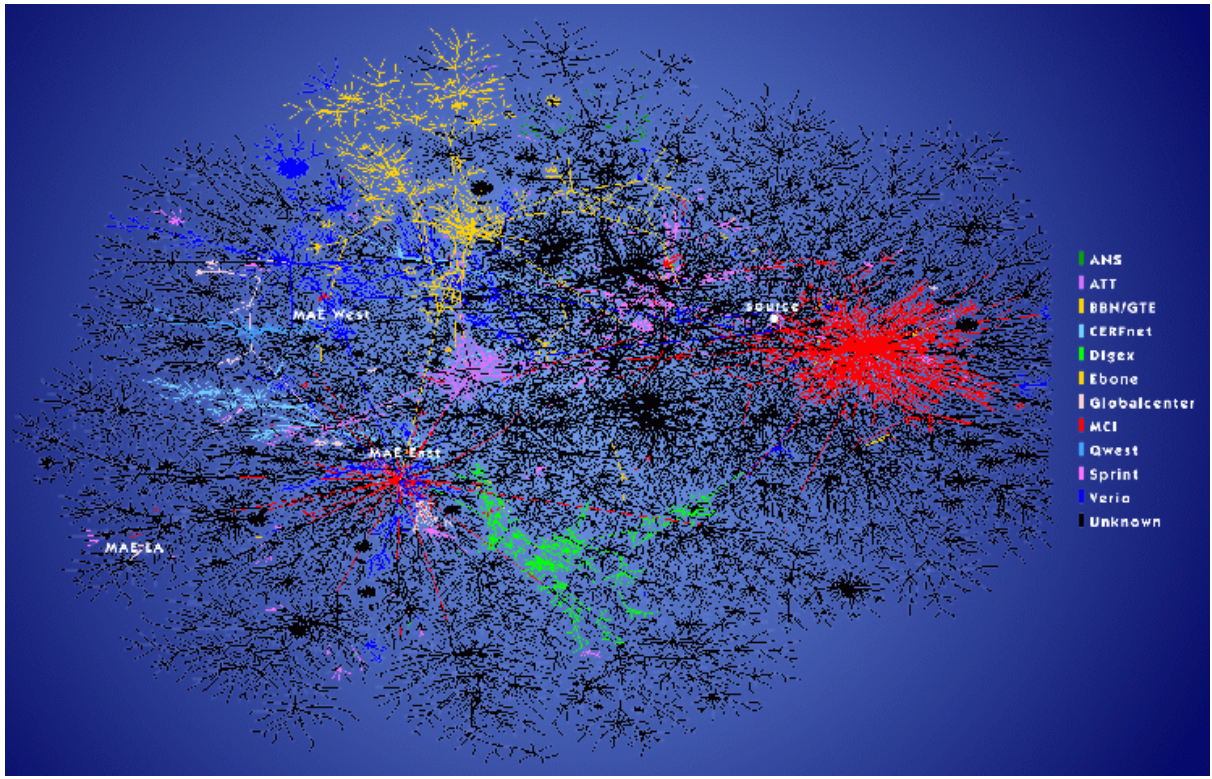


Doc FP = "fingerprint"
to test equivalence of pages

Figure from Manning et al 2008

How much of the Web is actually Crawled?

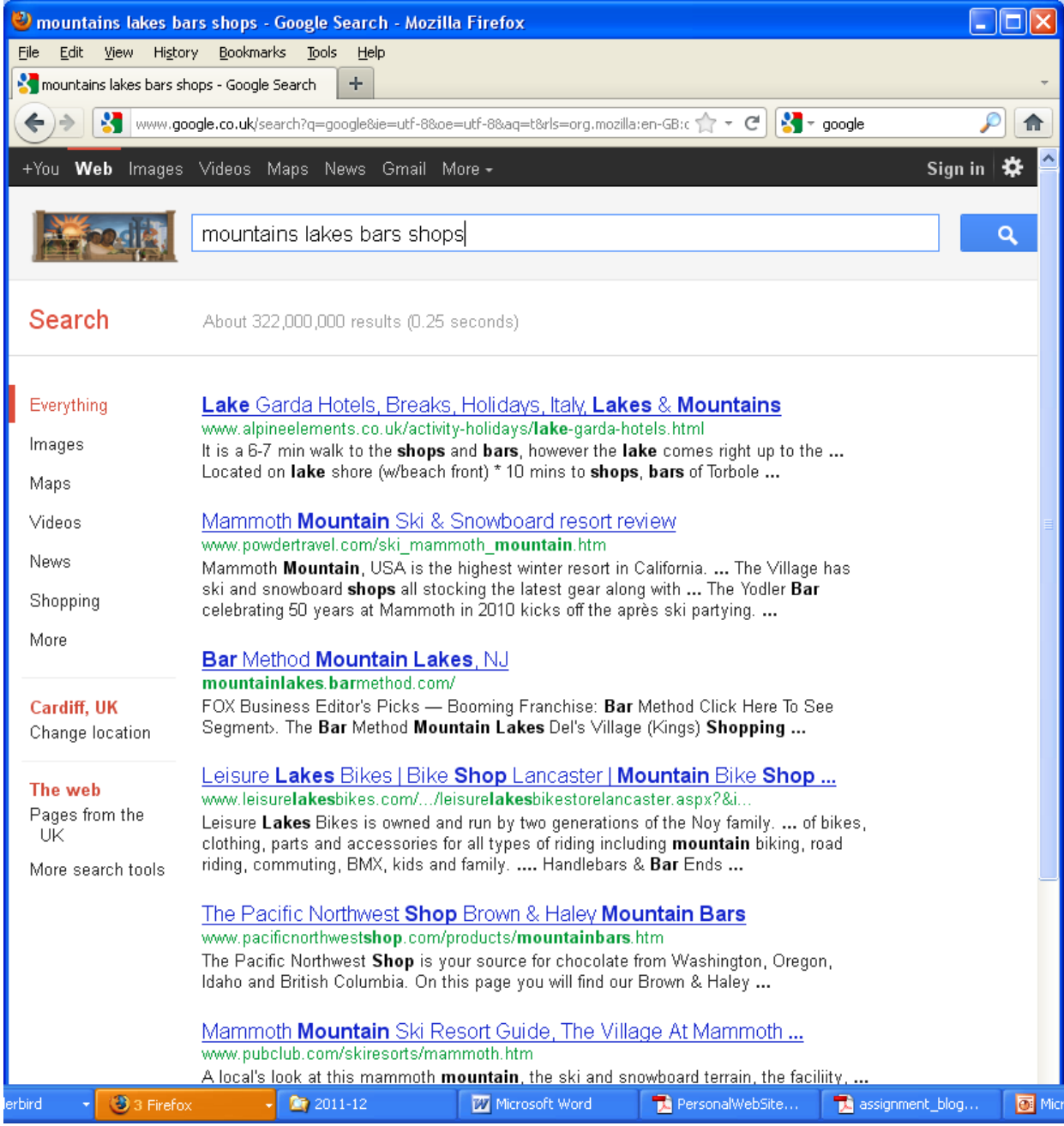
- The part of the Web that search engines know about or choose to index
- Only a fraction of the “Deep Web”
- Mostly HTML pages but some other file types too: PDF, Word, PPT, etc.



Indexing Documents

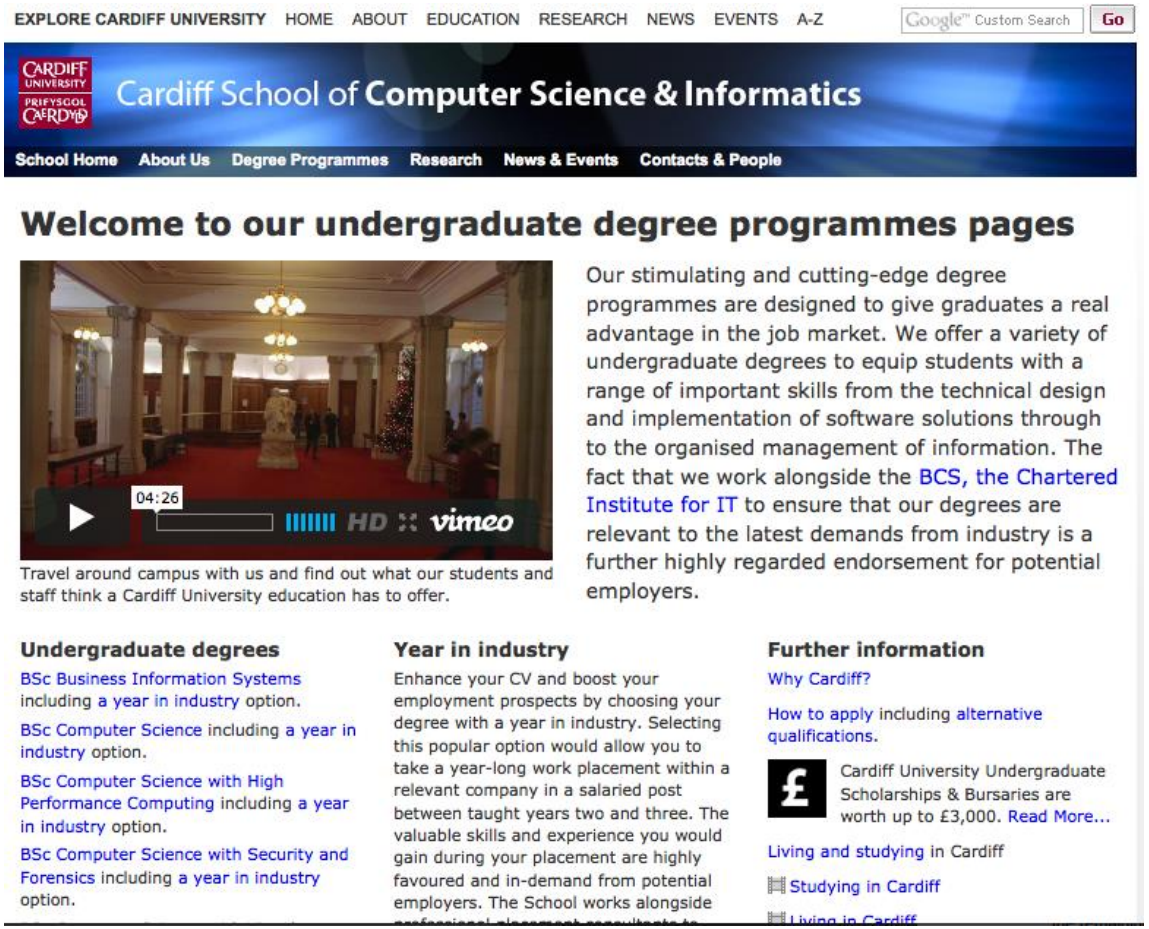
Objective:

Given some search terms, find relevant documents that contain them



What information should be indexed?

- ? All the words in the page ? + associated offset info
- ***What else?***

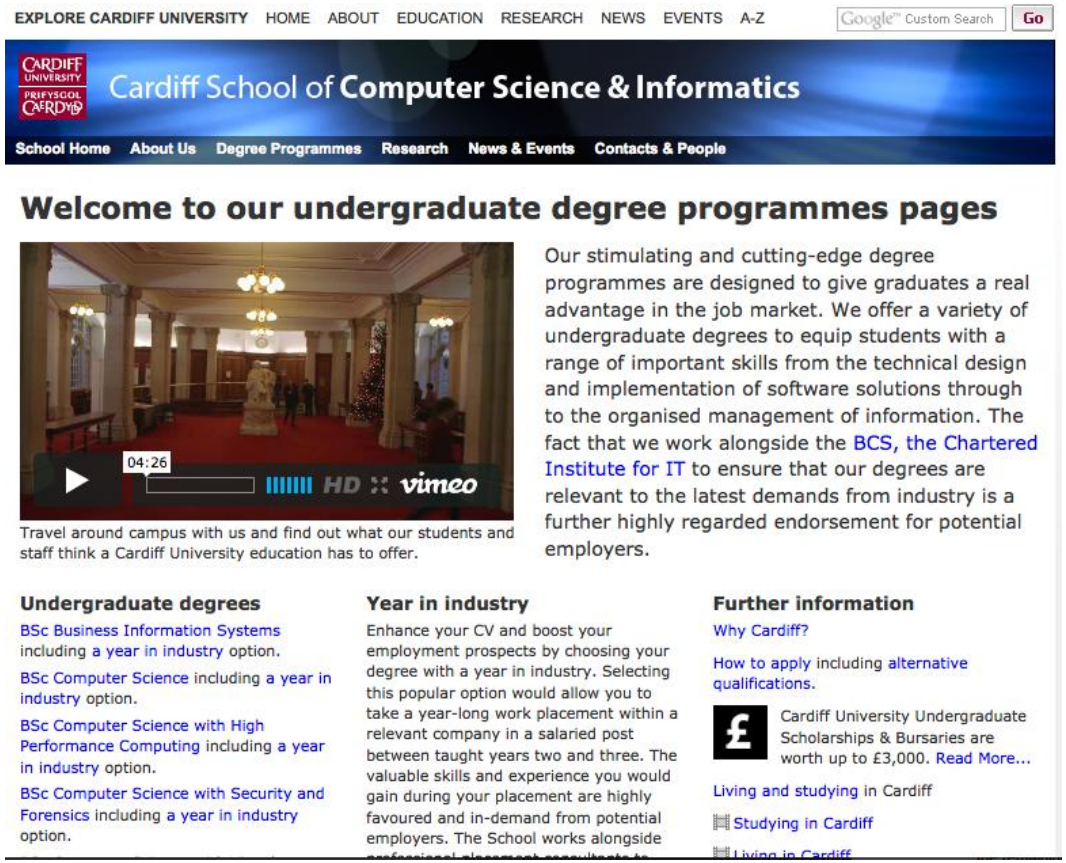


An **Inverted Index**

Associate words in a ‘dictionary’ with the documents they occur in

For each word,
list all the documents it occurs in.

stemming
for imprecise match
- cats -> cat[s]
- running -> run[ning]



Inverted Index Simple Example

Documents Ti = DocIDs

T1: The cow jumped over the moon

T2: How now blue cow

T3: Now the moon shines on the blue cow

How do we find which documents contain all of the terms:
“blue” “moon” “cow” ?

{2, 3} {1, 3} {1, 2, 3}

Terms	DocIDS
the	{1, 3}
cow	{1, 2, 3}
jumped	{1}
over	{1}
moon	{1, 3}
how	{2}
now	{2, 3}
blue	{2, 3}
shines	{3}
on	{3}

Inverted Index
Example with word offsets

- Offsets record where each word occurs in a document
  - pairs are document number + word number in doc
  - “jumped": {(1, 3)} means “jumped" is the 3rd word in document T1

phrases

How do we find documents with the phrase“now blue cow" ?

the	{ (1,1), (1,5), (3,2), (3,6) }
cow	{ (1,2), (2,4), (3,8) }
jumped	{ (1,3) }
over	{ (1,4) }
moon	{ (1,6), (3,3) }
how	{ (2,1) }
now	{ (2,2), (3,1) }
blue	{ (2,3), (3,7) }
shines	{ (3,4) }
on	{ (3,5) }

T1: The cow jumped over the moon
T2: How now blue cow
T3: Now the moon shines on the blue cow

Dictionaries/lexicons and Postings

- The set of terms is referred to as a Dictionary
- The Dictionary is sorted alphabetically *[But NOT in example below!]*
- The list of DocIDs is called a Postings List (sorted by ID)

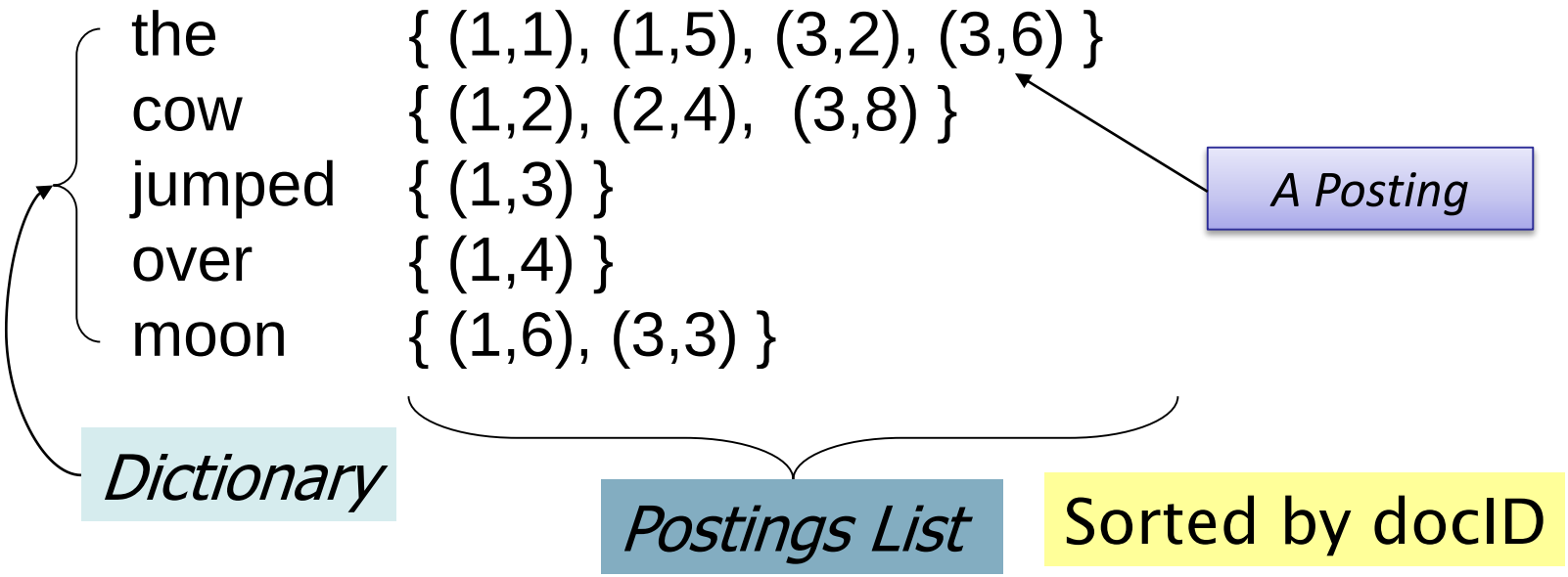
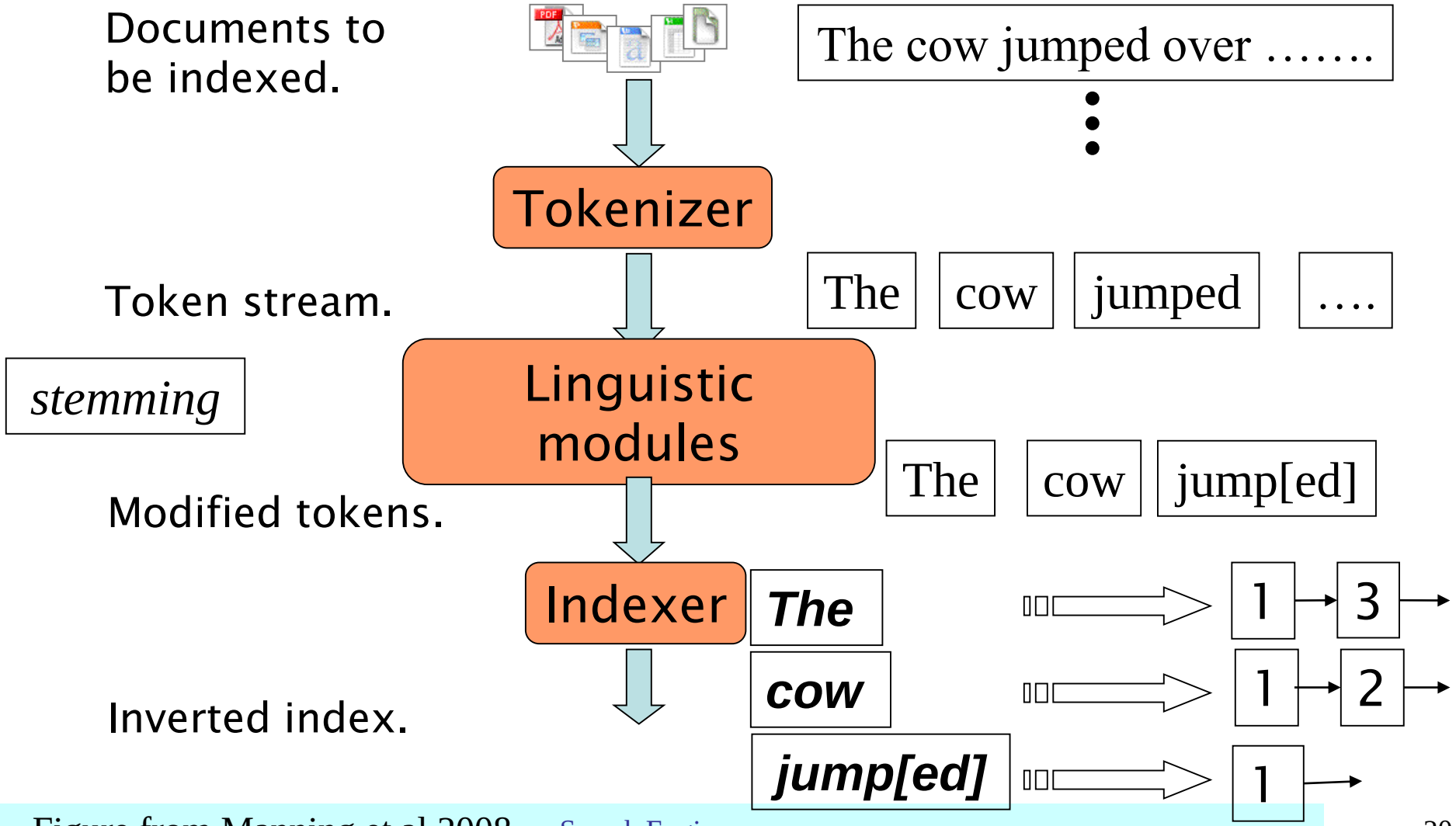


Figure from Manning et al 2008

Constructing the index



Building a sorted index

Doc 1

The cow jumped over the moon

Doc 2

How now blue cow

Doc 3

Now the moon shines on the blue cow

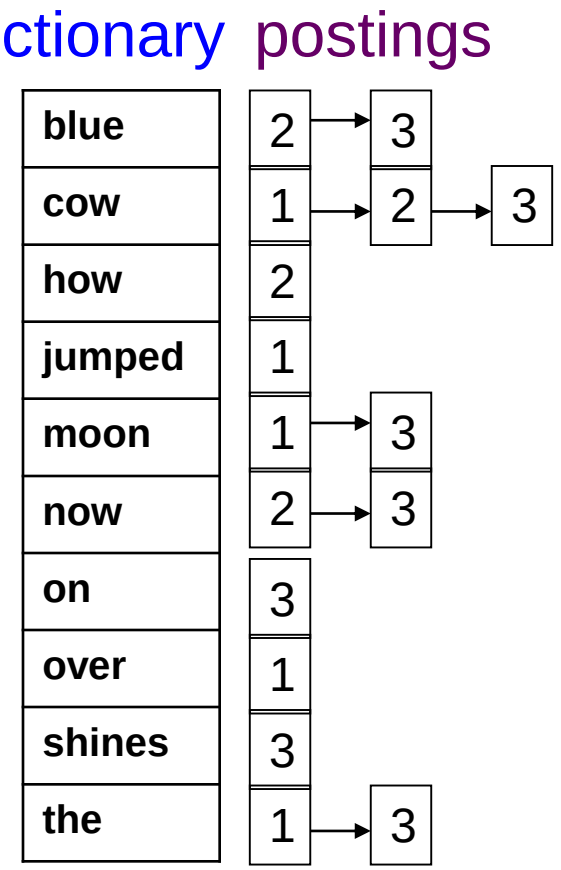
Term	docID
the	1
cow	1
jumped	1
over	1
moon	1
how	2
now	2
blue	2
cow	2
now	3
the	3
moon	3
shines	3
on	3
blue	3
cow	3

Term	docID
blue	2
blue	3
cow	1
cow	2
cow	3
how	2
jumped	1
moon	1
moon	3
now	2
now	3
on	3
over	1
shines	3
the	1
the	3

Merging terms

- Multiple term entries in a single document are merged.
- Split into Dictionary and Postings
- Doc. frequency information and offsets could also be added

Term	docID
blue	2
blue	3
cow	1
cow	2
cow	3
how	2
jumped	1
moon	1
moon	3
now	2
now	3
on	3
over	1
shines	3
the	1
the	3

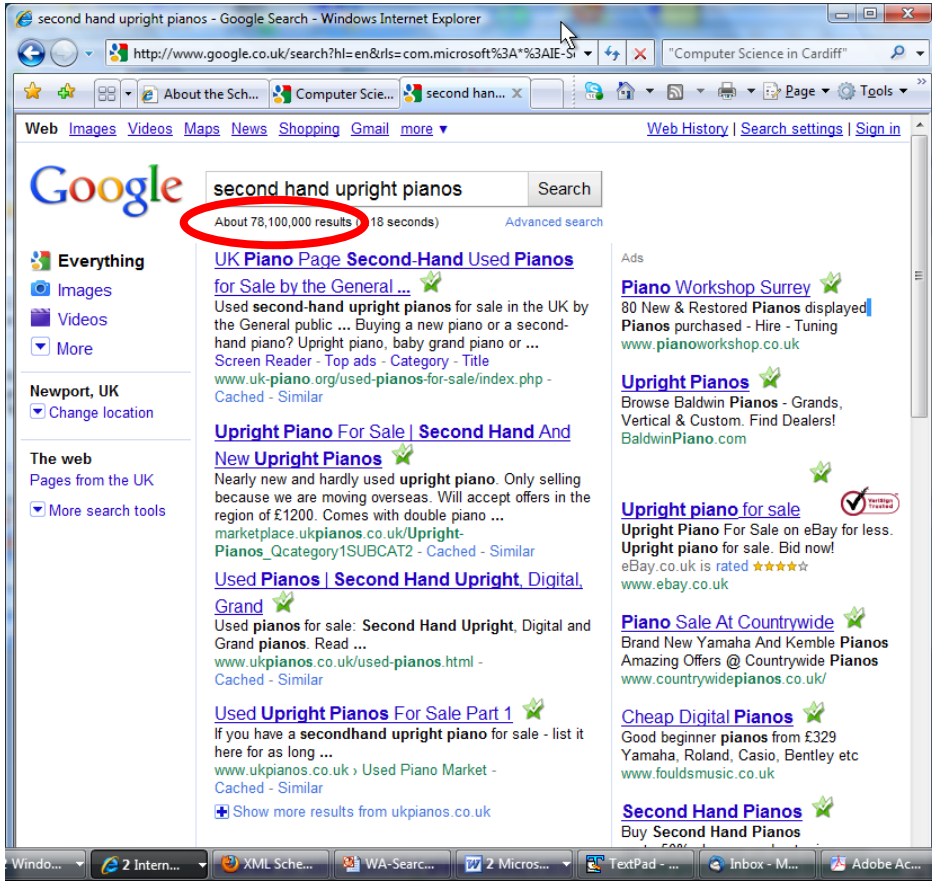


Inverted Index

- In reality, this index is HUGE
- May need to store the contents across multiple machines
  - Multiple versions (replication) of index with load balancer (to choose which one to use).
  - For each full index, partition documents into sub-indexes
    - Query all of them, and merge the results

Results Ranking

- For a given query may be thousands of documents that contain all the search terms
- Different search engines use different methods of ranking the results
  – not published in detail

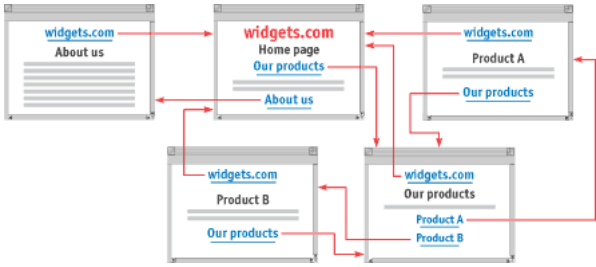


Some ranking criteria

For a given candidate result page, could use:

- Frequency of query terms on the page (term frequency ***tf***) and in general (document frequency ***df***)
- Proximity of matching words to one another (if query in quotes they must be together in the specified order)
- Location of terms within the page
- Location of terms within tags e.g. <title>, <h1>, anchor <a> text, meta tags…
- Anchor text on pages pointing to this one
- How many pages point to this one (*cf* PageRank)
- Click-through analysis: how often the page is clicked on
- How “fresh” is the page.
- +++*…???????*……..

PageRank Algorithm and the Importance of Linking

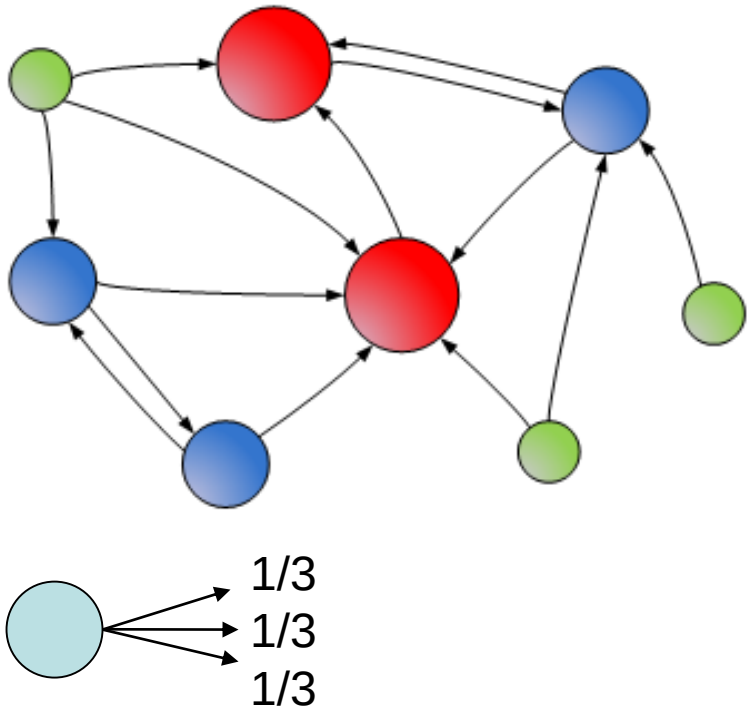


- Inspired by idea that relevance can be judged by *meta information* – i.e. not just the actual content of the document
  - *important pages are likely to be pointed to (linked) by many other pages –*
    so a link to a page is regarded as a vote for it.
  - *If the pointing page is 'important' then that strengthens its vote*
  - If the pointing page points to few pages then the importance of the destination / linked pages is greater

PageRank models a Random Surfer

The page rank PR(A) of a page A is probability that a "random surfer" would visit it

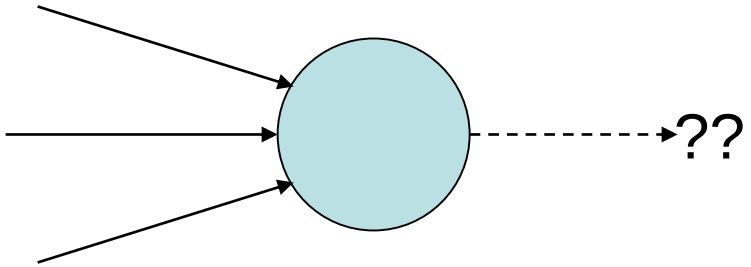
- Consider web pages to be nodes in a graph of N nodes where edges correspond to hyperlinks between pages
- A surfer at node A will proceed to any of the linked pages with probability 1/C where C is number of outgoing links



a page with many links to it is more likely to be visited

Teleporting

- If node A has no outgoing links the surfer moves randomly (teleports) to other nodes with probability *1/N*
- For nodes with outgoing links there is also a probability of teleporting to any other node (i.e. not just linked pages) with probability *a* (e.g. 0.1)



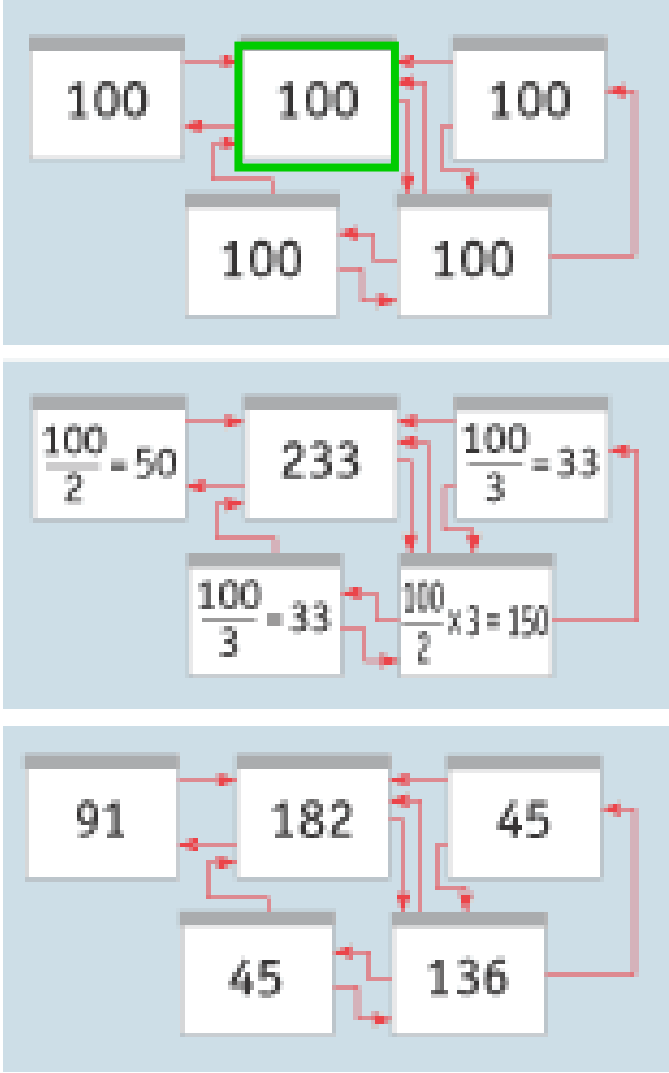
This modifies the *1/C* probability

PR(A) represents probability that a page is visited

PageRank cont.

Eg.: each page starts with PR= 100 points.

- The score PR(*A*) of a page *A* is recalculated by adding up scaled scores from each incoming link T_i .
  - This is the score PR(T_i) of the linking page T_i divided by its number of outgoing links C(T_i).
  - E.g, the page in green has 2 outgoing links and so its "points" are shared evenly by the 2 pages it links to.



- Keep repeating the score updates until no more changes.

$$PR(A) = (1-p) + p(PR(T1)/C(T1) + PR(T2)/C(T2) \ldots + PR(Tn)/C(Tn))$$

Where C(Ti) is outgoing links from Ti; p is (1 - teleporting probability)

Image from http://www.economist.com/science/tq/displayStory.cfm?story_id=3172188

Publication of PageRank

- Google founders Sergey Brin and Lawrence Page described the PageRank methods in:

The Anatomy of a Large-Scale Hypertextual Search Engine

http://ilpubs.stanford.edu:8090/361/

Spam technologies to manipulate ranking

Cloaking

Two versions of doc –

If SE robot, give page that will increase ranking, else provide the normal doc

Doorway pages

Pages optimized for a single keyword re-direct user to the real target (commercial) page (e.g. using meta refresh property)

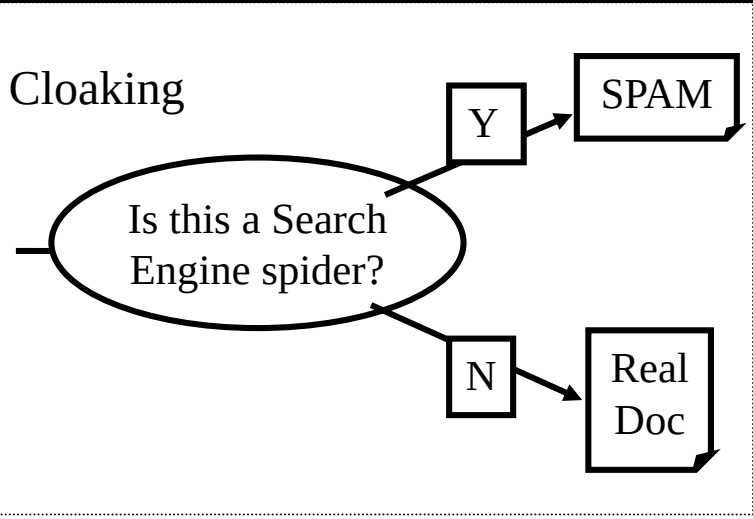
Keyword Spam

Misleading meta-keywords, excessive repetition of a term, engineered "anchor text"

Hidden (e.g. repetitive) text in background colours

Link spamming (→link farms)

Get multiple web pages to point to target



Meta-Keywords =
"… London hotels, hotel, holiday inn, hilton,
discount, booking, reservation,
sex, mp3,
britney spears, viagra, …"

This slide is based on an unknown source slide…

References

- Introduction to Information Retrieval. C. D. Manning, P. Raghavan, H. Schütze, Cambridge University Press. 2008.
- Managing Gigabytes: Compressing and Indexing Documents and Images. I.H. Whitten, A. Moffat, T.C. Bell, 1999.
- Modern Information Retrieval. Addison Wesley. 1999. R. Baeza-Yates, B. Ribeiro-Neto.
- http://news.netcraft.com/
- http://www.searchenginehistory.com/
- http://www.worldwidewebsize.com/
- http://www.searchengineland.com
- http://www.searchenginewatch.com
- http://www.searchenginejournal.com