I 202: Information Organization & Retrieval
Fall 2025

Class 22: How Web Search Works (IR intro, crawling, indexing)

Today’s Outline

- What is IR?
- Web Crawling
- Inverted Index
- Long Tail / Zipf’s Law

What is Information Retrieval (IR)?

"Information retrieval deals with the representation, storage, organization of, and access to information items."

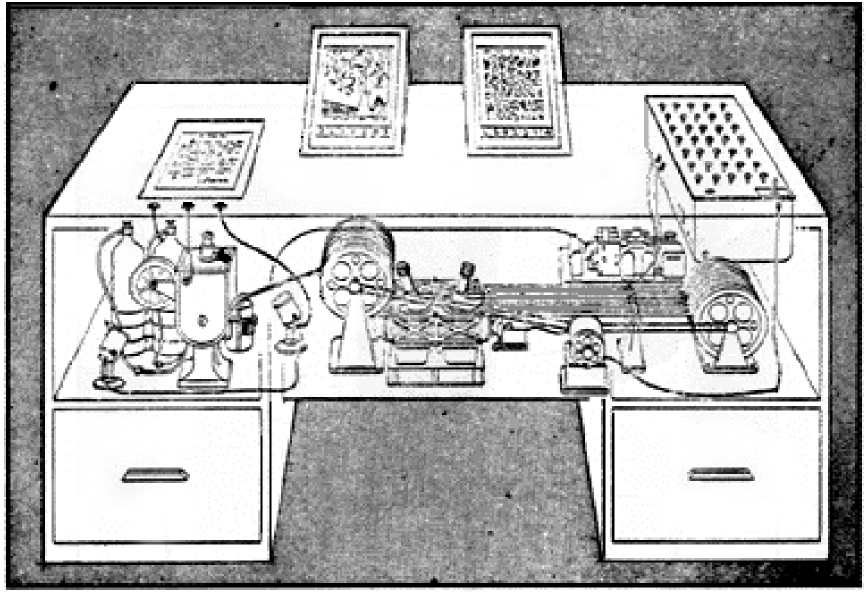
Chapter 1, Modern Information Retrieval, Baeza-Yates and Ribeiro-Neto 1999

IR History

- The “information overload” problem is not new!
- Origins in period immediately after World War II
  - *Tremendous scientific progress during the war*
  - *Rapid growth in amount of scientific publications available*
- The “Memex Machine”
  - *Conceived by Vannevar Bush, President Roosevelt's science advisor*
  - *Outlined in 1945 Atlantic Monthly article titled “As We May Think”*
  - *Foreshadows the development of hypertext (the Web) and information retrieval system*

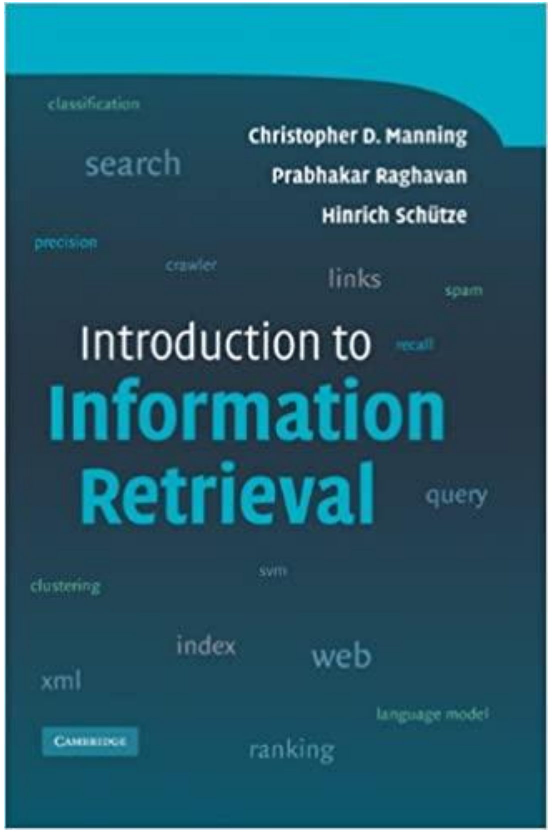
The Memex Machine

Vannevar Bush, 1945

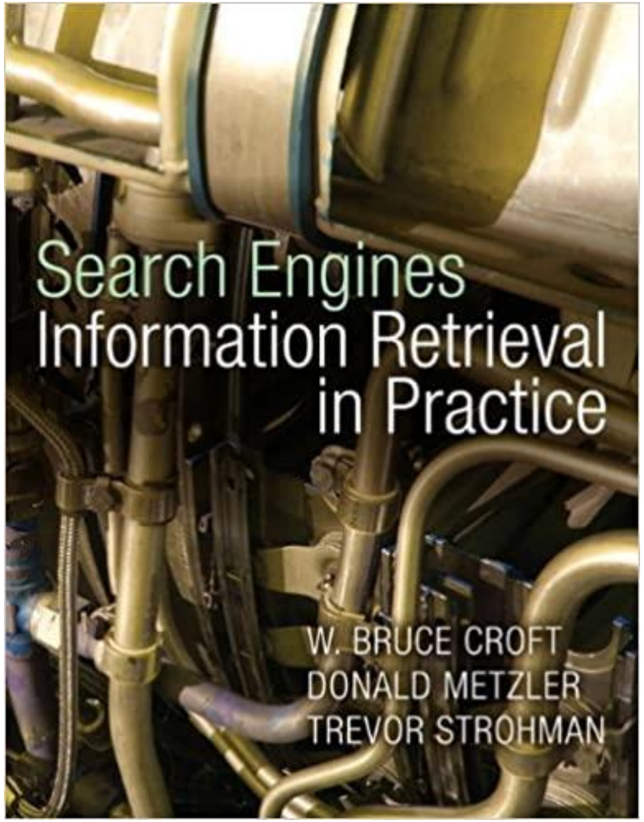


Memex in the form of a desk would instantly bring files and material on any subject to the operator's fingertips. Slanting translucent viewing screens magnify supermicrofilm filed by code numbers. At left is a mechanism which automatically photographs longhand notes, pictures and letters, then files them in the desk for future reference (*LIFE 19*(11), p. 123).

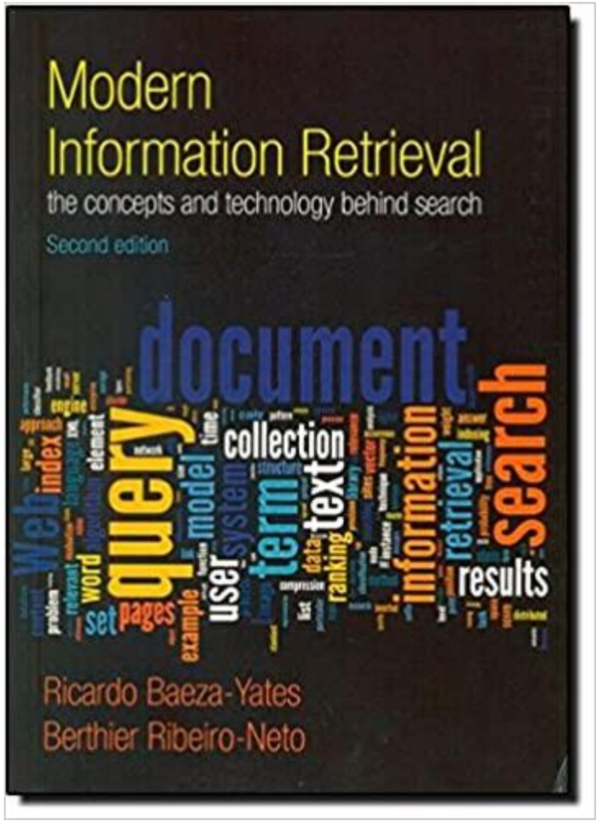
Information Retrieval Textbooks



2009

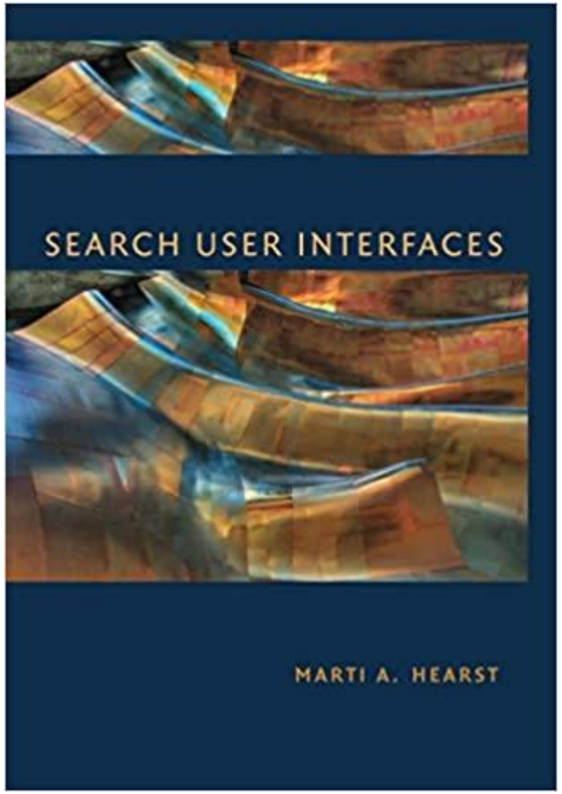


2010

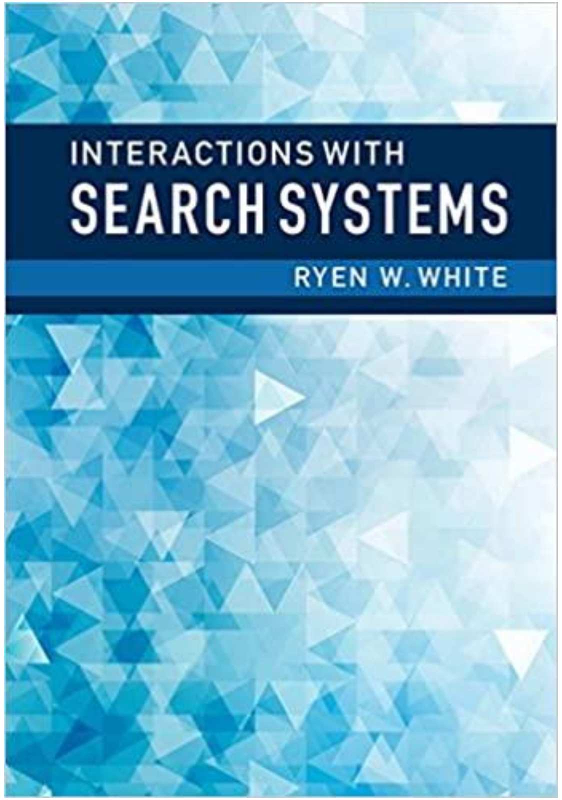


2011

Search Interface Textbooks (academic)

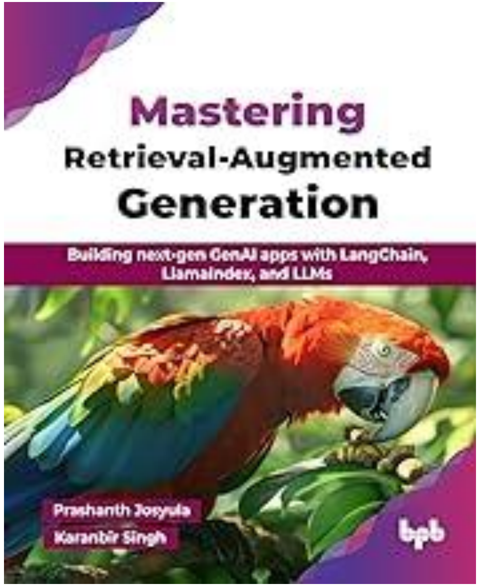


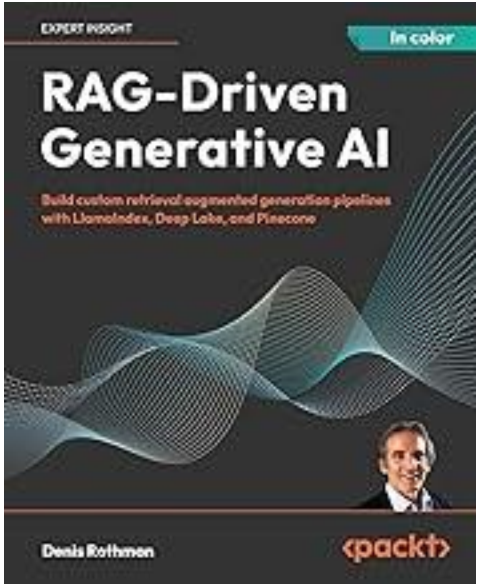
2009

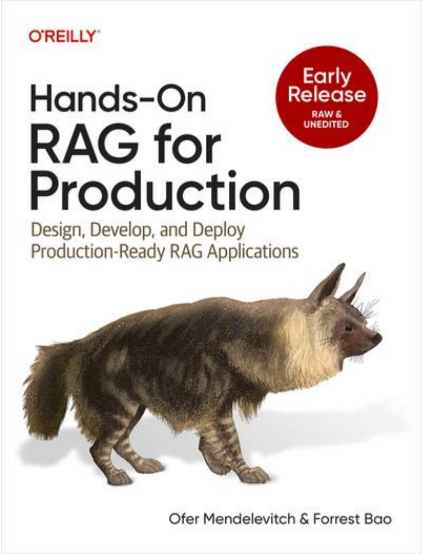


2016

Retrieval Augmented Generation
(I haven't read these … might not be good!)







2025

2024

DBMS vs. Information Retrieval

	Databases	IR
What we're retrieving	Structured data. Clear semantics based on a formal model.	Mostly unstructured. Free text with some metadata.
Queries we're posing	Formally (mathematically) defined queries. Unambiguous.	Vague, imprecise information needs (often expressed in natural language).
Results we get	Exact. Always correct in a formal sense.	Sometimes relevant, sometimes not.
Interaction with system	One-shot queries.	Interaction is important.
Other issues	Concurrency, recovery, atomicity are all critical.	Not usually relevant.

Discuss: How do Web Search Engines Work?

How Search Engines Work

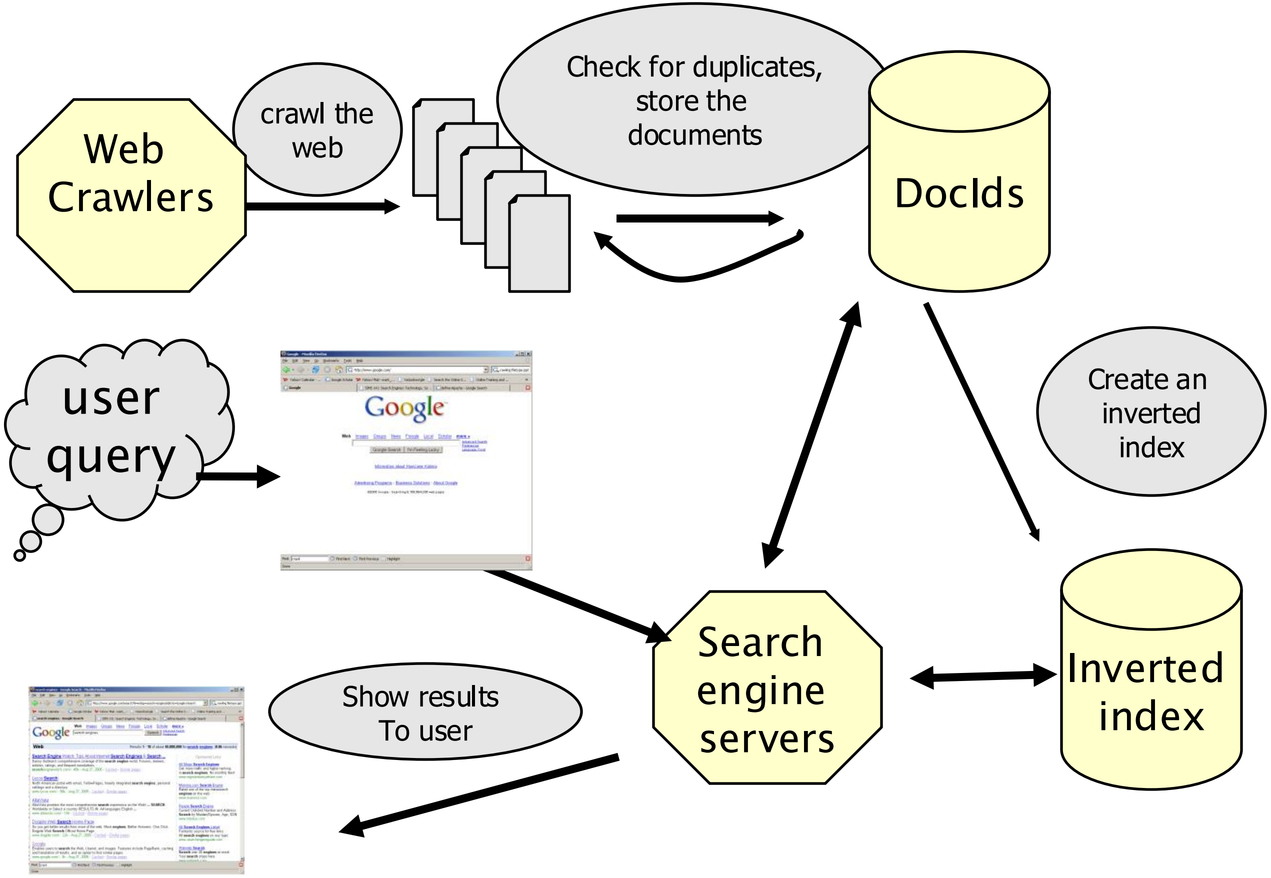
Three main parts:

i. Gather the contents of all web pages (using a program called a **crawler** or **spider**)

ii. Organize the contents of the pages in a way that allows efficient retrieval (**indexing**)

iii. Take in a query, determine which pages match, and show the results (**ranking** and **display** of results)

Standard Web Search Engine Architecture



The Web Is Enormous

WorldWideWebSize.com
DAILY ESTIMATED SIZE OF THE WORLD WIDE WEB

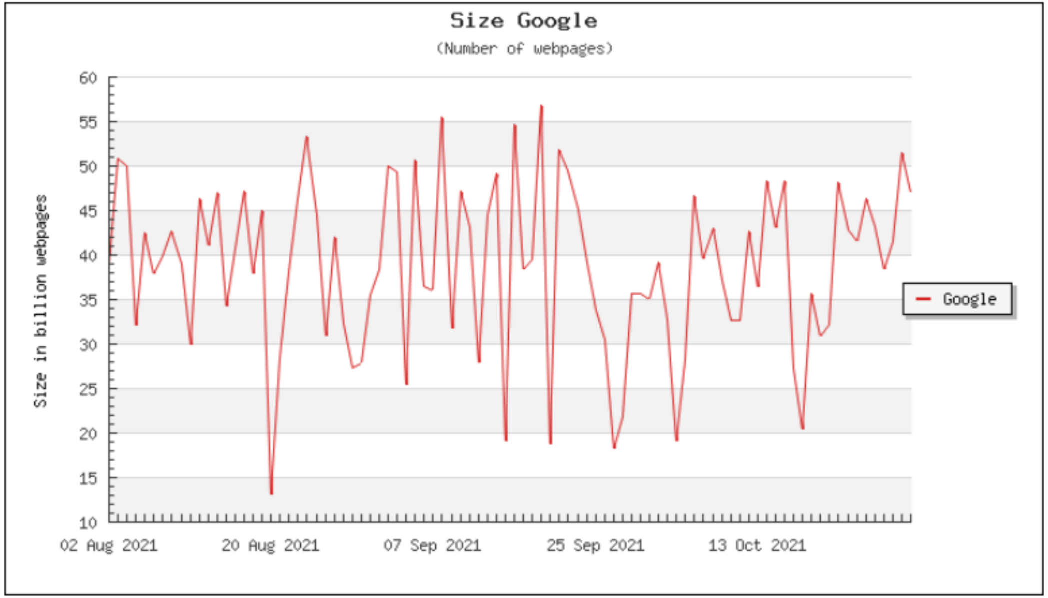
The size of the World Wide Web (The Internet)

The Indexed Web contains **at least 3.65 billion pages** (Sunday, 31 October, 2021).
The Dutch Indexed Web contains **at least 1600.64 million pages** (Sunday, 31 October, 2021).

Google

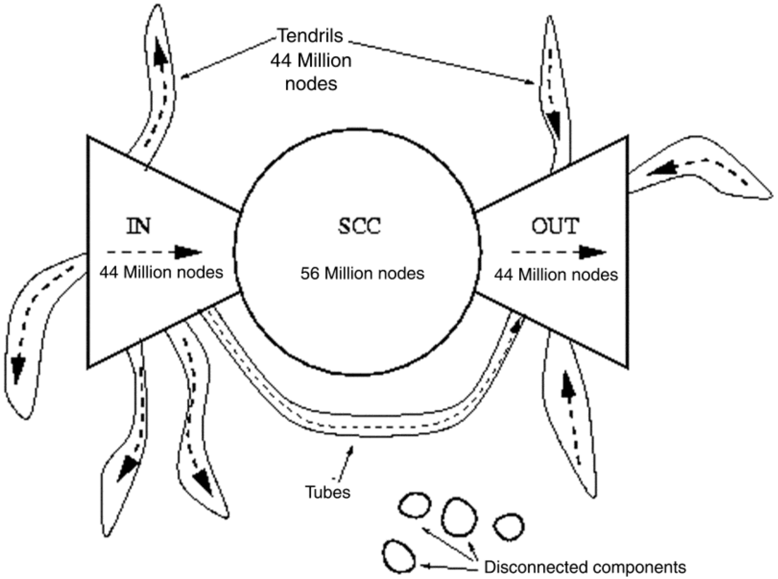
The size of the World Wide Web:
Estimated size of Google's index

Last Month | Last Three Months | Last Year | Last Two Years | Last Five Years



https://www.worldwidewebsize.com/

"The Web is a Bow-Tie", 2000



Broder, Andrei, et al. "Graph structure in the web." *Computer networks* 33.1-6 (2000): 309-320.

[*Nature*]() **volume 405**, page113 (2000)

personalized search needs may be met by dedicated science search portals. These webs within the web will concentrate many of the online resources you need within an easily navigable environment.

The various online repositories of the scientific literature may by then have adopted common standards to allow seamless searching across them. And the parallel development of 'intelligent' search software could mean that, as you type e-mails or word processing documents, your computer automatically delivers suggestions about relevant web resources, tailored to your particular interests and expertise.

Quality, not quantity

Portals are a hot topic on the web at the moment. The idea is to organize related content so that it can be searched in isolation from the web as a whole. This approach trades off the sheer scale of the available content against quality and ease of navigation. It also allows search engines that are overwhelmed by the public web to perform excellently.

Popular search engines have cottoned on to the trend. The Hotbot engine, for example, lets users search only academic sites with a domain name ending in '.edu'. "Soon you will see a whole slew of search engines specializing in particular sectors," predicts Sridhar Rajagopalan of IBM's Almaden Research Center in San Jose, California.

For the present, however, the biggest innovation in search engine technology takes its inspiration from the citation analyses used on the scientific literature. Conventional search engines use algorithms and simple rules of thumb to rank pages based on the frequency of the keywords specified in a query. But a new breed of engines is also exploiting the structure of the myriad links between web pages. Pages with many links pointing to them — akin to highly cited papers — are considered as 'authorities', and are ranked highest in search returns.

This approach has been pioneered by Sergey Brin and Lawrence Page, two graduate students in computer science at Stanford University in California. In less than a year, their Google search engine has become the most popular on the web, yielding more precise results for most queries than conventional engines — and transforming the lives of its developers. "I haven't finished my PhD," says Brin. "I'm afraid to say I've been too busy with Google."

Google's algorithms rank web pages by analysing their hyperlinks in a series of iterative cycles. "We don't just look at the number

Boy wonders: Brin (right) and Page's Google search engine is the most popular on the web.

Nature, which again suggests that *Nature* is a more important authority."

Whereas most search engines only associate the text of a link with the page the link is on, Google also associates it with the page the link points to. This allows it to cover many more pages than it actually crawls, even yielding links to sites that bar search engines' crawler programs.

Clever, a prototype search engine being developed at IBM Almaden, takes the citation analogy further. Like Google, it produces a ranking of authorities, but it also generates a list of 'hubs', pages that have many links to authorities. Hubs are akin to review articles, which cite many top papers. Those that link to many of the most highly cited authorities are given higher rankings than those that link to less well-respected sites. "Not all links are equal," says Rajagopalan, the driving force behind Clever. Users get not only the top hits, but the hubs provide a good starting point for browsing.

However, search strategies that rely on analysing hyperlinks are no panacea. New pages will have few links to them, and may be missed. And such strategies may be of little use to a scientist seeking highly specific information found on specialist sites with few incoming links. "All search engines have their weaknesses for certain queries," says Brin.

Effective searching requires a mix of techniques. If you want to trawl for background information before beginning a research project, you might use an engine like Google to identify key sites, in combination with

The web is a bow tie

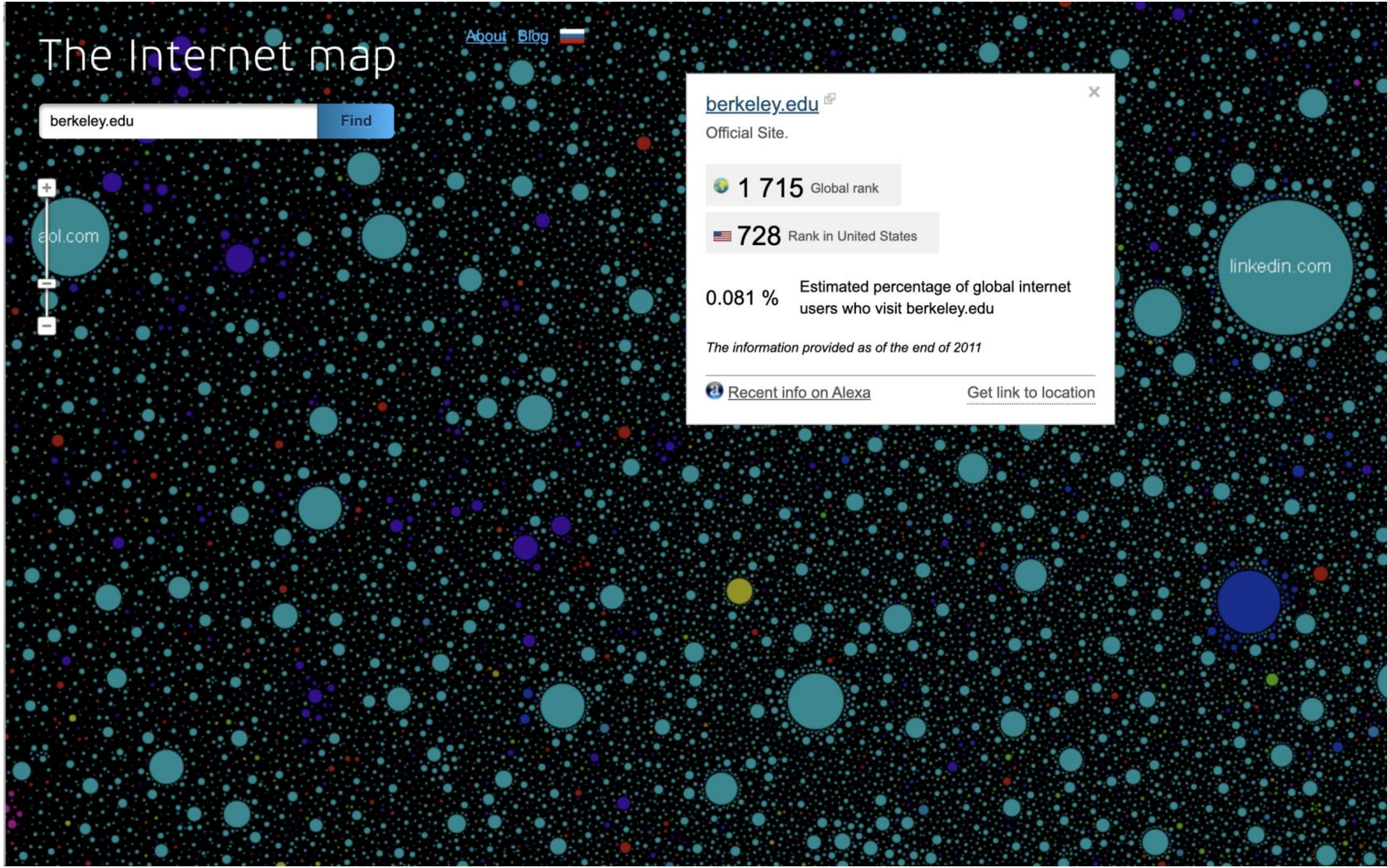
A study of the web's structure, five times larger than any attempted previously, reveals that it isn't the fully interconnected network that we've been led to believe. The study suggests that the chance of being able to surf between two randomly chosen pages is less than one in four.

Researchers from three Californian groups — at IBM's Almaden Research Center in San Jose, the Altavista search engine in San Mateo and Compaq Systems Research Center in Palo Alto — have analysed 200 million web pages and 1.5 billion hyperlinks. Their results, which will be presented next week at the World Wide Web 9 Conference in Amsterdam, indicate that the web is made up of four distinct components.

A central core contains pages between which users can surf easily. Another large cluster, labelled 'in', contains pages that link to the core but cannot be reached from it. These are often new pages that have not yet been linked to. A separate 'out' cluster consists of pages that can be reached from the core but do not link to it, such as corporate websites containing only internal links. Other groups of pages, called 'tendrils' and 'tubes', connect to either the in or out clusters, or both, but not to the core, whereas some pages are completely unconnected. To illustrate this structure, the researchers picture the web as a plot shaped like a bow tie with finger-like projections.

IN
44 million pages
Central core
56 million pages
OUT
44 million pages

The Web Is Enormous



Internet-map.net

i. Web Crawlers / Spiders

How to find web pages to visit and copy?

- Can start with a list of domain names, visit the home pages there.
- Look at the hyperlink on the home page, and follow those links to more pages.
- Keep a list of urls visited, and those still to be visited.
- Each time the program loads in a new HTML page, add the links in that page to the list to be crawled.

How does the World Wide Web Work?

- Internet vs WWW
- Server vs Router
- IP address vs Domain Name
- URL Structure and Network Protocols

How Does the World Wide Web Work?

sylvain kalache at wikicommons

Say a user named Oski using his computer at home (or in, say, Seoul) wants to find information about i202?

What happens when he:

Brings up a search engine home page?

Types his query?

First, we have to understand how the WWW works!

Then we can understand search engines.

Internet vs. WWW

- **Internet** and **Web** are not synonymous
- Internet is a global communication network connecting millions (billions?) of computers.
- World Wide Web (WWW) is one component of the Internet, along with e-mail, chat, etc.
- Now we'll talk about both.

Slide adapted from Lew & Davis

How Does the WWW Work?
(simplified explanation)

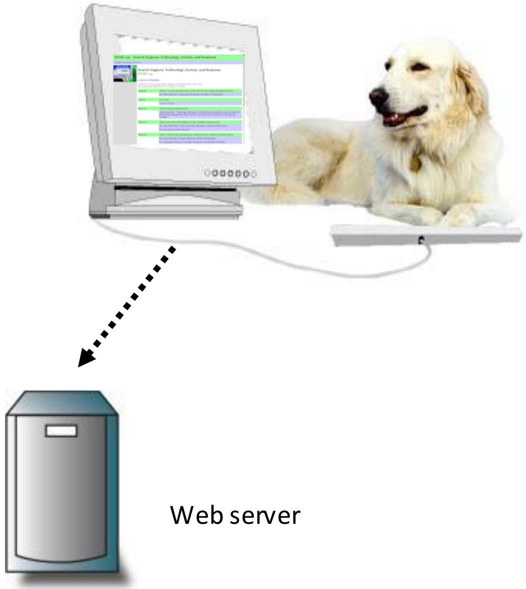
sylvain kalache at wikicommons

- Let's say Oski received email with the address for i202 (assume it is at ischool.berkeley.edu)
- He goes to a networked computer, and launches a web browser.
- He then types the address, known as a URL, into the address bar of the browser.
- What happens next?

(URL stands for Uniform Resource Locator)

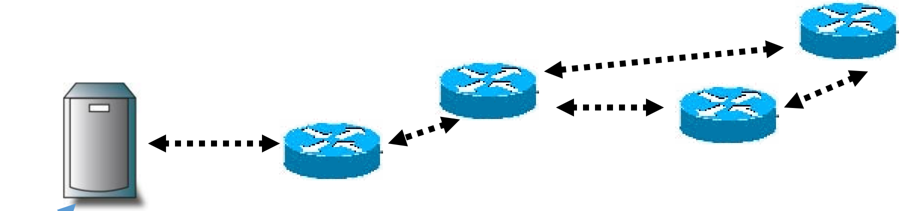
Uploading a Page

- Say Oski's instructor has written a web page on her laptop.
- She copies the page to a directory on a computer called herald, which is on the ischool local network.
- This computer is connected to the Internet and runs a program called Apache. This allows herald to act as a web server.



Routing Between Computers

- How does the computer at Oski's desk figure out where the i202 web pages are?
- In order for him to use the WWW, Oski's computer must be connected to another machine acting as a web server (via his ISP).
- This machine is in turn connected to other computers, some of which are routers.

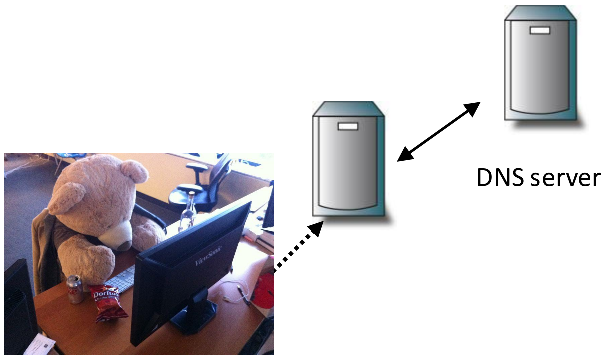


Routers figure out how to move information from one part of the network to another.

There are many different possible routes.

IP Address to Domain Name

- How do Oski's server and the routers know how to find the right server?
- First, the url has to be translated into a number known as an IP address.
- Oski's server connects to a Domain Names Server (DNS) that knows how to do the translation.



Converting Domain Names

- Domain names are for humans to read.
- The Internet actually uses numbers called **IP addresses** to describe network addresses.
- The Domain Name System (DNS) – resolves IP addresses into easily recognizable names
- For example:
  - *12.42.192.73* = *www.xyz.com*
- A domain name and its IP address refer to the same Web server.

Typical Domain Name

www.xyz.com

Server (host) name

Registered company domain name

Domain category (top-level domain)

Domain names are part of URLs, used in web pages.

Top-Level Domains

- com, biz, cc — commercial or company sites
- edu — educational institutions, typically universities
- org — organizations; originally meant for clubs, associations and nonprofit groups
- mil — U.S. military
- gov — U.S. civilian government
- net — network sites, including ISPs
- int — international organizations (rarely used)

Many other top-level domains are available

There is an interesting standards story about this!

Internet Addresses

- The internet is a network on which each computer must have a **unique address**.
- The Internet uses **IP addresses**; for example, herald's IP address is **128.32.78.23**
- Internet Protocol version 4 (IPv4) – supports 32-bit dotted quad IP address format
  - *Four sets of numbers, each set ranging from 0 to 255*
  - *UC Berkeley's LAN addresses range from 128.32.0.0 to 128.32.255.255*
  - *Other addresses in the iSchool LAN include* ***128.32.78.19***
- Using this setup, there are approximately 4 billion possible unique IP addresses
- Router software knows how to use the IP addresses to find the target computer.

IPV6 ADOPTION

Google IPv6

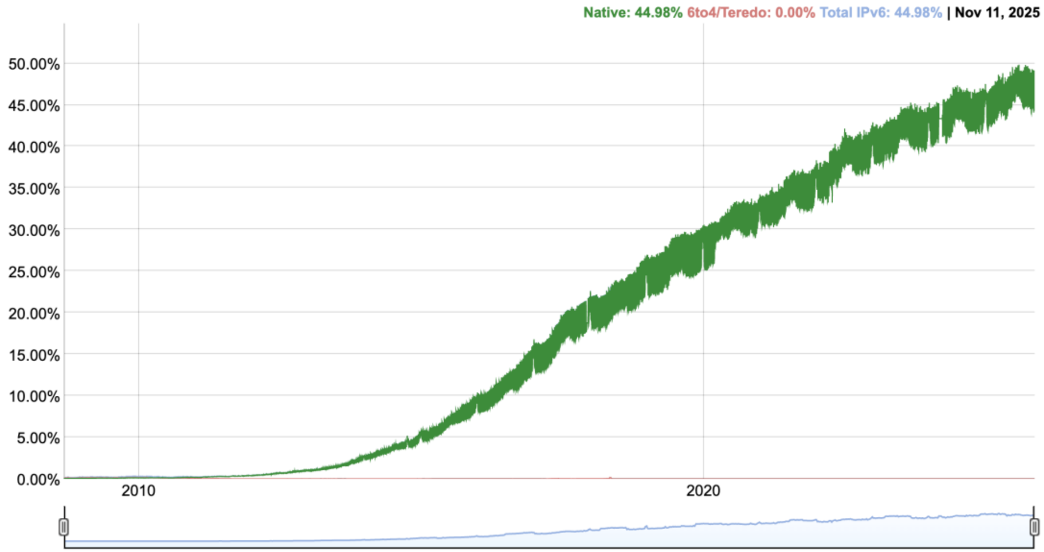
Statistics

Google collects statistics about IPv6 adoption in the Internet on an ongoing basis. We hope that publishing this information will help Internet providers, website owners, and policy makers as the industry rolls out IPv6.

IPv6 Adoption | Per-Country IPv6 adoption

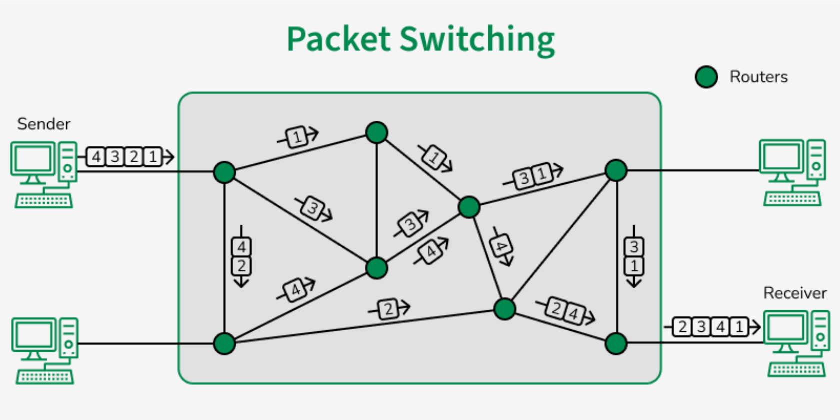
IPv6 Adoption

We are continuously measuring the availability of IPv6 connectivity among Google users. The graph shows the percentage of users that access Google over IPv6.



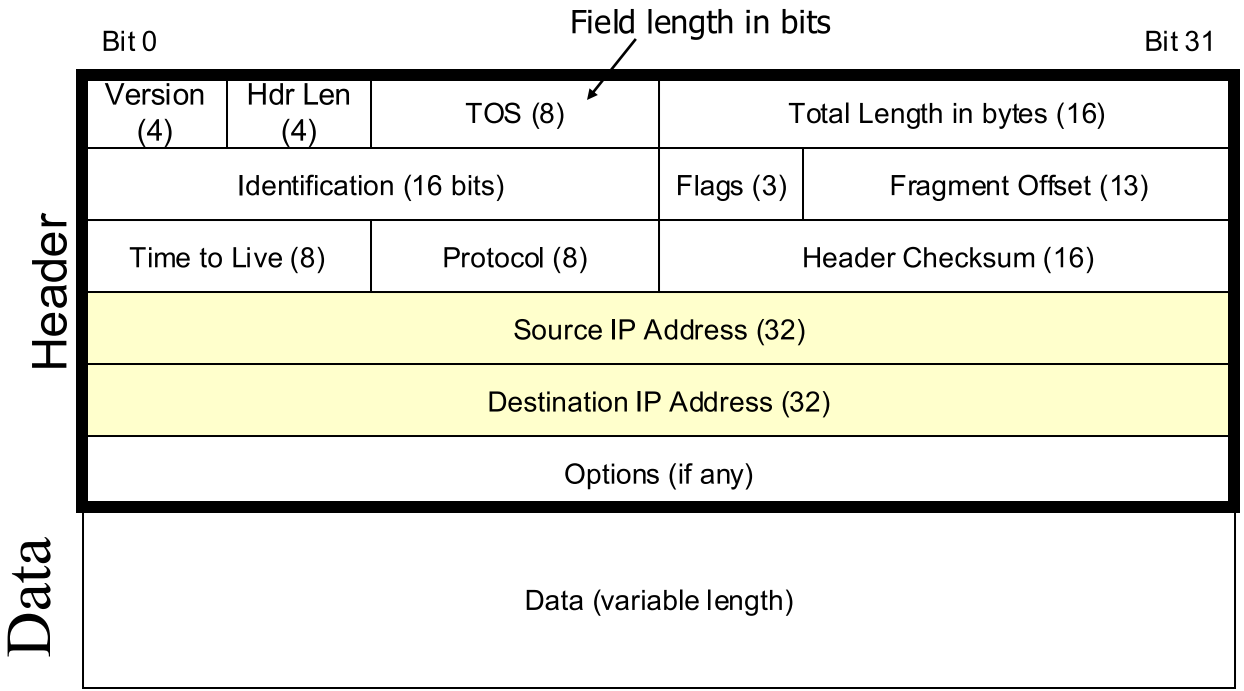
Network Protocols and Packets

- Network **Protocols**:
  - Protocol – an agreed-upon format for transmitting data between two devices
  - The Internet protocol is TCP/IP
  - The WWW protocol is HTTP
- Network **Packets**:
  - Typically, a message is broken up into smaller pieces and re-assembled at the receiving end.
  - These pieces of information, surrounded by address information, are called packets



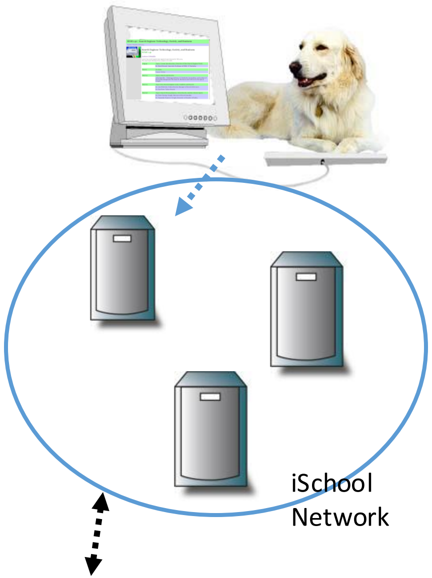
Slide adapted from CIW foundations
https://www.geeksforgeeks.org/computer-networks/difference-between-message-and-packet-switching/

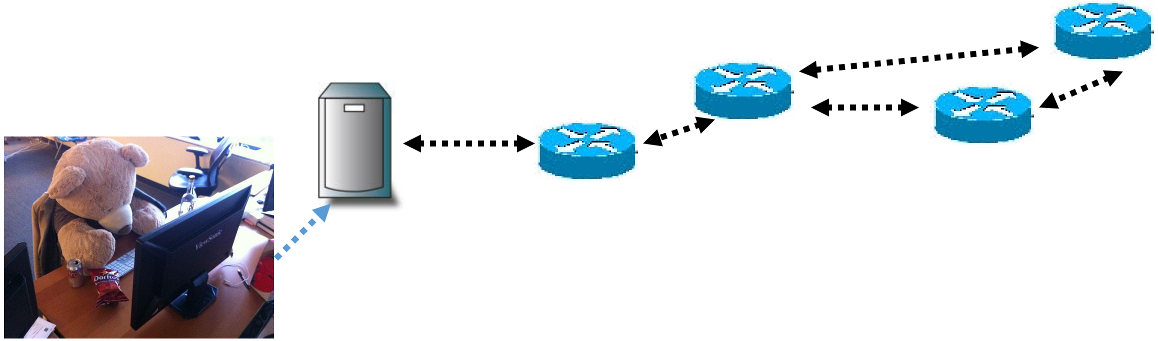
IP Packet Format (v4)



Using the URL

- What happens now that the request for information from Oski's browser has been received by the web server *herald* at www.ischool.berkeley.edu?
- The web server processes the url to figure out which page on the server is requested.
- It then sends all the information from that page back to the requesting address.





Reading a URL

http://courses.ischool.berkeley.edu/i202/f21/index.html

http:// = HyperText Transfer Protocol

courses = service name (often is www)

.ischool = host name

.berkeley = primary domain name

.edu/ = top level domain

i202/ = directory name

f21/ = directory names

index.html = file name of web page

Slide adapted from Lew & Davis

HTTP Request: Example

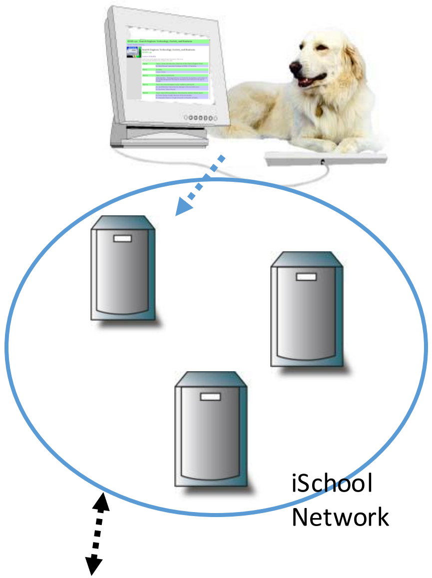
This information is received by the web server at www.ischool.berkeley.edu :

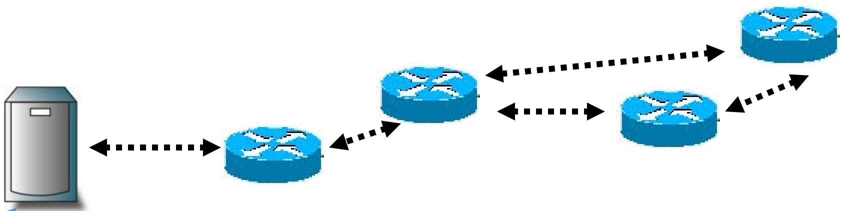
Request line	GET i202/f21/index.html HTTP/1.1<CRLF>
Request header	Host: courses.ischool.berkeley.edu <CRLF>
Blank line	<CRLF>

Because HTTP is built on TCP/IP, the web server knows which IP address to send the contents of the web page back to.

Serving a Web Page

- When Oski typed in the url for the i202 home page, this was turned into an HTTP request and routed to the web server in Berkeley.
- The web server then decomposed the url and figured out which web page in its directories was being asked for.
- The server then sends the HTML contents of the page back to Oski's IP address.





Oski's browser receives these HTML contents and renders the page in graphical form.

If he clicks on the hyperlink to the syllabus, a similar sequence of events will happen.

How Do You Think a Web Crawler Works?

Yogesh Gosavi, Unsplash

i. Web Crawlers / Spiders

How to find web pages to visit and copy?

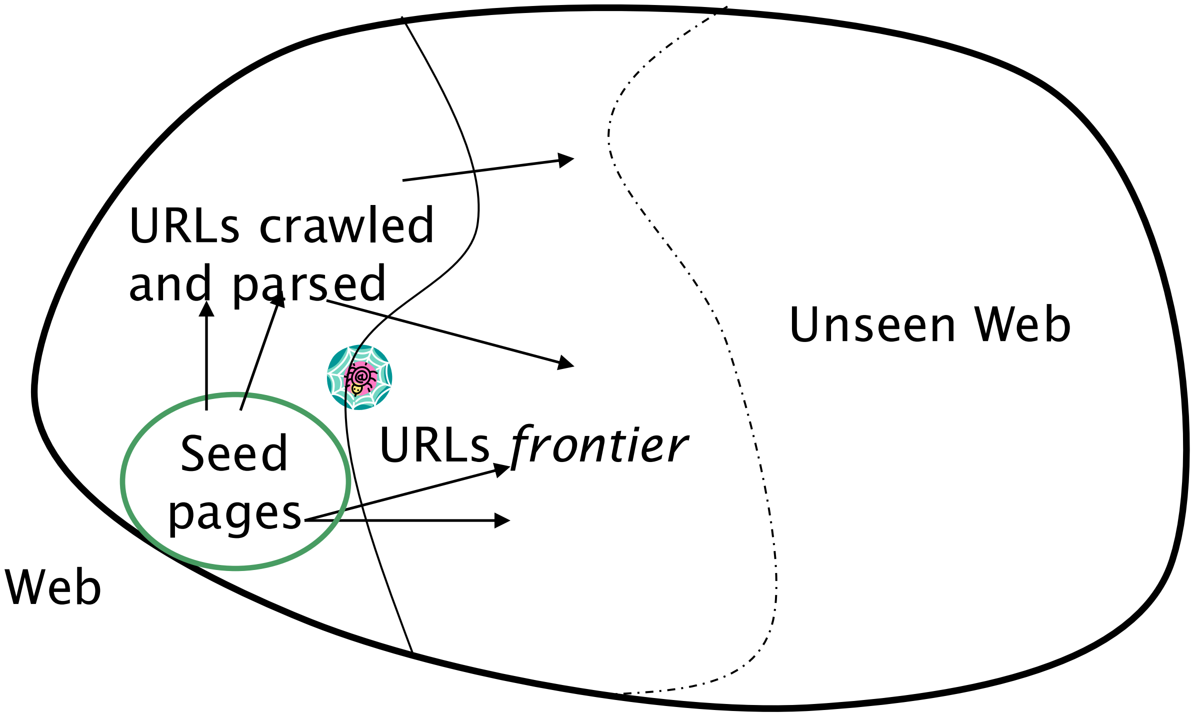
- Can start with a list of domain names, visit the home pages there.
- Look at the hyperlink on the home page, and follow those links to more pages.
- Keep a list of urls visited, and those still to be visited.
- Each time the program loads in a new HTML page, add the links in that page to the list to be crawled.

Slide adapted from Lew & Davis

Retrieving Web Pages

- Web crawler client program connects to a *domain name system* (DNS) server
- DNS server translates the hostname into an *internet protocol* (IP) address
- Crawler then attempts to connect to server host using specific *port*
- After connection, crawler sends an HTTP request to the web server to request a page (usually a GET request)

Crawling picture



WEB CRAWLING

- Web crawlers spend a lot of time waiting for responses to requests
- To reduce this inefficiency, web crawlers use threads and fetch hundreds of pages at once
- Crawlers could potentially flood sites with requests for pages
- To avoid this problem, web crawlers use *politeness policies*
  - *e.g., delay between requests to same web server*

FOUR "LAWS" OF CRAWLING

- A Crawler must show identification
- A Crawler must obey the robots exclusion standard

  Use the robots.txt file to give instructions to the crawler

  http://www.robotstxt.org/wc/norobots.html

- A Crawler must not hog resources
- A Crawler must report errors

```
User-agent: *
Disallow: /private/
Disallow: /confidential/
Disallow: /other/
Allow: /other/public/

User-agent: FavoredCrawler
Disallow:

Sitemap: http://mysite.com/sitemap.xml.gz
```

SPIDER BEHAVIOUR VARIES

- Parts of a web page that are indexed
- How deeply a site is indexed
- Types of files indexed
- How frequently the site is spidered

Slide adapted from Lew & Davis

Lots of tricky aspects

- Servers are often down or slow
- Hyperlinks can get the crawler into cycles
- Some websites have junk in the web pages
- Many pages have dynamic content
- The web is HUGE

"FRESHNESS"

- Need to keep checking pages
  - *Pages change*
    - At different frequencies
    - Who is the fastest changing?
    - Pages are removed
  - *Many search engines **cache** the pages (store a copy on their own servers)*

DEEP WEB

- Sites that are difficult for a crawler to find are collectively referred to as the *Deep* (or *hidden*) *Web*
  - *much larger than conventional Web*
- Three broad categories:
  - *private sites*
    - no incoming links, or may require log in with a valid account
  - *form results*
    - sites that can be reached only after entering some data into a form
  - *scripted pages*
    - pages that use JavaScript, Flash, or another client-side language to generate links

SITEMAPS

- Sitemaps contain lists of URLs and data about those URLs, such as modification time and modification frequency
- Generated by web server administrators
- Tells crawler about pages it might not otherwise find
- Gives crawler a hint about when to check a page for changes

Sitemap Example

```xml
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
  <url>
    <loc>http://www.company.com/</loc>
    <lastmod>2008-01-15</lastmod>
    <changefreq>monthly</changefreq>
    <priority>0.7</priority>
  </url>
  <url>
    <loc>http://www.company.com/items?item=truck</loc>
    <changefreq>weekly</changefreq>
  </url>
  <url>
    <loc>http://www.company.com/items?item=bicycle</loc>
    <changefreq>daily</changefreq>
  </url>
</urlset>
```

Dynamically generated pages

Slide from Croft et al., Information Retrieval in Practice

Document Feeds

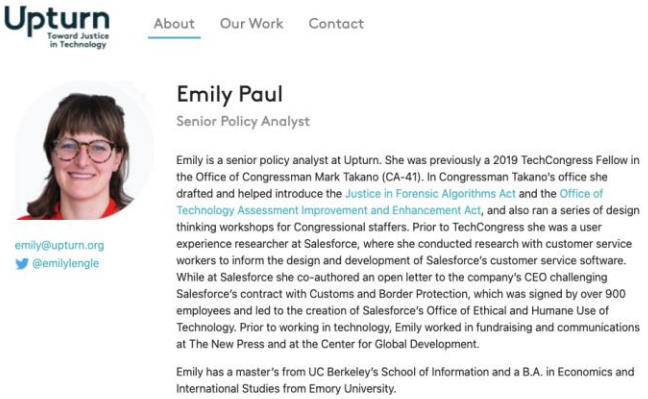
- Many documents are *published*
  - *created at a fixed time and rarely updated again*
  - *e.g., news articles, blog posts, press releases, email*
- Published documents from a single source can be ordered in a sequence called a *document feed*
  - *new documents found by examining the end of the feed*
  - *RSS used to be popular*
  - *Social media (seems to) have replaced it*

DISTRIBUTED CRAWLING

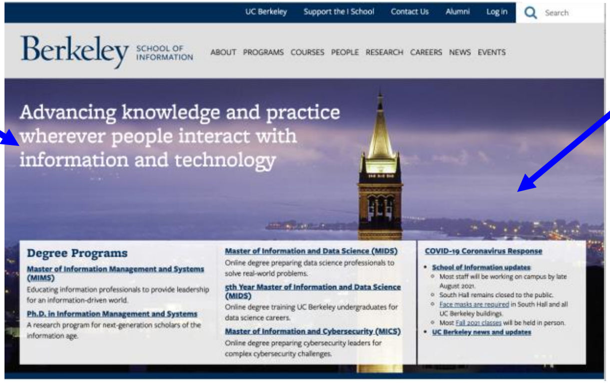
- Useful to use multiple computers for crawling
  - *Helps to put the crawler closer to the sites it crawls*
  - *Reduces the number of sites the crawler has to keep track of*
  - *Reduces computing resources required*
- Distributed crawler uses a hash function to assign URLs to crawling computers
  - *hash function should be computed on the host part of each URL*

The Importance of HTML Anchor Text

UCB School of Information



A terrific place to get a masters degree



The anchor text (in green) summarizes what the website is about.

How Search Engines Work

Three main parts:

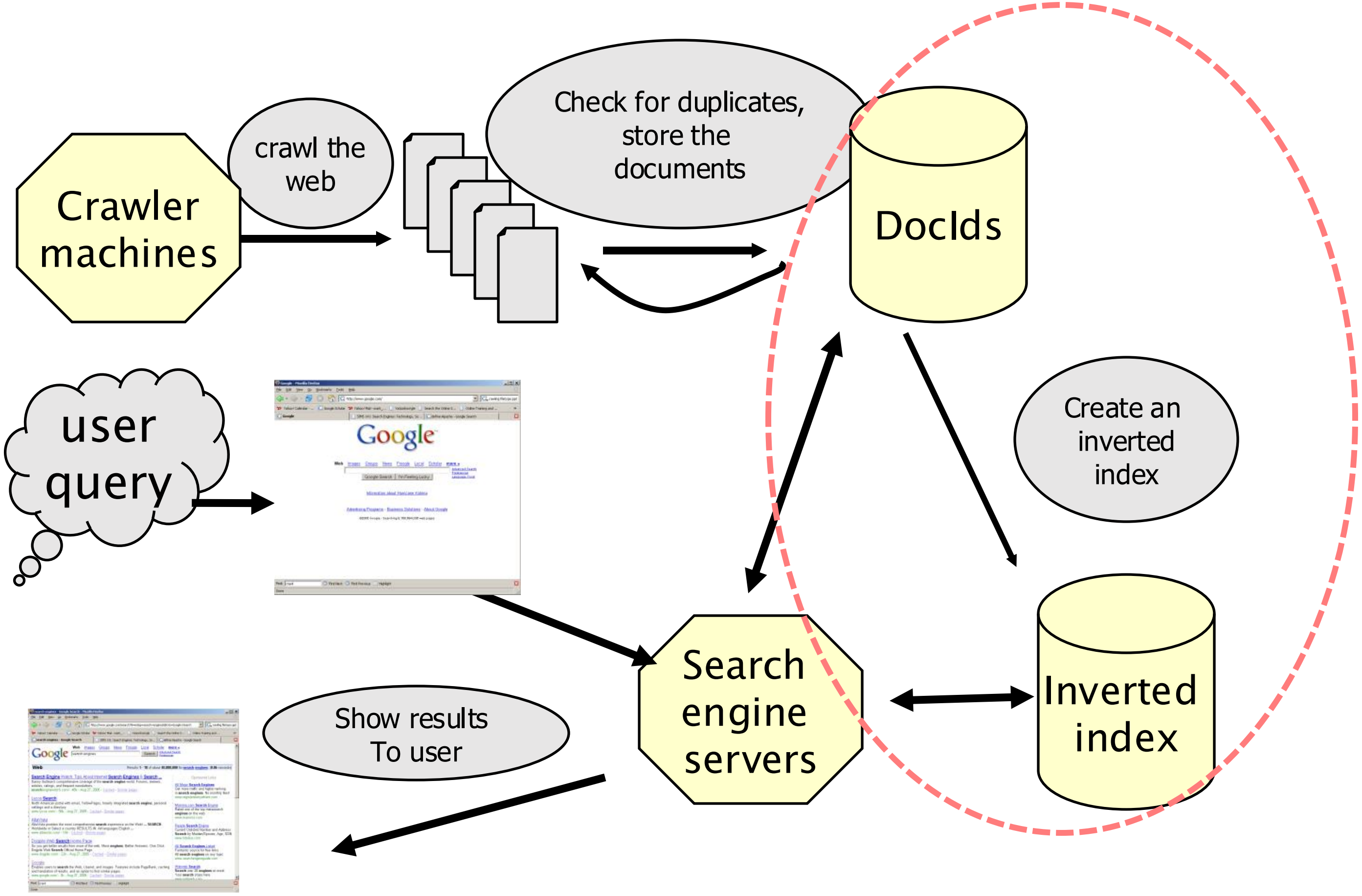
i. Gather the contents of all web pages (using a program called a **crawler** or **spider**)

ii. Organize the contents of the pages in a way that allows efficient retrieval (**indexing**)

iii. Take in a query, determine which pages match, and show the results (**ranking** and **display** of results)

Inverted Index

Standard Web Search Engine Architecture



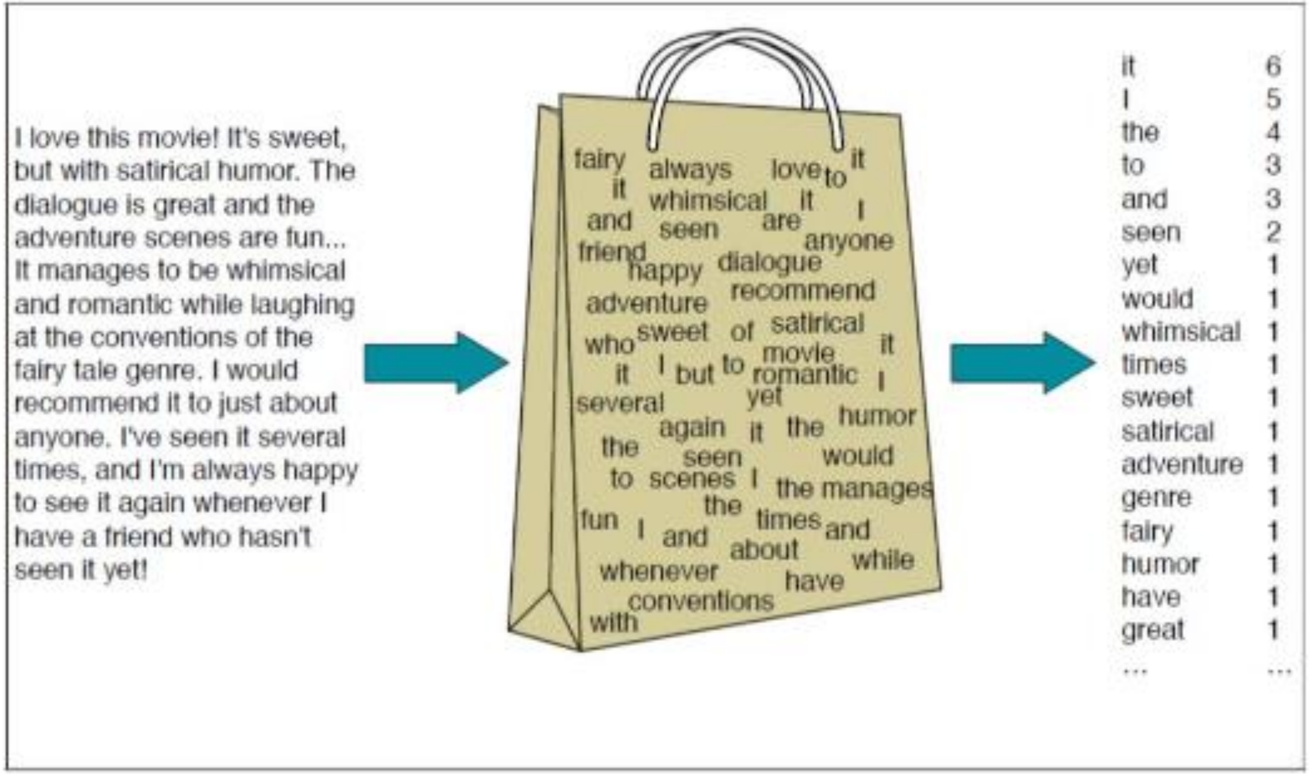
II. INDEX

Record information about each page

- List of words
  - *In the title?*
  - *How far down in the page?*
  - *Was the word in boldface?*
- URLs of pages pointing to this one
- Anchor text on pages pointing to this one

Slide adapted from Lew & Davis

Bag of Words Representation



Jurafsky & Martin slides

A Simple Representation of Text

- How do we represent the complexities of language?
  - Keeping in mind that computers don't "understand" documents or queries
- A simple, yet effective approach: create a"bag of words"
  - Treat all the words in a document as index terms for that document
  - Assign a "weight" to each term based on its "importance"
  - Disregard order, structure, meaning, etc. of the words

Sample Document

McDonald's slims down spuds

Fast-food chain to reduce certain types of fat in its french fries with new cooking oil.

NEW YORK (CNN/Money) - McDonald's Corp. is cutting the amount of "bad" fat in its french fries nearly in half, the fast-food chain said Tuesday as it moves to make all its fried menu items healthier.

But does that mean the popular shoestring fries won't taste the same? The company says no. "It's a win-win for our customers because they are getting the same great french-fry taste along with an even healthier nutrition profile," said Mike Roberts, president of McDonald's USA.

But others are not so sure. McDonald's will not specifically discuss the kind of oil it plans to use, but at least one nutrition expert says playing with the formula could mean a different taste.

Shares of Oak Brook, Ill.-based McDonald's (MCD: down $0.54 to $23.22, Research, Estimates) were lower Tuesday afternoon. It was unclear Tuesday whether competitors Burger King and Wendy's International (WEN: down $0.80 to $34.91, Research, Estimates) would follow suit. Neither company could immediately be reached for comment.

…

16 times: said

14 times: McDonalds

11 times: fries

6 times each: company french nutrition

5 times each: food oil percent reduce taste

…

“Bag of Words”

WHY DOES "BAG OF WORDS" WORK?

- Words alone tell us a lot about content

  Random: beating takes points falling another Dow 355

  Alphabetical: 355 another beating Dow falling points

  Actual: Dow takes another beating, falling 355 points

- BUT: ignoring word order & context can be misleading

  junior college is not the same as college junior

  building … code: software or architecture regulations?

The Document / Term Matrix is Sparse

Term	Doc 1	Doc 2	Doc 3	Doc 4	Doc 5	Doc 6	Doc 7	Doc 8
aid	0	0	0	1	0	0	0	1
all	0	1	0	1	0	1	0	0
back	1	0	1	0	0	0	1	0
brown	1	0	1	0	1	0	1	0
come	0	1	0	1	0	1	0	1
dog	0	0	1	0	1	0	0	0
fox	0	0	1	0	1	0	1	0
good	0	1	0	1	0	1	0	1
jump	0	0	1	0	0	0	0	0
lazy	1	0	1	0	1	0	1	0
men	0	1	0	1	0	0	0	1
now	0	1	0	0	0	1	0	1
over	1	0	1	0	1	0	1	1
party	0	0	0	0	0	1	0	1
quick	1	0	1	0	0	0	0	0
their	1	0	0	0	1	0	1	0
time	0	1	0	1	0	1	0	0

Notice all of the zeros – tha tis a lot of wasted space

Inverted Index

- In reality, this index is HUGE
- Need to store the contents across many machines
- Need to do optimization tricks to make lookup fast.
- How and why to build it?

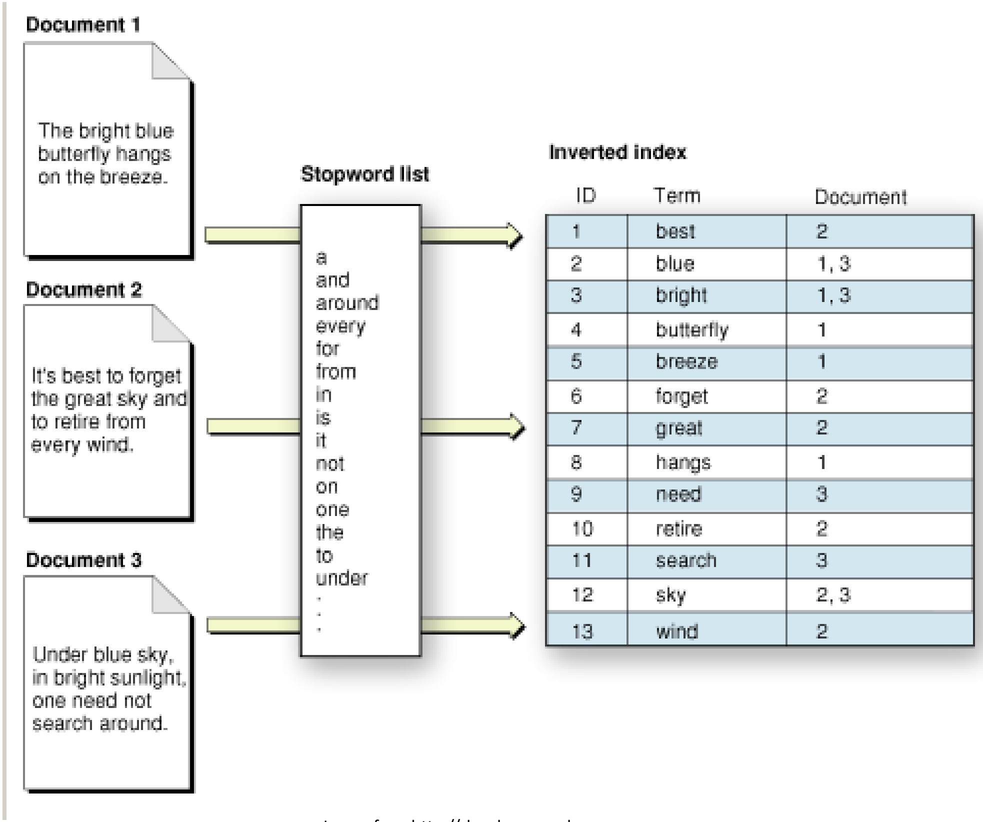
We Need A Smaller, Faster Data Structure

- Can we make this data structure smaller, keeping in mind the need for fast retrieval?
- Observations:
  - *The nature of the search problem requires us to quickly find which documents contain a term*
  - *The term-document matrix is very sparse*
  - *Some terms are more useful than others*
- Solution: The **Inverted Index**

INVERTED INDEX

- How to store the words for fast lookup
- Basic steps:
  - *Make a "dictionary" of all the words in all of the web pages*
  - *For each word, list all the documents it occurs in.*
  - *Often omit very common words*
    - **"stop words"**
  - *Sometimes **stem** the words*
    - (also called **morphological analysis**)
    - cats -> cat
    - running -> run

Inverted Index Example



ID	Term	Document
1	best	2
2	blue	1, 3
3	bright	1, 3
4	butterfly	1
5	breeze	1
6	forget	2
7	great	2
8	hangs	1
9	need	3
10	retire	2
11	search	3
12	sky	2, 3
13	wind	2

Image from http://developer.apple.com
/documentation/UserExperience/Conceptual/SearchKitConcepts/searchKit_basics/chapter_2_section_2.html

INVERTED INDEX

- This is the primary data structure for text indexes
- Main Idea:
  - *Invert documents into a big index*
- Basic steps:
  - *Make a "dictionary" of all the tokens in the collection*
  - *For each token, list all the docs it occurs in.*
  - *Do a few things to reduce redundancy in the data structure*

An Inverted Index Has Terms (Dictionary) and Postings

Term	Postings
aid	4, 8
all	2, 4, 6
back	1, 3, 7
brown	1, 3, 5, 7
come	2, 4, 6, 8
dog	3, 5
fox	3, 5, 7
good	2, 4, 6, 8
jump	3
lazy	1, 3, 5, 7
men	2, 4, 8
now	2, 6, 8
over	1, 3, 5, 7, 8
party	6, 8
quick	1, 3
their	1, 5, 7
time	2, 4, 6

Creating Postings

Term	Doc 1	Doc 2	Doc 3	Doc 4	Doc 5	Doc 6	Doc 7	Doc 8	Postings
aid	0	0	0	1	0	0	0	1	4, 8
all	0	1	0	1	0	1	0	0	2, 4, 6
back	1	0	1	0	0	0	1	0	1, 3, 7
brown	1	0	1	0	1	0	1	0	1, 3, 5, 7
come	0	1	0	1	0	1	0	1	2, 4, 6, 8
dog	0	0	1	0	1	0	0	0	3, 5
fox	0	0	1	0	1	0	1	0	3, 5, 7
good	0	1	0	1	0	1	0	1	2, 4, 6, 8
jump	0	0	1	0	0	0	0	0	3
lazy	1	0	1	0	1	0	1	0	1, 3, 5, 7
men	0	1	0	1	0	0	0	1	2, 4, 8
now	0	1	0	0	0	1	0	1	2, 6, 8
over	1	0	1	0	1	0	1	1	1, 3, 5, 7, 8
party	0	0	0	0	0	1	0	1	6, 8
quick	1	0	1	0	0	0	0	0	1, 3
their	1	0	0	0	1	0	1	0	1, 5, 7
time	0	1	0	1	0	1	0	0	2, 4, 6

What goes in the postings?

- Boolean retrieval
  - *Just the document number*
- Ranked Retrieval
  - *Document number and term weight (tf.idf, ...)*
- Proximity operators
  - *Word offsets for each occurrence of the term*

Using The Inverted Index in The Retrieval Process

- During retrieval:
  - *Find the relevant postings based on query terms*
  - *Manipulate the postings based on the query*
  - *Return appropriate documents*

How big are the postings?

- Very compact for Boolean retrieval
  - *About 10% of the size of the documents*
- Not much larger for ranked retrieval
  - *Perhaps 20% of collection size*
- Enormous for proximity operators
  - *Sometimes larger than the document collection*

Further Compressing the Index

- Postings can still be quite large
  - *Especially if you have a large collection*

e.g., 1 million documents → 20-bit document numbers

- Idea: encode differences instead of document numbers

37, 42, 43, 48, 97, 98, 243 →
37, 5, 1, 5, 49, 1, 145

- Many other ways to compress the postings

- What about dropping unimportant terms from the index?
  - *How much space does stopword removal save?*

DECOUPLING THE INVERTED INDEX

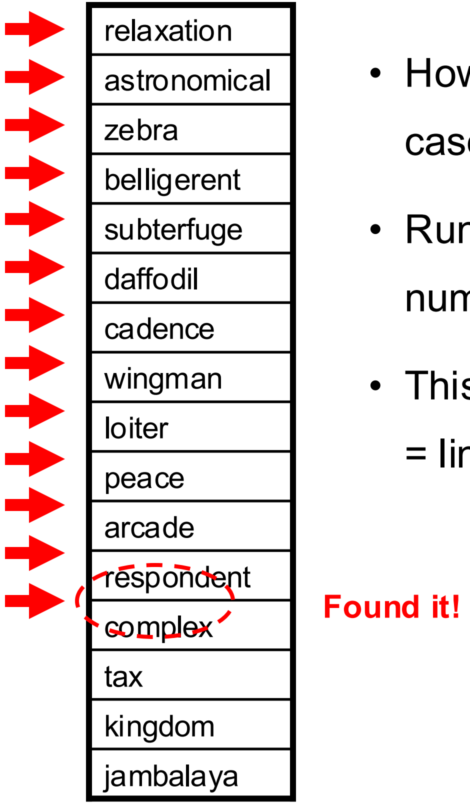
The term Index	Postings
aid	4, 8
all	2, 4, 6
back	1, 3, 7
brown	1, 3, 5, 7
come	2, 4, 6, 8
dog	3, 5
fox	3, 5, 7
good	2, 4, 6, 8
jump	3
lazy	1, 3, 5, 7
men	2, 4, 8
now	2, 6, 8
over	1, 3, 5, 7, 8
party	6, 8
quick	1, 3
their	1, 5, 7
time	2, 4, 6

Terms in the Collection

- Let's focus on the term index
- The postings are relatively simple
  - *During indexing: once you find the correct postings, add information from current document*
  - *During retrieval: once you find the correct postings, manipulate based on query operator*
- Questions
  - *How do you find the correct posting quickly?*
  - *What happens when you come across a new term?*

Linear Dictionary Lookup

Suppose we want to find the word "complex"



- How long does this take, in the worst case?
- Running time is proportional to number of entries in the dictionary
- This algorithm is O(*n*) = linear time algorithm

With a Sorted Dictionary

Let’s try again, except this time with a sorted dictionary: find “complex”

arcade
astronomical
belligerent
cadence
complex
daffodil
jambalaya
kingdom
loiter
peace
relaxation
respondent
subterfuge
tax
wingman
zebra

Found it!

- How long does this take, in the worst case?

Binary Search: Analysis

- Algorithm:
  - *Look in the middle entry of a region, call this x*
  - *If the entry you're looking for comes before x, then look in first half, otherwise look in second half*
  - *Repeat until you find what you're looking for*
- Analysis:
  - *Each time we look up an entry, we cut down the number to consider by a half*
  - *How many times can you divide a number by 2?*
- This algorithm is $O(\lg n)$

Which is faster?

- Two algorithms:
  - *O(n): Sequentially search through every entry*
  - *O(lg n): Binary search*
- Big-O notation
  - *Tells us the asymptotic worst case running time of an algorithm*
  - *Allows us to compare the speed of different algorithms*

NEXT TIME

- Considerations for Ranking
- Boolean Ranking
- Vector Space Rankng