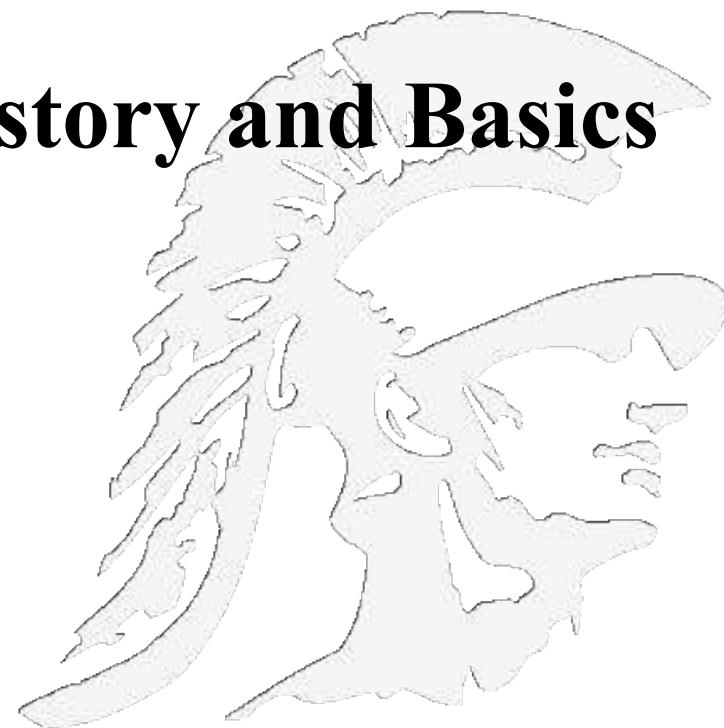


Search Engine History and Basics



A Brief Chronology of Search Engines

- 1991
 - Gopher, Archie, Veronica

early search engines, non-web

- 1993
 - Wanderer,
 - ALIWeb
 - Excite

<http://www.excite.com/>

powerful indexing

- 1994
 - Galaxy
 - Yahoo
 - Lycos
 - WebCrawler
 - Alta Vista

<http://www.galaxy.com/>
<http://www.yahoo.com/>
<http://www.lycos.com/>
<http://www.webcrawler.com/>
<http://www.altavista.com/>

Early searchable directory
 Sophisticated searchable directory
 Improved query matching
 Includes full text of pages
 a large index

- 1995
 - Infoseek
 - Metacrawler
 - SavvySearch
 - LookSmart

<http://www.infoseek.com/>
<http://www.metacrawler.com/>
<http://www.savvysearch.com/>
<http://www.looksmart.com>

included in Netscape Navigator
 combines results from other engines
 combines results from other engines
 convenient organization

- 1996
 - Inktomi
 - HotBot,

<http://www.inktomi.com>
<http://www.hotbot.com/>

a large index using commodity hardware
 a large index

- 1997
 - AskJeeves

<http://www.askjeeves.com>

fancy query processing

- 1998
 - Goto
 - Google

<http://www.goto.com>
<http://www.google.com>

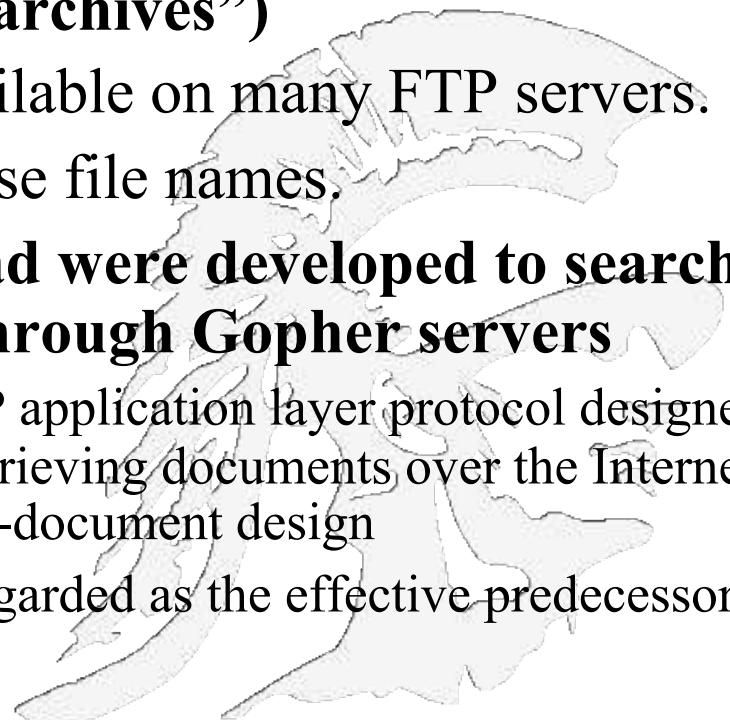
introduces auctioning of positions
 ranking using content and links

- Today there are hundreds of search engines, many are specialized
- See Search Engine History

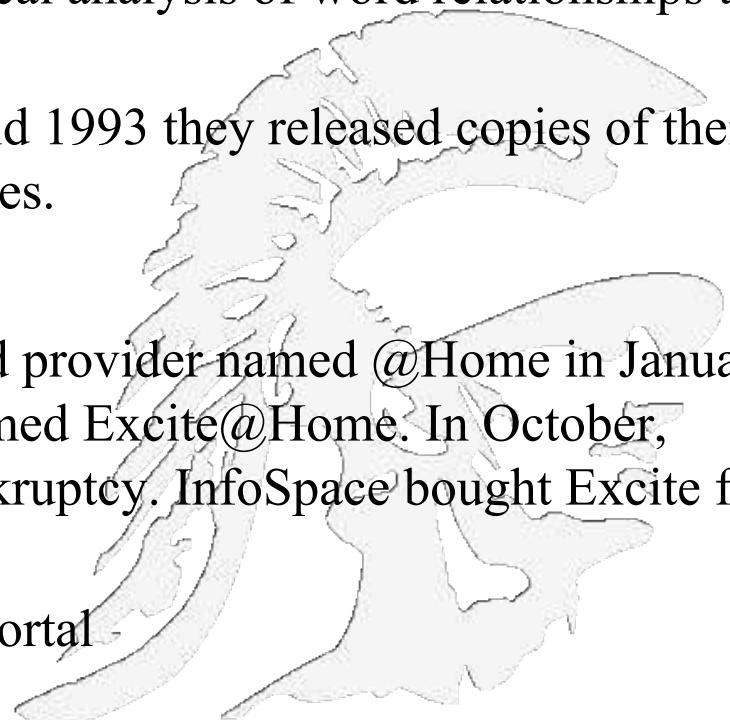
- A very long web page describing the history of search

Archie, Veronica, Gopher

- By late 1980's many files were available by anonymous FTP.
- In 1990, Alan Emtage, P. Deutsch, et al of McGill Univ. developed Archie (short for “archives”)
 - Assembled lists of files available on many FTP servers.
 - Allowed regex search of these file names.
- In 1993, Veronica and Jughead were developed to search names of text files available through Gopher servers
 - The **Gopher protocol** is a TCP/IP application layer protocol designed for distributing, searching, and retrieving documents over the Internet. Strongly oriented towards a menu-document design
 - The Gopher ecosystem is often regarded as the effective predecessor of the World Wide Web

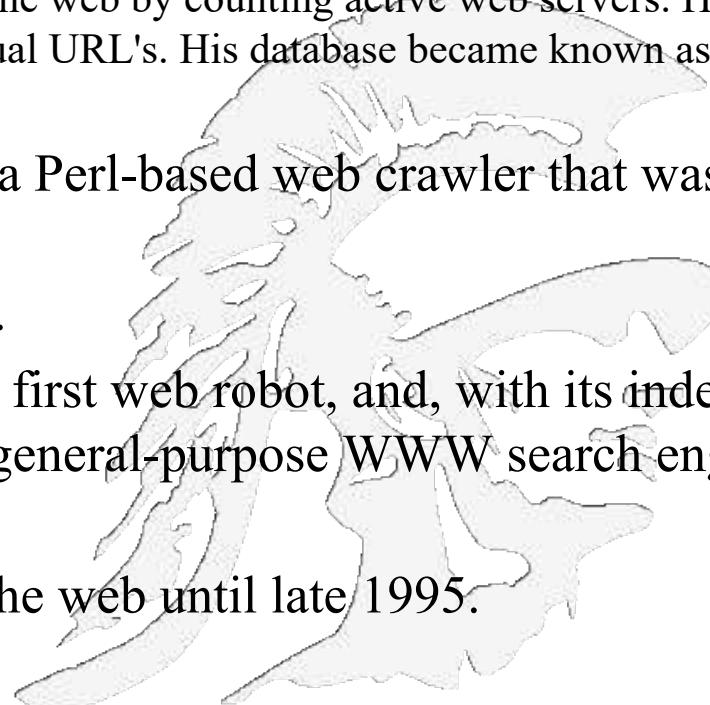


- Excite came from the project Architext, which was started in February, 1993 by six Stanford undergrad students.
 - They had the idea of using statistical analysis of word relationships to make searching more efficient.
 - They were soon funded, and in mid 1993 they released copies of their search software for use on web sites.
- Later developments
 - Excite was bought by a broadband provider named @Home in January, 1999 for \$6.5 billion, and was named Excite@Home. In October, 2001 Excite@Home filed for bankruptcy. InfoSpace bought Excite from bankruptcy court for \$10 million
 - www.excite.com still exists as a portal

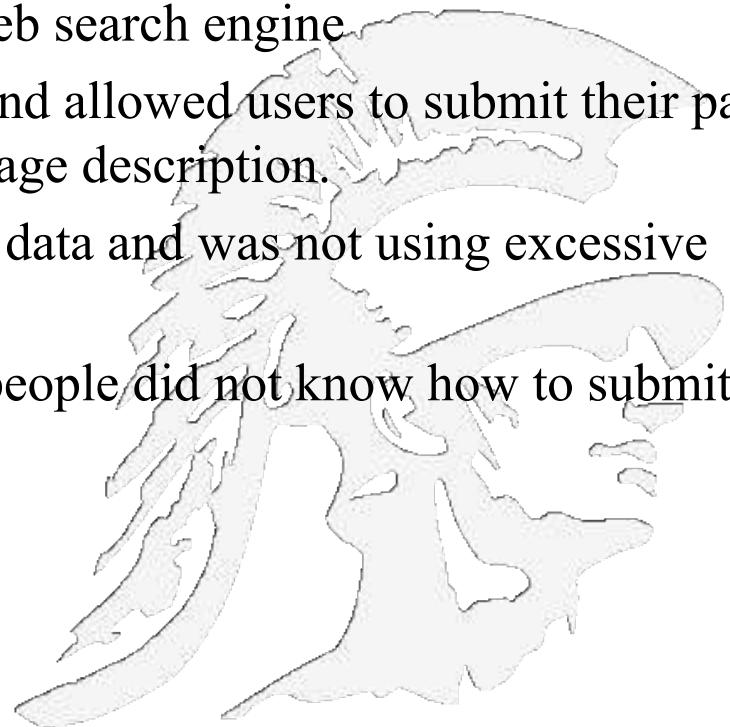


World Wide Web Wanderer

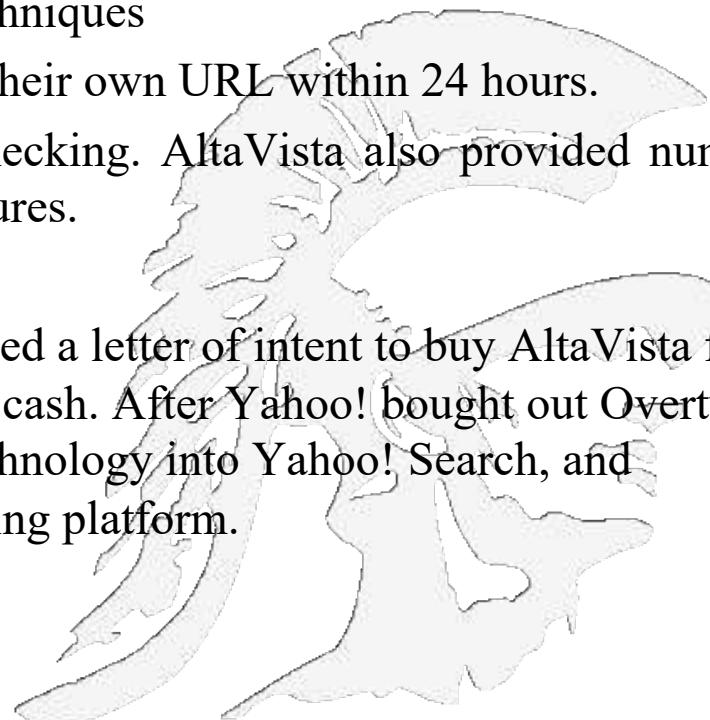
- In June 1993 Matthew Gray while at MIT introduced the World Wide Web Wanderer.
 - Initial goal was to measure the growth of the web by counting active web servers. He soon upgraded the software to capture actual URL's. His database became known as the Wandex.
- The World Wide Web Wanderer was a Perl-based web crawler that was first deployed in June 1993
- Matthew Gray now works for Google.
- While the Wanderer was probably the first web robot, and, with its index, clearly had the potential to become a general-purpose WWW search engine it never went that far
- The Wanderer charted the growth of the web until late 1995.



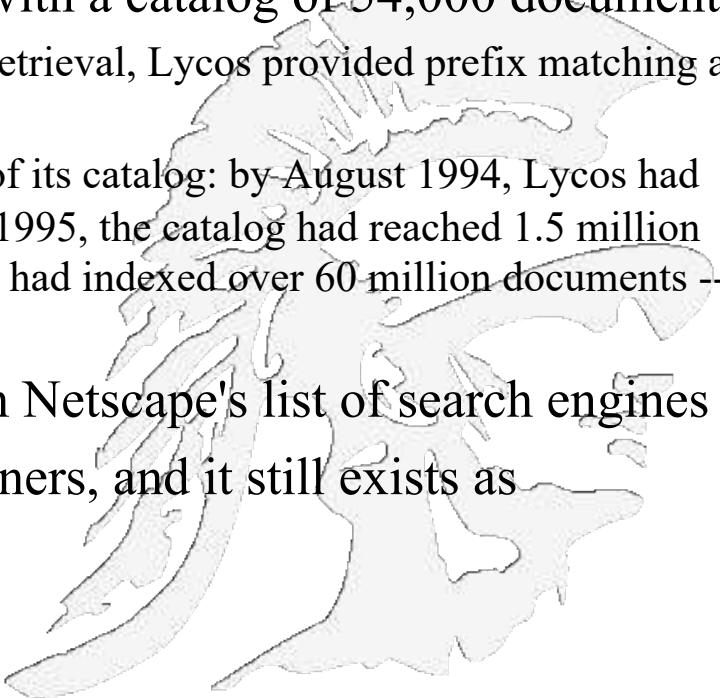
- In November of 1993 Martijn Koster created “Archie-Like Indexing of the Web”, or ALIWEB in response to the Wanderer.
 - Some consider it to be the first Web search engine
- ALIWEB crawled meta information and allowed users to submit their pages they wanted indexed with their own page description.
- This meant it needed no bot to collect data and was not using excessive bandwidth.
- One downside of ALIWEB was that people did not know how to submit their site



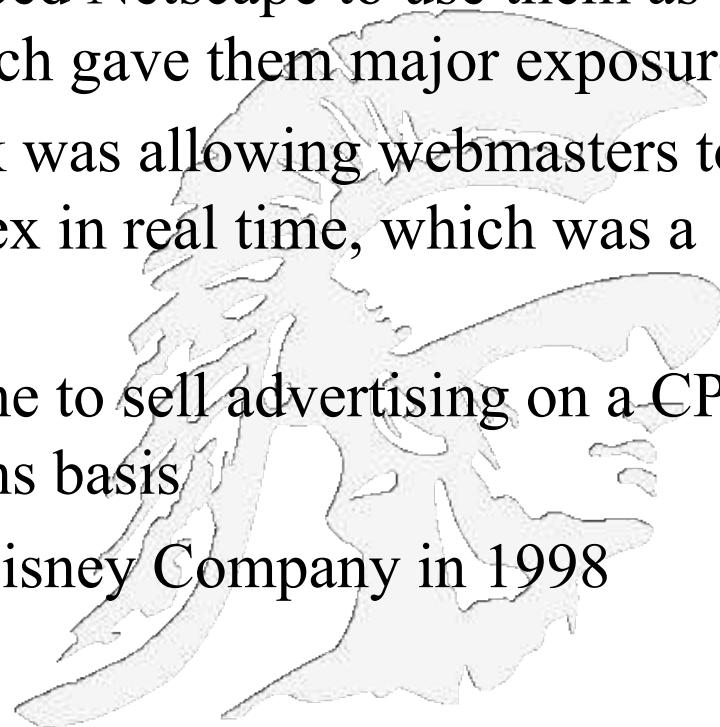
- AltaVista debut online came during December, 1995. AltaVista brought many important features to the web scene.
 - They were the first to allow natural language queries
 - They offered advanced searching techniques
 - They allowed users to add or delete their own URL within 24 hours.
 - They even allowed inbound link checking. AltaVista also provided numerous search tips and advanced search features.
- Later developments
 - On February 18, 2003, Overture signed a letter of intent to buy AltaVista for \$80 million in stock and \$60 million cash. After Yahoo! bought out Overture they rolled some of the AltaVista technology into Yahoo! Search, and occasionally used AltaVista as a testing platform.



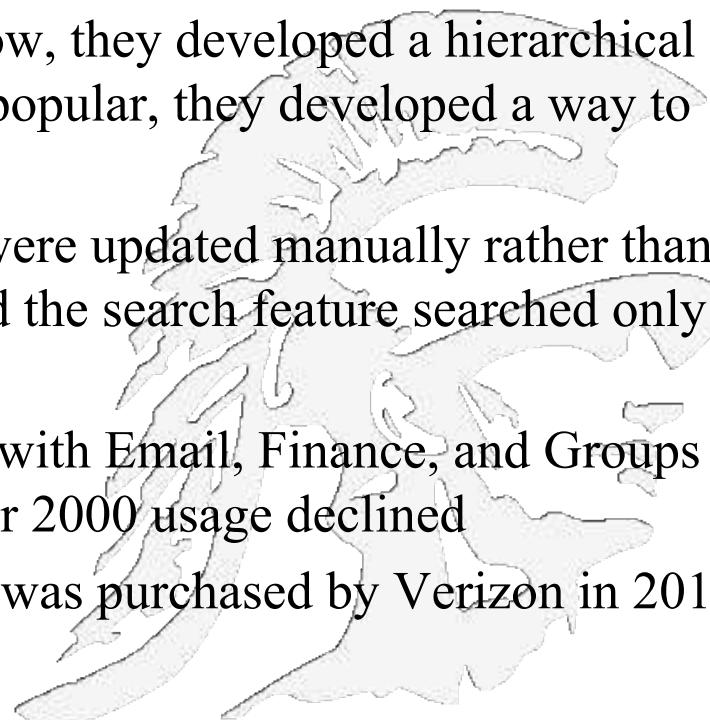
- Lycos was designed at Carnegie Mellon University around July of 1994. Michael Loren Mauldin was responsible for this search engine and was the chief scientist at Lycos Inc in the early years.
- On July 20, 1994, Lycos went public with a catalog of 54,000 documents.
 - In addition to providing ranked relevance retrieval, Lycos provided prefix matching and word proximity bonuses.
 - Lycos' main difference was the sheer size of its catalog: by August 1994, Lycos had identified 394,000 documents; by January 1995, the catalog had reached 1.5 million documents; and by November 1996, Lycos had indexed over 60 million documents -- more than any other Web search engine.
- In October 1994, Lycos ranked first on Netscape's list of search engines
- Lycos has gone through a series of owners, and it still exists as www.lycos.com



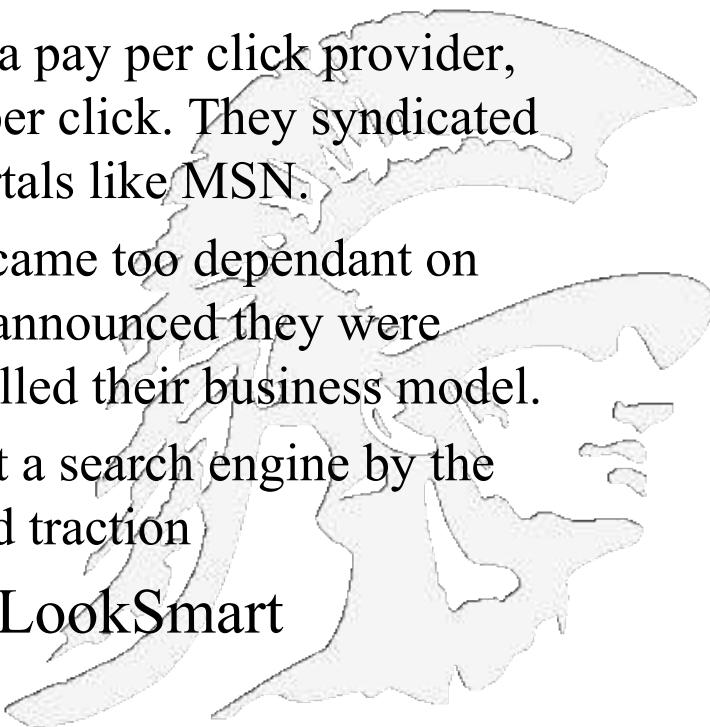
- Infoseek also started out in 1994, founded by Steve Kirsch
- In December 1995 they convinced Netscape to use them as their default search engine, which gave them major exposure.
- One popular feature of Infoseek was allowing webmasters to submit a page to the search index in real time, which was a search spammer's paradise
- They were the first search engine to sell advertising on a CPM (Cost per Thousand) impressions basis
- Infoseek was bought by Walt Disney Company in 1998



- In 1994, two Stanford Ph.D. students David Filo and Jerry Yang posted web pages with links on them, organized into a topical hierarchy.
- As the number of links began to grow, they developed a hierarchical listing. As the pages become more popular, they developed a way to search through all of the links.
- Early on all the links on the pages were updated manually rather than automatically by spider or robot and the search feature searched only those links
- Yahoo home page acted as a portal with Email, Finance, and Groups being very successful; however after 2000 usage declined
- After many years of decline Yahoo was purchased by Verizon in 2017 for \$4.48 billion, and it lives on



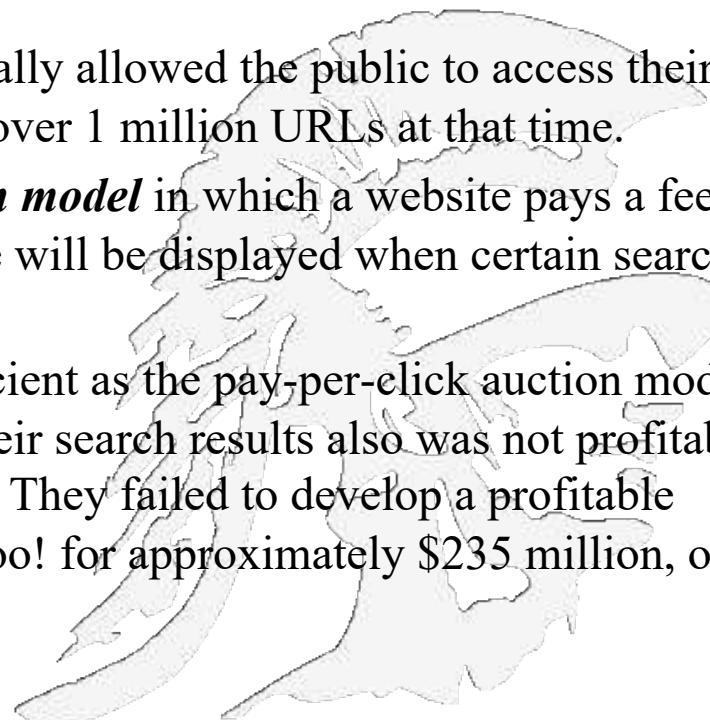
- Looksmart was founded in 1995 in Australia. They competed with the Yahoo! Directory by frequently increasing their inclusion rates
- Later developments
 - In 2002 Looksmart transitioned into a pay per click provider, which charged listed sites a flat fee per click. They syndicated those paid listings to some major portals like MSN.
 - The problem was that Looksmart became too dependant on MSN, and in 2003, when Microsoft announced they were dumping Looksmart that basically killed their business model.
 - In March of 2002, Looksmart bought a search engine by the name of WiseNut, but it never gained traction
- See <https://en.wikipedia.org/wiki/LookSmart>





Inktomi

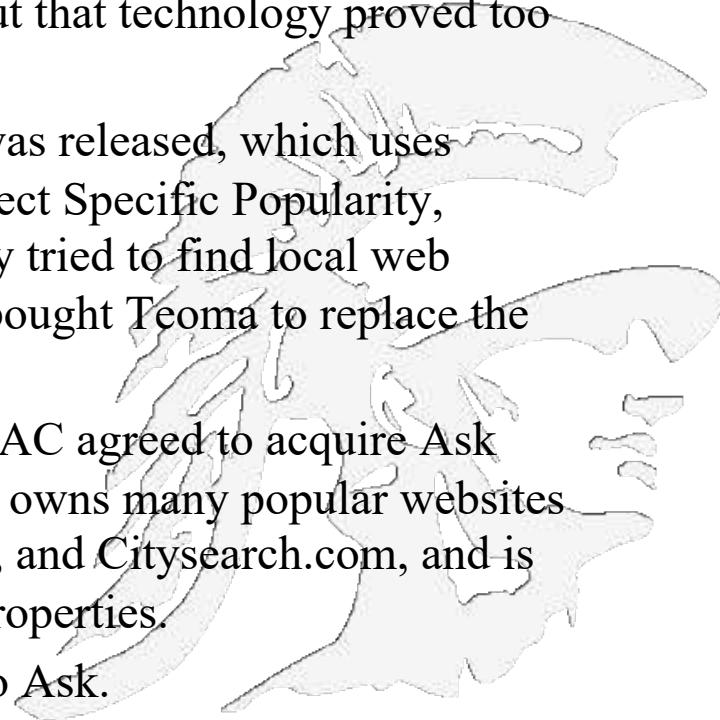
- The Inktomi Corporation came about on May 20, 1996 with its search engine Hotbot. Two Cal Berkeley cohorts created Inktomi from the improved technology gained from their research
- Later developments
 - In October of 2001 Inktomi accidentally allowed the public to access their database of spam sites, which listed over 1 million URLs at that time.
 - Inktomi pioneered ***the paid inclusion model*** in which a website pays a fee to the search engine that guarantees the site will be displayed when certain search terms are entered
 - The model was nowhere near as efficient as the pay-per-click auction model developed by Overture. Licensing their search results also was not profitable enough to pay for their scaling costs. They failed to develop a profitable business model, and sold out to Yahoo! for approximately \$235 million, or \$1.65 a share, in December of 2003.



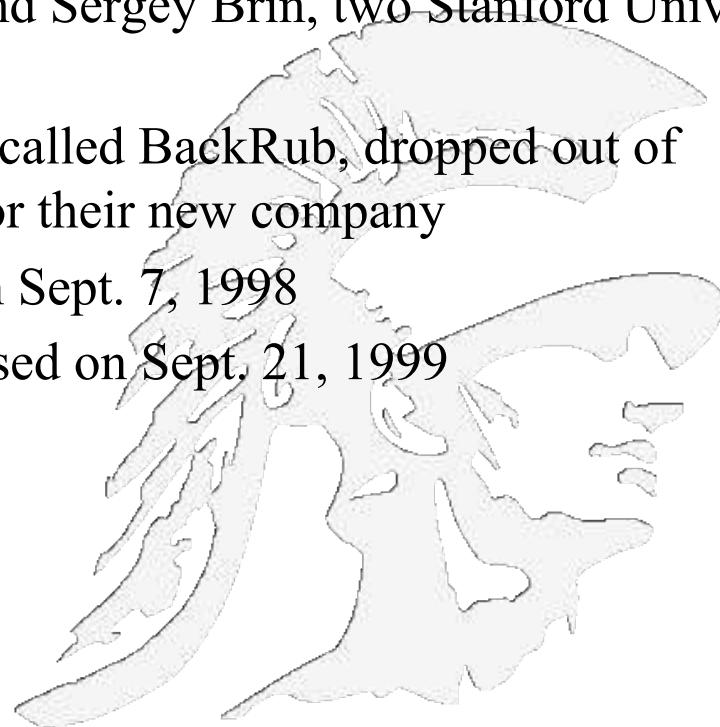
*<http://searchenginewatch.com/article/2066745/Inktomi-Spam-Database-Left-Open-To-Public>



- In April of 1997 Ask Jeeves was launched as a natural language search engine.
 - Ask Jeeves used human editors to try to match search queries.
 - Ask was powered by DirectHit for a while, which aimed to rank results based on their popularity, but that technology proved too easy to spam.
 - In 2000 the Teoma search engine was released, which uses clustering to organize sites by Subject Specific Popularity, which is another way of saying they tried to find local web communities. In 2001 Ask Jeeves bought Teoma to replace the DirectHit search technology.
 - On March 21, 2005 Barry Diller's IAC agreed to acquire Ask Jeeves for 1.85 billion dollars. IAC owns many popular websites like Match.com, Ticketmaster.com, and Citysearch.com, and is promoting Ask across their other properties.
 - In 2006 Ask Jeeves was renamed to Ask.



- Google is a play on the word Googol, coined by Milton Sirotta; it refers to a 1 followed by 100 zeros, 10000000.....0
- A googol is bigger than the number of atoms in the universe
- Google was founded by Larry Page and Sergey Brin, two Stanford Univ. Computer Science graduate students
- In 1998 they built a prototype system called BackRub, dropped out of school, and tried to attract investors for their new company
- Google Inc. released a beta version on Sept. 7, 1998
- www.google.com was officially released on Sept. 21, 1999



A Brief Chronology of Search Engines

Algorithmic Search Era

Gopher/Archie/
Veronica - Early
Internet Search
Engines

Yahoo

Lycos

Excite

Alta Vista

Inktomi

Hotbot

Ask Jeeves

Google

Google
begins
Adword and
Pay-Per-Click

1991

1992

1993

1994

1995

1996

1997

1998

1999

2000

2001

2002

2003

Overture (goto.com) is the first to combine sponsored (paid) search results with conventional search results

Goto.com
introduces
pay-per-click
search results

Google
begins to include
pay-per-click
search results

Paid Search Era

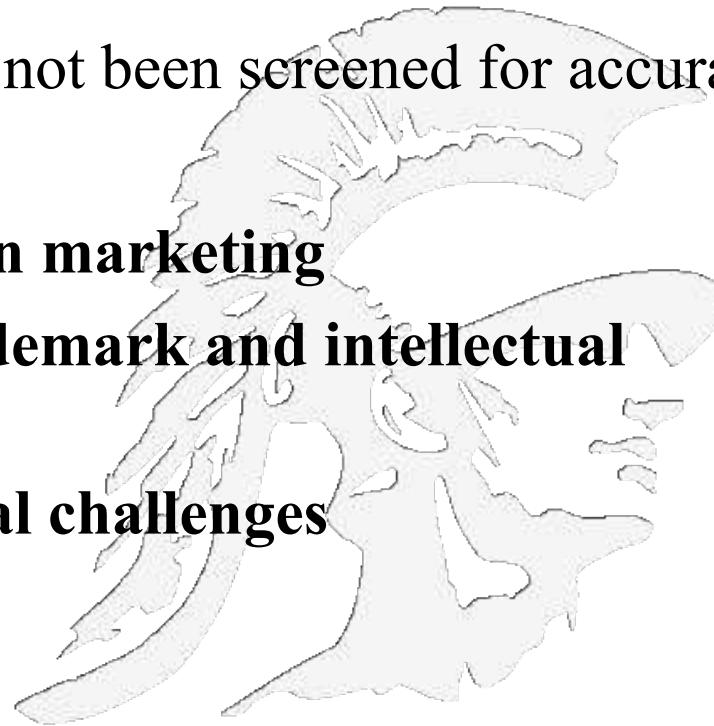
(paid search became pervasive)

Search Engine Basic Behavior



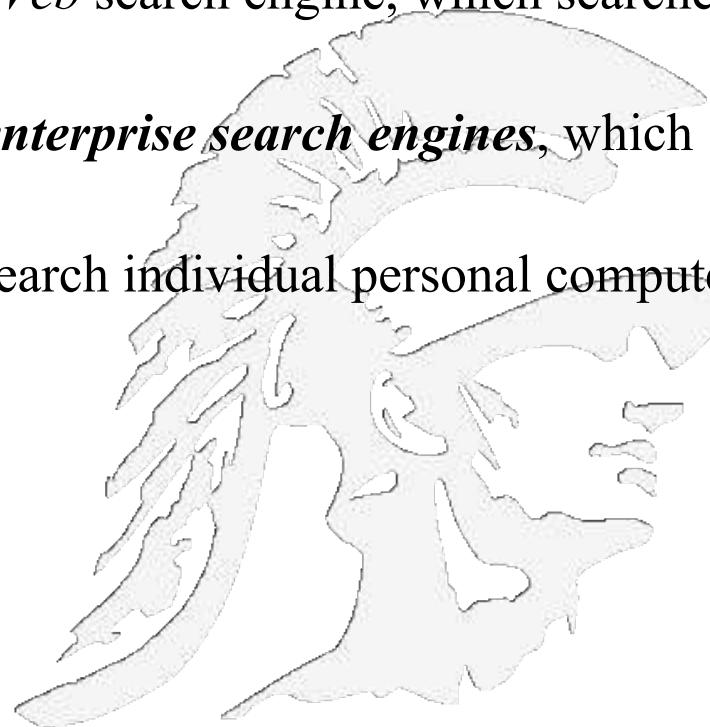
What is Web Search?

- **Providing access to heterogeneous, distributed information that is publicly available on the World Wide Web**
 - Information comes in many different formats
 - Most of the information has not been screened for accuracy
- **Multi-billion dollar business**
- **Source of new opportunities in marketing**
- **Strains the boundaries of trademark and intellectual property laws**
- **A source of unending technical challenges**

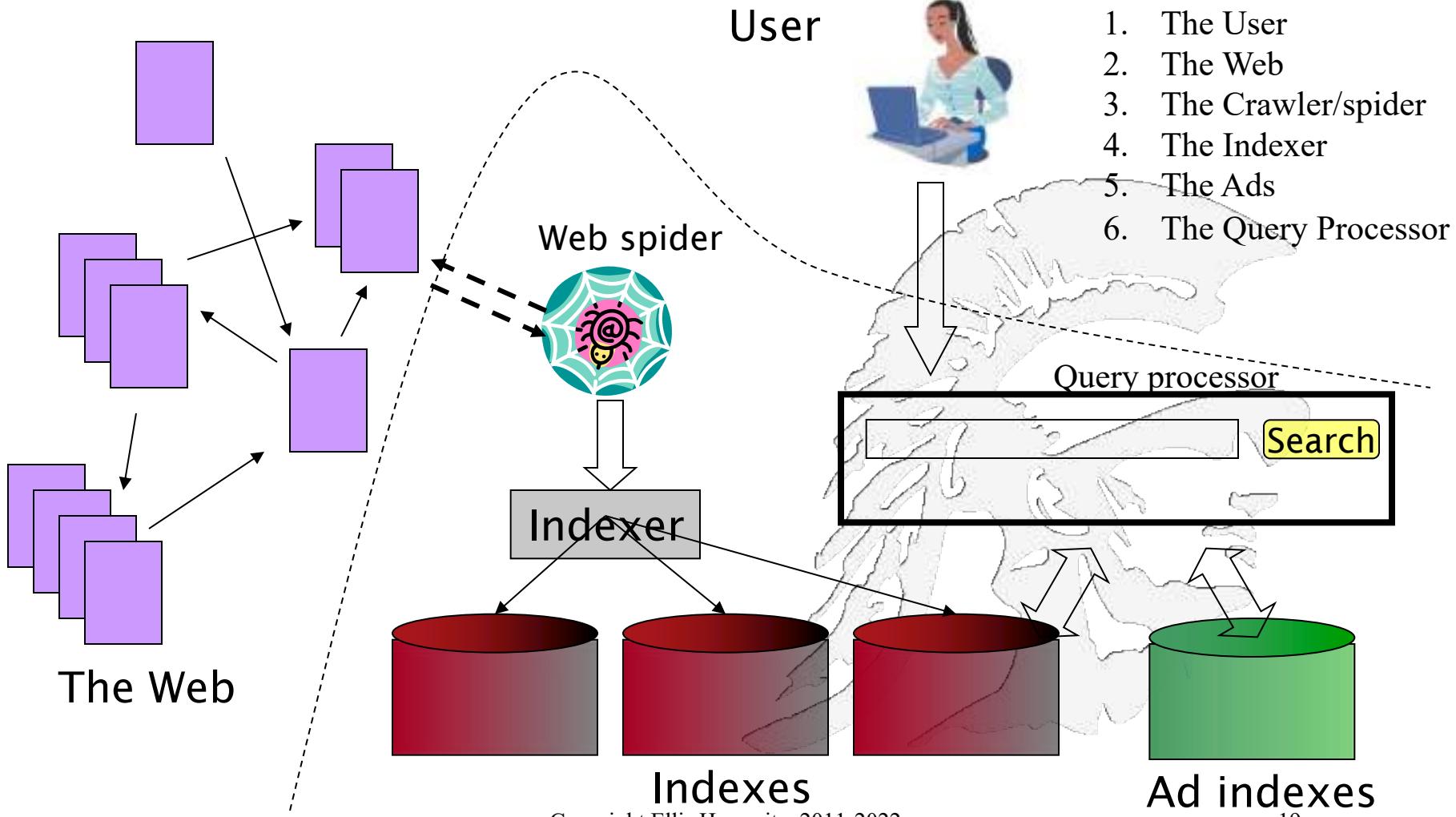


Web Search Engine Definitions

- “A search engine is a program designed to help find information stored on a computer system such as the World Wide Web, inside a corporate or proprietary network or a personal computer” *wikipedia*
 - *search engine* usually refers to a *Web search engine*, which searches for information on the public Web.
 - Other kinds of search engine are *enterprise search engines*, which search on intranets,
 - *personal search engines*, which search individual personal computers

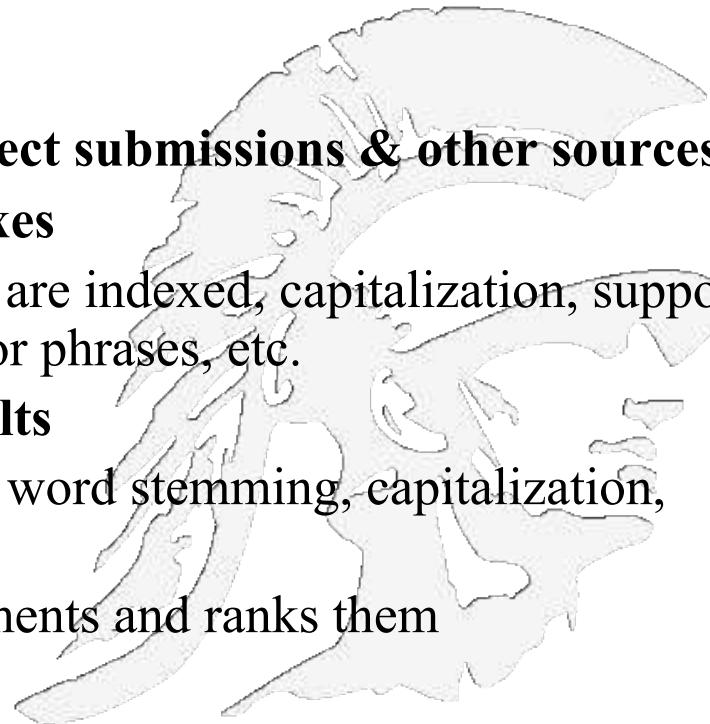


Basic Web Search Internals

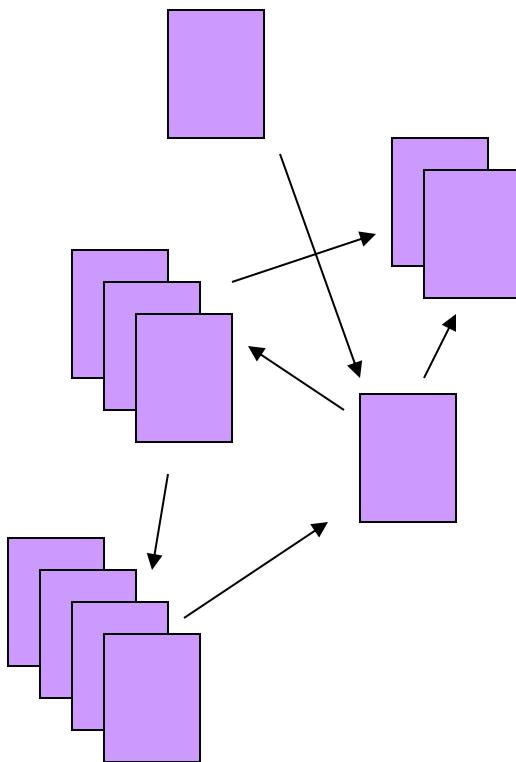


Web Search Engine Elements

- ***Spider* (a.k.a. crawler/robot) – builds **corpus**
 - Collects web pages recursively
 - For each known URL, fetch the page, parse it, and extract new URLs
 - Repeat
 - Additional pages come from direct submissions & other sources**
- The ***indexer*** – creates inverted indexes
 - Various policies wrt which words are indexed, capitalization, support for Unicode, stemming, support for phrases, etc.
- ***Query processor*** – serves query results
 - **Front end** – query reformulation, word stemming, capitalization, optimization of Booleans, etc.
 - **Back end** – finds matching documents and ranks them

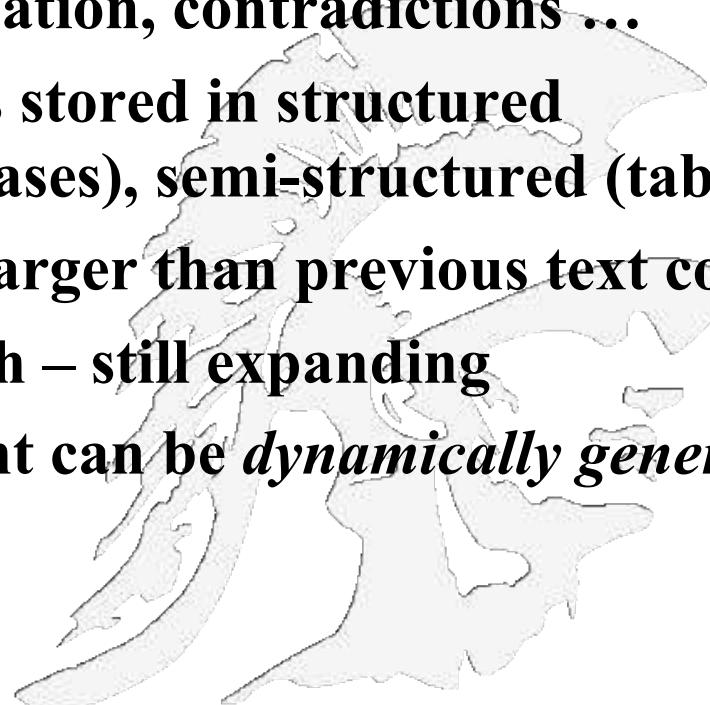


The Web



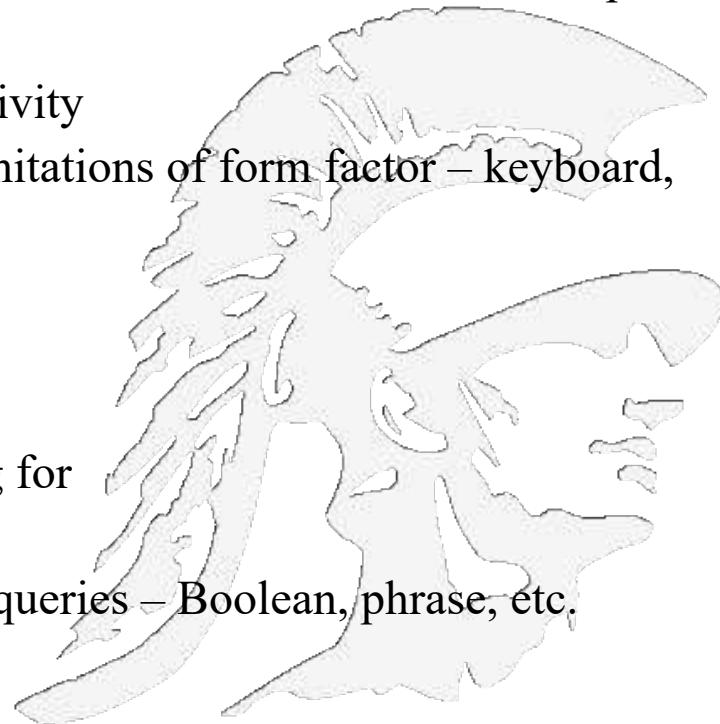
The Web

- No design/co-ordination
- Distributed content creation, linking
- Content includes truth, lies, obsolete information, contradictions ...
- Data is stored in structured (databases), semi-structured (tables)...
- Scale larger than previous text corpora
- Growth – still expanding
- Content can be *dynamically generated*



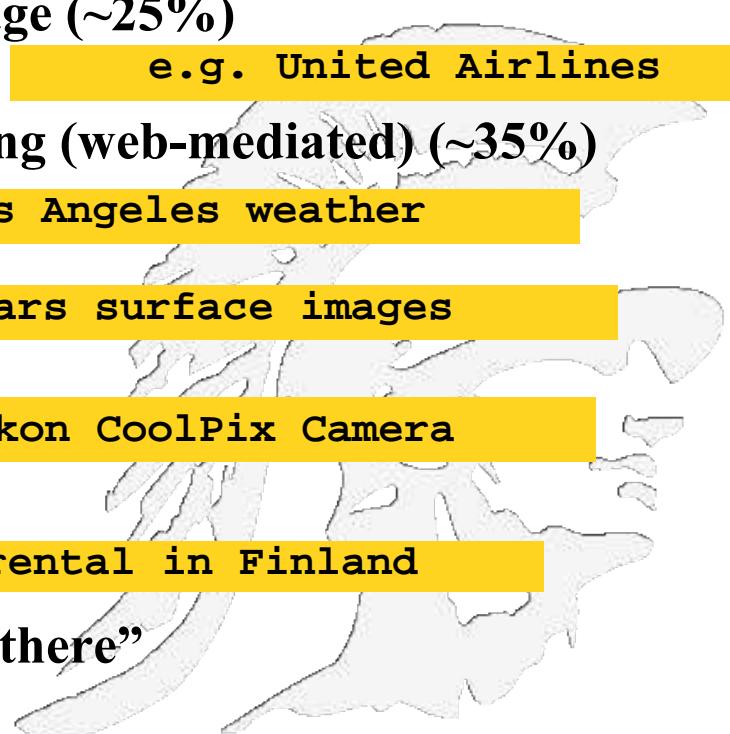
The User

- **Diverse in background/training**
 - Users sometimes cannot tell the difference between a search bar from the URL address field (**Chrome conflates the two**)
 - Users rarely use the scroll bar, so key results must be at or near the top
- **Diverse in access methodology**
 - Increasingly, high bandwidth connectivity
 - Growing segment of mobile users: limitations of form factor – keyboard, display
- **Diverse in search methodology**
 - Search, search + browse,
 - Average query length ~ 2.5 terms
 - Has to do with what they're searching for
- **Poor comprehension of syntax**
 - Early engines offered rich syntax for queries – Boolean, phrase, etc.
 - Current engines hide these



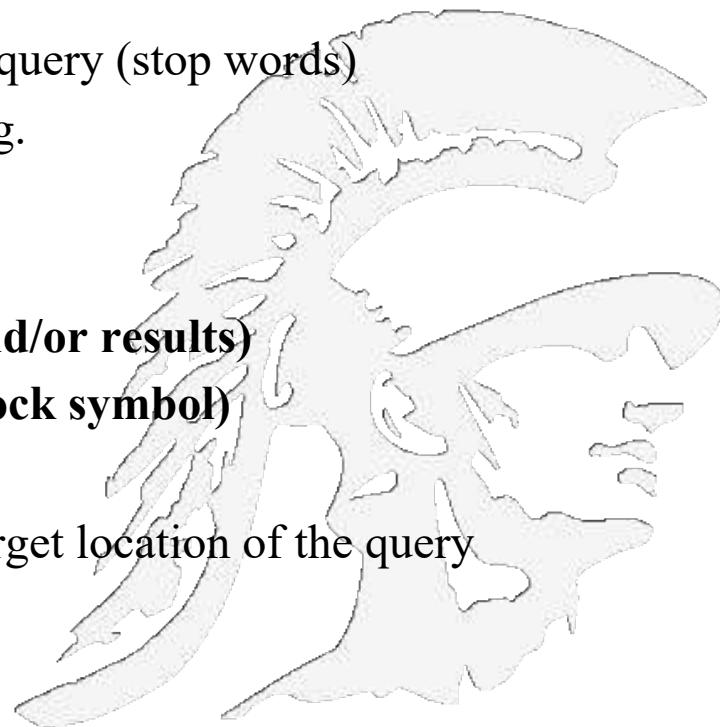
User's Information Needs Are Diverse

- **Informational** – want to learn about something (~40%)
e.g. Low hemoglobin
- **Navigational** – want to go to that page (~25%)
e.g. United Airlines
- **Transactional** – want to do something (web-mediated) (~35%)
 - Access a service Los Angeles weather
 - Downloads Mars surface images
 - Shop Nikon CoolPix Camera
- **Gray areas**
 - Find a good hub Car rental in Finland
 - Exploratory search “see what’s there”

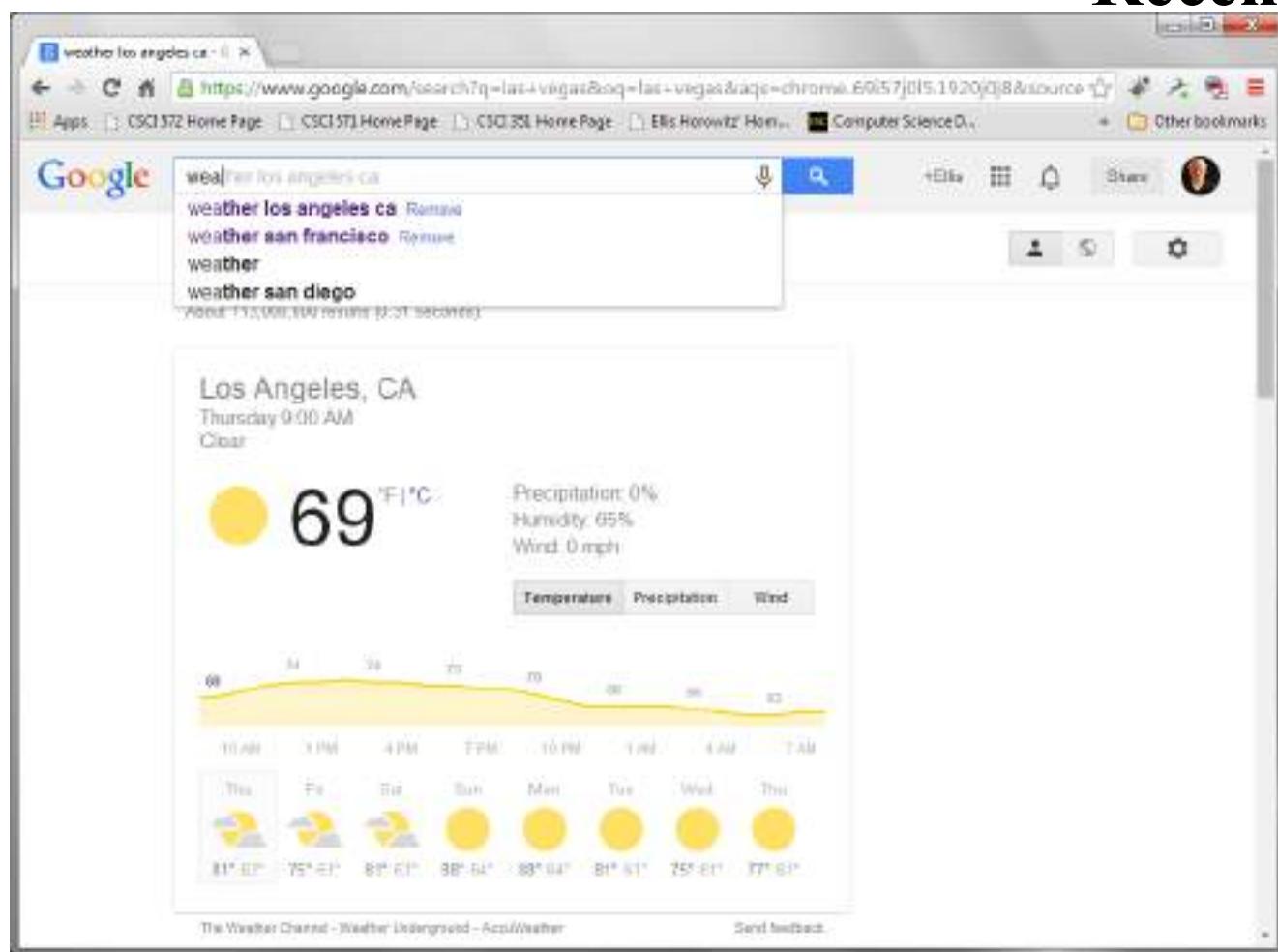


Query Processing is Now Extremely Clever

- Query processing involves much more than just matching query terms with document terms
- Semantic analysis of the query includes:
 1. Determining the language of the query
 2. Filtering of unnecessary words from the query (stop words)
 3. Looking for specific types of queries, e.g.
 - **Personalities (triggered on names)**
 - **Cities (travel info, maps)**
 - **Medical info (triggered on names and/or results)**
 - **Stock quotes, news (triggered on stock symbol)**
 - **Company info ...**
 4. Determining the user's location or the target location of the query
 5. Remembering previous queries
 6. Maintaining a user profile



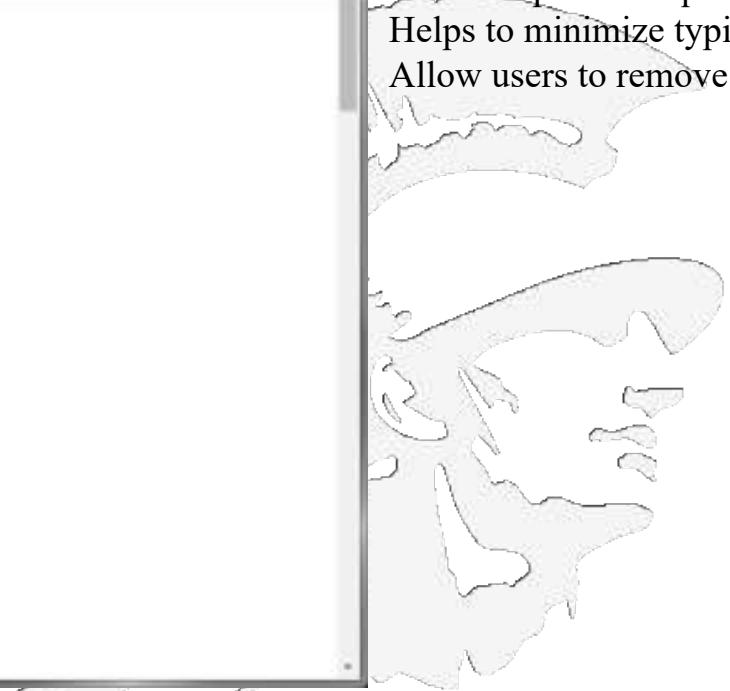
Google Maintains Your Recent Query History



A screenshot of a Google search results page. The search query "weather los angeles ca" is entered in the search bar. Below the search bar, a dropdown menu shows recent queries: "weather los angeles ca", "weather san francisco", "weather", and "weather san diego". The main search results show the weather forecast for Los Angeles, CA, with a current temperature of 69°F/15°C, 0% precipitation, 65% humidity, and 0 mph wind. A graph shows the temperature fluctuating between 68°F and 73°F over the next 24 hours. Below the graph, a weekly forecast is provided for Los Angeles, showing temperatures and weather conditions for each day from Friday to Thursday.

Day	Temp (F)	Temp (C)	Condition
Fri	61°	16°	Partly Cloudy
Sat	63°	17°	Partly Cloudy
Sun	68°	20°	Sunny
Mon	68°	20°	Sunny
Tue	61°	16°	Sunny
Wed	61°	16°	Sunny
Thu	67°	19°	Sunny

Maintain previous queries
Helps to minimize typing
Allow users to remove old ones



Results are Holistic A Person Query

File Edit View History Bookmarks Tools Help

george clooney - Google Search

<https://www.google.com/search?q=george+clooney>

Google george clooney

Web News Images Videos Shopping More Search tools

About 37,700,000 results (0.22 seconds)

In the news

 [George Clooney, Amal Alamuddin Honeymoon in New British Home](#)
Us Magazine · 4 hours ago
Newlyweds George Clooney and Amal Alamuddin have skipped the traditional far-flung

[People: George Clooney's Wedding Cost About \\$1.6 Million](#)
Yahoo! Voices · 2 days ago

[George Clooney & Amal Alamuddin Could Nab His & Hers Nobel Peace Prize, Friend Predicts](#)
People Magazine · 20 hours ago

[More news for george clooney](#)

George Clooney - Wikipedia, the free encyclopedia
en.wikipedia.org/wiki/George_Clooney · Wikipedia ·
George Timothy Clooney (born May 6, 1961) is an American actor and filmmaker. He has received three Golden Globe Awards for his work as an actor and two Academy Awards, one for acting and the other for producing.
[Wikipedia](#)

George Clooney & Amal Alamuddin Could Nab His & Hers ...
www.people.com/article/george-clooney-amal-alamuddin-nobel-peace-prize · People · 23 hours ago · SEE 26 MORE CLOONEY WEDDING PHOTOS! Subscribe now to PEOPLE'S digital edition and get 36 BONUS photos from inside the ...

George Clooney - IMDb
www.imdb.com/name/nm000123/ · Internet Movie Database ·



[More images](#)

George Clooney
Actor

George Timothy Clooney is an American actor and filmmaker. He has received three Golden Globe Awards for his work as an actor and two Academy Awards, one for acting and the other for producing.

Born: May 6, 1961 (age 58), Lexington, KY

Height: 5' 11" (1.80 m)

Spouse: Amal Alamuddin (m. 2014); Talia Balsam (m. 1989–1993)

Siblings: Adela Clooney

Parents: Nina Bruce Warren, Nick Clooney

Includes the following:

Latest news

Biography

Photos

Basic facts

born

married

parents

career

Results are Holistic A Place Query

Google Search results for "las vegas":

About 168,000,000 results (0.41 seconds)

Las Vegas - VEGAS.com™
www.vegas.com/ Las Vegas Shows, Hotels and more. The Official VEGAS Travel Site™
 Vegas.com has 767 followers on Google+
 Las Vegas Air + Hotel - Las Vegas Hotels - Headliners and Concerts

Attractions in Las Vegas - TravelNevada.com
www.travelnvada.com/ Order Your Free Visitor's Guide & Plan Your Vacation To Las Vegas!
 Travel Nevada has 1,238 followers on Google+

Las Vegas 5 Day Room Sale - SouthPointCasino.com
www.southpointcasino.com/ South Point Hotel Casino Spa Rates From \$36 Sun-Thurs \$60 Fr. & Sat.
 Restaurants - Hotel Packages & Promos - Nightlife & Entertainment - Casino

(Official City of Las Vegas Web Site)
www.lasvegashappy.com/ Las Vegas
 The City of Las Vegas (Official Government Site) ... City of Las Vegas: Serving You Online Rather Than In Line; Photo: Downtown Las Vegas

Las Vegas Hotels, Shows, Casinos, Restaurants, Maps and ...
www.lasvegas.com/ Your official What happens in Vegas, stays in Vegas resource. Plan hotels and things to do for your trip on the only official website of Las Vegas.
 Shows & Events - Las Vegas Hotels - Air + Hotel Packages - Special Offers & Deals




Map data ©2011 Google

Las Vegas

26,003 followers on Google+

[Follow](#)

Las Vegas, officially the City of Las Vegas and often known as simply Vegas, is the most populous city in the U.S. state of Nevada and the county seat of Clark County. Wikipedia

Founded: May 11, 1905

Area: 136.9 sq miles (352 km²)

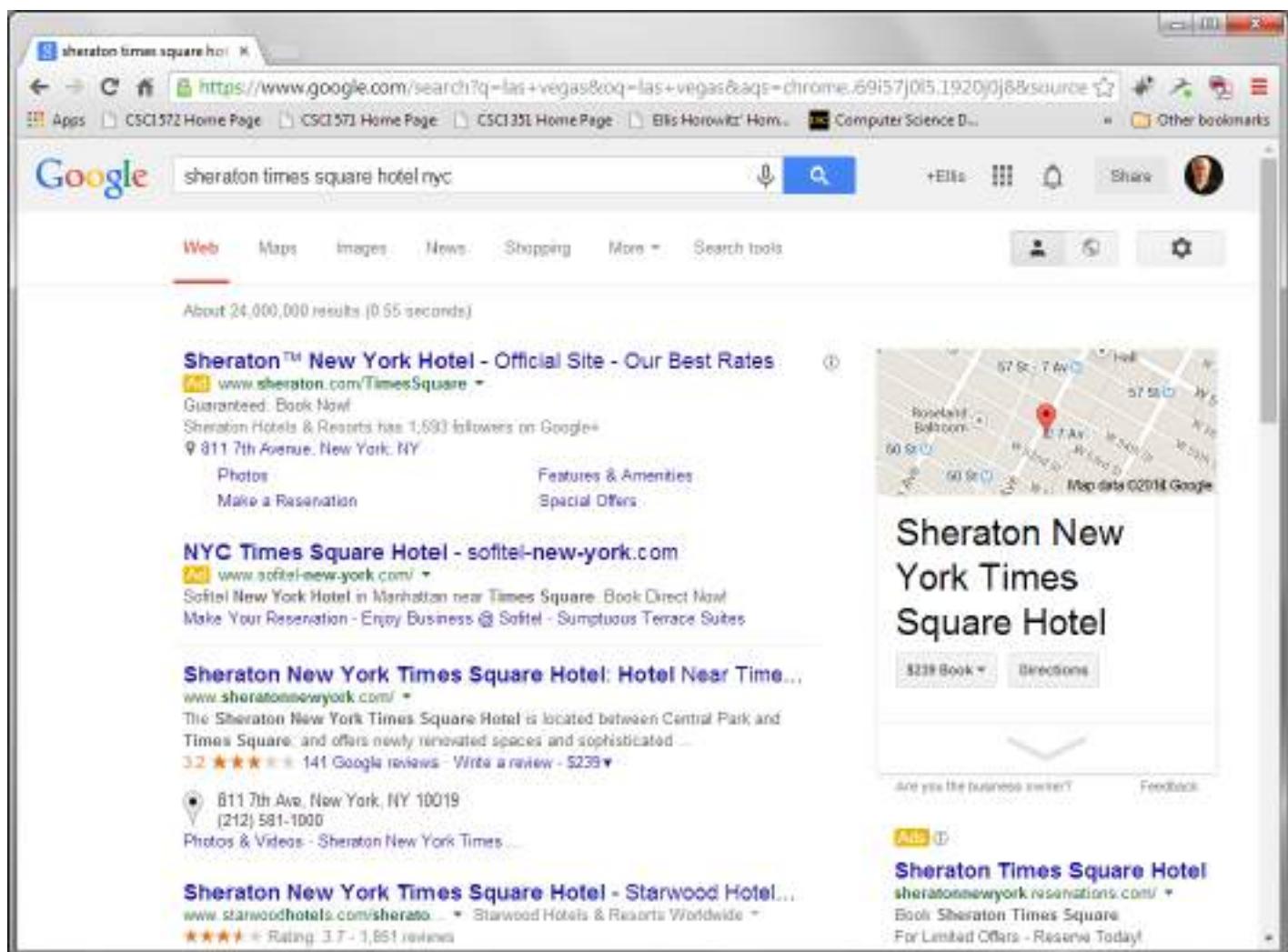
Weather: 87°F (31°C), Wind N at 5 mph (8 km/h), 5% Humidity

Local time: Thursday 9:32 AM
Dowload time: 2012-09-10 09:40:00

Includes the following:

Official site
 Map
 Essential facts
 founded
 area
 weather
 time
 population

Results are Holistic An Hotel Query



The screenshot shows a Google search results page for the query "sheraton times square hotel nyc". The results include:

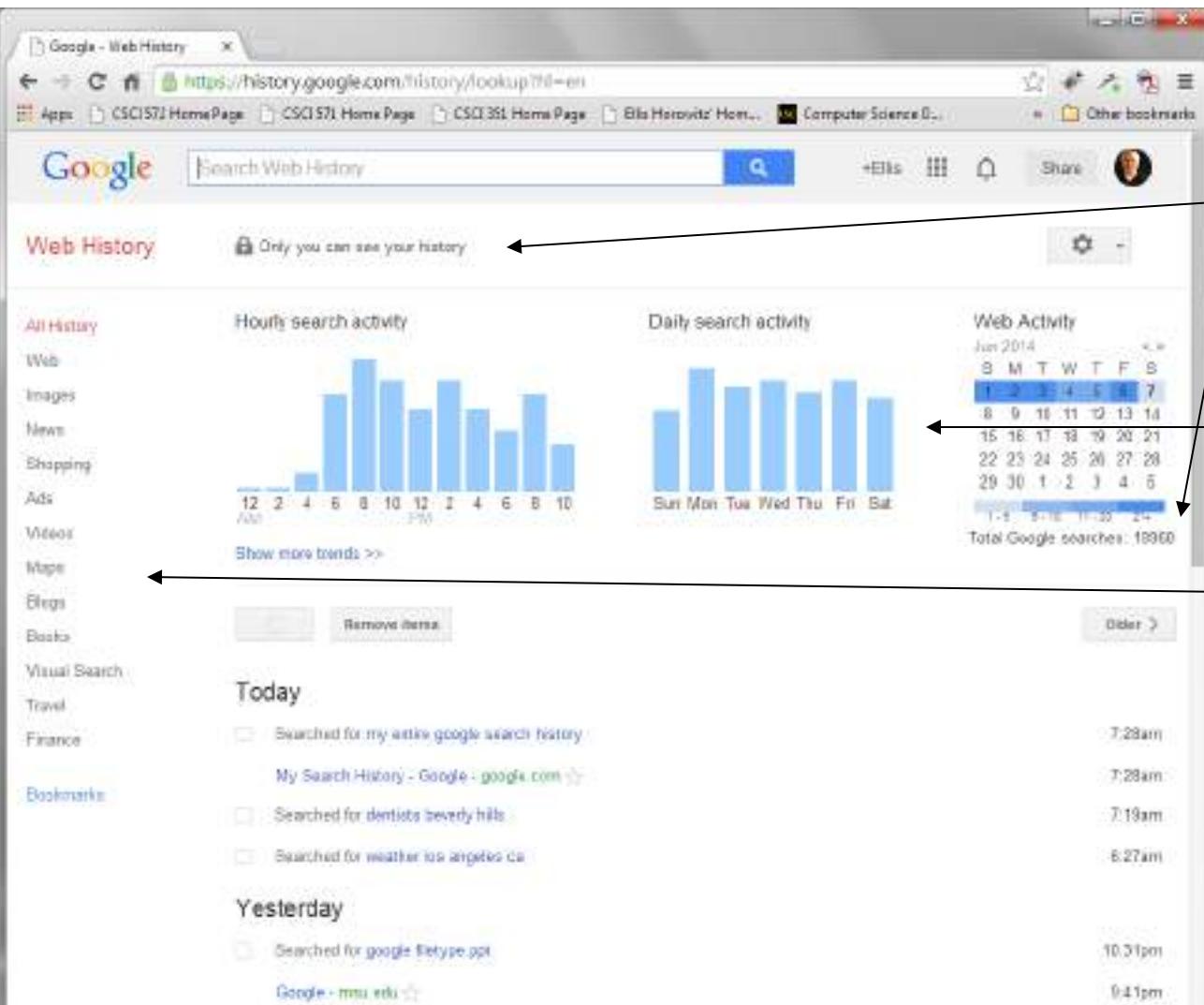
- Sheraton™ New York Hotel - Official Site - Our Best Rates**
www.sheraton.com/TimesSquare •
Guaranteed Book Now!
Sheraton Hotels & Resorts has 1,693 followers on Google+
811 7th Avenue, New York, NY
Photos Features & Amenities
Make a Reservation Special Offers
- NYC Times Square Hotel - softel-new-york.com**
www.softel-new-york.com •
Softel New York Hotel in Manhattan near Times Square. Book Direct Now!
Make Your Reservation - Enjoy Business @ Softel - Sumptuous Terrace Suites
- Sheraton New York Times Square Hotel: Hotel Near Time...**
www.sheratonnewyork.com •
The Sheraton New York Times Square Hotel is located between Central Park and Times Square, and offers newly renovated spaces and sophisticated...
3.2 ★★★★ 141 Google reviews - Write a review - \$239 •
811 7th Ave, New York, NY 10019
(212) 581-1000
Photos & Videos - Sheraton New York Times...
- Sheraton New York Times Square Hotel - Starwood Hotel...**
www.starwoodhotels.com/sheraton • Starwood Hotels & Resorts Worldwide •
★★★☆ Rating: 3.7 - 1,861 reviews

To the right of the search results, there is a map showing the location of the Sheraton New York Times Square Hotel at 811 7th Avenue, New York, NY 10019. Below the map is a snippet from the hotel's Google My Business profile:

Sheraton New York Times Square Hotel
sheratonnewyork.reservations.com •
Book: Sheraton Times Square
For Limited Offers - Reserve Today!

Includes the following:
Main hotel website
Map
Address
Phone number
Price of a room
Directions

Google Retains a User's Entire Query History!



The screenshot shows the Google Web History interface. On the left, a sidebar lists categories like All History, Web, Images, News, Shopping, Ads, Videos, Maps, Blogs, Books, Visual Search, Travel, Finance, and Bookmarks. The main area displays two bar charts: 'Hourly search activity' and 'Daily search activity'. Below these are sections for 'Today' and 'Yesterday', each listing search queries with their times. A summary on the right shows 'Web Activity' for June 2014, with a total of 18,960 queries.

Category	Value
Total Google searches	18,960
Hourly search activity (approximate values)	12, 2, 4, 6, 8, 10, 12, 2, 4, 6, 8, 10
Daily search activity (approximate values)	Sun: 4, Mon: 8, Tue: 7, Wed: 7, Thu: 6, Fri: 7, Sat: 5
Today's queries (approximate times)	Searched for my active google search history (7:28am), My Search History - Google - google.com (7:28am), Searched for dentists beverly hills (7:19am), Searched for weather los angeles ca (6:27am)
Yesterday's queries (approximate times)	Searched for google filetype:pdf (10:31pm), Google - mru.edu (9:41pm)

They claim that only I can see my history;
I have issued a total of 18,960 queries;

Graphs show my queries by hour and by week;

I can view my Web queries as distinct from my Image queries or my News queries, etc

As a result, Google now knows a great deal about us!

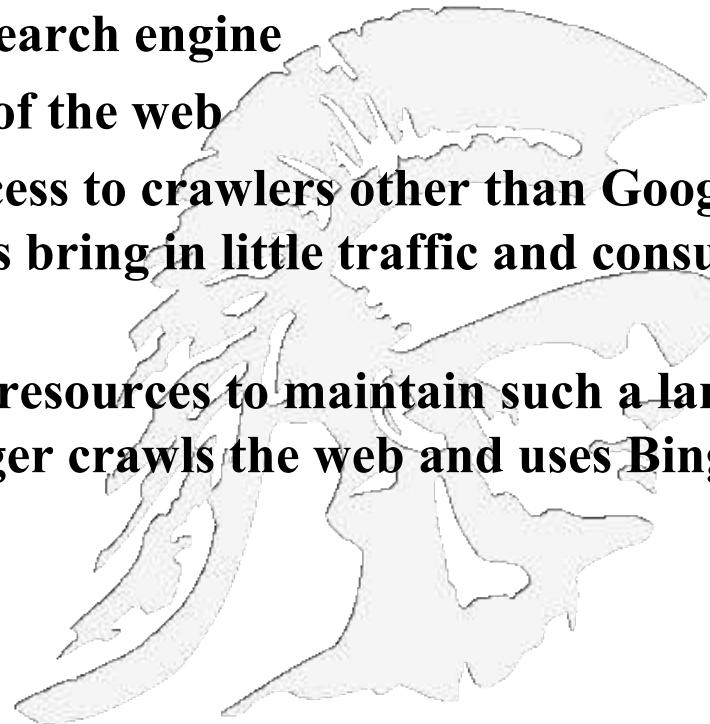
Search Engines are an Industry

- The search engine industry is 20+ years old, having started with WebCrawler and Lycos in 1994 who sold banner ads as their business model
- Search engine revenue today
 - **Google:** 2021: \$257 Billion; 2020: \$181 Billion; 2019: \$162 Billion; 2018: \$116 Billion; 2017: \$109 Billion; 2016: \$90 Billion; 2015: \$74.5 Billion; 2014: \$66 Billion; 2013: \$37 Billion
 - **Baidu:** 2021: \$31 Billion; 2020: \$16.4 Billion; 2019: \$15 Billion; : \$11.3 Billion; 2017: \$13 Billion; 2016: \$10.1 Billion; 2015: \$10.2 Billion; 2014: 8.0 Billion
 - **Yahoo:** 2021: 5.2Billion; 2019: 6.97Billion; 2018: 3.03 Billion; 2017: 3.0 Billion; 2016: 2.98 Billion; 2015: \$4.9 Billion; 2014: 4.6 Billion; 2013: 4.6Billion
 - **Bing:** 2020 \$7.74 Billion; 2019: \$7.63 Billion; 2018: \$7.01 Billion
 - Microsoft says that in Q1 2016 Bing became profitable

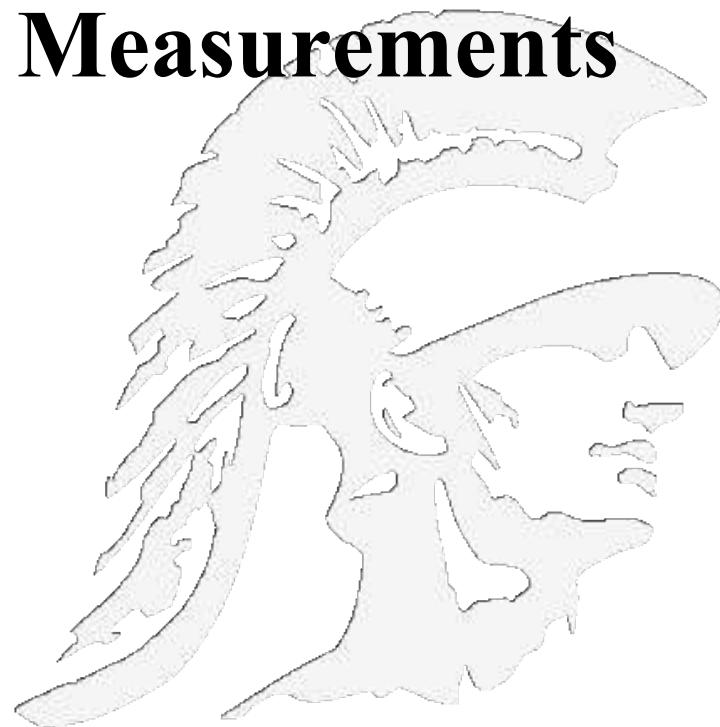


Google is a Monopoly Gatekeeper for the Internet

- The US is suing Google for anti-Trust violations
 - Google claims it has strong competition in search!!!
 - Google has sweetheart deals with Apple and pays Apple \$8+ Billion/year to be their default search engine
- Google maintains the largest index of the web
 - Some websites actually deny access to crawlers other than Google and Bing as these other crawlers bring in little traffic and consume server cycles
 - Only Google and Bing have the resources to maintain such a large index, e.g. DuckDuckGo no longer crawls the web and uses Bing's index

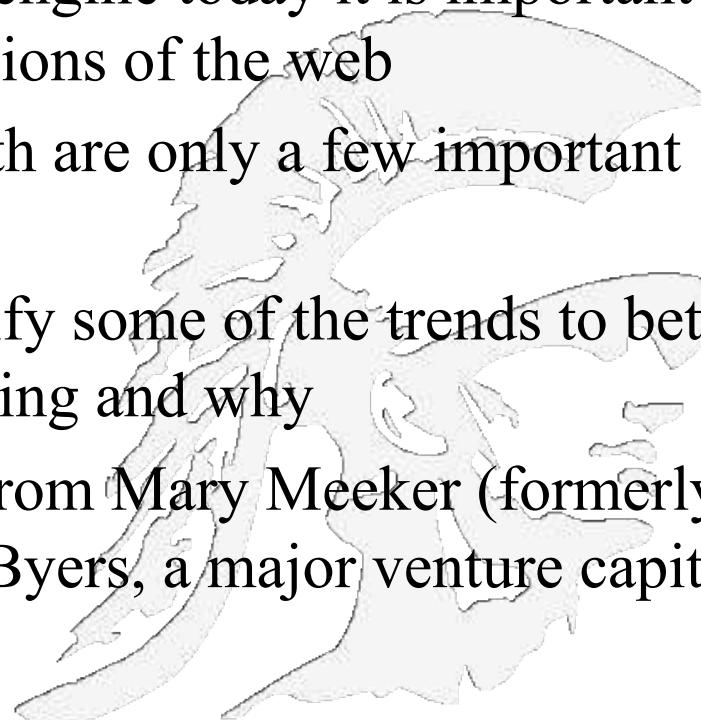


Web Trends and Measurements

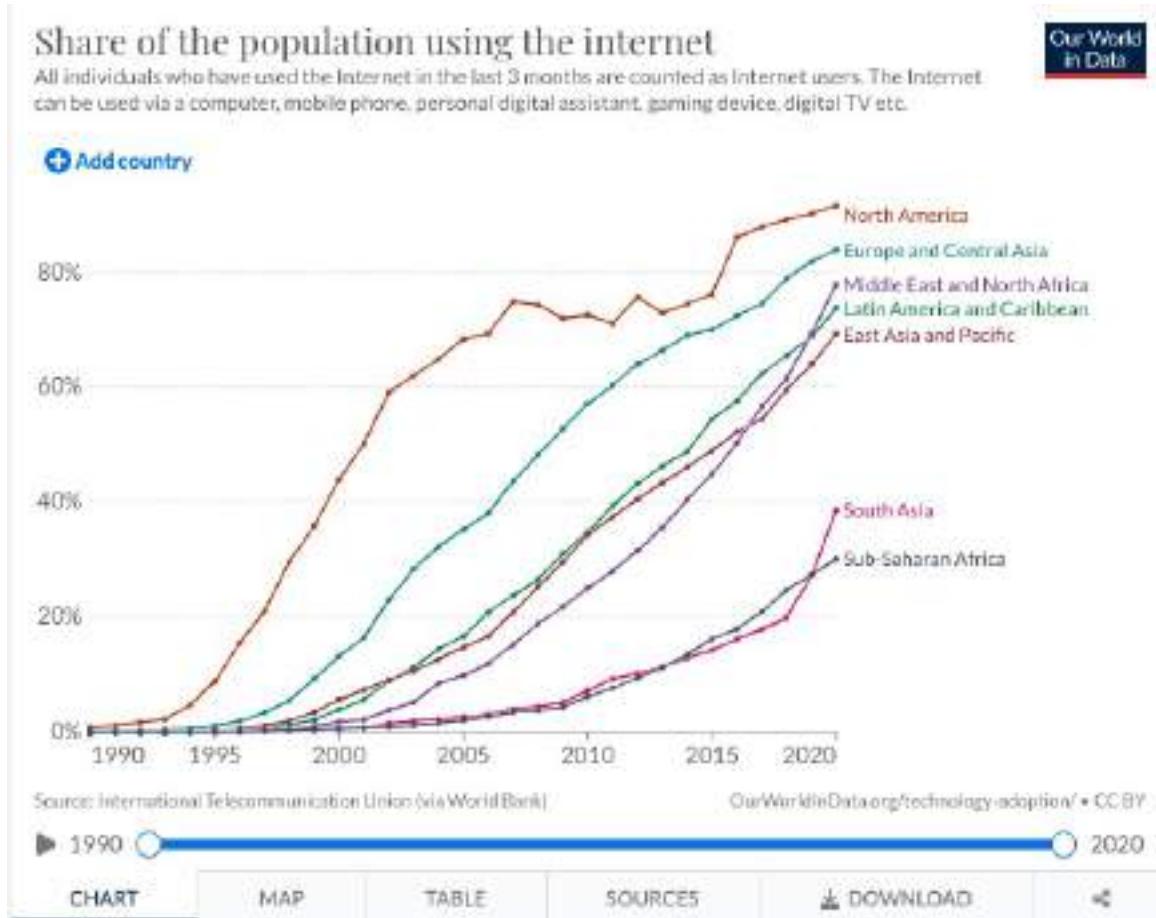


Web Trends

- Web has changed dramatically over the last 30+ years
- If one is building a web search engine today it is important to understand the different dimensions of the web
 - Scale, complexity and growth are only a few important factors
- In today's lecture I try to quantify some of the trends to better understand where the web is going and why
- many of the early slides come from Mary Meeker (formerly of) Kleiner, Perkins, Caufield and Byers, a major venture capital firm, <http://www.kpcb.com/>



A total of **5.03 billion** people around the world use the internet today – equivalent to 63.1 percent of the world's total population

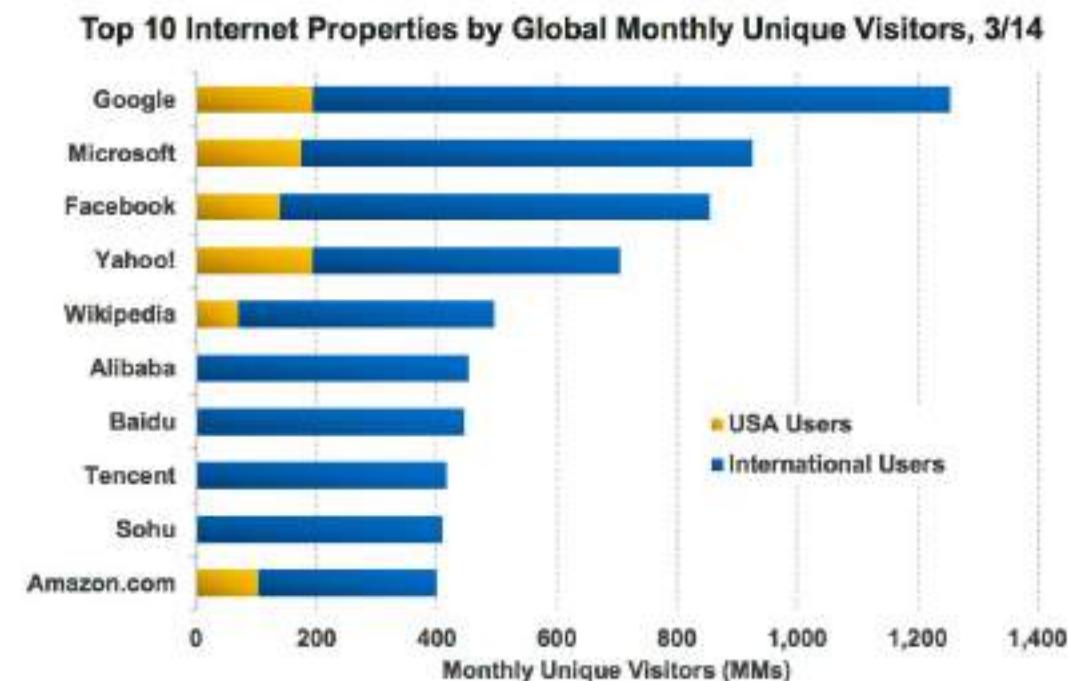


The US leads in the development of highly popular Internet websites;

Baidu is a Chinese search engine

Tencent is a Chinese holding company of Internet properties, among the most popular being, QQ, for chatting;
 Sohu.com Inc. is a Chinese online media, search, gaming, community and mobile service group.

**3/14 – 6 of Top 10 Global Internet Properties ‘Made in USA’...
 >86% of Their Users Outside America...China Rising Fast**



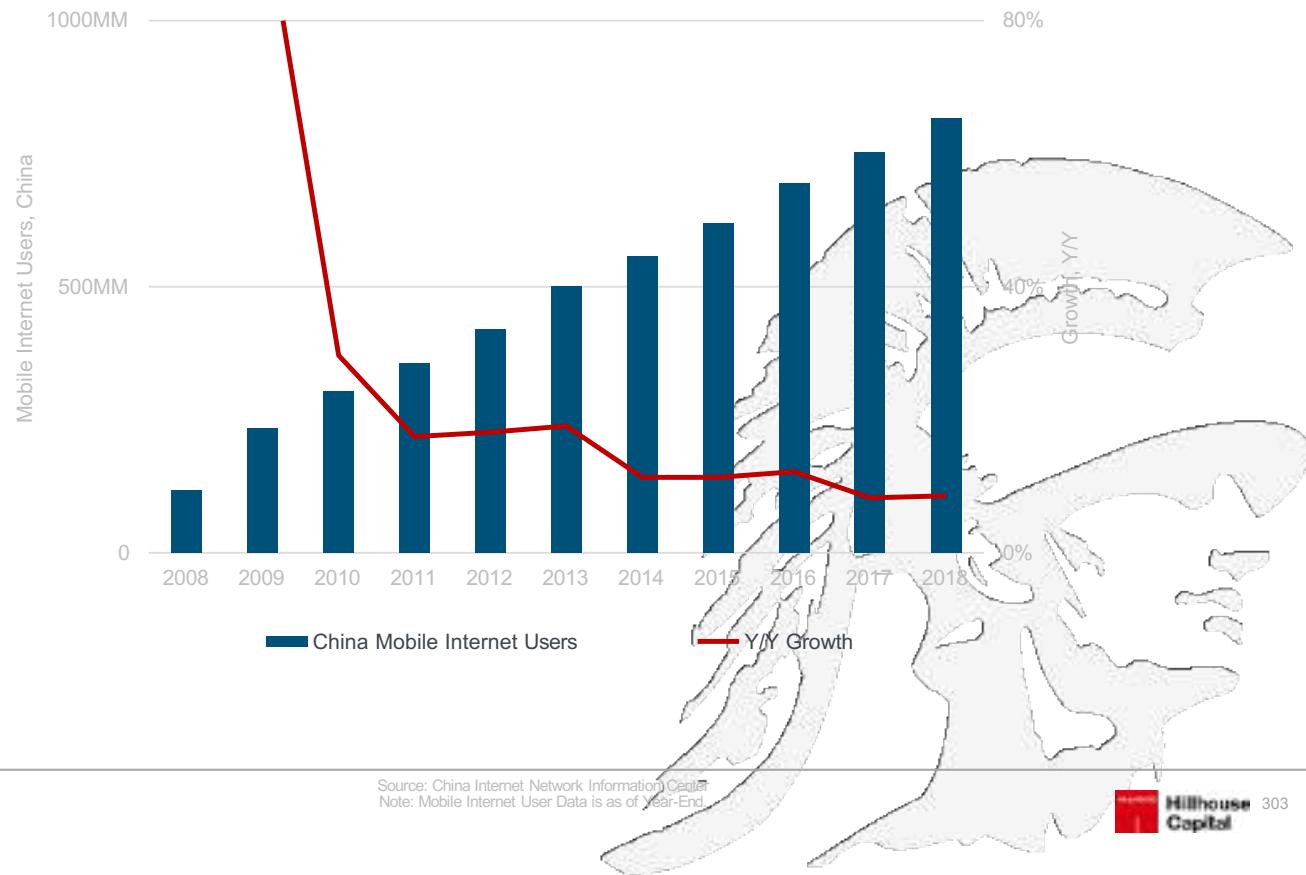
@KPCB Source: comScore, 3/14

131



China Mobile Internet Users = 817MM.+9% vs. +8% Y/Y

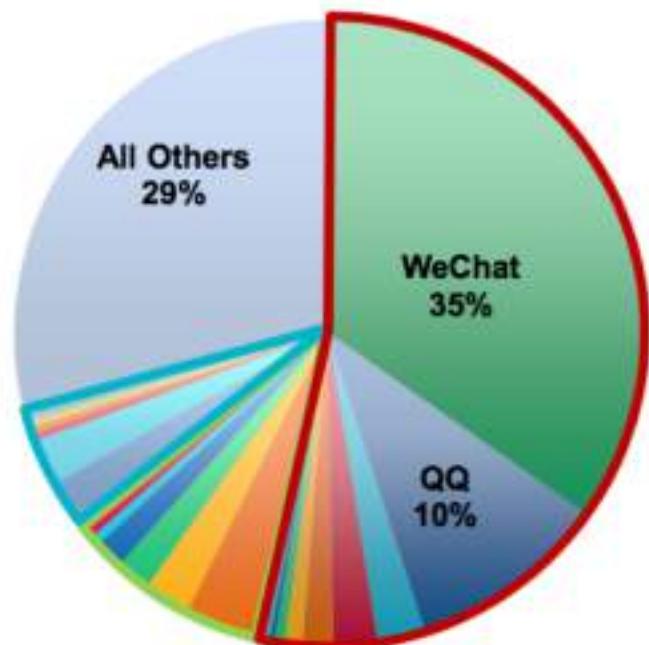
China Mobile Internet Users vs. Y/Y Growth



China Mobile Internet Usage Leaders...

Tencent + Alibaba + Baidu = 71% of Mobile Time Spent

Share of Mobile Time Spent, April 2016
 Daily Mobile Time Spent = ~200 Minutes per User, Average



Tencent

Alibaba

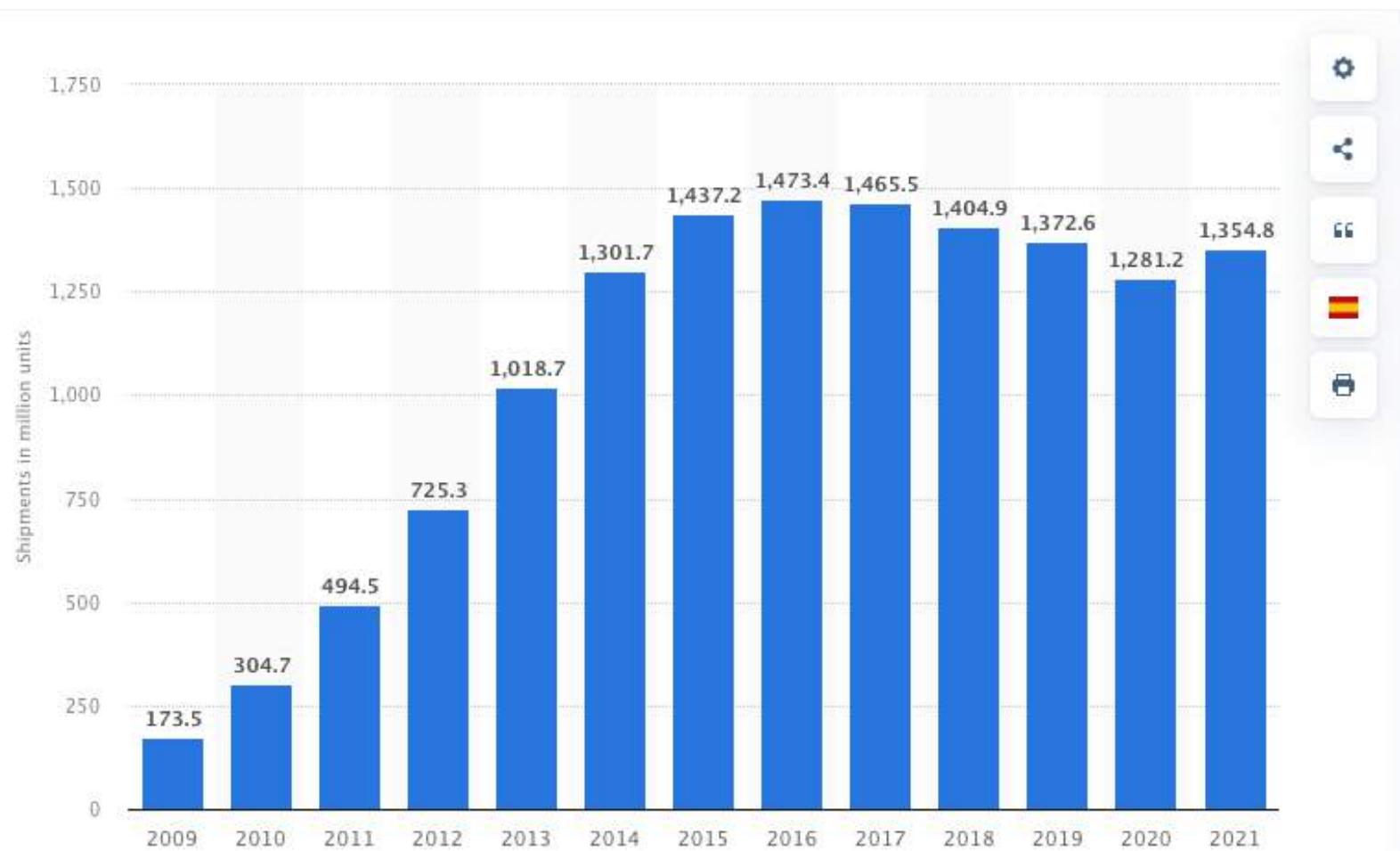
Baidu

- WeChat
- QQ
- QQ Browser
- Tencent Video
- Tencent News
- Tencent Games
- QQ Music
- JD.com
- QQ Reading

- UCWeb Browser
- Taobao
- Weibo
- YouKu Video
- Momo
- Shuqi Novel
- AliPay
- AutoNavi

- Mobile Baidu
- iQiyi / PPS Video
- Baidu Browser
- Baidu Tieba
- 91 Desktop
- Baidu Maps
- All Other

Global smartphone shipments from 2009 to 2021



World's Content is Increasingly Findable + Shared + Tagged -
Digital Info Created + Shared up 9x in Five Years

There has been exponential growth in online information;

1 Zettabyte = 1,024 Exabytes

1 Exabyte = 1,024 Petabytes

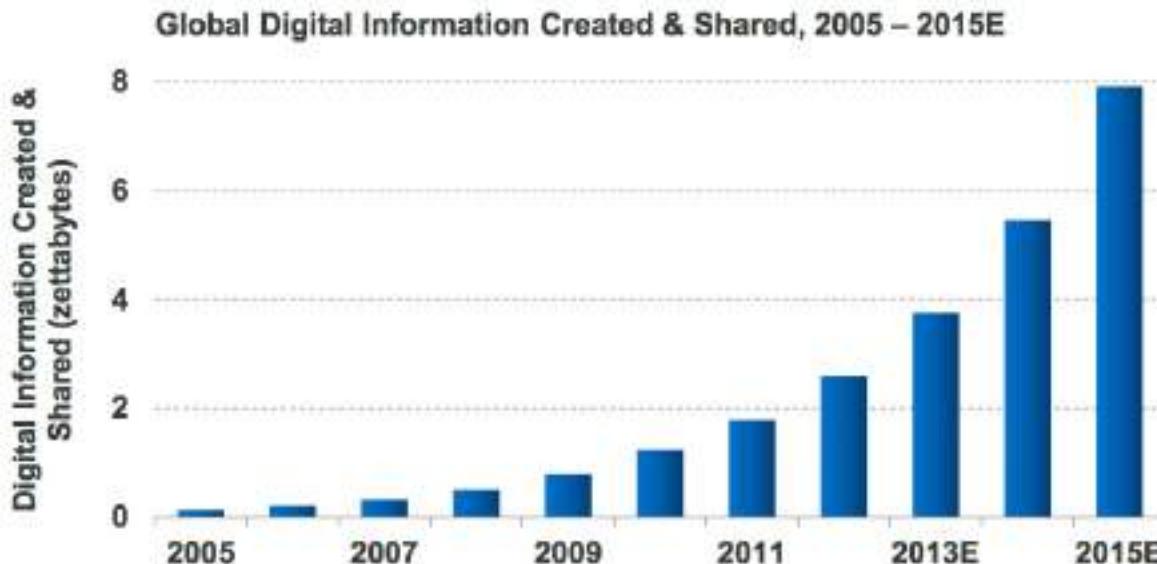
1 Petabyte = 1,024 Terabytes

1 Terabyte = 1,024 Gigabytes

or

1 Zettabyte = 1,000,000,000,000
gigabytes

*Amount of global digital information created & shared
– from documents to pictures to tweets –
grew 9x in five years to nearly 2 zettabytes* in 2011, per IDC.*

 KPCB

Note: * 1 zettabyte = 1 trillion gigabytes. Source: IDC report 'Extracting Value from Chaos' 6/11. 11



Photos Alone = 1.8B+ Uploaded & Shared Per Day... Growth Remains Robust as New Real-Time Platforms Emerge

500 million photos are uploaded every day and that number is doubling every year

Yahoo has recently made a major upgrade to **Flickr**

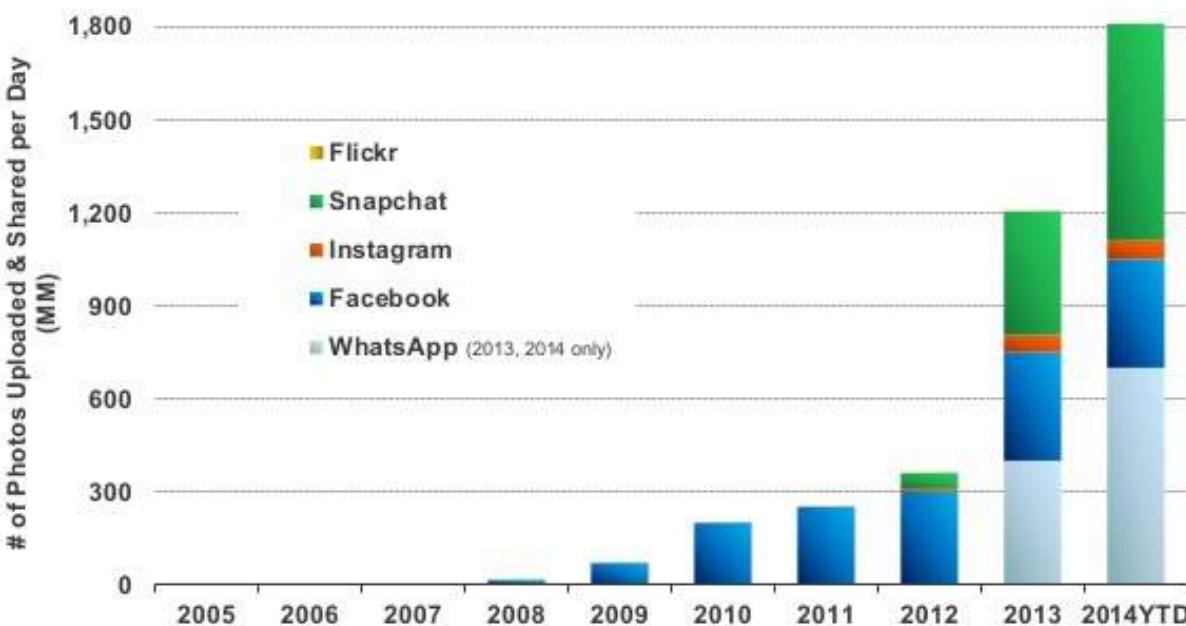
Instagram was in 2010 purchased by Facebook for \$1 billion

Snapchat is a photo messaging application developed by two Stanford students (\$9B valuation);



bobby Murphy - Evan Spiegel

Daily Number of Photos Uploaded & Shared on Select Platforms,
2005 – 2014YTD



@KPCB

Source: KPCB estimates based on publicly disclosed company data. 2014 YTD data per latest as of 5/14.

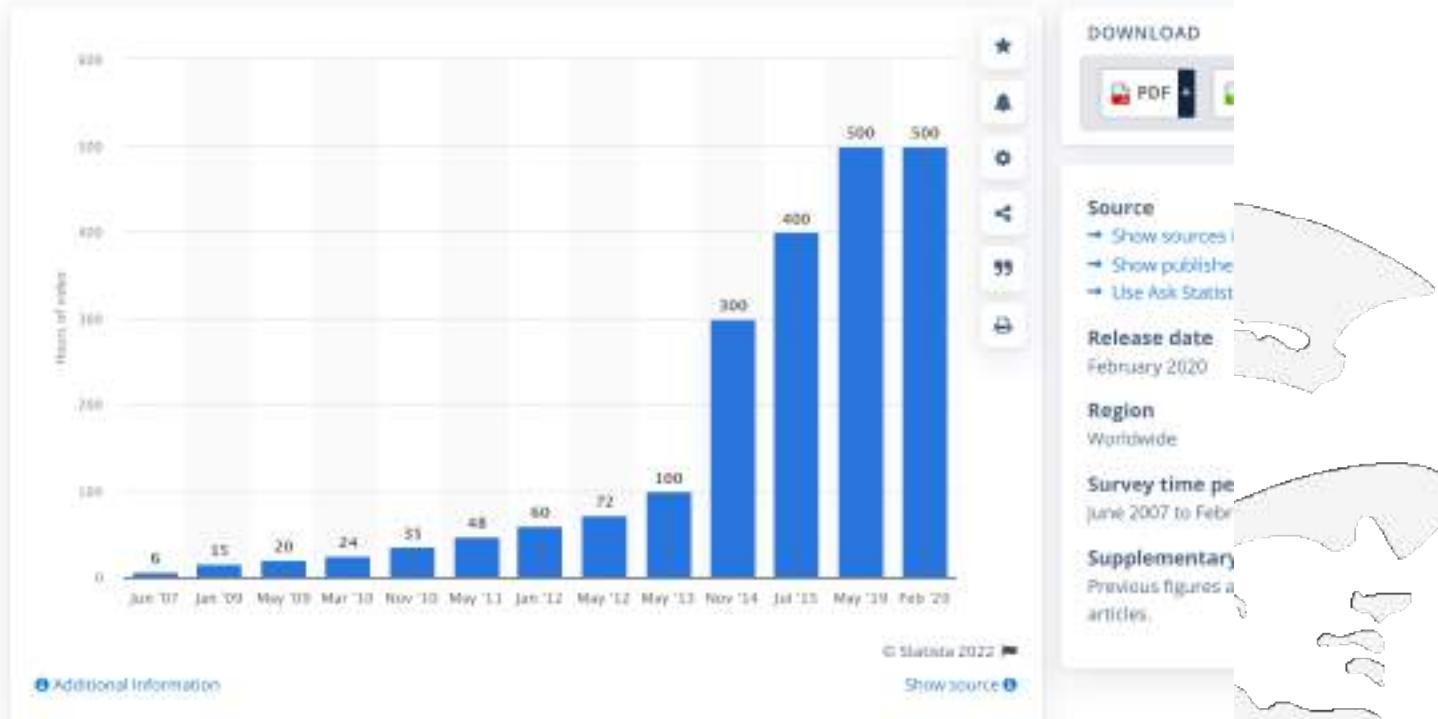
62



Content Uploaded to YouTube

Internet > Online Video & Entertainment

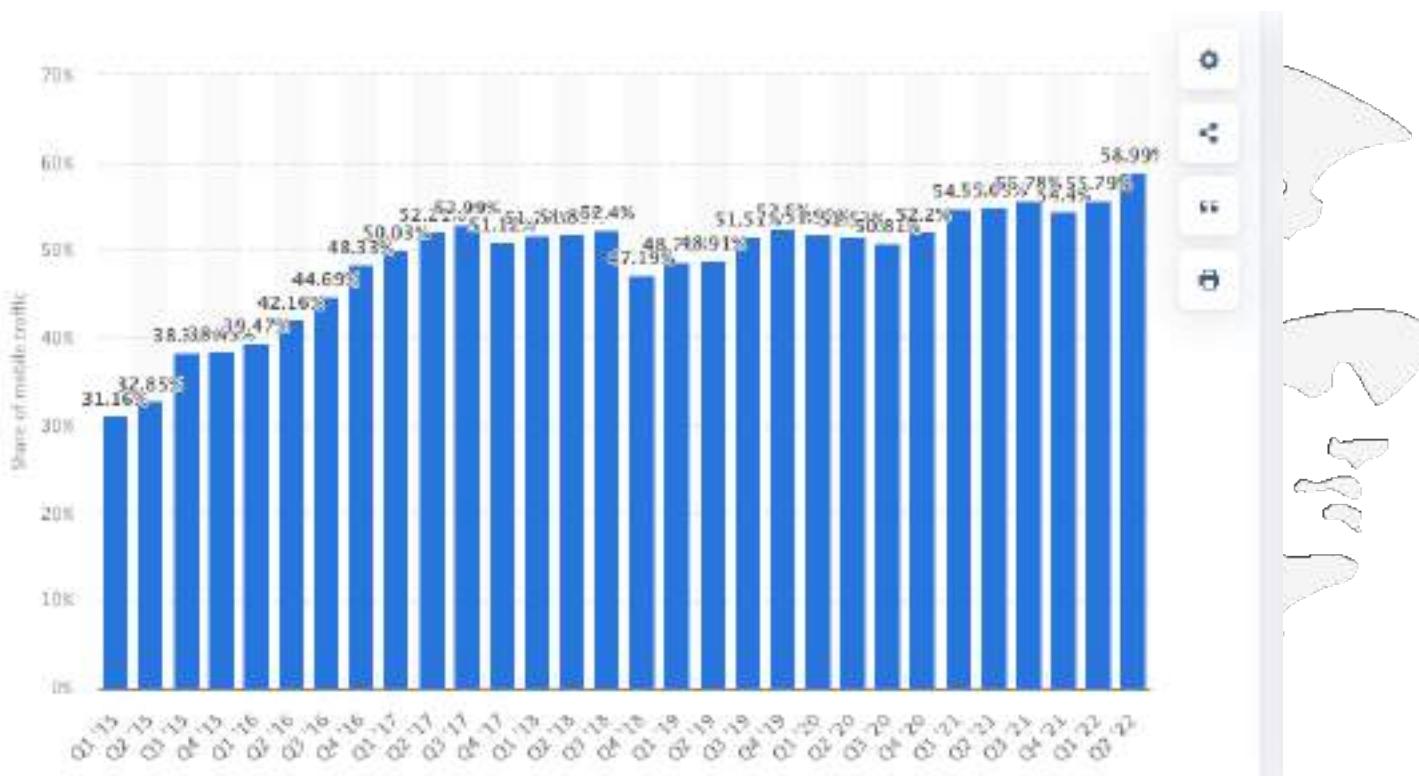
Hours of video uploaded to YouTube every minute as of February 2020



- As of February 2020, more than 500 hours of video were uploaded to YouTube every minute.
- This equates to approximately 30,000 hours of newly uploaded content per hour.
- The number of video content hours uploaded every 60 seconds grew by around 40 percent between 2014 and 2020.

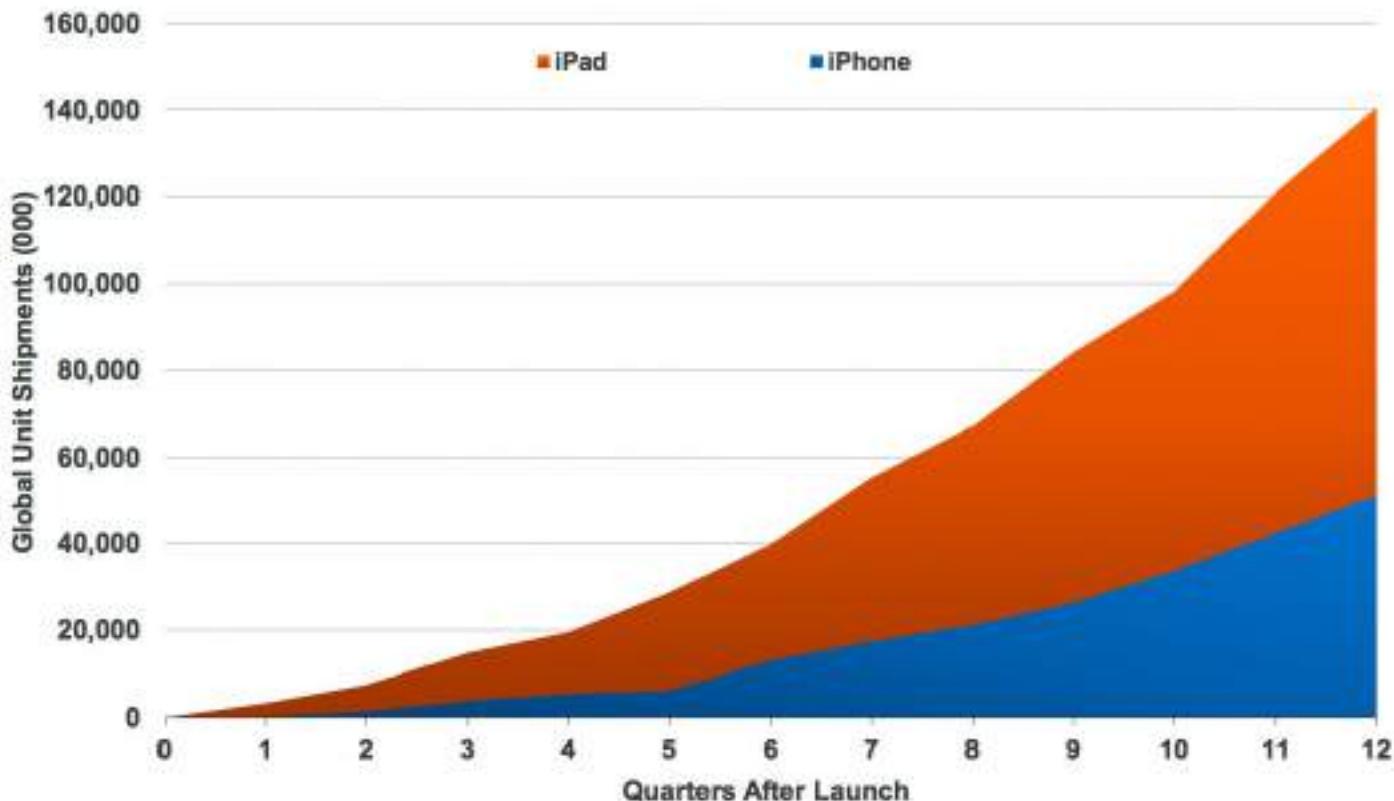
Mobile Accounts for 50% of Web Traffic

In the second quarter of 2022, mobile devices (excluding tablets) generated 58.99 percent of global website traffic, consistently hovering around the 50 percent mark since the beginning of 2017 before permanently surpassing it in 2020.



Tablet Growth = More Rapid than Smartphones, iPad = ~3x iPhone Growth

First 12 Quarters Cumulative Unit Shipments, iPhone vs. iPad



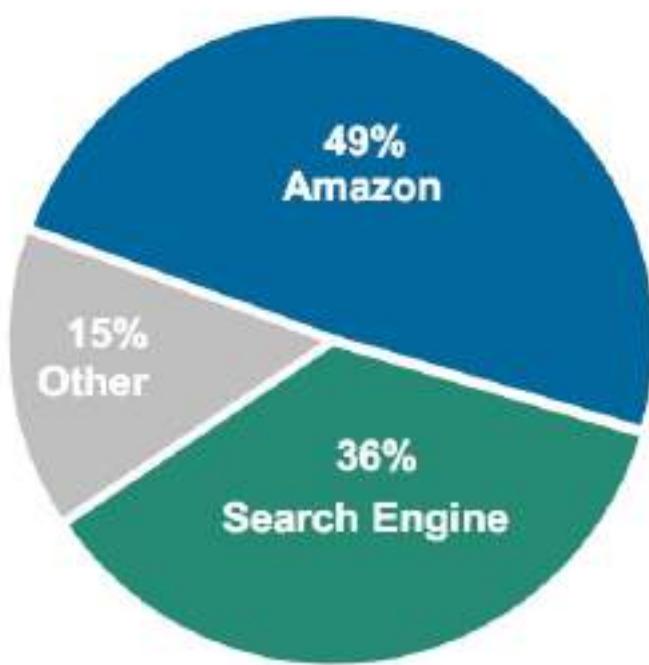
Source: Apple, as of CQ1:13 (12 quarters post iPad launch).
Launch Dates: iPhone (6/29/07), iPad (4/3/10).

44

KPCB

Product Finding =
Often Starts @ Search (Amazon + Google...)

Where Do You Begin Your Product Search?



Technology Cycles – Still Early Cycle on Smartphones + Tablets, Now Wearables Coming on Strong, Faster than Typical 10-Year Cycle

Technology Cycles Have Tended to Last Ten Years

Mainframe Computing
1960s



Mini Computing
1970s



Personal Computing
1980s



Desktop Internet Computing
1990s



Mobile Internet Computing
2000s



Wearable / Everywhere Computing
2014+



Others?

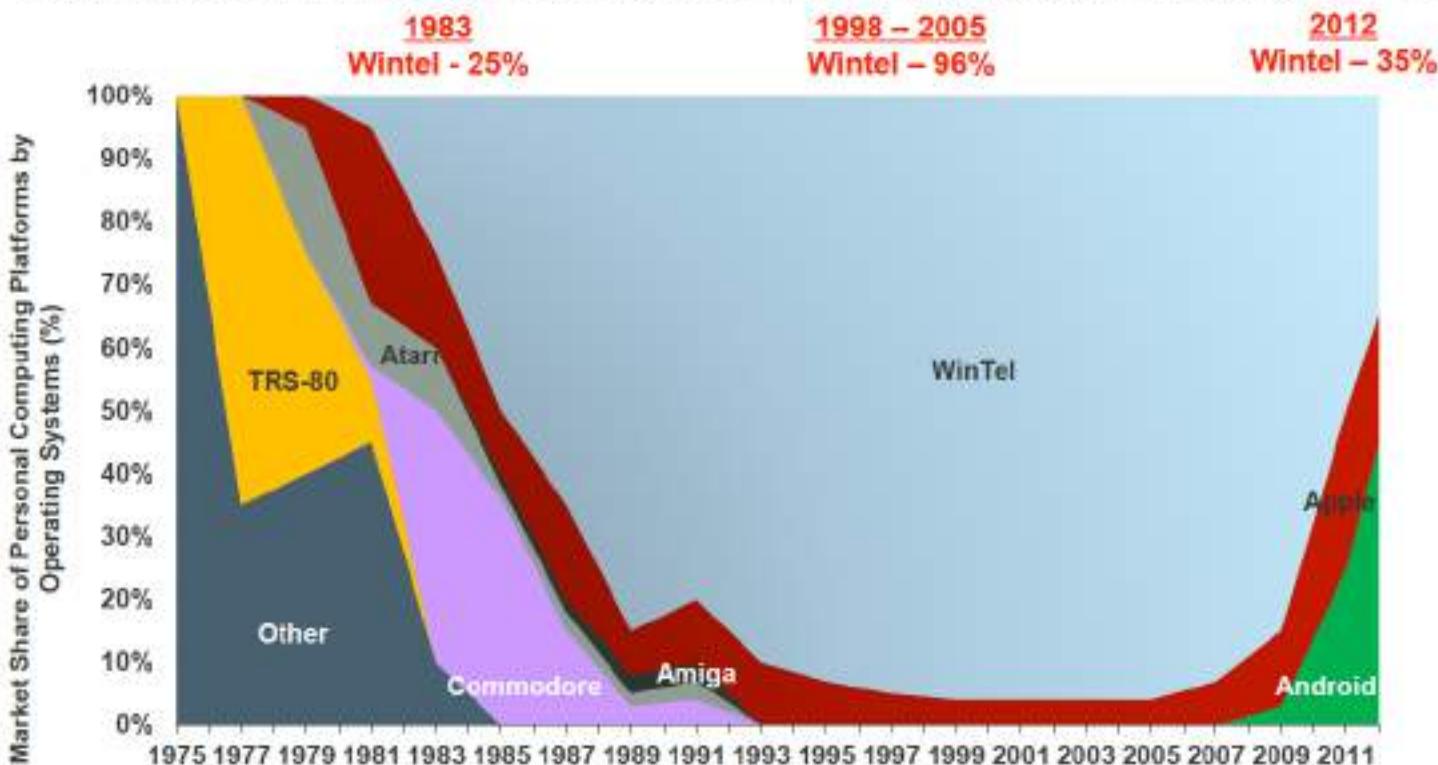
KPCB

Image Source: Computerscienceslab.com, Wikipedia, IBM, Apple, Google, NTT docomo, Google, Jawbone, Pebble.

49

Re-Imagination of Computing Operating Systems - iOS + Android = 60% Share vs. 35% for Windows

Global Market Share of Personal Computing Platforms by Operating System Shipments, 1975 – 2012



KPCB

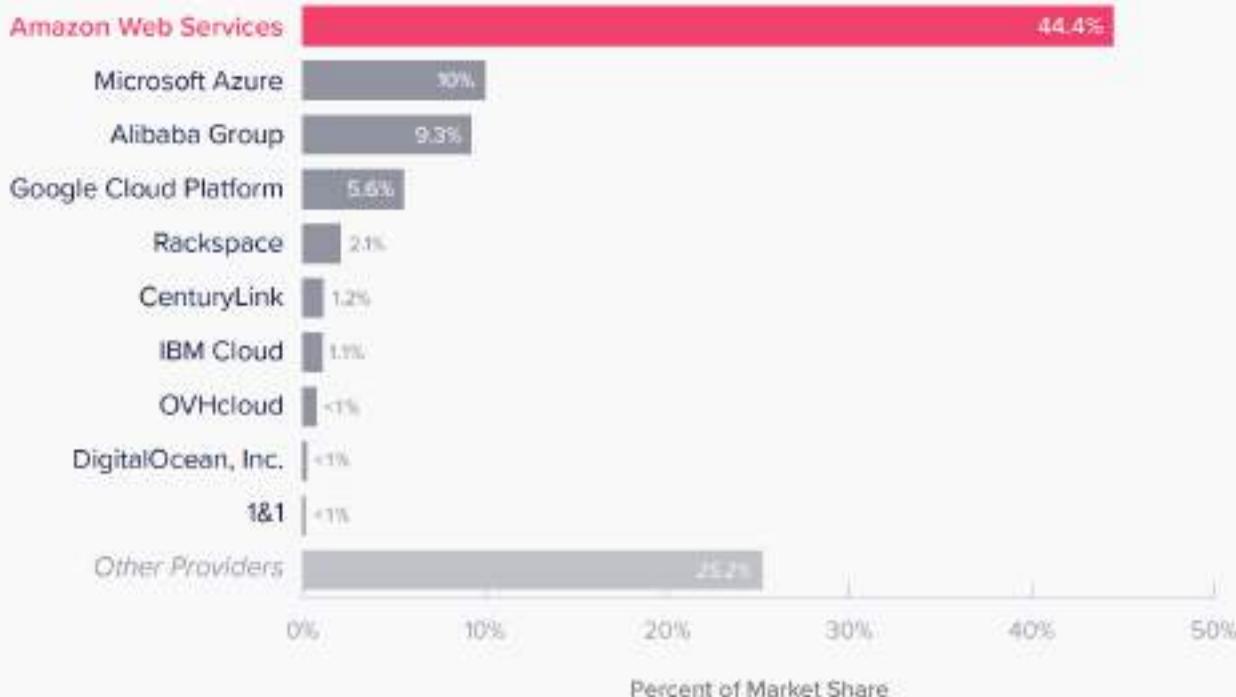
Source: Asymco.com (as of 2011), Public Filings, Morgan Stanley Research, Gartner for 2012 data.

109

Major Cloud Providers

Market Share Of Leading Cloud Hosting Providers

Top 10 Providers by Total 2020 Market Share

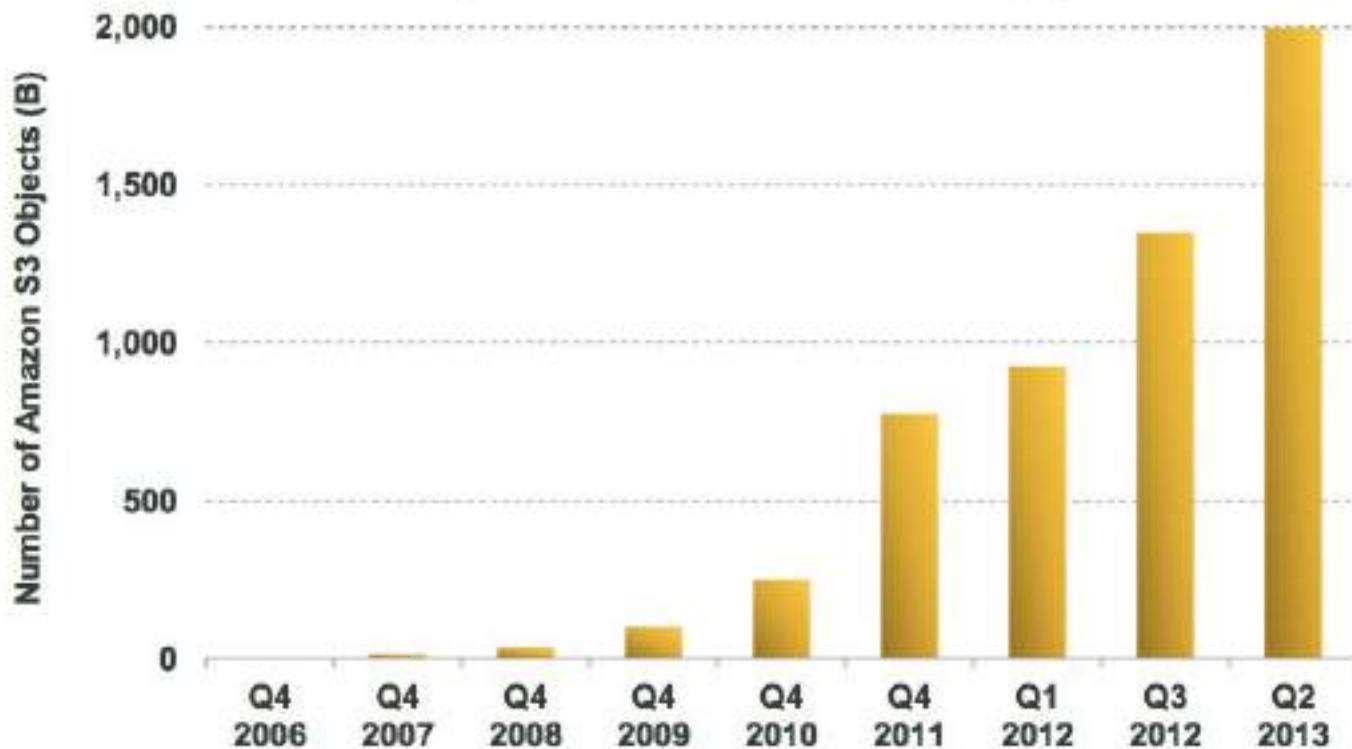


Source: Intricately data, April 2021

...While The Cloud Rises

Amazon Web Services (AWS) Leading Cloud Charge...

Objects Stored in Amazon S3* (B)



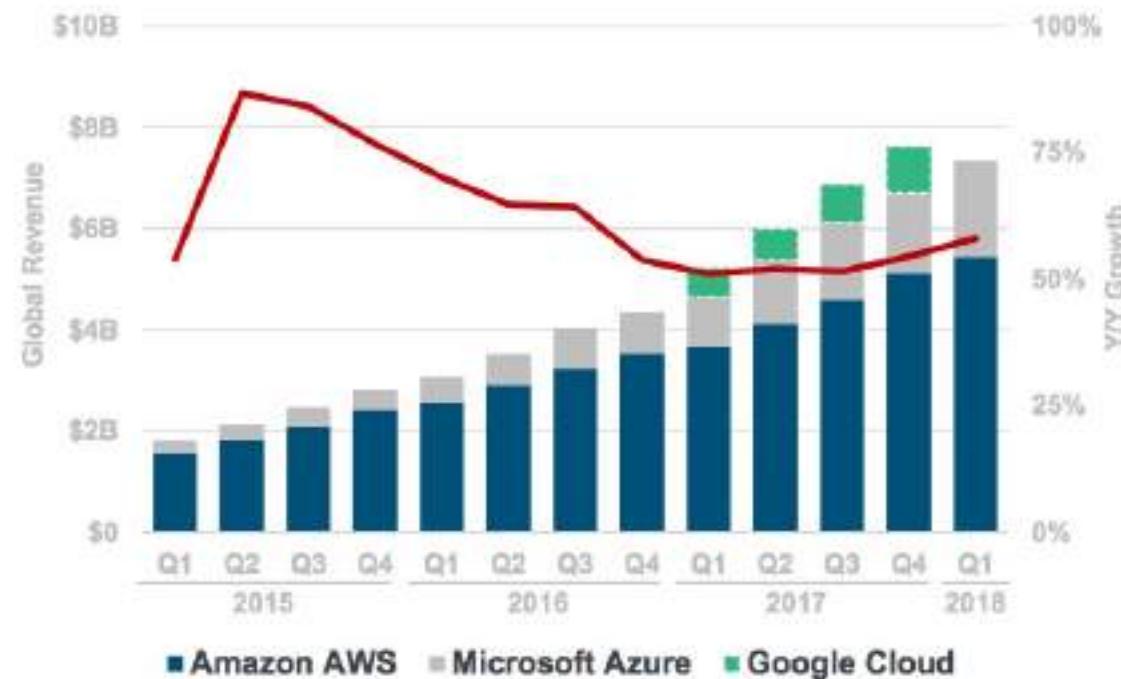
*Note: S3 is AWS' storage product and used as proxy for AWS scale / growth.
Source: Company data.



74

...Computing Big Bangs Volume Effects = Cloud Revenue Re-Accelerating +58% vs. +54% Q/Q

Cloud Service Revenue – Amazon + Microsoft + Google



KLEINER PERKINS
2018
INTERNET TRENDS

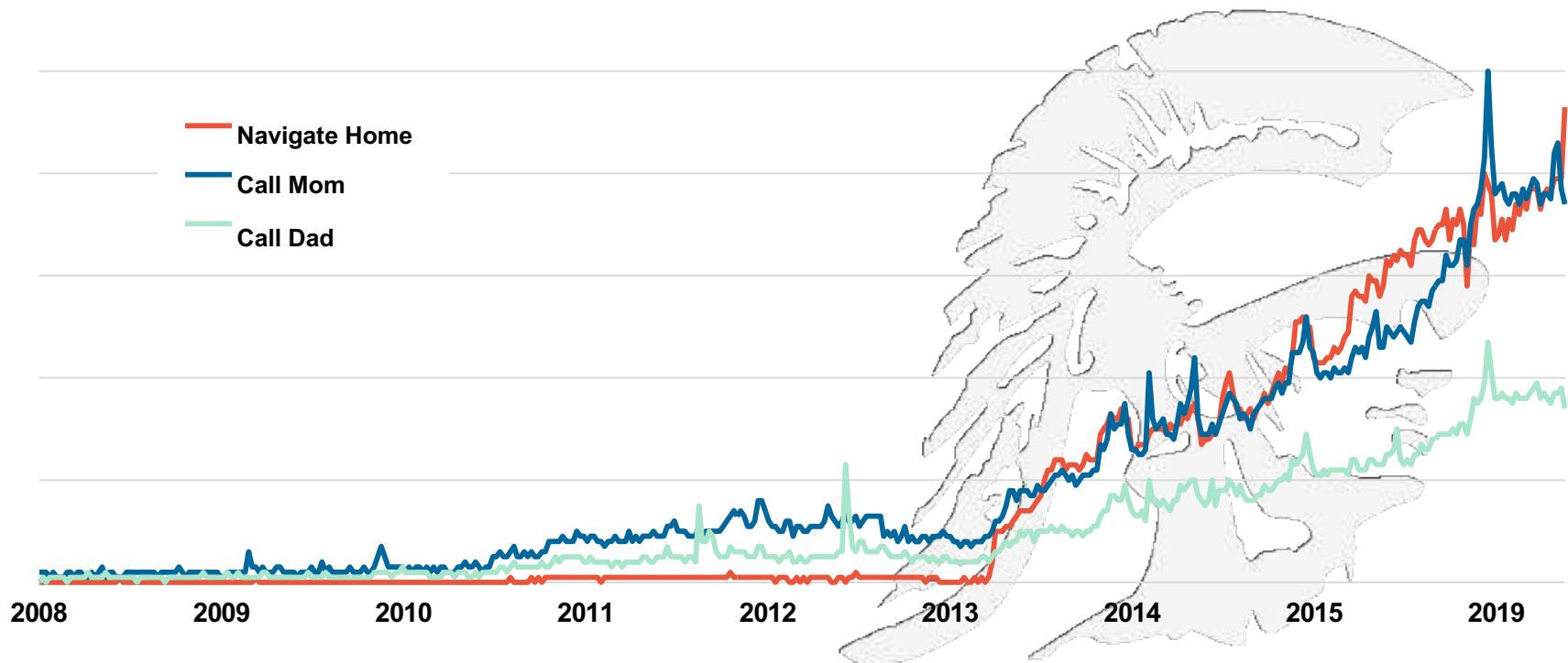
Sources: Amazon AWS = Company filings; Microsoft Azure = Keith Weiss @ Morgan Stanley (4/18); Google Cloud = Brad Routh @ Morgan Stanley (5/18). Total: Google Cloud revenue assumed to Y/Y growth rate published due to limited availability information.

182

Google Voice Search Queries

Google Trends imply queries associated with voice-related commands have risen >35x since 2008 after launch of iPhone & Google Voice Search

Google Trends, Worldwide, 2008 – 2019

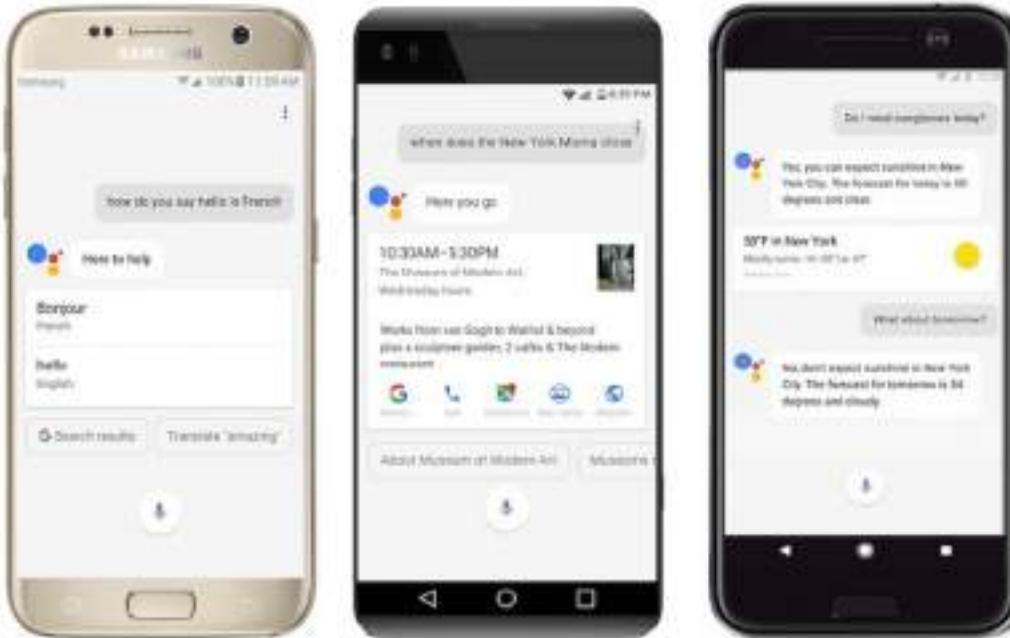


Voice-Based Mobile Platform Front-Ends = Voice Can Replace Typing

Google Assistant

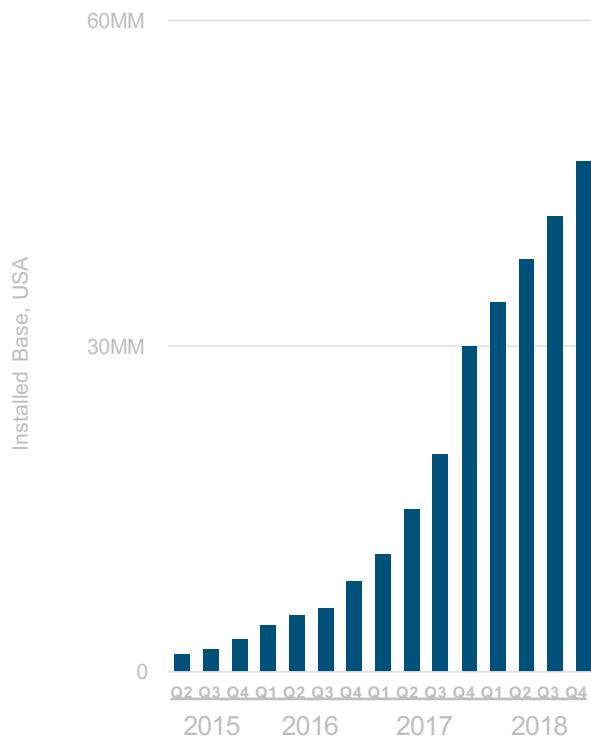
Nearly 70% of Requests are Natural / Conversational Language, 5/17

20% of Mobile Queries Made via Voice, 5/16

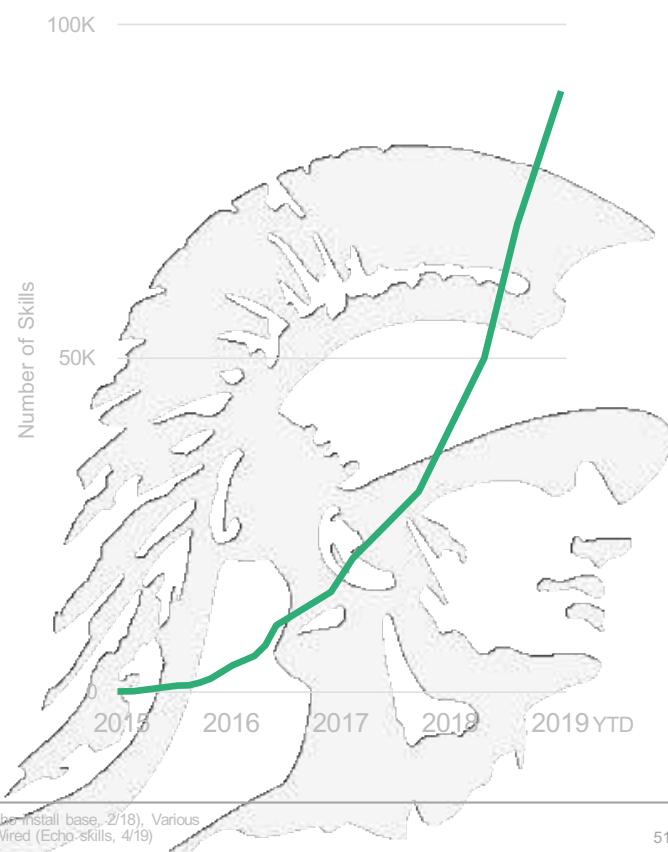


.Voice = 47MM Amazon Echo Base + ~2x in One Year

Amazon Echo Installed Base



Amazon Echo Skills



BOND
Internet Trends
2019

Source: Consumer Intelligence Research Partners LLC (Echo install base, 2/18), Various media outlets including Geekwire, TechCrunch & Wired (Echo-skills, 4/19)

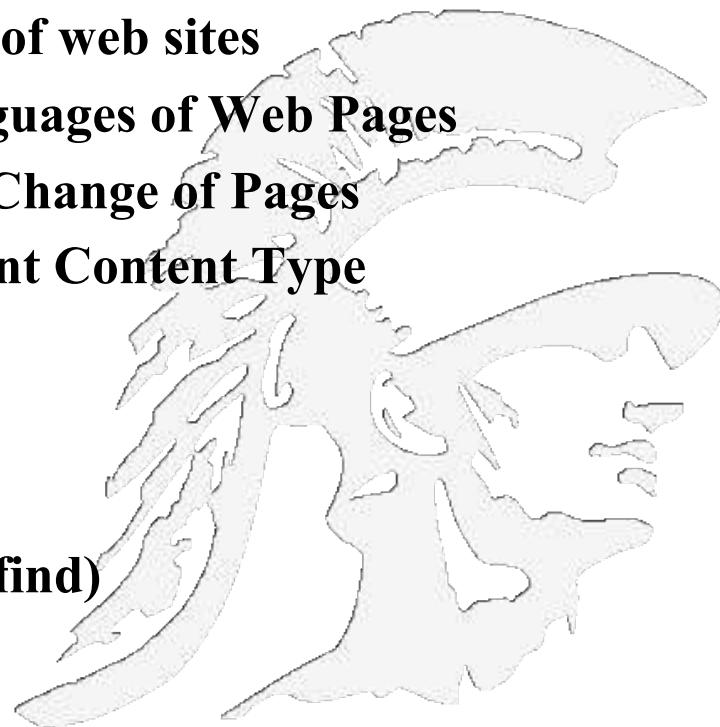
51

Summary of Recent Trends in Web/Internet Development

- Growth in number of users connected
- Growth in Smartphone use
- Growth in digital data, especially photos and video
- Growth in Social Media as an advertising platform
- Transition from desktop/laptop use to mobile
- Growth in tablet usage over desktops/laptops
- Decreased dominance of Microsoft Windows
- Move away from server farms to cloud computing
- Growth in voice communication with devices

Measuring the Web

- The World Wide Web (the Web, the publicly accessible web) is so dynamic it is hard to describe it and have the description be valid for very long
- In this lecture we look at what is known,
 - Measuring the Web by number of web sites
 - Measuring the Web by the Languages of Web Pages
 - Measuring the Web by Rate of Change of Pages
 - Measuring the Web by Document Content Type
 - Measuring the Web by linkage
 - Measuring the Web as a Graph
 - Measuring the Web by Content
 - (using the best statistics we can find)



Number of Websites

Jan. 2020:

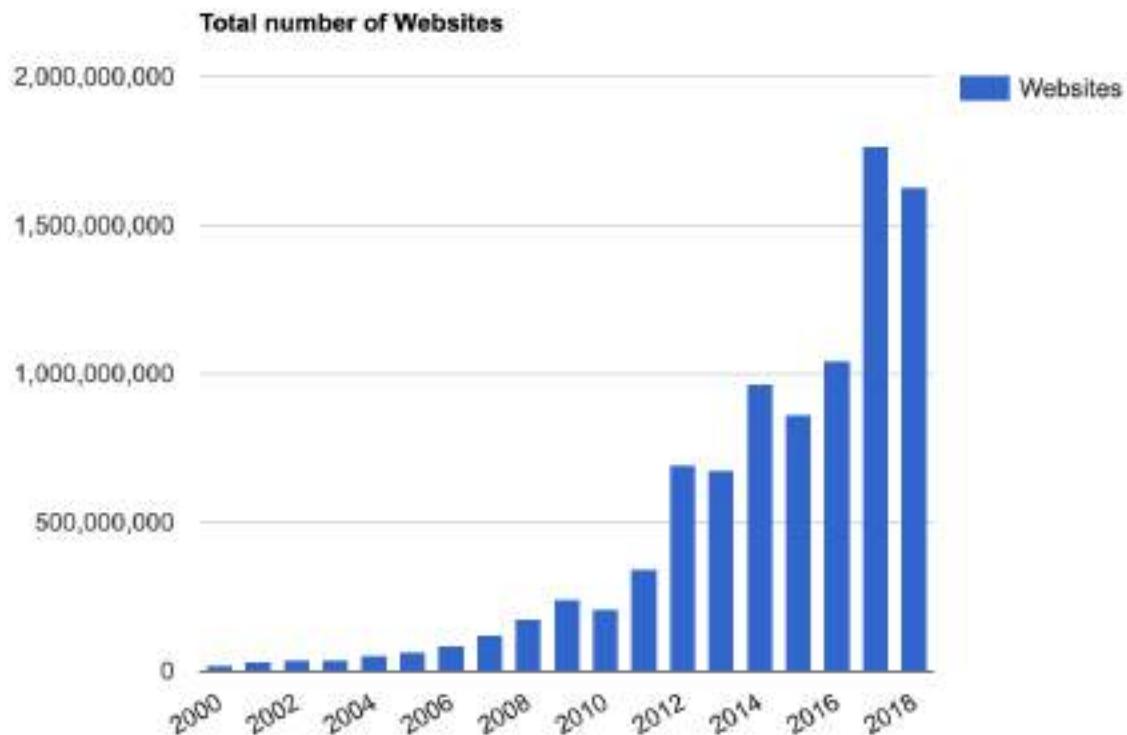
~1.7 Billion sites

nginx web server
had the largest
growth;

Over 50% of websites
Are hosted either by
Apache or nginx;

But Microsoft web
servers still power
43.2% of all sites

Around 75% of websites
are not active, but parked



<http://www.internetlivestats.com/total-number-of-websites/>

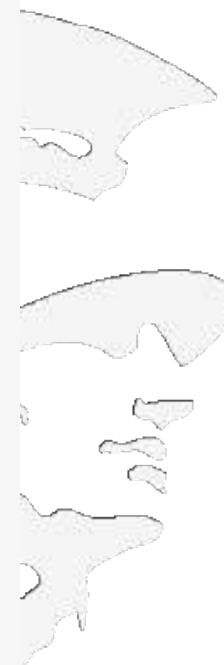
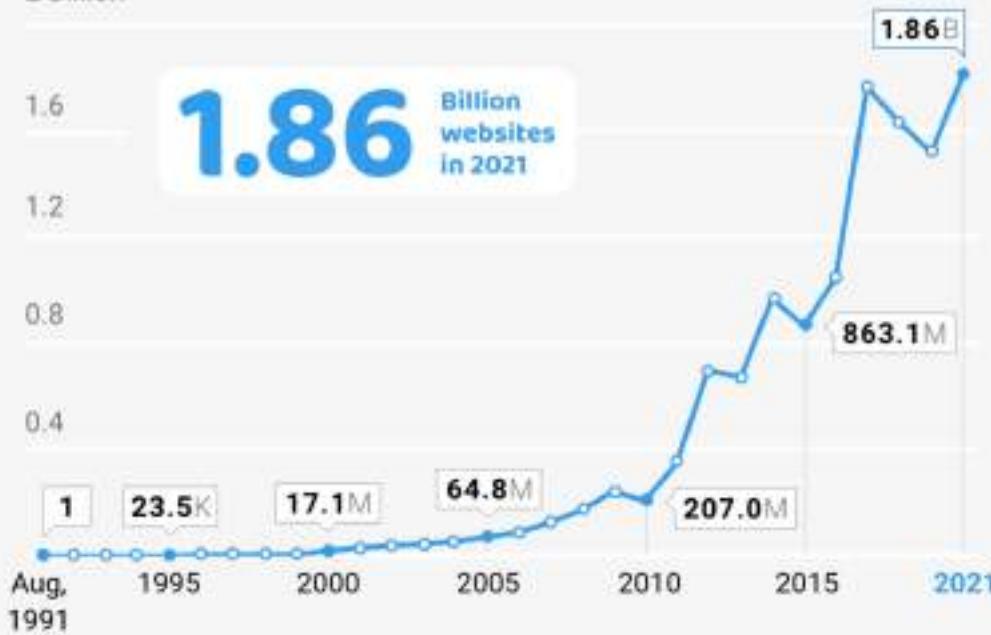
Number of websites in the world



The global number of websites has more than doubled from 2015 to 2021. Websites growth rate from 1991 to 2021

No. of websites:

2 Billion



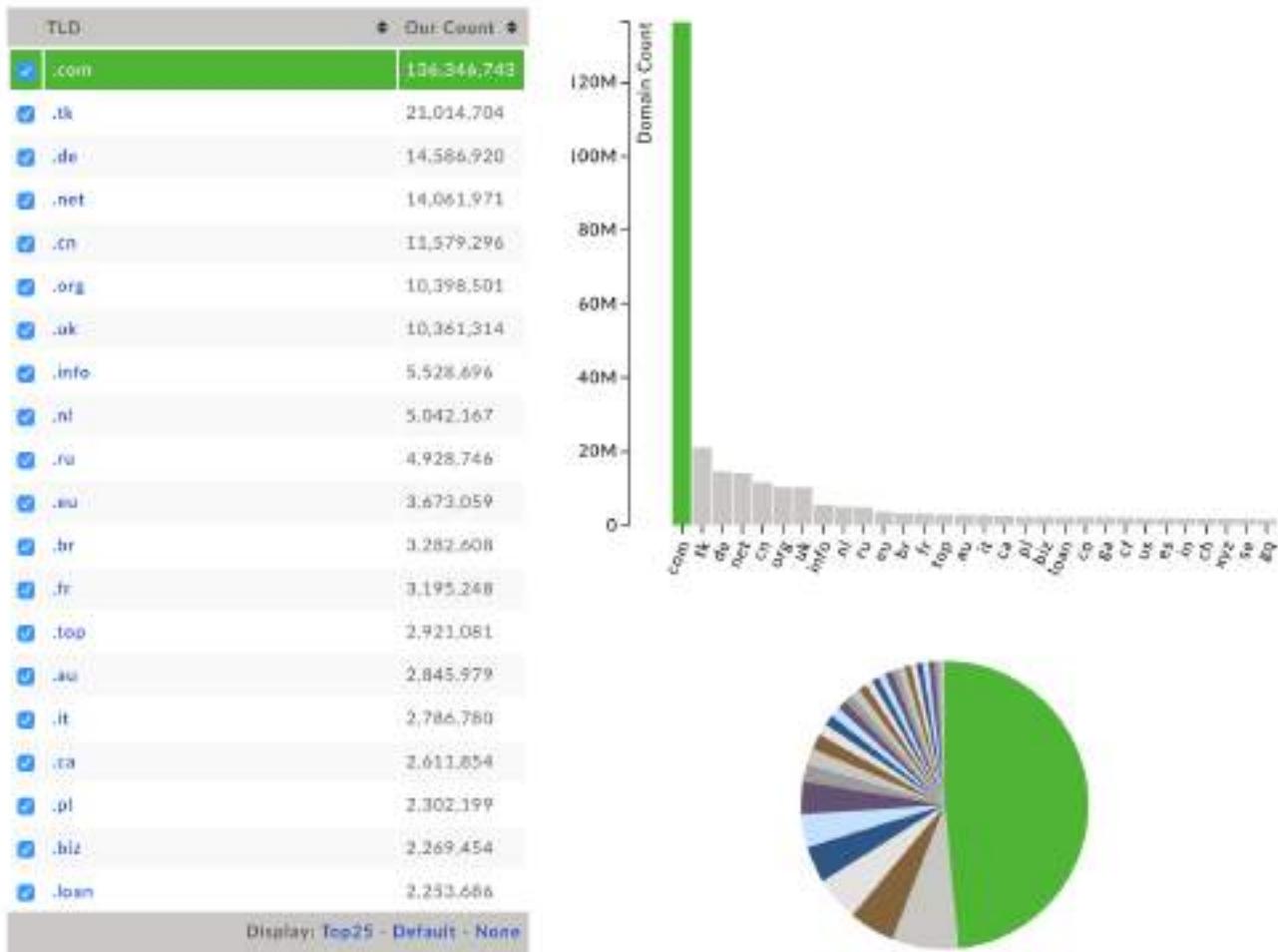
Distribution Across TLDs

136 million in .com,
21million in .tk, 14
million in .de, etc

what is .tk and why is
it so large? (Tokelau)

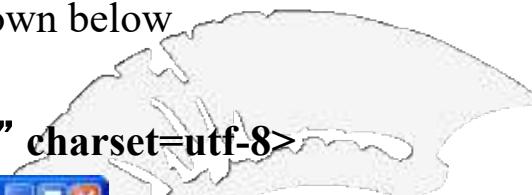
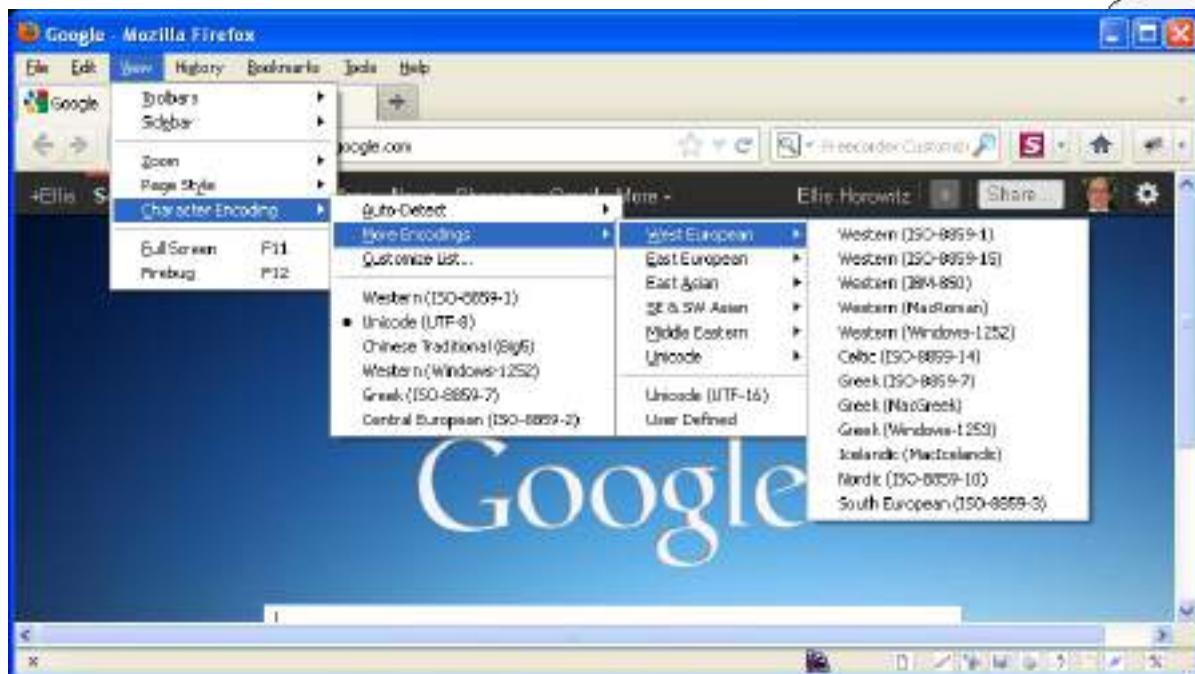
Domain Count Statistics for TLDs

This page displays the count of all Domains in each TLD. For Registry's publishing a domain count, "Our Count" should closely match their published record. For registry's that don't provide a zone file or publish an up-to-date record, Our Count represents all domains we know about, which is usually more accurate.



Web Page Language Diversity

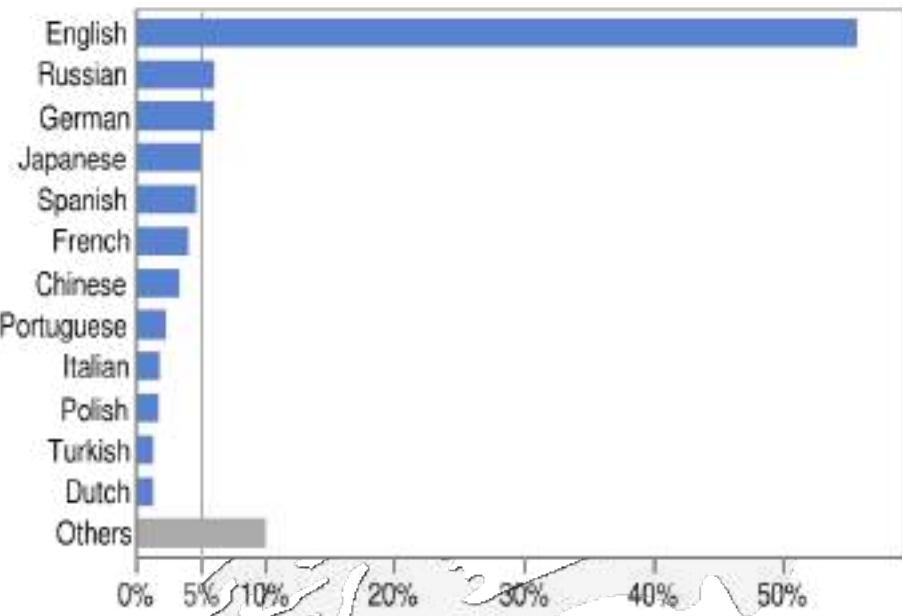
- The Web contains pages in many different languages
- Characters in a language are encoded such that each character is paired with a number
- Unicode and its parallel standard, the ISO/IEC 10646 Universal Character Set, together constitute a modern, unified character encoding.
- Most modern web browsers feature automatic character encoding detection. In Firefox, for example, see the View/Character Encoding submenu, shown below
- In HTML one can specify the character encoding using
- `<meta http-equiv="Content-Type content="text/html" charset=utf-8>`



- If charset is missing ISO-8859-1 is taken as the default unless there is a browser setting;
- Websites in non-western languages typically use UTF-8

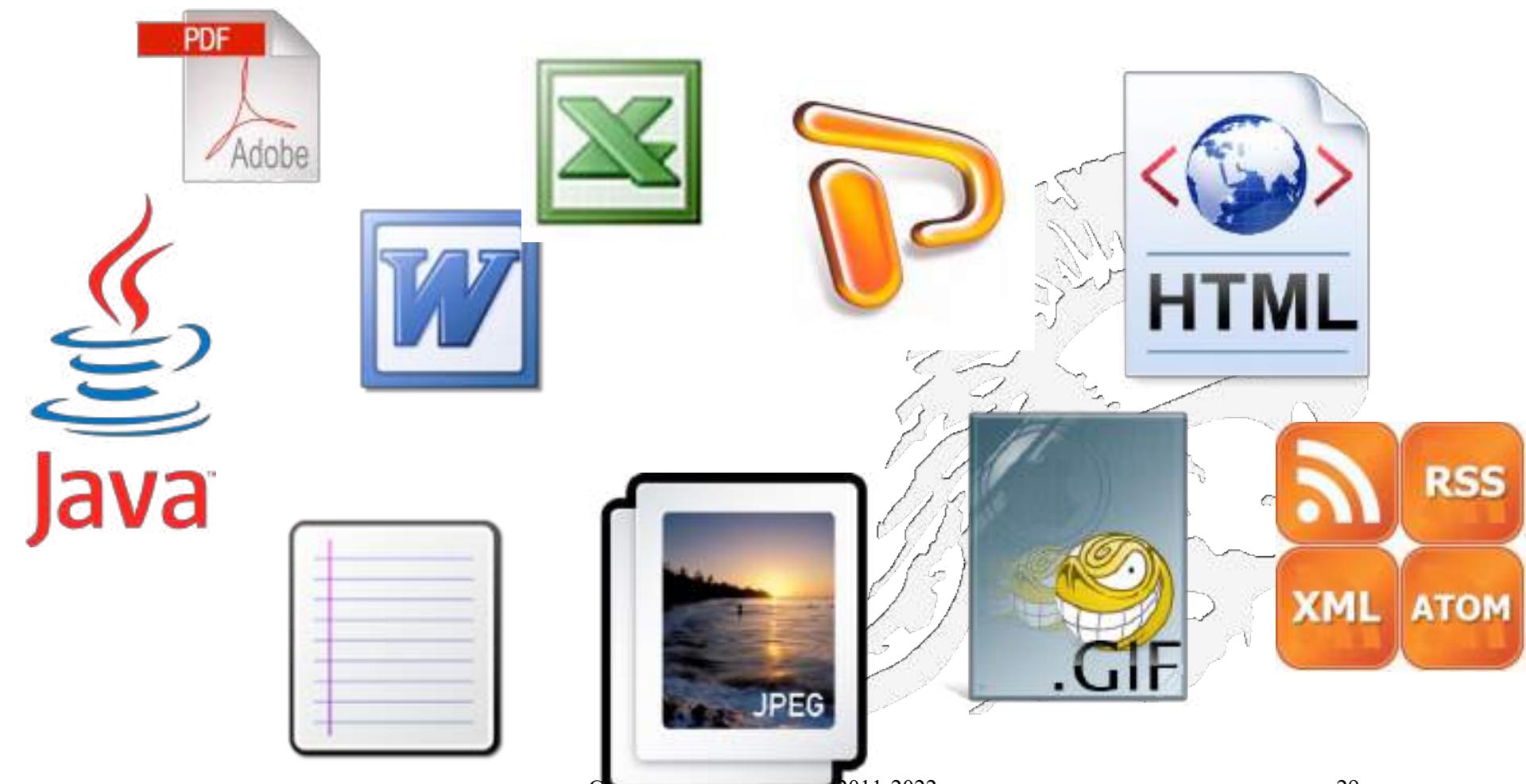
Measuring Language Diversity

- It is estimated that about 40,000 different languages have been created by human beings
- Only between 6,000-9,000 are still in use
- Study done by the United Nations
 - <http://unesdoc.unesco.org/images/0018/001870/187016e.pdf>
 - The methodology was to examine the pages on a search engine and attempt to identify the primary language in which the page is written
 - Conclusions
 - From 1996 – 2008 English was predominant, occupying roughly 80% of web pages
 - At the same time the number of Internet users who had English as their primary language dropped from 80% to 40%



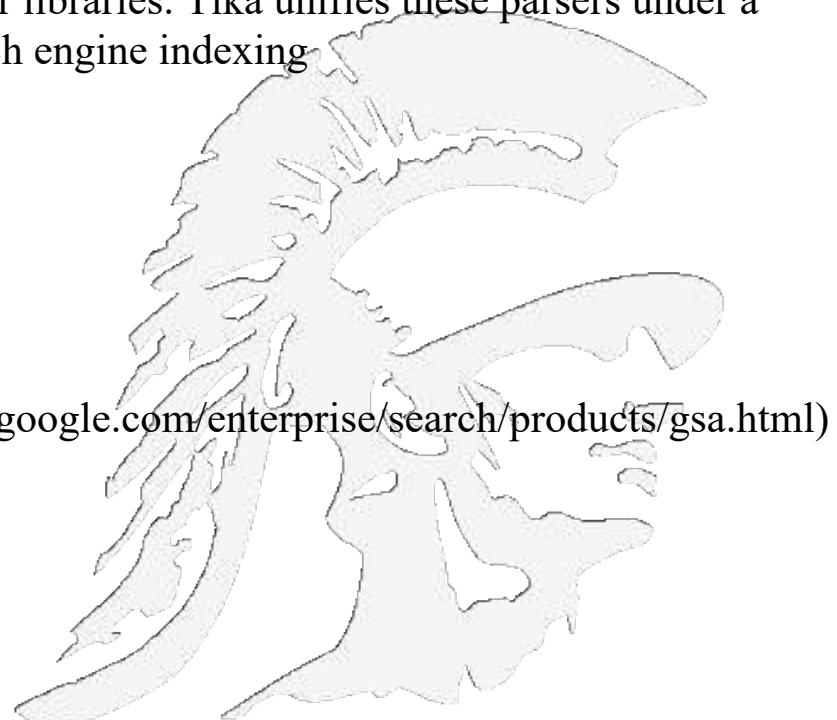
Content languages for websites as of 12 March 2014 [2]

Complexity of Data Types



Proliferation of Content Types Available

- By some accounts, there are 16,000 to 51,000 content types*
- What to do with content types?
 - Parse them
 - How? The Apache Tika™ toolkit detects and extracts metadata and text content from various documents using existing parser libraries. Tika unifies these parsers under a single interface. Tika is useful for search engine indexing
 - Extract their text and structure
 - Index their metadata
 - Use an indexing technology like
 - Lucene, <http://lucene.apache.org/>
 - Solr, <http://lucene.apache.org/solr/>
 - Google Search Appliance (<http://www.google.com/enterprise/search/products/gsa.html>)
 - Identify what language they belong to
 - N-grams



*<http://fileext.com/> (see if you can name the top 20 file extensions)

Content Types Indexed by Google

What file types can Google index? support.google.com/webmasters/bin/answer.py?hl=en&answer=35287

Google

Webmaster Tools

What file types can Google index?

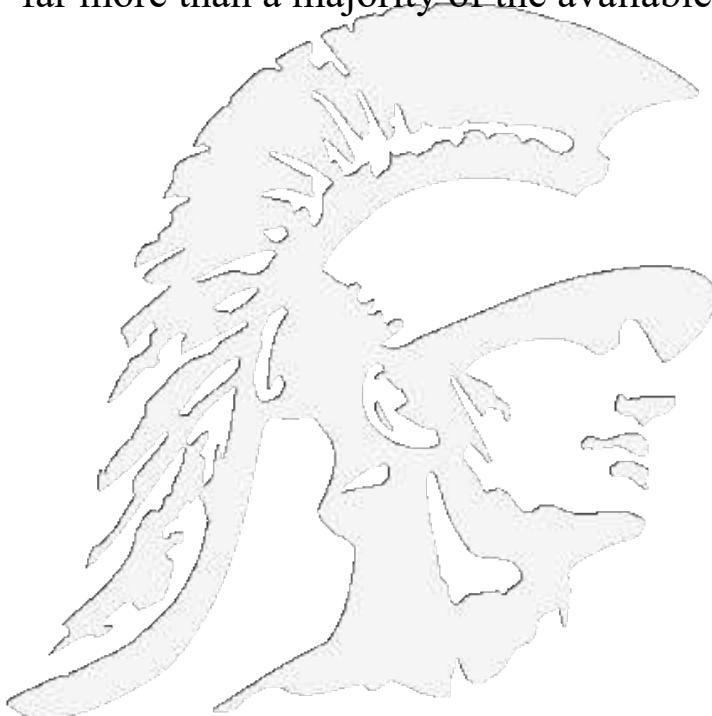
Google can index the content of most types of pages and files. The most common file types we index include:

- Adobe Flash [swf]
- Adobe Portable Document Format [pdf]
- Adobe PostScript [ps]
- Autodesk Design Web Format [dwf]
- Google Earth [kmz, kmx]
- GPS Exchange Format [gpx]
- Microsoft Word [doc]
- Microsoft Excel [xls, xlsx]
- Microsoft PowerPoint [ppt, pptx]
- Microsoft Word [doc, docx]
- OpenOffice presentation [odp]
- OpenOffice spreadsheet [ods]
- OpenOffice text [odt]
- Rich Text Format [rtf, wri]
- Scalable Vector Graphics [svg]
- TwinklText [txt]
- Text [txt, txtz, other file extensions], including source code in various programming languages
 - Basic source code [bas]
 - C/C++ source code [c, cc, cpp, cxx, h, hxx]
 - C# source code [cs]
 - Java source code [java]
 - Perl source code [pl]
 - Python source code [py]
- XML [xml]

Related

- Flickr and other photo formats
- Google+ Webmaster FAQ
- Search Engine Optimization (SEO)
- Webmaster FAQ
- Meta tags
- Image Sitemaps

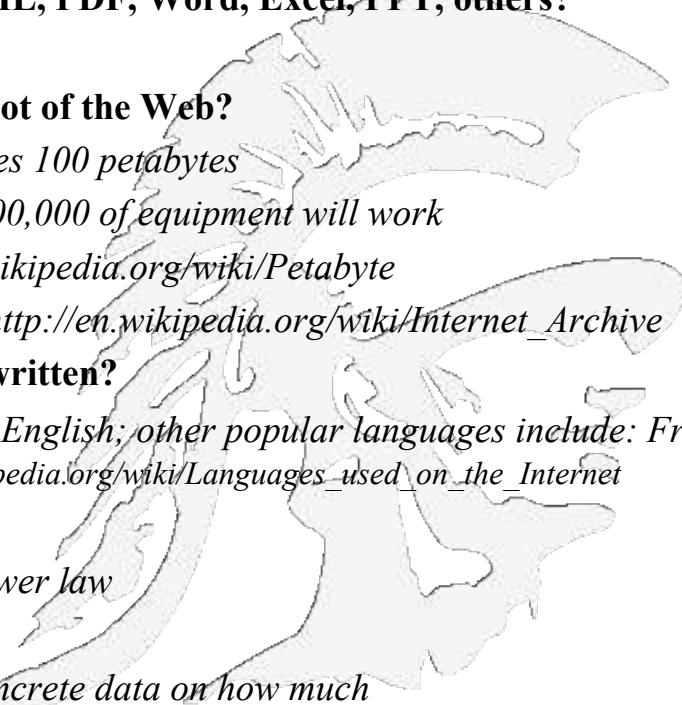
Considering the fact that there are thousands of file types of content stored on the web, Google actually indexes only a small number, less than 3 dozen, but they may well constitute far more than a majority of the available content



<http://support.google.com/webmasters/bin/answer.py?hl=en&answer=35287>

A Summary of Some Web Facts

- **How many websites?** $\sim 1.86 \text{ billion}$
- **How are they distributed across TLDs or across countries?**
 - *112 million out of 148 million belong to .com or about 72%*
- **How many web pages are there?** *30 trillion unique URLs from Google found in 2012,*
see <http://googleblog.blogspot.com/2008/07/we-knew-web-was-big.html>
- **Which content types hold the most information:** HTML, PDF, Word, Excel, PPT, others?
 - *There are thousands of different content types*
- **How much storage is required to hold a single snapshot of the Web?**
 - *1 trillion web pages at 100K bytes per page requires 100 petabytes*
 - *1 petabyte storage costs under \$1,000, so \$100,000 of equipment will work*
 - *Google processes 24 petabytes per day, <http://en.wikipedia.org/wiki/Petabyte>*
 - *The Internet Archive has more than 10 petabytes, http://en.wikipedia.org/wiki/Internet_Archive*
- **What are the languages in which the documents are written?**
 - *According to the Internet Archive, about 55% is in English; other popular languages include: French, German, Spanish and Chinese, also see http://en.wikipedia.org/wiki/Languages_used_on_the_Internet*
- **General properties of the Web graph**
 - *In-degree and out-degree distribution follows a power law*
- **Categories of Content: pornography, spam, mirrors**
 - *Presumably there is a lot of the above, but little concrete data on how much*



Manual Hierarchical Web Taxonomies

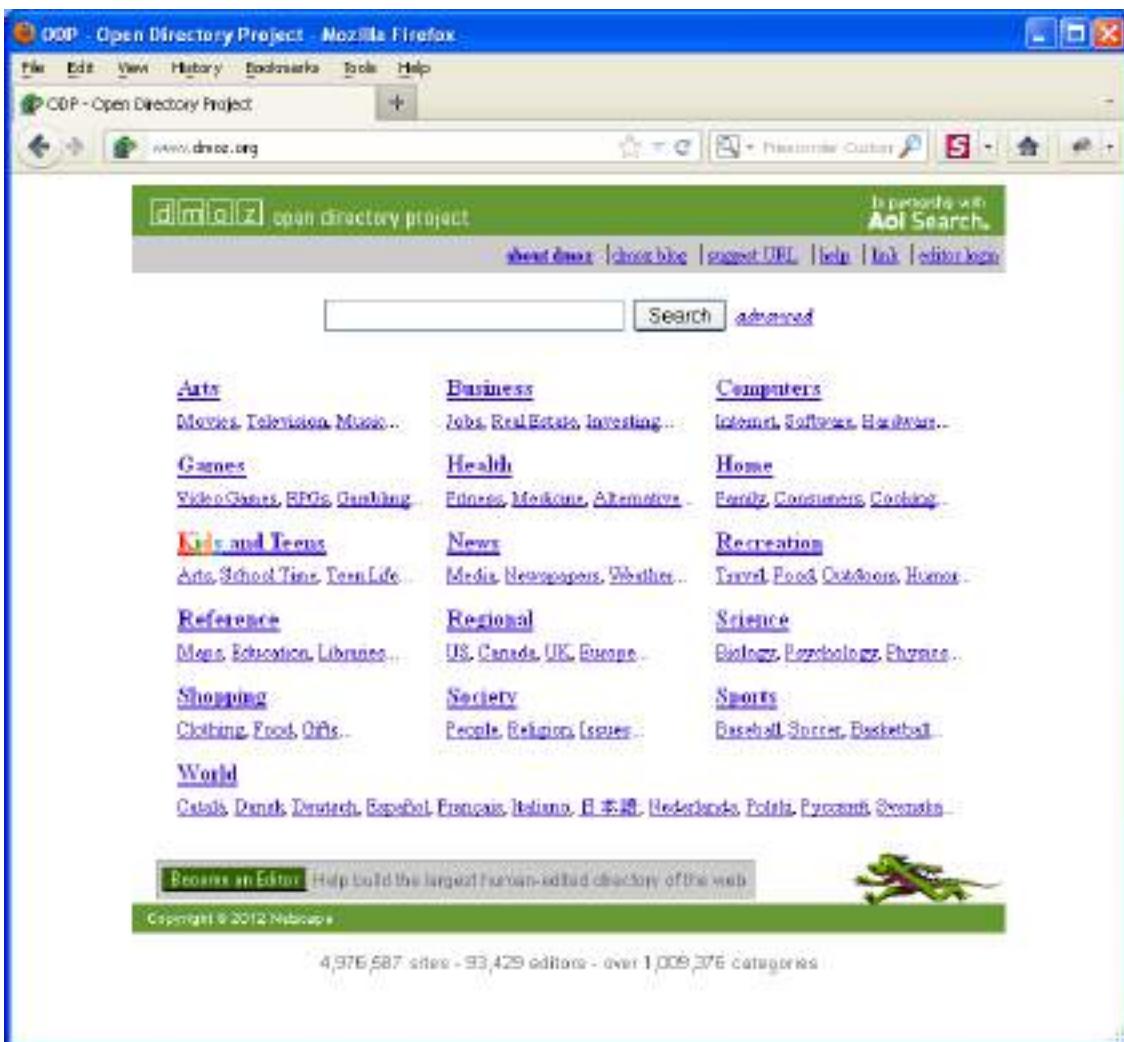
The screenshot shows a Mozilla Firefox browser window with the title bar "european cars - Yahoo! Directory Search Results - Mozilla Firefox". The menu bar includes "File", "Edit", "View", "History", "Bookmarks", "Tools", and "Help". The toolbar has icons for Back, Forward, Stop, Home, and Search. The address bar shows the URL "drsearch.yahoo.com(secure)_rl=000Gdnrc2ASPsEAuIwz00". The main content area displays the "YAHOO! DIRECTORY" search results for "european cars" with 100 results. On the left, there is a "FILTER" sidebar with options like "Show All", "Regional (9)", "Business and Economy (10)", "Recreation (11)", "Arts (3)", and "None...". Below that is a "FILTER BY TIME" section with "Any time", "Last 3 months", "Last 6 months", and "Last year". The main search results are listed as follows:

- Also try:** [european car parts](#), [european cars for sale](#), [More...](#)
- European Car Sharing**
Umbrella organization for car sharing companies in Europe.
Category: [Business and Economy](#)>[Shopping and Services](#)>[Automotive](#)>[Car Sharing](#)
www.carssharing.org
- European New Car Assessment Programme (EuroNCAP)**
Aims to provide motoring consumers with a realistic and independent assessment of the safety performance of cars sold in Europe.
Category: [Recreation](#)>[Automotive](#)>[Driving](#)>[Safety](#)>[Organizations](#)
www.euroncap.com
- European Car Free Day**
Take place September 22, 2000, to protest problems of urban mobility, air pollution, and noise.
Category: [Recreation](#)>[Travel](#)>[Transportation](#)>[Auto-Free Transportation](#)>[Organizations](#)
www.22september.org
- ClassicDriver.com - The European Car Webzine**
Focuses on prestige marques, includes articles, web broadcasting, screen savers, dealer guide, and more.
Category: [Recreation](#)>[Automotive](#)>[News and Media](#)>[Magazines](#)
www.classicdrivers.com

- **Yahoo** originally used human editors to assemble a large hierarchically structured directory of web pages.

Yahoo still retains the hierarchy as seen to the left; under european cars we see categories: regional with 93 matches, business & economy with 79 matches, etc

Open Directory Project



- **Open Directory Project, known as DMoZ, is an effort to organize the web according to an ontology;**
- **An approach similar to Yahoo's;**
- **Based on the distributed labor of volunteer editors ("net-citizens provide the collective brain").**
- **Used by most other search engines.**
- **Started by Netscape.**
 - <http://www.dmoz.org/>
- **Distributes its data using RDF format**
- **DMOZ shut down in 2016**

Drilling Down By Category

Open Directory - Science - Mozilla Firefox

File Edit View Bookmarks Tools Help

Open Directory - Science

www.theg.org/Science

open directory project

about about:home blog Import about:open Data

Search the entire directory

[\[A\]](#) [\[B\]](#) [\[C\]](#) [\[D\]](#) [\[E\]](#) [\[F\]](#) [\[G\]](#) [\[H\]](#) [\[I\]](#) [\[J\]](#) [\[K\]](#) [\[L\]](#) [\[M\]](#) [\[N\]](#) [\[O\]](#) [\[P\]](#) [\[Q\]](#) [\[R\]](#) [\[S\]](#) [\[T\]](#) [\[U\]](#) [\[V\]](#) [\[W\]](#) [\[X\]](#) [\[Y\]](#) [\[Z\]](#)

Top: Science (204,480)

- [Aerospace \(2,673\)](#)
- [Astronomy and Alternative Science \(404\)](#)
- [Astrochemistry \(3,395\)](#)
- [Biology \(31,912\)](#)
- [Chemistry \(4,771\)](#)
- [Computer Science@ \(1,971\)](#)
- [Earth Sciences \(6,098\)](#)
- [Academic Departments \(9\)](#)
- [By Region \(8\)](#)
- [Chats and Forums \(16\)](#)
- [Directories \(27\)](#)
- [Educational Resources \(752\)](#)
- [Employment \(68\)](#)
- [Events \(54\)](#)
- [History@ \(246\)](#)
- [Instruments and Supplies \(3,379\)](#)
- [Libraries@ \(81\)](#)
- [Mathematics and Technology \(167\)](#)
- [Environment \(6,571\)](#)
- [Math \(9,541\)](#)
- [Physics \(4,811\)](#)
- [Science in Society \(561\)](#)
- [Social Sciences \(9,493\)](#)
- [Technology \(10,566\)](#)
- [Women@ \(253\)](#)
- [Museums@ \(473\)](#)
- [News and Media \(354\)](#)
- [Organizations \(172\)](#)
- [People \(8\)](#)
- [Publications \(247\)](#)
- [Reference \(382\)](#)
- [Research Groups and Centers \(57\)](#)
- [Search Engines \(9\)](#)
- [Software \(383\)](#)
- [Weblogs@ \(177\)](#)

Selecting Category “Science”

Open Directory - Computer: Computer Science - Mozilla Firefox

File Edit View Bookmarks Tools Help

Open Directory - Computer: Computer Sci...

www.theg.org/computer/Computer_Science

open directory project

about about:home blog Import about:open Data

Search the entire directory

Top: Computer: Computer Science (1,971)

- [Academic Departments \(553\)](#)
- [Conferences \(203\)](#)
- [Directories \(4\)](#)
- [Organizations \(77\)](#)
- [People \(271\)](#)
- [Publications \(87\)](#)
- [Reference \(4\)](#)
- [Research Institutes \(74\)](#)
- [Artificial Intelligence@ \(1,794\)](#)
- [Artificial Life@ \(331\)](#)
- [Computational Geometry \(51\)](#)
- [Computer Graphics \(39\)](#)
- [Database Theory \(42\)](#)
- [Distributed Computing \(225\)](#)
- [Parallel Computing \(387\)](#)
- [Software Engineering@ \(114\)](#)
- [Theoretical \(361\)](#)

See also:

- [Computer: Algorithms \(204\)](#)
- [Computer: Programming \(15,619\)](#)
- [Science: Math \(9,541\)](#)
- [Science: Technology: Electrical Engineering \(237\)](#)

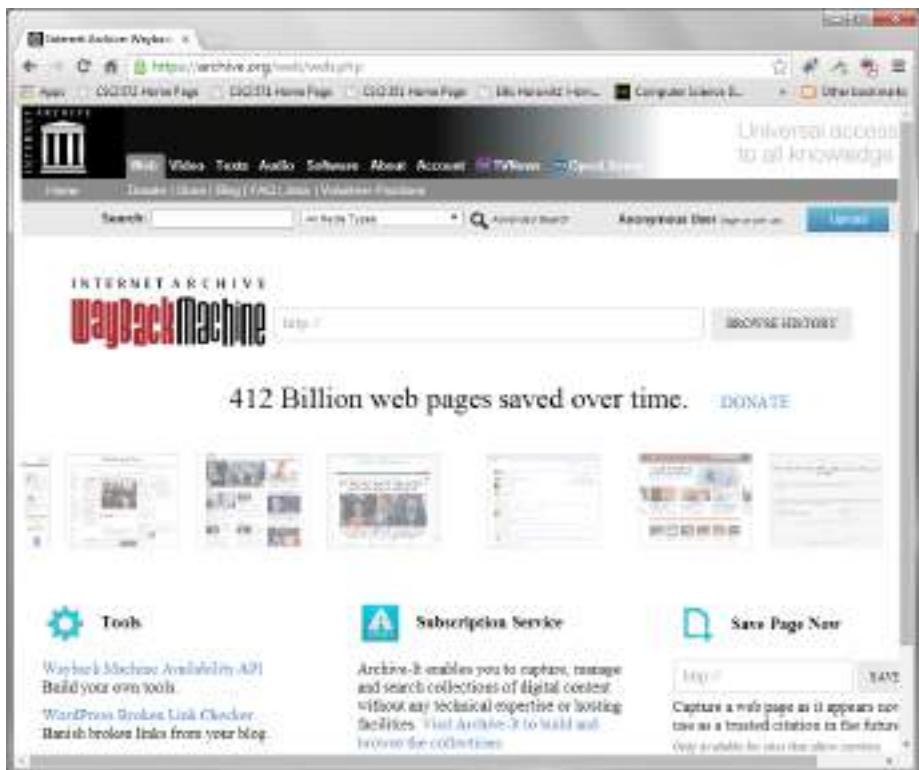
The category in other languages:



Selecting Category “Computer Science”

- The Internet Archive has been taking a snapshot of the World Wide Web every two months since 1997 – has used **Apache Nutch**
- The results are made available through [the Wayback Machine](#),
- Its database is approximately 4.5 petabytes
- The founder is Brewster Kahle
- For the past 13 years, the Internet Archive has been growing rapidly, most recently by about 100TB of data per month.
- Their crawler surveys the web every two months. The algorithm first performs a broad crawl that starts with a few "seed sites," such as Yahoo's directory. After snapping a shot of the home page, it then moves to any referable pages within the site until there are no more pages to capture. If there are any links on those pages, the algorithm automatically opens them and archives that content as well.

Internet Archive



Surface Web

SURFACE WEB

Google

Bing

Wikipedia

Academic Information

Medical Records

Legal Documents

Scientific Reports

Subscription Information

DEEP WEB

*Contains 90% of the information on
the Internet, but is not accessible
by Surface Web crawlers.*

Social Media

Multilingual Databases

Financial Records

Government Resources

Competitor Websites

Organization-specific
Repositories

(DARK WEB)

A part of the Deep Web accessible only through certain browsers such as Tor designed to ensure anonymity. Deep Web Technologies has zero involvement with the Dark Web.

Illegal Information

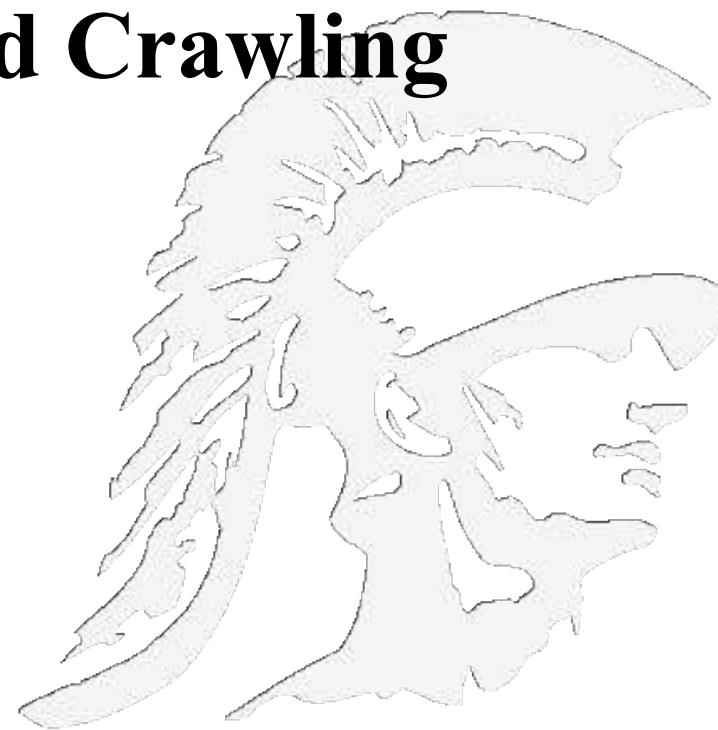
TOR-Encrypted sites

Drug Trafficking sites

Political Protests

Private Communications

Crawlers and Crawling



There are Many Crawlers

- A web crawler is a computer program that visits web pages in an organized way

- Sometimes called a spider or robot

- A list of web crawlers can be found at

http://en.wikipedia.org/wiki/Web_crawler

Google's crawler is called googlebot, see

<http://support.google.com/webmasters/bin/answer.py?hl=en&answer=182072>

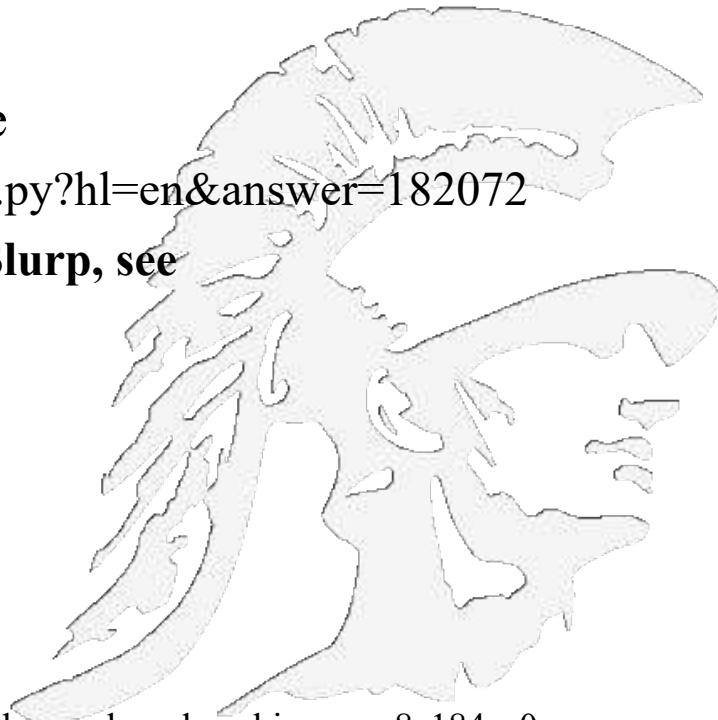
- Yahoo's web crawler is/was called Yahoo! Slurp, see

http://en.wikipedia.org/wiki/Yahoo!_Search

- Bing uses five crawlers

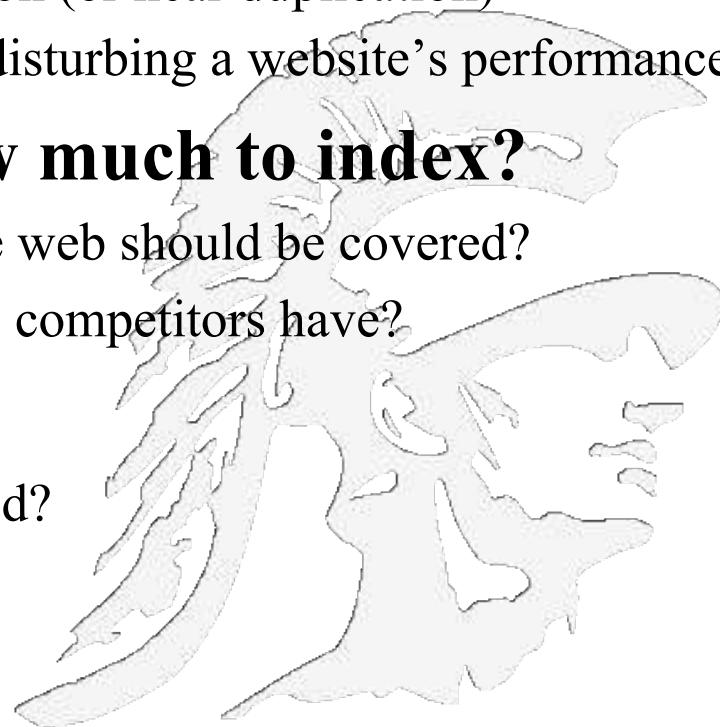
- Bingbot, standard crawler
 - Adidxbot, used by Bing Ads
 - MSNbot, remnant from MSN, but still in use
 - MSNBotMedia, crawls images and video
 - BingPreview, generates page snapshots

- For details see: <http://www.bing.com/webmaster/help/which-crawlers-does-bing-use-8c184ec0>



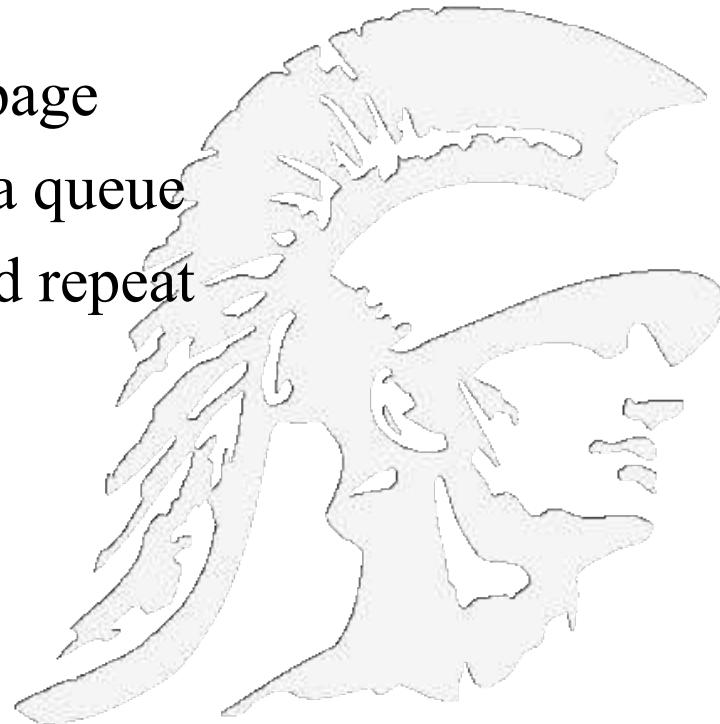
Web Crawling Issues

- **How to crawl?**
 - *Quality*: how to find the “Best” pages first
 - *Efficiency*: how to avoid duplication (or near duplication)
 - *Etiquette*: behave politely by not disturbing a website’s performance
- **How much to crawl? How much to index?**
 - *Coverage*: What percentage of the web should be covered?
 - *Relative Coverage*: How much do competitors have?
- **How often to crawl?**
 - *Freshness*: How much has changed?
 - How much has really changed?



Simplest Crawler Operation

- Initialize (begin with known “seed” pages)
- Loop: Fetch and parse a page
 - Place the page in a database
 - Extract the URLs within the page
 - Place the extracted URLs on a queue
 - Fetch a URL on the queue and repeat



Crawling Picture

20

Web crawling and indexes

20.1 Overview

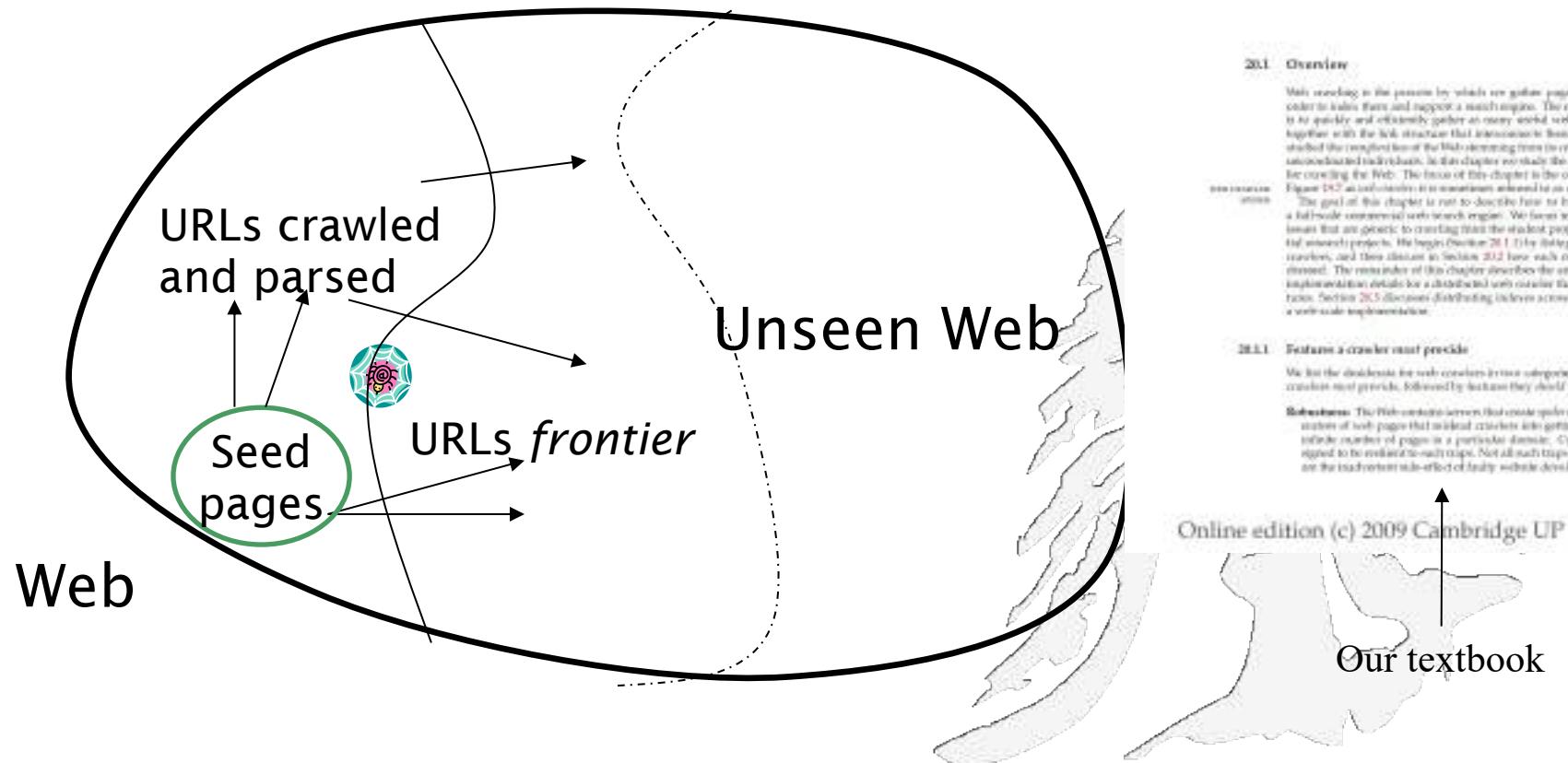
Web crawling is the process by which we gather pages from the Web, in order to index them and support a search engine. The objective of crawling is to quickly and efficiently gather as many useful web pages as possible, together with the link structure that interconnects them. In Chapter 19 we studied the crawled part of the Web, stemming from its creation by millions of uncoordinated individuals. In this chapter we study the resulting difficulties for crawling the Web. The focus of this chapter is the component shown in Figure 20.2, which is the crawler itself, as it is sometimes referred to as a spider.

The goal of this chapter is not to describe how to build the crawler for a full-scale commercial search engine. We focus instead on a range of issues that are generic to crawling from the student project code to substantive research projects. We begin (Section 20.1.1) by listing requirements for seed crawlers, and then discuss in Section 20.2 how each of these requirements is addressed. The remainder of this chapter describes the architecture and some implementation details for a distributed web crawler that satisfies these features. Section 20.3 discusses distributing indexing across many machines for a web-scale implementation.

20.1.1 Features a crawler must provide

We list the demands for web crawlers in four categories. Features that web crawlers must provide, followed by features they should provide.

Reliability: The Web contains servers that create spikes traps, which are generators of lots of pages that suddenly crawl into getting stuck, freezing at a fairly number of pages in a particular domain. Crawlers need to be designed to be resilient to such traps. Not all such traps are malicious; some are the inadvertent side-effect of faulty website development.

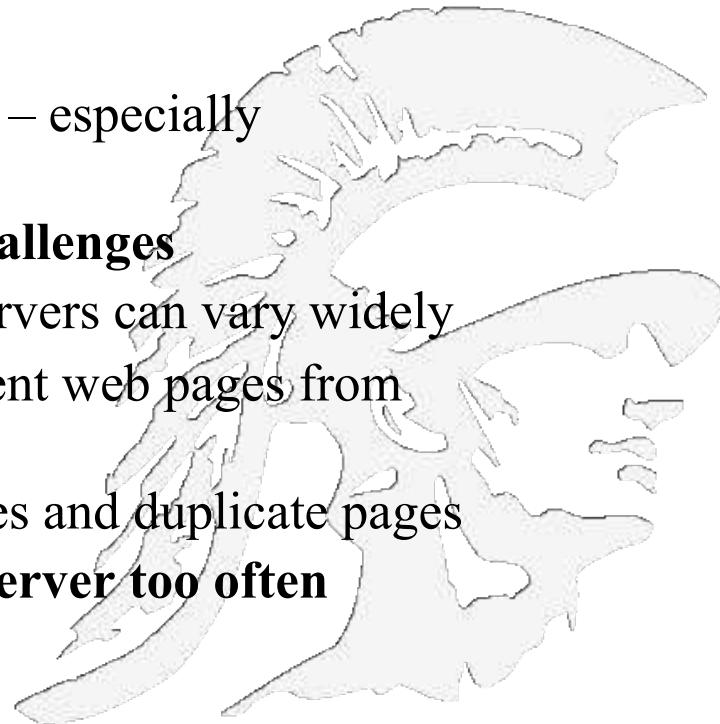


Online edition (c) 2009 Cambridge UP

Our textbook

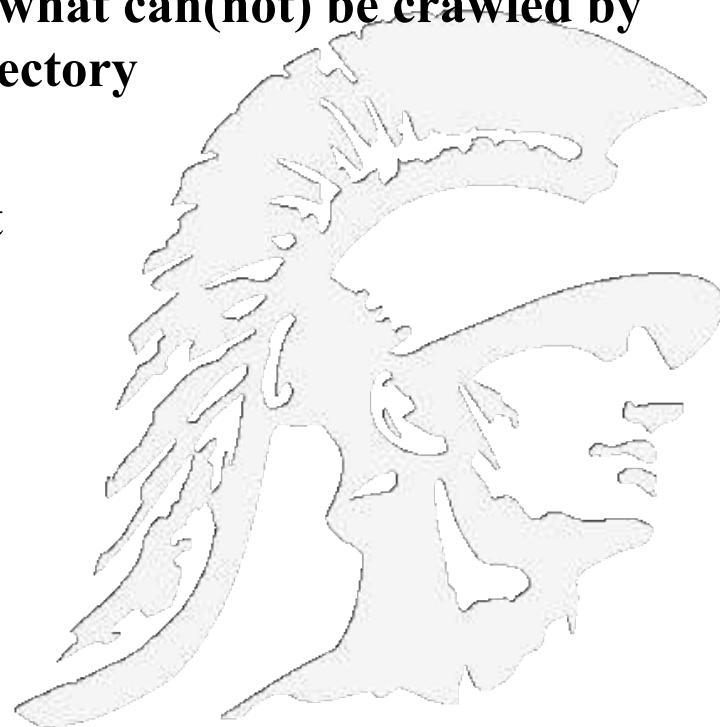
Simple Picture – Complications

- Crawling the entire web isn't feasible with one machine
 - But all of the above steps can be distributed
- Challenges
 - Handling/Avoiding malicious pages
 - Some pages contain spam
 - Some pages contain spider traps – especially dynamically generated pages
 - Even non-malicious pages pose challenges
 - Latency/bandwidth to remote servers can vary widely
 - Robots.txt stipulations can prevent web pages from being visited
 - How can one avoid mirrored sites and duplicate pages
 - Maintain politeness – don't hit a server too often



Robots.txt

- There is a protocol that defines the limitations for a web crawler as it visits a website; its definition is here
 - <http://www.robotstxt.org/orig.html>
- The website announces its request on what can(not) be crawled by placing a robots.txt file in the root directory
 - e.g. see
<http://www.ticketmaster.com/robots.txt>



Robots.txt Example

- No robot visiting this domain should visit any URL starting with "/yoursite/temp/":

User-agent: *

Disallow: /yoursite/temp/

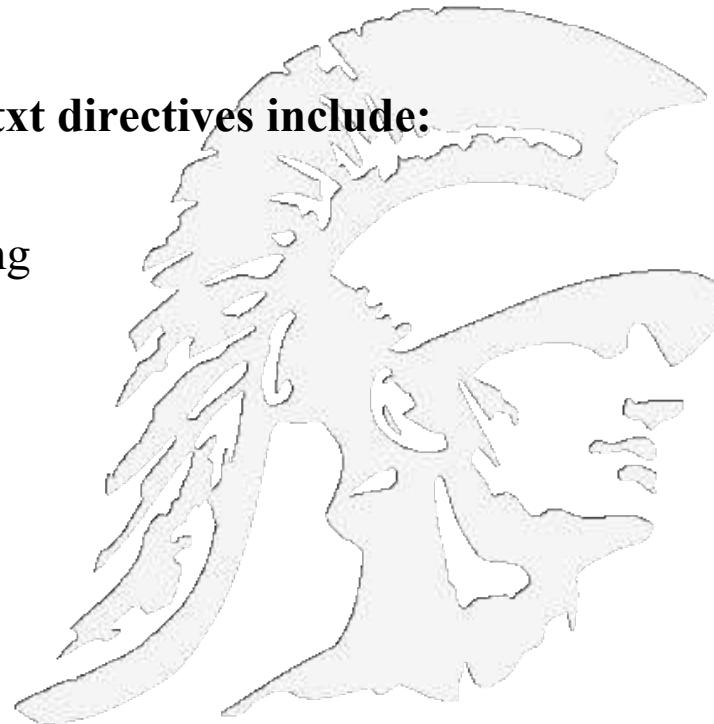
- Directives are case sensitive
- Additional symbols allowed in the robots.txt directives include:
 - '*' - matches a sequence of characters
 - '\$' - anchors at the end of the URL string
- Example of '*':

User-agent: Slurp

Allow: /public*/

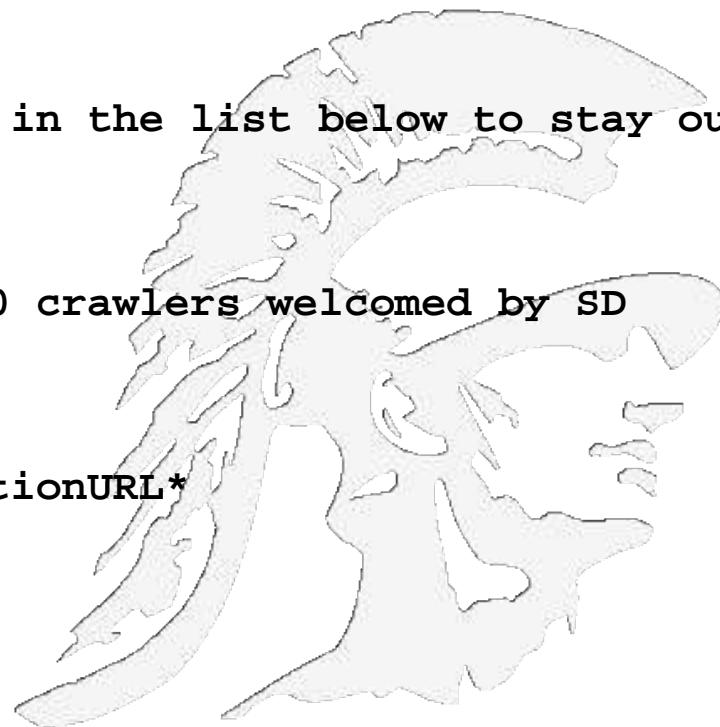
Disallow: /*_print*.html

Disallow: /*?sessionid



Websites Give Preference to Googlebot

- In researching how websites treat crawlers, Zack Maril looked at 17 million robots.txt files and discovered many places where Google is given an advantage; e.g. see
- <https://www.sciencedirect.com/robots.txt>
- E.g.
- # go away ? tell all others not in the list below to stay out!
- User-agent: *
- Disallow: /
- # As of 02/09/2021, there are 10 crawlers welcomed by SD
- User-agent: Googlebot
- Disallow: /cache/MiamiImageURL/
- Disallow: /science?_ob=MiamiCaptionURL*



A Study of Robots.txt Files

- Determining bias to search engines from robots.txt, Giles, Sun, Zhuang, Penn State*

Favored Robots (Sample size $N = 2925$)				
robot name	N_{favor}	N_{disf}	$\Delta P(r)$	σ
google	877	25	0.2913	0.0084
yahoo	631	34	0.2041	0.0075
msn	349	9	0.1162	0.0059
scooter	341	15	0.1104	0.0058
lycos	91	5	0.0294	0.0031
netmechanic	84	10	0.0253	0.0029
htdig	15	3	0.0041	0.0012
teoma	13	3	0.0034	0.0011
oodlebot*	8	0	0.0027	0.0010
momspider	6	0	0.0021	0.0008

Disfavored Robots (Sample size $N = 2925$)				
robot name	N_{favor}	N_{disf}	$\Delta P(r)$	σ
msiecrawler	0	85	-0.0291	0.0031
ia_archiver	7	55	-0.0164	0.0023
cherrypicker	0	37	-0.0126	0.0021
emailiphon	3	34	-0.0106	0.0019
roverbot	2	27	-0.0085	0.0017
psbot	0	23	-0.0079	0.0016
webzip	0	21	-0.0072	0.0016
wget	1	22	-0.0072	0.0016
linkwalker	2	20	-0.0062	0.0015
asterias	0	18	-0.0062	0.0015

Table 2. Top 10 favored and disfavored robots. σ is the standard deviation of $\Delta P(r)$.

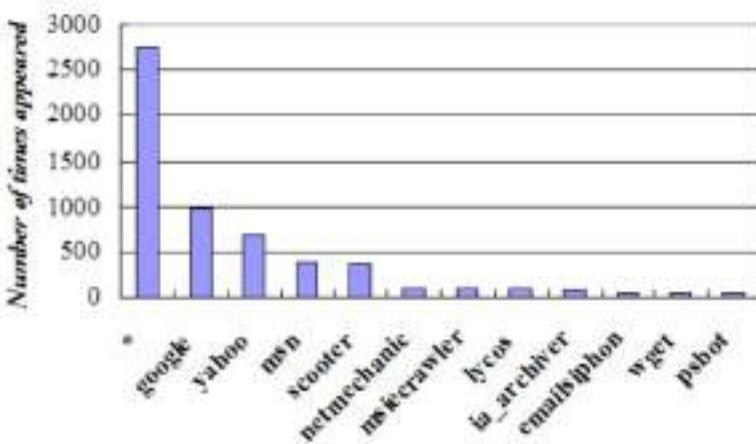
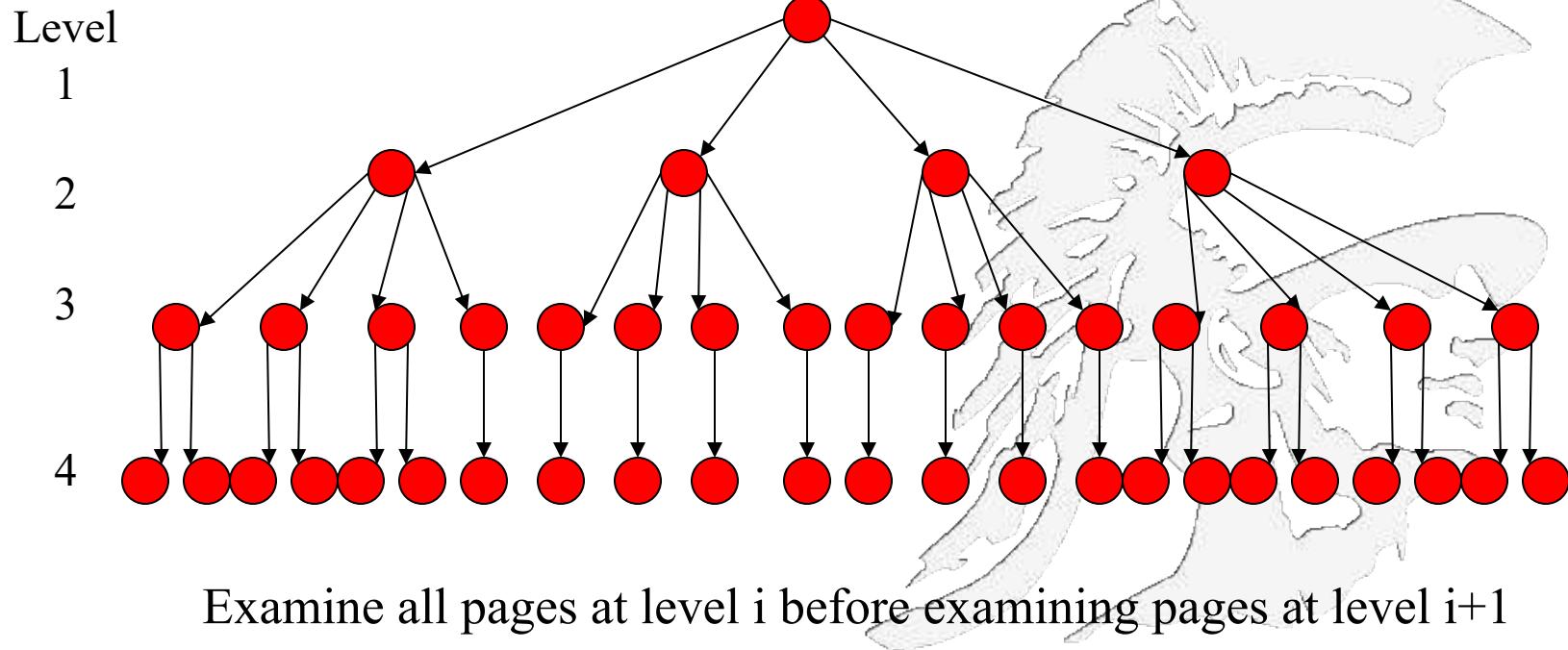


Figure 2. Most frequently used robot names in robots.txt files. The height of the bar represents the number of times a robot appeared in our dataset.



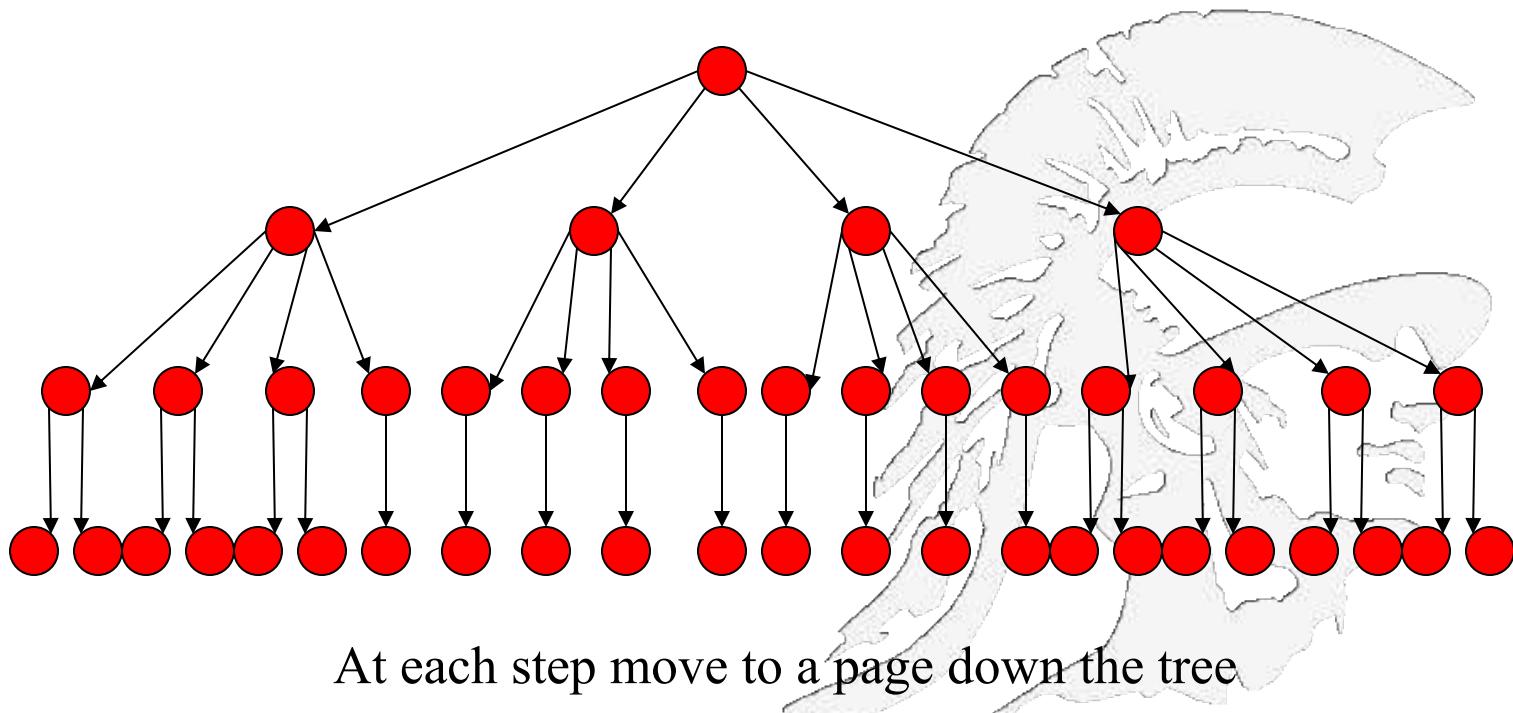
Basic Search Strategies

Breadth-first Search

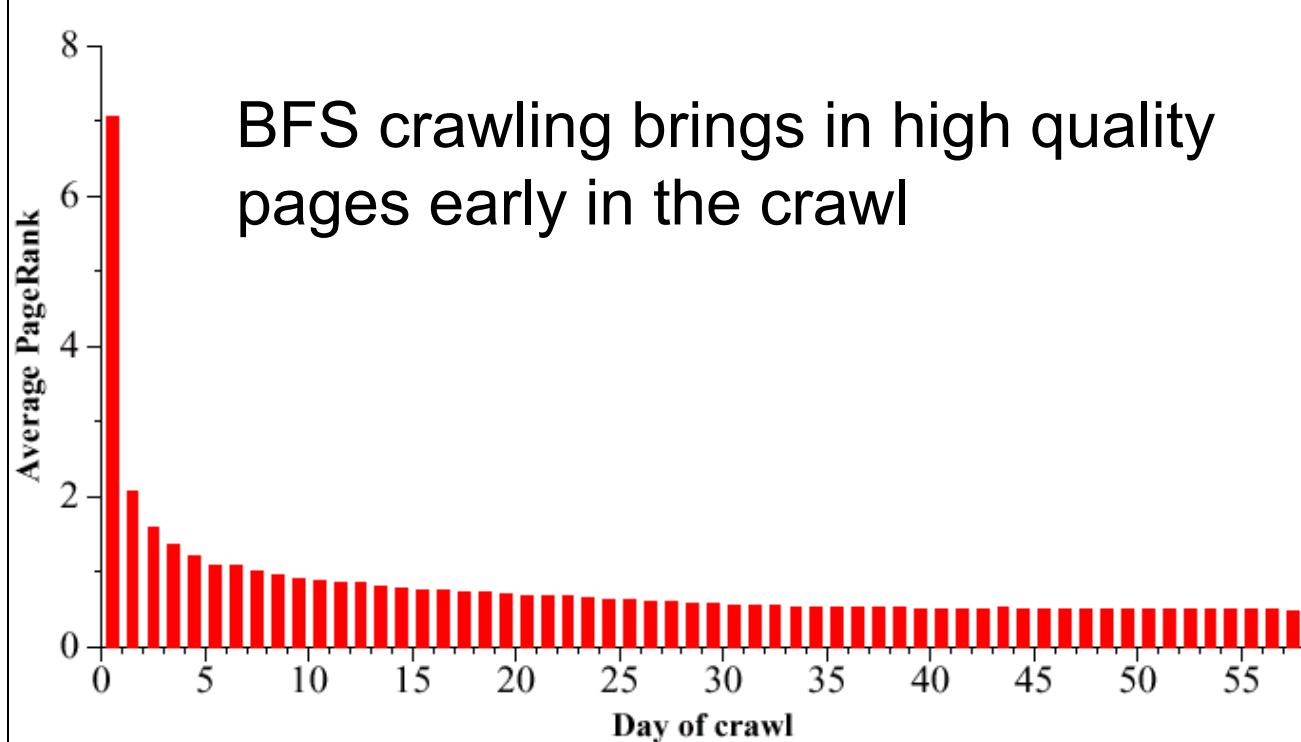


Basic Search Strategies (cont)

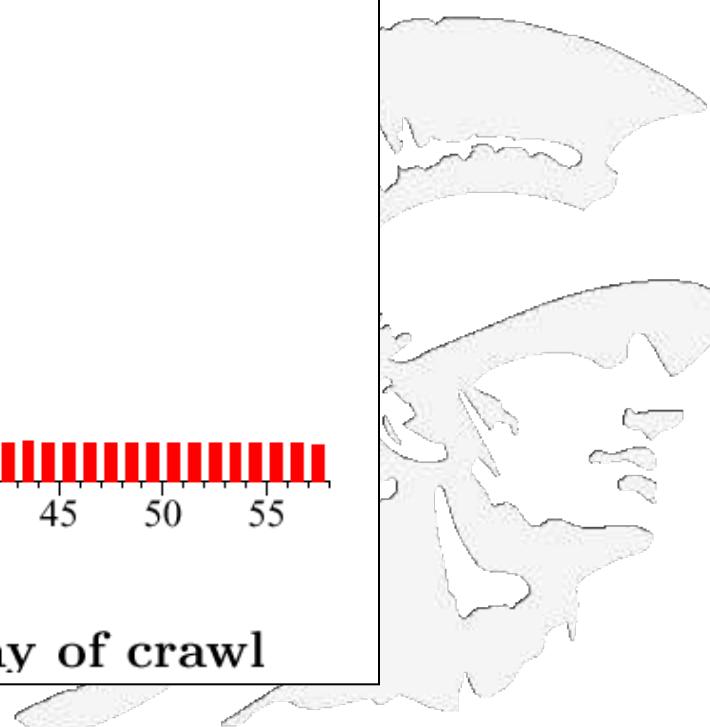
Depth-first Search



Web Wide Crawl (328M pages) [Najo01]



BFS crawling brings in high quality pages early in the crawl



Average PageRank score by day of crawl

Page Rank is an algorithm developed by Google for determining the value of a page

Crawling Algorithm – Version 2

Initialize queue (Q) with initial set of known URL's.

Loop until Q empty or page or time limit exhausted:

Pop a URL, call it L, from the front of Q.

If L is not an HTML page (e.g. .gif, .jpeg,)

continue the loop

If L has already been visited, continue the loop.

Download page, P, for L

If cannot download P (e.g. 404 error, robot excluded)

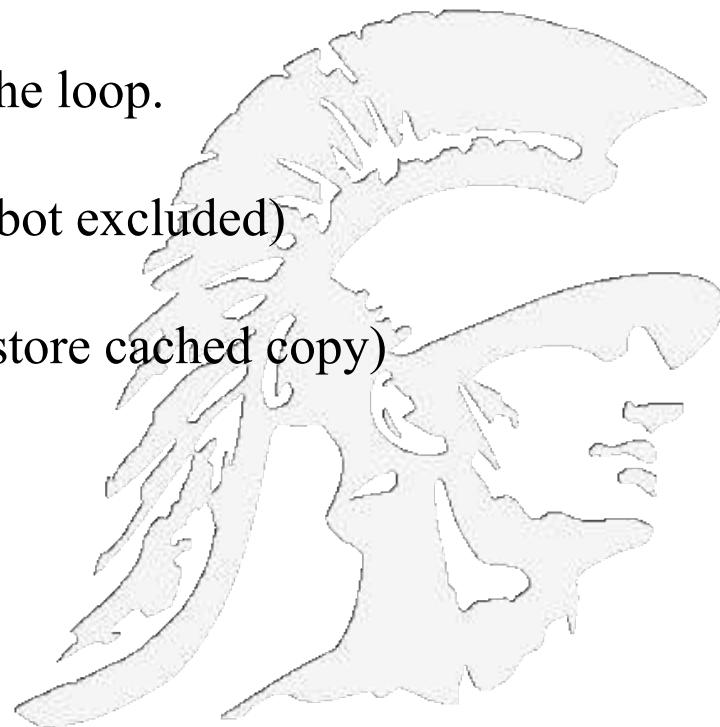
continue loop

Index P (e.g. add to inverted index and store cached copy)

Parse P to obtain list of new links N.

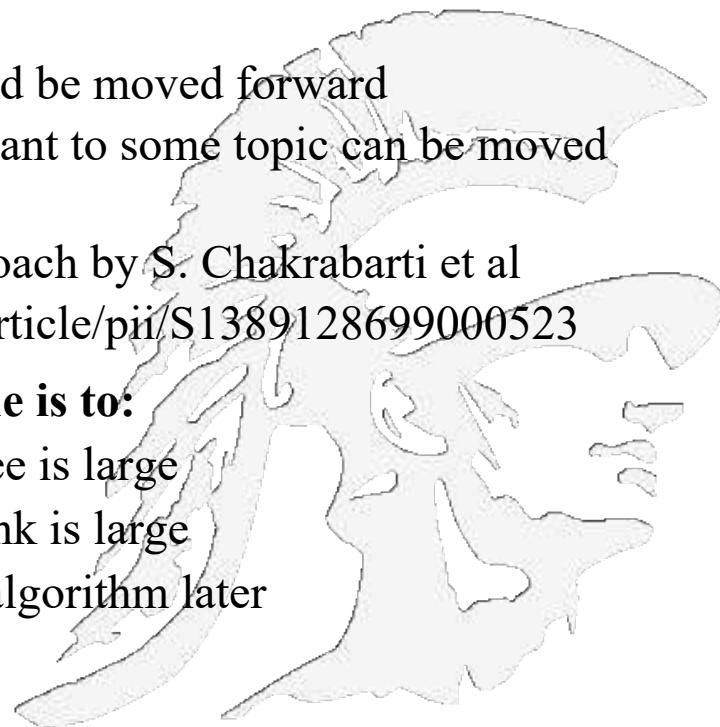
Append N to the end of Q

End loop



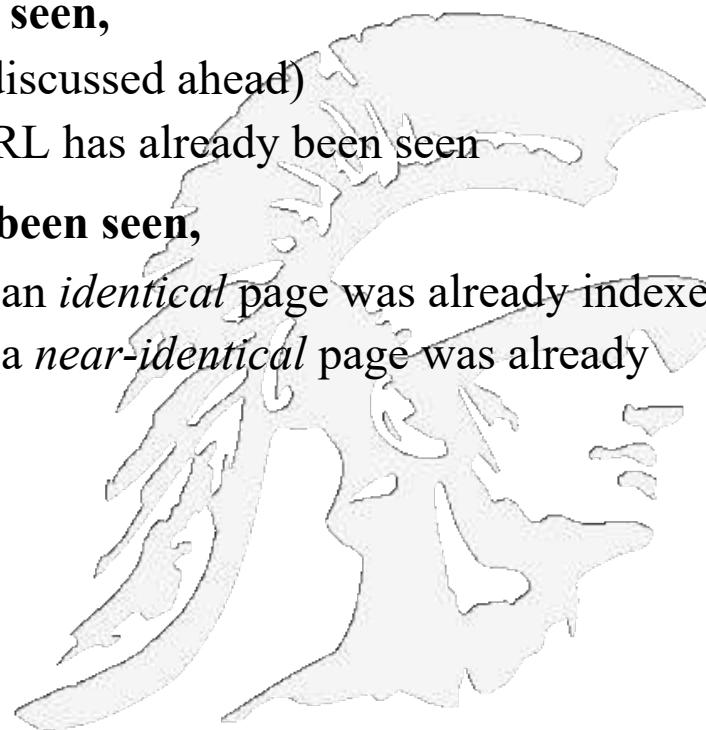
Queueing Strategy

- How new links are added to the queue determines the search strategy.
- FIFO (append to end of Q) gives breadth-first search.
- LIFO (add to front of Q) gives depth-first search.
- Heuristically ordering the Q gives a “focused crawler” that directs its search towards “interesting” pages; e.g.
 - A document that changes frequently could be moved forward
 - A document whose content appears relevant to some topic can be moved forward
 - e.g. see Focused Crawling: A New Approach by S. Chakrabarti et al
 - <https://www.sciencedirect.com/science/article/pii/S1389128699000523>
- One way to re-order the URLs on the queue is to:
 - Move forward URLs whose In-degree is large
 - Move forward URLs whose PageRank is large
 - We will discuss the PageRank algorithm later



Avoiding Page Duplication

- A crawler must detect when revisiting a page that has already been crawled (Remember: the web is a graph not a tree).
- Therefore, a crawler must efficiently index URLs as well as already visited pages
- To determine if a URL has already been seen,
 - Must store URLs in a standard format (discussed ahead)
 - Must develop a fast way to check if a URL has already been seen
- To determine if a new page has already been seen,
 - Must develop a fast way to determine if an *identical* page was already indexed
 - Must develop a fast way to determine if a *near-identical* page was already indexed



Link Extraction

- **Must find all links in a page and extract URLs;**

```
var links = document.querySelectorAll("a");
for (var i = 0; i < links.length; i++) {
    var link = links[i].getAttribute("href");
    console.log(link); }
```

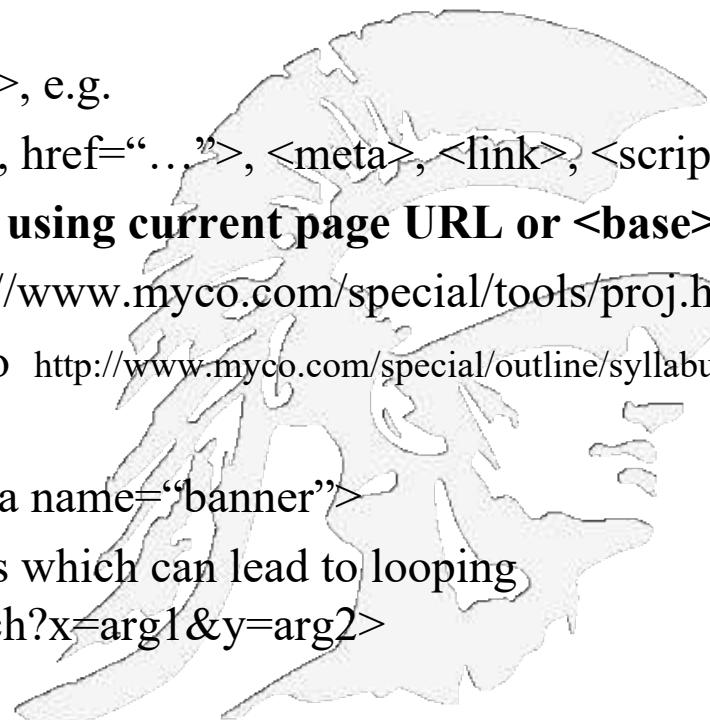
- But URLs occur in tags other than <a>, e.g.
- <frame src="site-index.html">, <area href="...">, <meta>, <link>, <script>

- **Relative URL's must be completed, e.g. using current page URL or <base> tag**

- to http://www.myco.com/special/tools/proj.html
- to http://www.myco.com/special/outline/syllabus.html

- **Two Anomalies**

1. Some anchors don't have links, e.g.
2. Some anchors produce dynamic pages which can lead to looping



Representing URLs

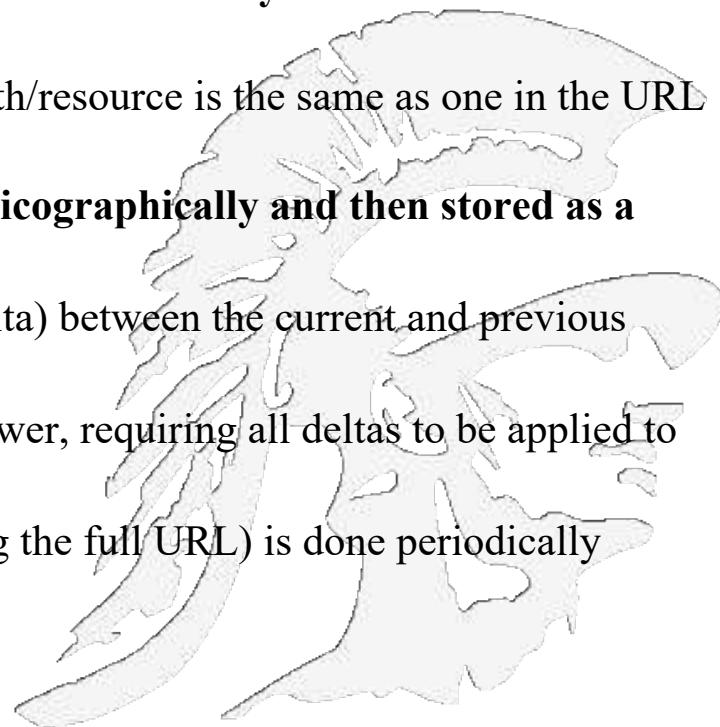
- URLs are rather long, 80 bytes on the average, implying 1 trillion URLs will require 80 Terabytes
 - Recently Google reported finding 30 trillion unique URLs, which by the above would require 2400 terabytes (or 2.4 petabytes) to store

1. One Proposed Method: To determine if a new URL has already been seen

- First hash on host/domain name, then
- Use a trie data structure to determine if the path/resource is the same as one in the URL database

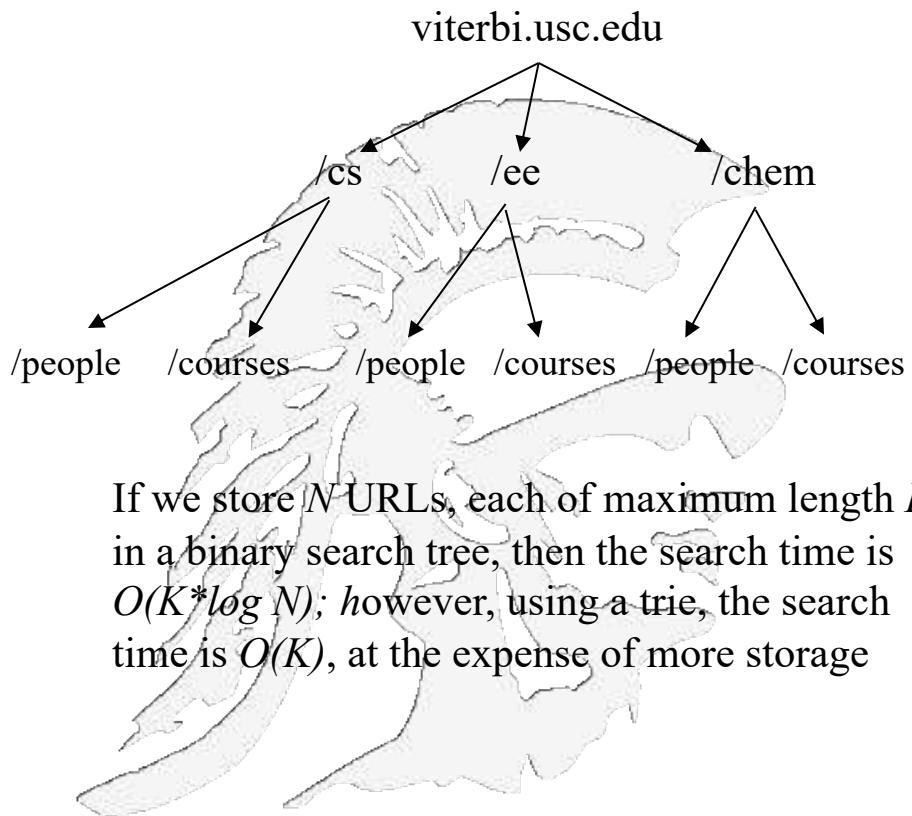
2. Another Proposed Method: URLs are sorted lexicographically and then stored as a delta-encoded text file

- Each entry is stored as the difference (delta) between the current and previous URL; this substantially reduces storage
- However, restoring the actual URL is slower, requiring all deltas to be applied to the initial URL
- To improve speed, checkpointing (storing the full URL) is done periodically



Trie for URL Exact Matching

- Simplest (and worst) algorithm to determine if a new URL is in your set
 - `grep -i <search_url> <url_file>`
 - For N URLs and maximum length K , time is $O(NK)$
- Characteristics of tries
 - They share the same prefix among multiple “words”
 - Each path from the root to a leaf corresponds to one “word”
 - *Endmarker symbol, \$, at the ends of all words*
 - To avoid confusion between words with almost identical elements
 - Assume all words are \$ terminated



Why Normalizing URLs is Important

- For example, all the following URLs have the same meaning (return the same web page), but different hashes:
 - `http://www.google.com`
 - `http://www.google.com/`
 - `https://www.google.com`
 - `www.google.com`
 - `google.com`
 - `google.com/`
 - `google.com.`



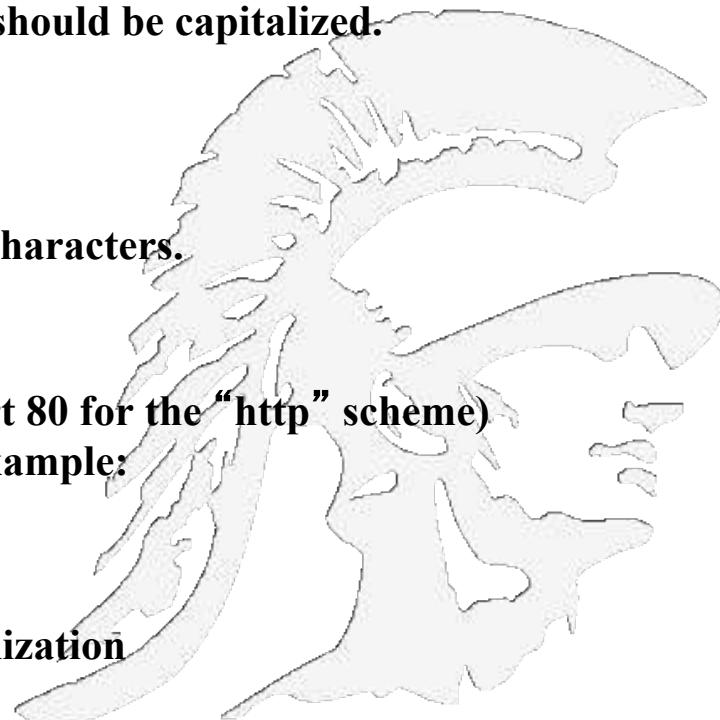
Normalizing URLs (4 rules)

1. Convert the scheme and host to lower case. The scheme and host components of the URL are case-insensitive.
 - HTTP://www.Example.com/ → http://www.example.com/
2. Capitalize letters in escape sequences. All letters within a percent-encoding triplet (e.g., "%3A") are case-insensitive, and should be capitalized.

Example:

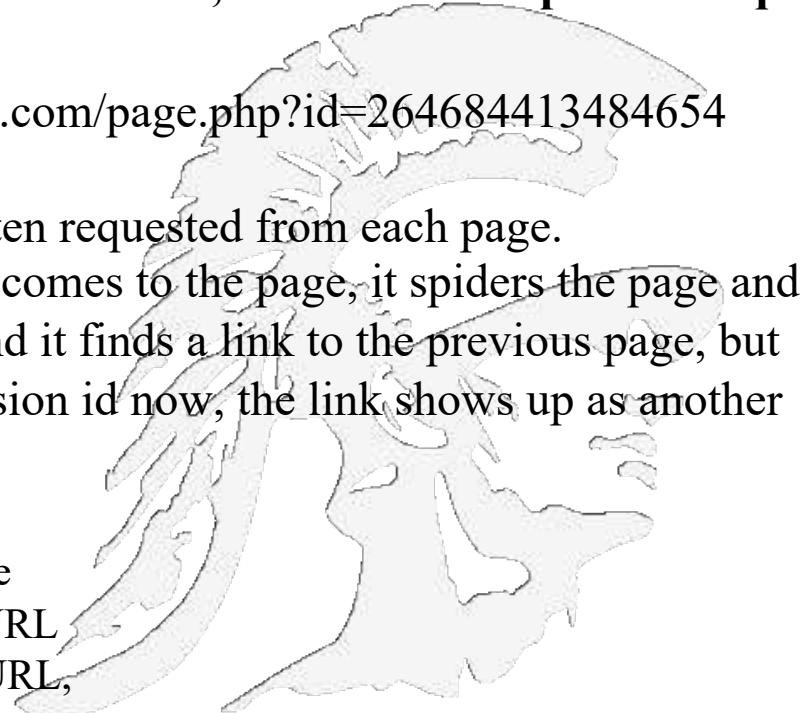
 - http://www.example.com/a%c2%b1b →
http://www.example.com/a%C2%B1b
3. Decode percent-encoded octets of unreserved characters.

http://www.example.com/%7Eusername/ →
http://www.example.com/~username/
4. Remove the default port. The default port (port 80 for the “http” scheme) may be removed from (or added to) a URL. Example:
 - http://www.example.com:80/bar.html →
http://www.example.com/bar.html
 - See https://en.wikipedia.org/wiki/URL_normalization



Avoiding Spider Traps

- A spider trap is when a crawler re-visits the same page over and over again
- The most well-known spider trap is the one created by the use of Session ID's
 - J2EE, ASP, .NET, and PHP all provide session ID management
- A Session ID is often used to keep track of visitors, and some sites puts a unique ID in the URL:
 - An example is www.webmasterworld.com/page.php?id=264684413484654 (**Note** this URL doesn't exist).
Each user gets a unique ID and it's often requested from each page.
The problem here is when Googlebot comes to the page, it spiders the page and then leaves, it goes to another page and it finds a link to the previous page, but since it has been given a different session id now, the link shows up as another URL.



- One way to avoid such traps is for the crawler to be careful when the querystring "ID=" is present in the URL
- Another technique is to monitor the length of the URL, and stop if the length gets "too long"

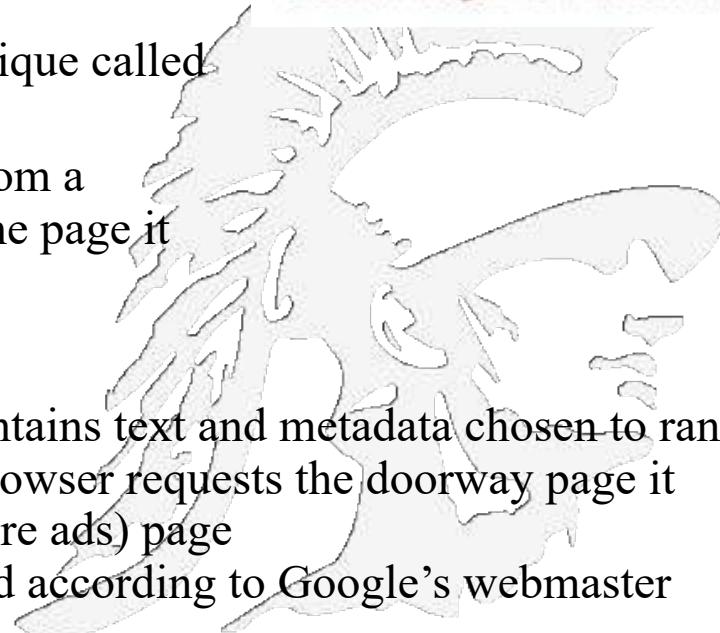
Handling Spam Web Pages

- The **first generation** of spam web pages consisted of pages with a high number of repeated terms, so as to score high on search engines that ranked by word frequency
 - Words were typically rendered in the same color as the background, so as to not be visible, but still count
- The **second generation** of spam used a technique called *cloaking*;
 - When the web server detects a request from a crawler, it returns a different page than the page it returns from a user request
 - The page is mistakenly indexed
- A third generation, called a doorway page, contains text and metadata chosen to rank highly on certain search keywords, but when a browser requests the doorway page it instead gets a more “commercially oriented” (more ads) page
- *Cloaking and doorway pages* are not permitted according to Google’s webmaster suggestions

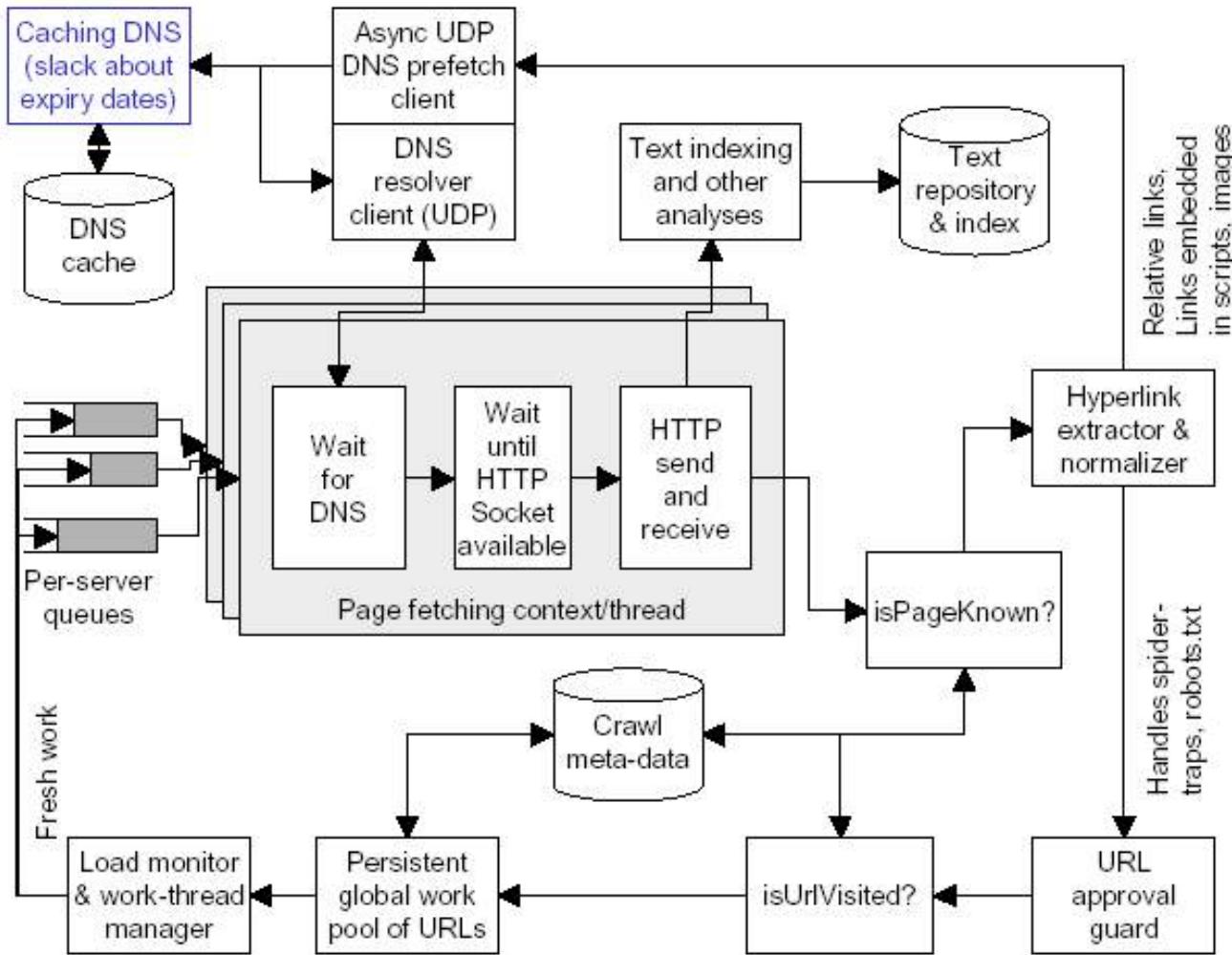
See <http://support.google.com/webmasters/bin/answer.py?hl=en&answer=66355>

Google's example of keyword stuffing:

We sell **custom cigar humidors**. Our **custom cigar humidors** are handmade. If you're thinking of buying a **custom cigar humidor**, please contact our **custom cigar humidor** specialists at custom.cigar.humidor@example.com.



The Mercator Web Crawler



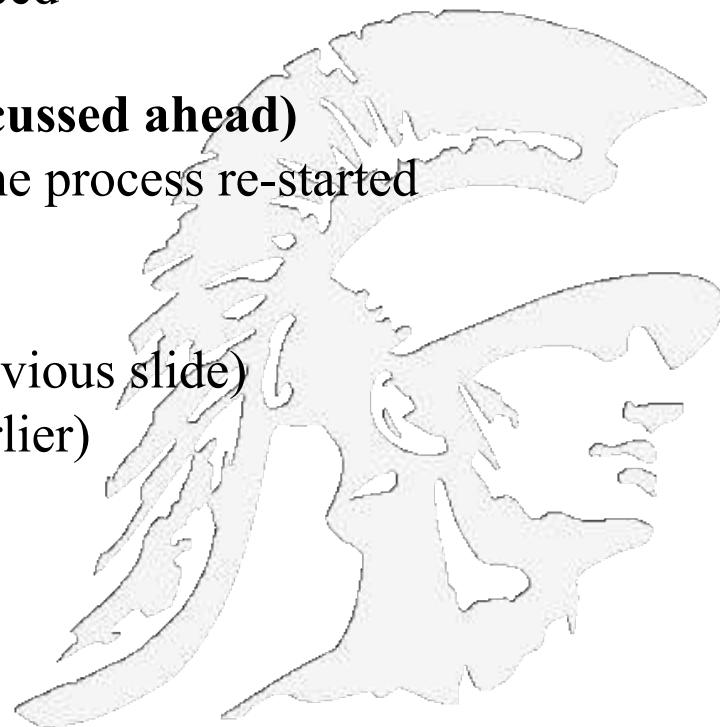
The diagram points out all of the key elements of a crawler;

Notice

1. The DNS caching server
2. Use of UDP for DNS
3. Load and thread monitor
4. Parallel threads waiting for a page to download

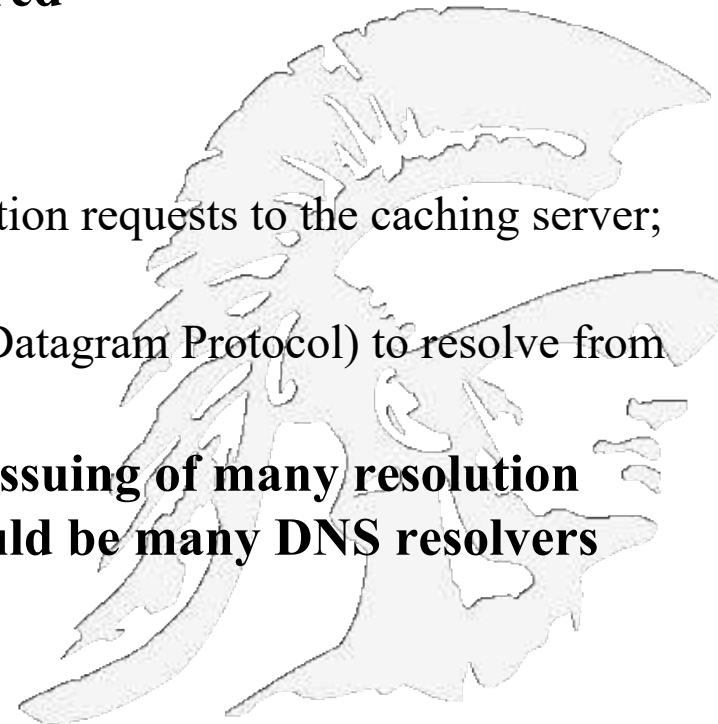
Measuring and Tuning a Crawler

- **Measuring and tuning a crawler for peak performance eventually reduces to**
 - Improving URL parsing speed
 - Improving network bandwidth speed
 - Improving fault tolerance
- **More Issues (some of which are discussed ahead)**
 - Refresh Strategies: how often is the process re-started
 - Detecting duplicate pages
 - Detecting mirror sites
 - Speeding up DNS lookup (see previous slide)
 - URL normalization (discussed earlier)
 - Handling malformed HTML



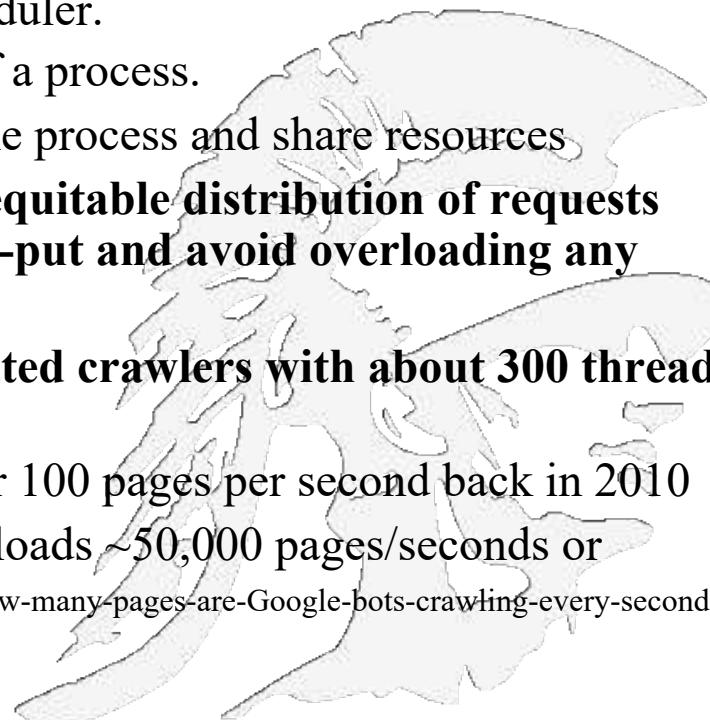
DNS caching, pre-fetching and resolution

- *A common operating system's implementation of DNS lookup is blocking: only one outstanding request at a time; so*
- 1. **DNS caching:** build a caching server that retains IP-domain name mappings previously discovered
- 2. **Pre-fetching client**
 - once a page is parsed,
 - immediately make DNS resolution requests to the caching server; and
 - if unresolved, use UDP (User Datagram Protocol) to resolve from the DNS server
- 3. **Customize the crawler so it allows issuing of many resolution requests simultaneously; there should be many DNS resolvers**



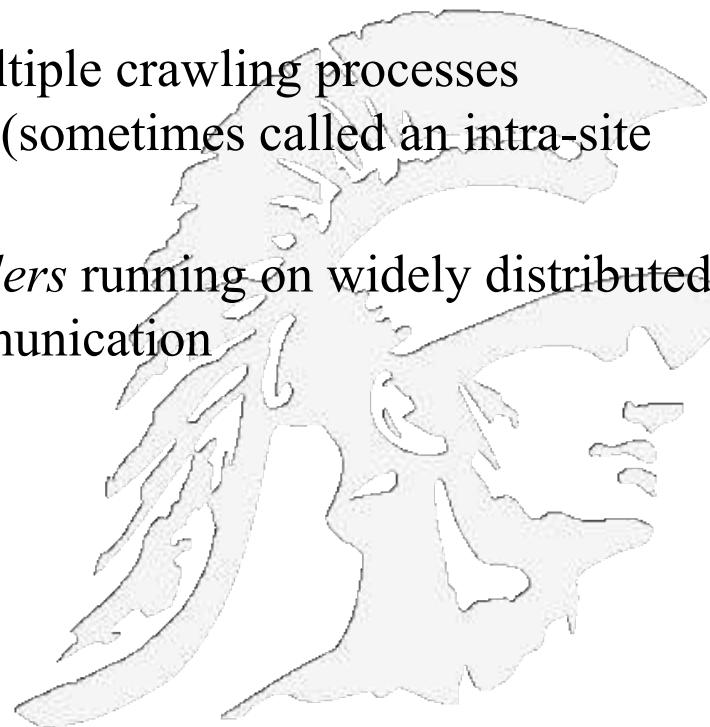
Multi-Threaded Crawling

- One bottleneck is network delay in downloading individual pages.
- It is best to have multiple threads running in parallel each requesting a page from a different host.
 - a **thread** of execution is the smallest sequence of programmed instructions that can be managed independently by a scheduler.
 - In most cases, a thread is a component of a process.
 - Multiple threads can exist within the same process and share resources
- Distribute URL's to threads to guarantee equitable distribution of requests across different hosts to maximize through-put and avoid overloading any single server.
- Early Google spider had multiple coordinated crawlers with about 300 threads each,
 - together they were able to download over 100 pages per second back in 2010
 - It is estimated that in 2021 Google downloads ~50,000 pages/seconds or 4billion+ in a day, see <https://www.quora.com/How-many-pages-are-Google-bots-crawling-every-second>



Distributed Crawling Approaches

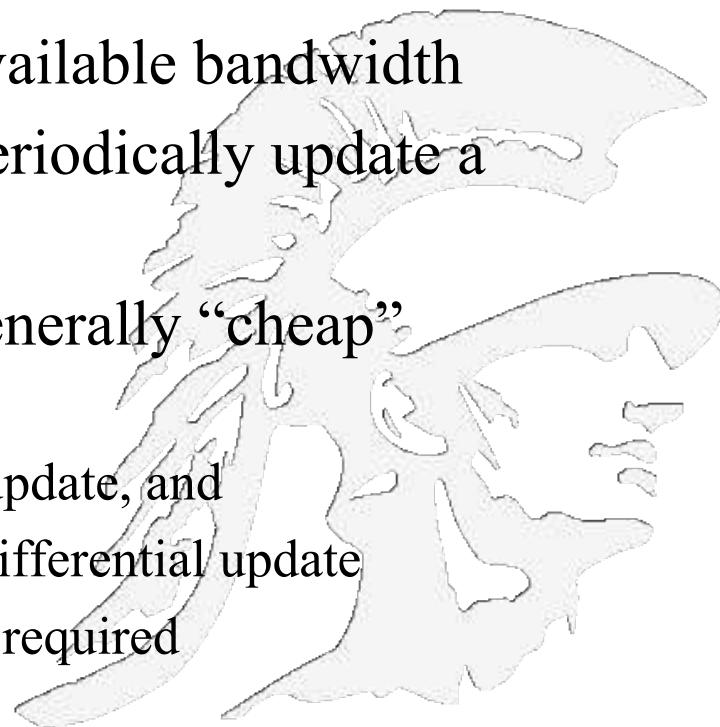
- Once the crawler program itself has been optimized, the next issue to decide is how many crawlers will be running at any time
- Scenario 1: A *centralized crawler* controlling a set of parallel crawlers all running on a LAN
 - A *parallel crawler* consists of multiple crawling processes communicating via local network (sometimes called an intra-site parallel crawler)
- Scenario 2: A *distributed set of crawlers* running on widely distributed machines, with or without cross communication



Distributed Model

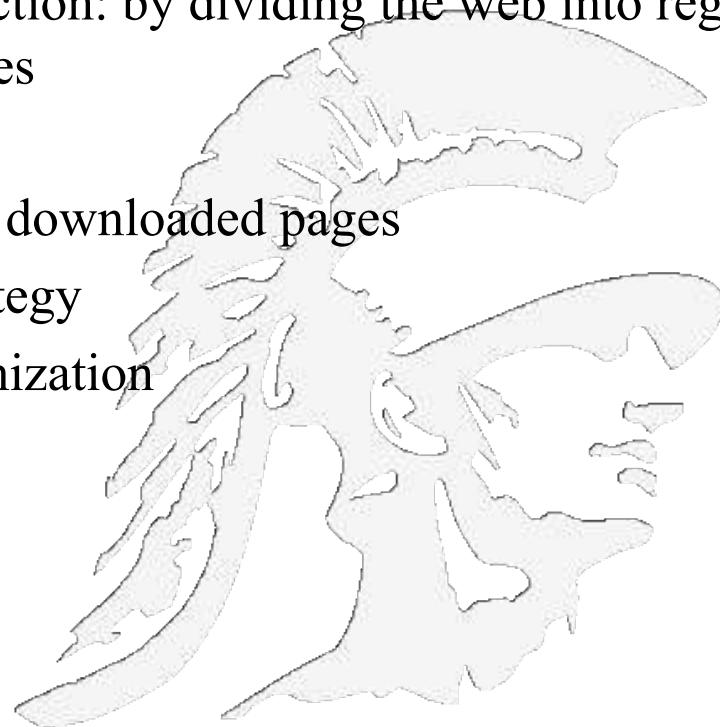
- If crawlers are running in diverse geographic locations, how do we organize them

- By country, by region, by available bandwidth
- Distributed crawlers must periodically update a master index
- But incremental update is generally “cheap”
 - Why? Because
 - a. you can compress the update, and
 - b. you need only send a differential update both of which will limit the required communication



Issues and Benefits of Distributed Crawling

- **Benefits:**
 - scalability: for large-scale web-crawls
 - costs: use of cheaper machines
 - network-load dispersion and reduction: by dividing the web into regions and crawling only the nearest pages
- **Issues:**
 - overlap: minimization of multiple downloaded pages
 - quality: depends on the crawl strategy
 - communication bandwidth: minimization



Coordination of Distributed Crawling

- Three strategies

1. **Independent:**

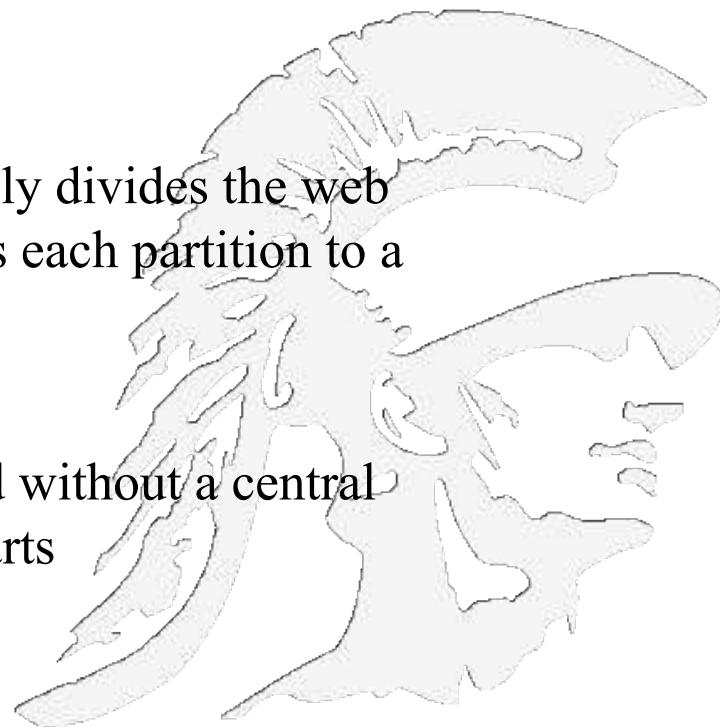
- ▶ no coordination, every process follows its extracted links

2. **Dynamic assignment:**

- ▶ a central coordinator dynamically divides the web into small partitions and assigns each partition to a process

3. **Static assignment:**

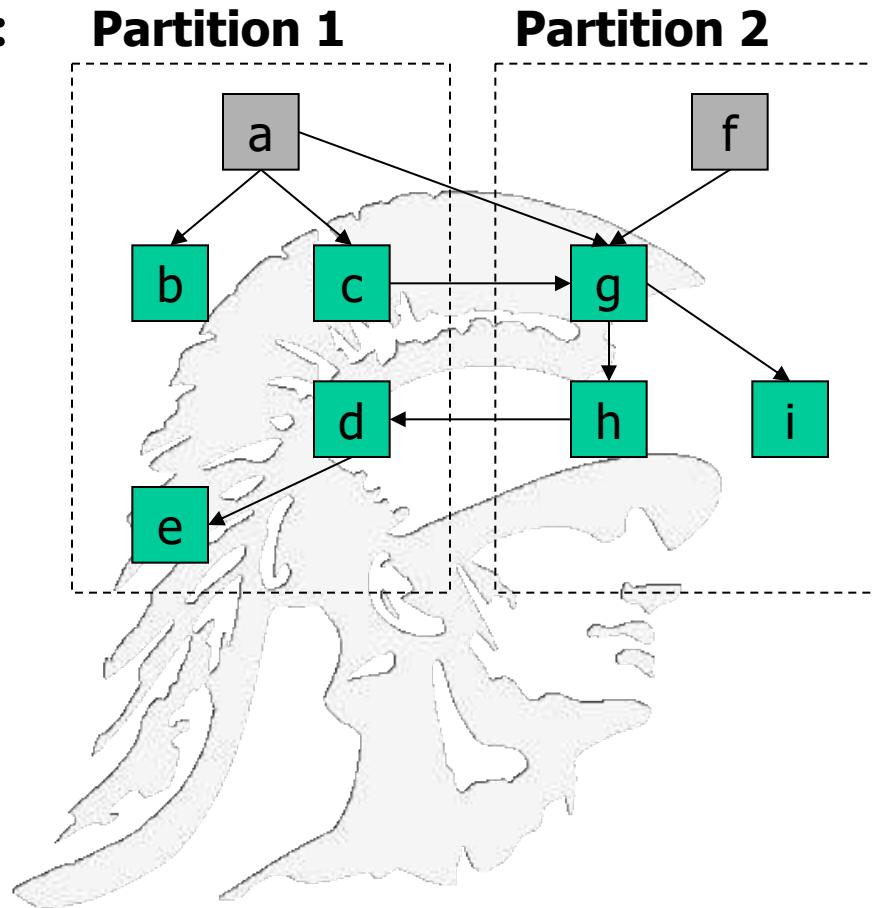
- ▶ Web is partitioned and assigned without a central coordinator before the crawl starts



Static Assignment

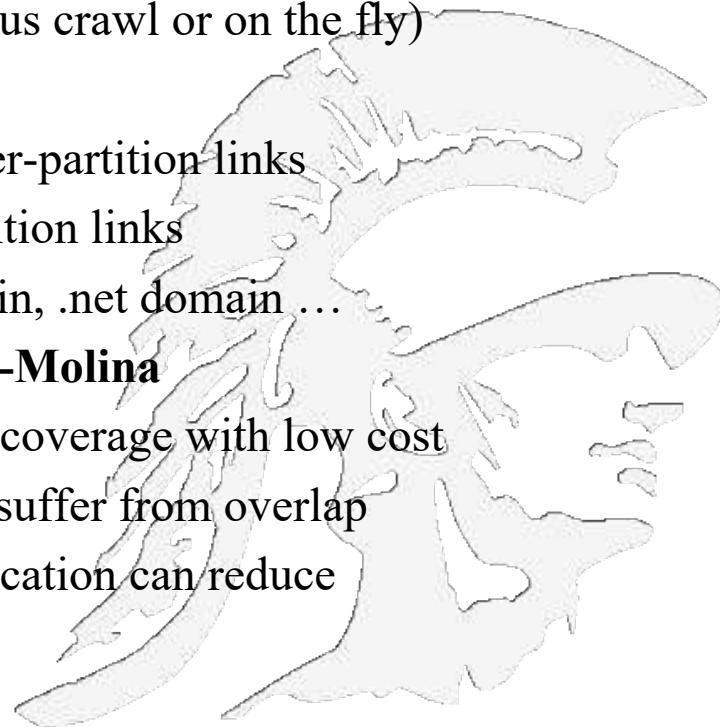
Links from one partition to another (inter-partition links) can be handled in one of three ways:

1. *Firewall mode:*
a process does not follow any inter-partition link
2. *Cross-over mode:*
a process also follows inter-partition links and possibly discovers also more pages in its partition
3. *Exchange mode:*
processes exchange inter-partition URLs; this mode requires communication

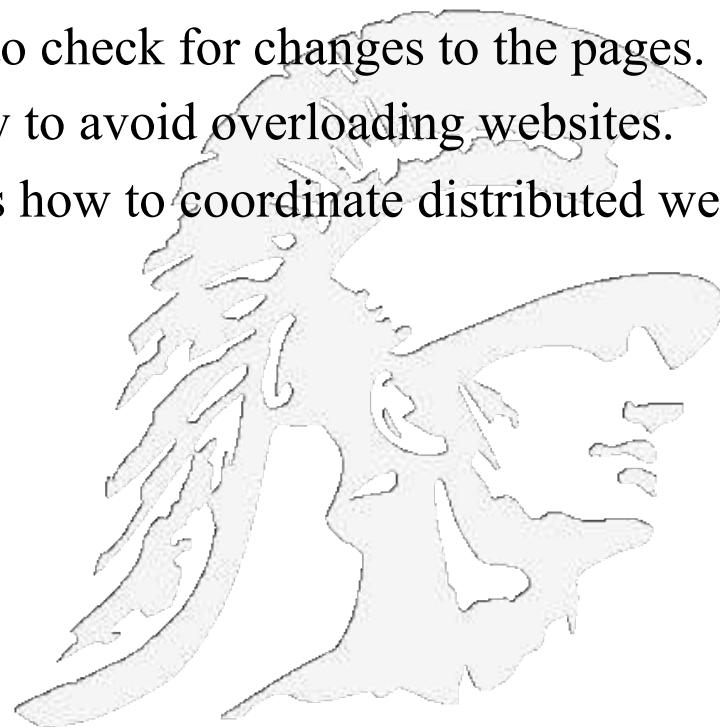


Classification of Parallel Crawlers

- If exchange mode is used, communication can be limited by:
 - Batch communication: every process collects some URLs and sends them in a batch
 - Replication: the k most popular URLs are replicated at each process and are not exchanged (previous crawl or on the fly)
- Some ways to partition the Web:
 - URL-hash based: this yields many inter-partition links
 - Site-hash based: reduces the inter partition links
 - Hierarchical: by TLD, e.g. .com domain, .net domain ...
- General Conclusions of Cho and Garcia-Molina
 - Firewall crawlers attain good, general coverage with low cost
 - Cross-over ensures 100% quality, but suffer from overlap
 - Replicating URLs and batch communication can reduce overhead

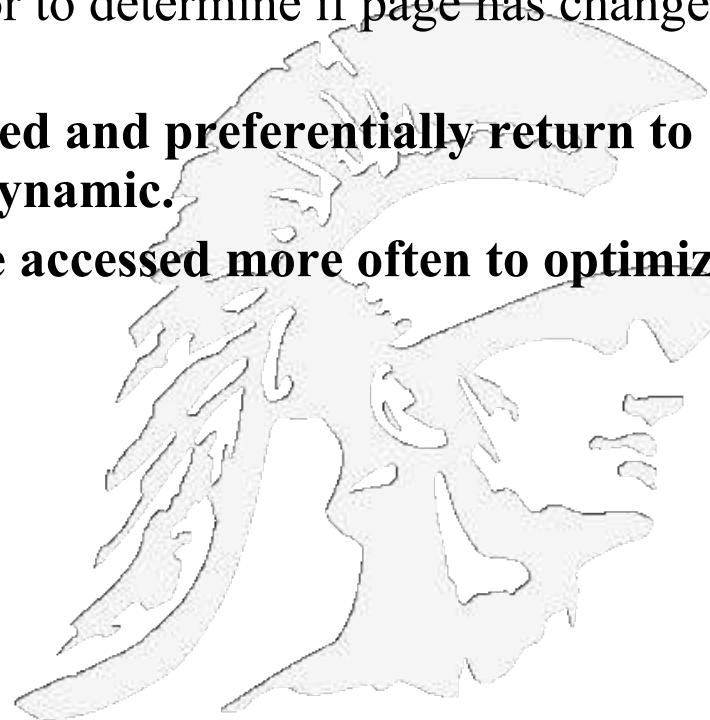


- **The behavior of a Web crawler is the outcome of a combination of policies:**
 - A *selection policy* that states which pages to download.
 - A *re-visit policy* that states when to check for changes to the pages.
 - A *politeness policy* that states how to avoid overloading websites.
 - A *parallelization policy* that states how to coordinate distributed web crawlers.



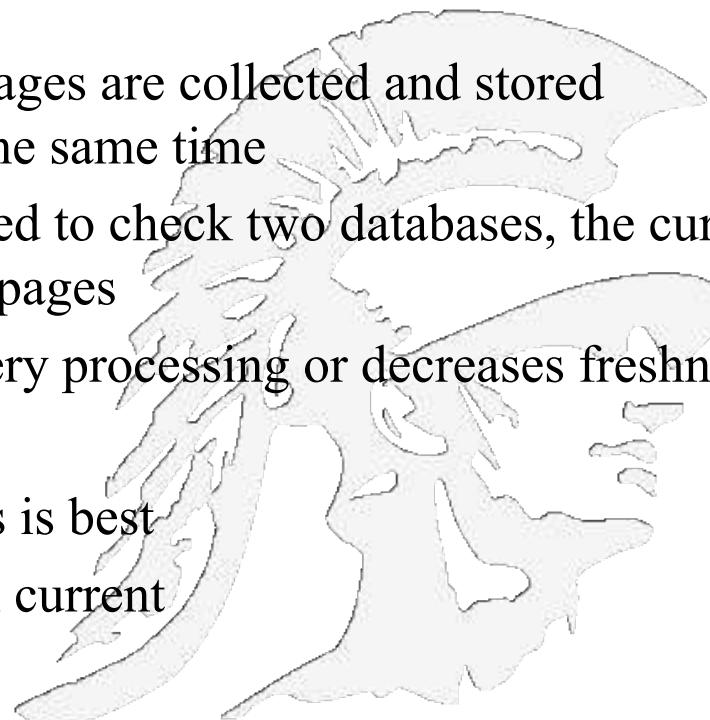
Keeping Spidered Pages Up to Date

- Web is very dynamic: many new pages, updated pages, deleted pages, etc.
- Periodically check crawled pages for updates and deletions:
 - Just look at LastModified indicator to determine if page has changed, only reload entire page if needed
- Track how often each page is updated and preferentially return to pages which are historically more dynamic.
- Preferentially update pages that are accessed more often to optimize freshness of more popular pages.



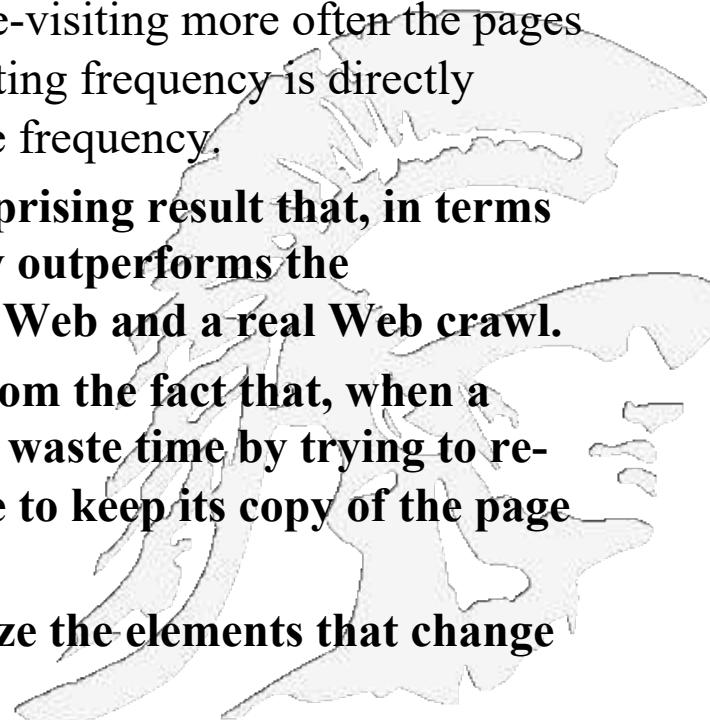
Implications for a Web Crawler

- **A steady crawler runs continuously without pause**
 - Typically search engines use multiple crawlers
- **When a crawler replaces an old version by a new page, does it do it “in-place” or “shadowing”**
 - Shadowing implies a new set of pages are collected and stored separately and all are updated at the same time
 - The above implies that queries need to check two databases, the current database and the database of new pages
 - Shadowing either slows down query processing or decreases freshness
- **Conclusions:**
 - running multiple types of crawlers is best
 - Updating in-place keeps the index current



Cho and Garcia-Molina, 2000

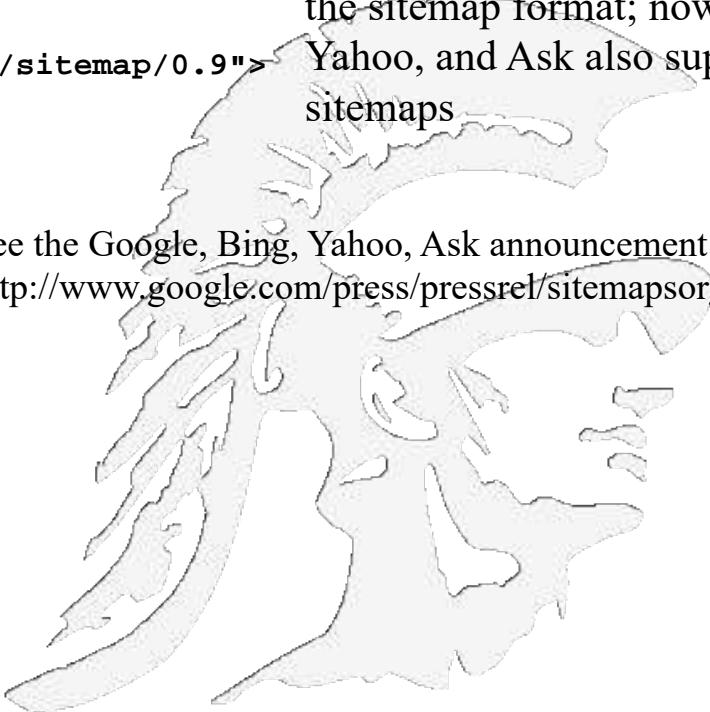
- Two simple re-visiting policies
 - Uniform policy: This involves re-visiting all pages in the collection with the same frequency, regardless of their rates of change.
 - Proportional policy: This involves re-visiting more often the pages that change more frequently. The visiting frequency is directly proportional to the (estimated) change frequency.
- Cho and Garcia-Molina proved the surprising result that, in terms of average freshness, the uniform policy outperforms the proportional policy in both a simulated Web and a real Web crawl.
- The explanation for this result comes from the fact that, when a page changes too often, the crawler will waste time by trying to re-crawl it too fast and still will not be able to keep its copy of the page fresh.
- To improve freshness, we should penalize the elements that change too often



Help the Search Engine Crawler Creating a SiteMap

- A sitemap is a list of pages of a web site accessible to crawlers
- This helps search engine crawlers find pages on the site
- XML is used as the standard for representing sitemaps
- Here is an example of an XML sitemap for a three page website

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
<url>
  <loc>http://www.example.com/?id=who</loc>
  <lastmod>2009-09-22</lastmod>
  <changefreq>monthly</changefreq>
  <priority>0.8</priority> </url>
<url>
  <loc>http://www.example.com/?id=what</loc>
  <lastmod>2009-09-22</lastmod>
  <changefreq>monthly</changefreq>
  <priority>0.5</priority> </url>
<url>
  <loc>http://www.example.com/?id=how</loc>
  <lastmod>2009-09-22</lastmod>
  <changefreq>monthly</changefreq>
  <priority>0.5</priority> </url>
</urlset>
```



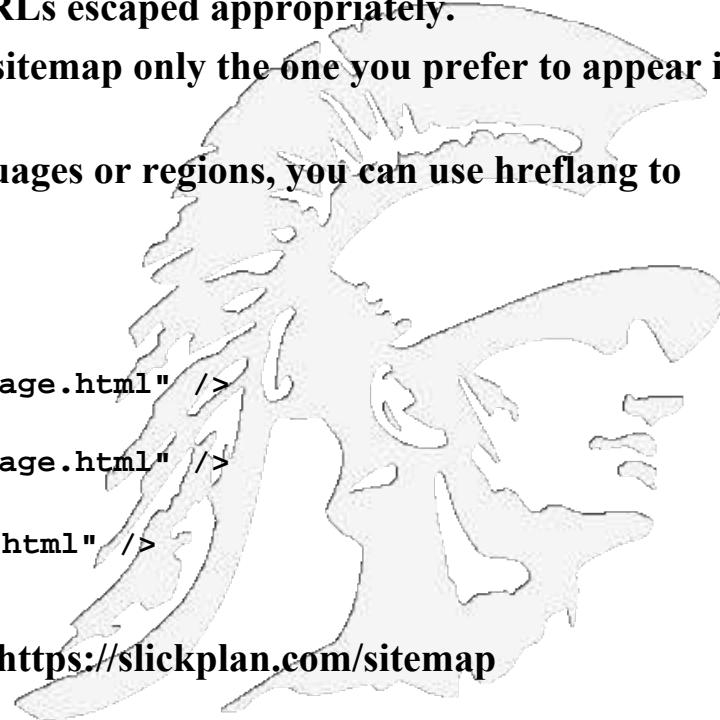
Back in 2006 Google introduced the sitemap format; now Bing, Yahoo, and Ask also support sitemaps

See the Google, Bing, Yahoo, Ask announcement:
<http://www.google.com/press/pressrel/sitemapsorg.html>

General Sitemap Guidelines

- Use consistent, fully-qualified URLs. Google will crawl your URLs exactly as listed
- A sitemap can be posted anywhere on your site, but a sitemap affects only descendants of the parent directory
- Don't include session IDs from URLs in your sitemap.
- Sitemap files must be UTF-8 encoded, and URLs escaped appropriately.
- If you have two versions of a page, list in the sitemap only the one you prefer to appear in search results
- If you have alternate pages for different languages or regions, you can use hreflang to indicate the alternate URLs.

```
<head>
  <title>Widgets, Inc</title>
  <link rel="alternate" hreflang="en-gb"
        href="https://en-gb.example.com/page.html" />
  <link rel="alternate" hreflang="en-us"
        href="https://en-us.example.com/page.html" />
  <link rel="alternate" hreflang="en"
        href="https://en.example.com/page.html" />
```
- There are many sitemap generator tools, e.g. <https://slickplan.com/sitemap>



Google Crawlers

- Google now uses multiple crawlers
 - APIs-Google
 - AdSense
 - AdsBot Mobile Web Android
 - AdsBot Mobile Web
 - AdsBot
 - Googlebot Images
 - Googlebot News
 - Googlebot Video
 - Googlebot (desktop)
 - Googlebot (smartphone)
 - Mobile AdSense
 - Mobile Apps Android
 - Feedfetcher
 - Google Read Aloud

Crawler	User agent token (product token)	Full user agent string
APIs-Google	APIs-Google-1.0	APIs-Google (https://www.gstatic.google.com/webmasters/tools/api.html?do=google.html)
AdSense	AdSense- partner -Google	AdSense- partner -Google
AdsBot Mobile Web Android	AdsBot-Google-Mobile	AdsBot-Google-Mobile (Android 5.0; DR-020A1; Apple-iPhone; 3G; Google Chrome Mobile Safari; AppleWebKit/536.23;KHTML, like Gecko; Chrome/22.0.1229.75; U; zh-CN; Google-Mobile) (https://www.google.com/mobile/adsbot.html)
AdsBot Mobile Web	AdsBot-Google-Mobile-2.0	AdsBot-Google-Mobile-2.0 (iPhone; CPU iPhone OS 9.1 like Mac OS X; AppleWebKit/536.23;KHTML, like Gecko; Chrome/22.0.1229.75; U; zh-CN; Google-Mobile) (https://www.google.com/mobile/adsbot.html)
AdsBot	AdsBot-Google	AdsBot-Google (https://www.google.com/adsbot.html)
Googlebot Images	Googlebot-Image	Googlebot-Image (Googlebot-Image/1.0)
Googlebot News	Googlebot-News	Googlebot-News (Googlebot-News/1.0)
Googlebot Video	Googlebot-Video	Googlebot-Video (Googlebot-Video/1.0)
Googlebot (Desktop)	Googlebot	Googlebot (Googlebot/2.0; compatible/Googlebot/2.0; https://www.google.com/search/genie.html)

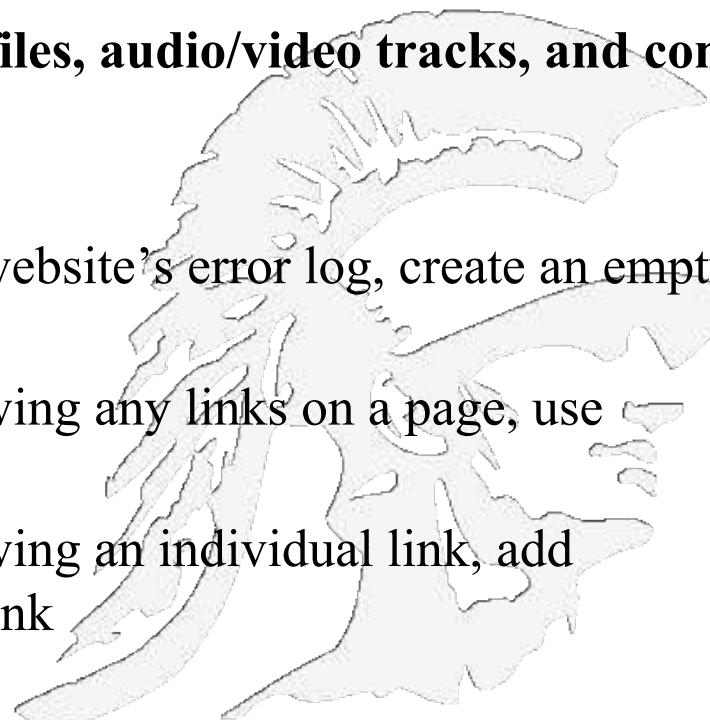
For details see

<https://support.google.com/webmasters/answer/1061943?hl=en>

see also Google's tool for checking how Googlebot sees your website
<https://support.google.com/webmasters/answer/6066468?rd=2>

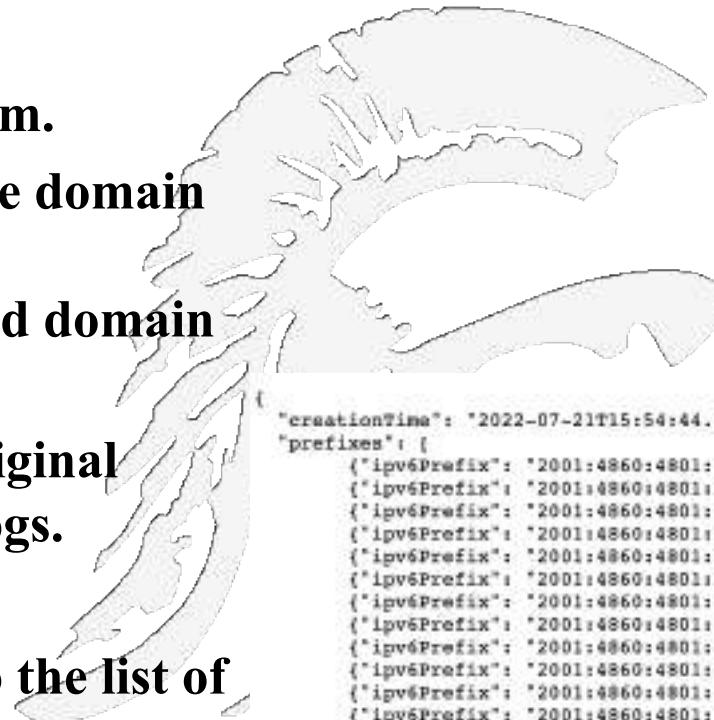
Google's Googlebot

- Begins with a list of webpage URLs generated from previous crawls
- Uses Sitemap data provided by webmasters
- Many versions of Googlebot are run on multiple machines located near the site they are indexing
- Googlebot cannot see within Flash files, audio/video tracks, and content within programs
- Advice
 - To prevent “File not found” in a website’s error log, create an empty robots.txt file
 - To prevent Googlebot from following any links on a page, use “nofollow” meta tag
 - To prevent Googlebot from following an individual link, add “rel=‘nofollow’” attribute to the link



To verify That The Visitor is Googlebot

- **Method 1**
 1. Run a reverse DNS lookup on the accessing IP address from your logs, using the host command.
 2. Verify that the domain name is either googlebot.com or google.com.
 3. Run a forward DNS lookup on the domain name retrieved in step 1 using the host command on the retrieved domain name.
 4. Verify that it's the same as the original accessing IP address from your logs.
- **Method 2**
 - match the crawler's IP address to the list of Googlebot IP addresses



```
"creationTime": "2022-07-21T15:54:44.265580",
"prefixes": [
    {"ipv6Prefix": "2001:4860:4801:10::/64"},
    {"ipv6Prefix": "2001:4860:4801:11::/64"},
    {"ipv6Prefix": "2001:4860:4801:12::/64"},
    {"ipv6Prefix": "2001:4860:4801:13::/64"},
    {"ipv6Prefix": "2001:4860:4801:14::/64"},
    {"ipv6Prefix": "2001:4860:4801:15::/64"},
    {"ipv6Prefix": "2001:4860:4801:16::/64"},
    {"ipv6Prefix": "2001:4860:4801:17::/64"},
    {"ipv6Prefix": "2001:4860:4801:18::/64"},
    {"ipv6Prefix": "2001:4860:4801:19::/64"},
    {"ipv6Prefix": "2001:4860:4801:1a::/64"},
    {"ipv6Prefix": "2001:4860:4801:1b::/64"}]
```

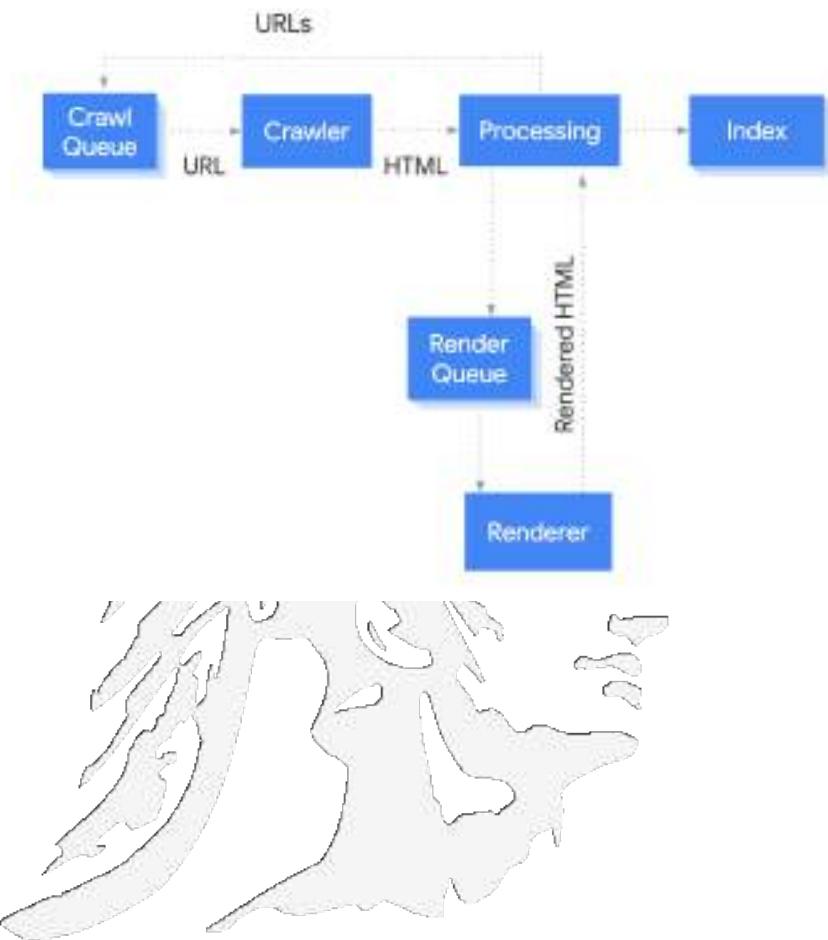
Controlling How Often Googlebot Visits Your Site

- The term *crawl rate* means how many requests per second Googlebot makes to your site when it is crawling it: for example, 5 requests per second.
- You cannot change how often Google crawls your site, but if you want Google to crawl new or updated content on your site, you can request a recrawl; for details see
 - <https://developers.google.com/search/docs/advanced/crawling/ask-google-to-recrawl>
- You can reduce the crawl rate by
 - returning pages with 500, 503 or 429 http status codes
 - Setting a new rate in the Search Console
 - A video discussing Google's Crawl Status Report can be found here:
<https://support.google.com/webmasters/answer/9679690>

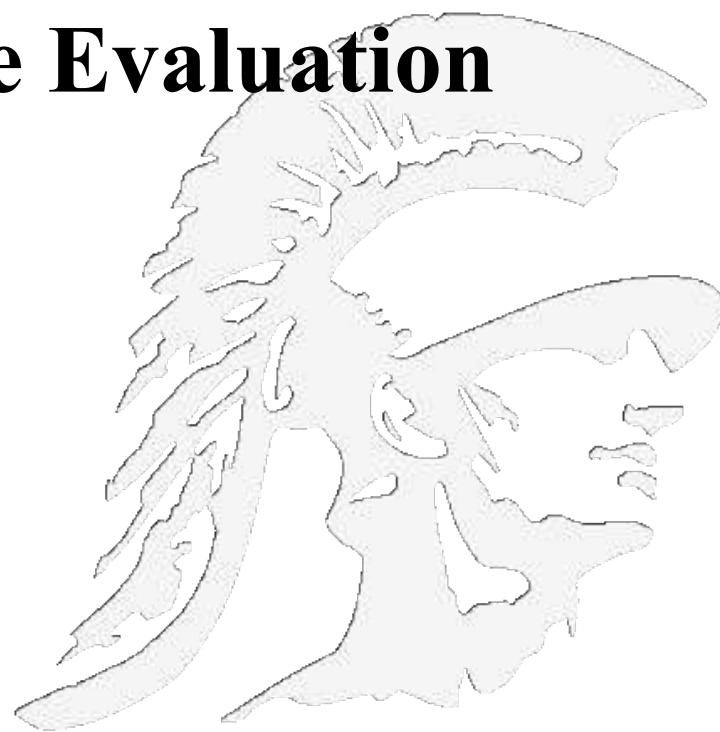


Googlebot is Really Chrome Re-packaged

- **Why:** browsers don't just render the DOM hierarchy of HTML, they include transformations via CSS and JavaScript, and for Googlebot to extract the most meaningful features from a web page it would be necessary to have access to these transformations
 - Therefore Googlebot must understand and execute JavaScript code
- **Conclusion:** Googlebot and Chrome share a great deal of code
- **Googlebot processes web pages with JavaScript in 3 phases**
 1. Crawling – processing all links
 2. Rendering – executing JS and then looping back to 1
 3. Indexing
- As of 2019 Googlebot runs the latest Chromium rendering engine
- **Note: Server-side rendering saves Googlebot from rendering the page**



Search Engine Evaluation

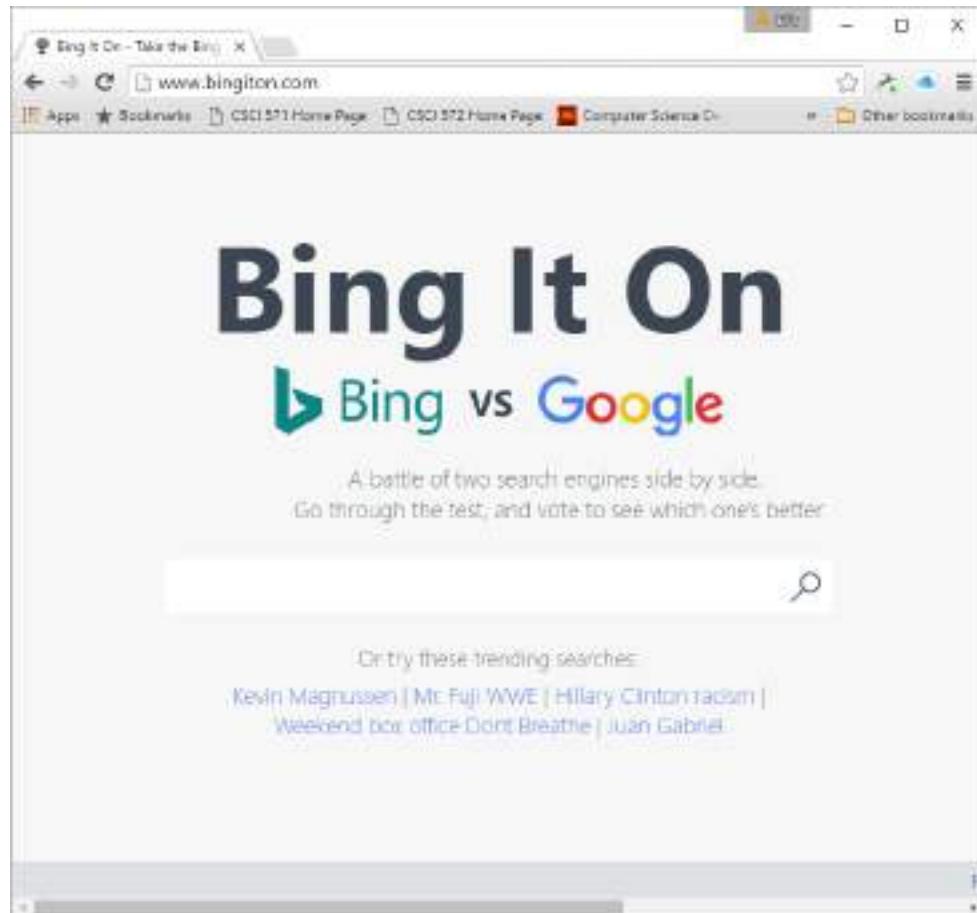


Outline

- **Defining precision/recall**
- **Mean Average Precision**
- **Harmonic Mean and F Measure**
- **Discounted Cumulative Gain**
- **Elements of Good Search Results**
- **Google's Search Quality Guidelines**
- **Using log files for evaluation**
- **A/B Testing**



Comparing Bing and Google



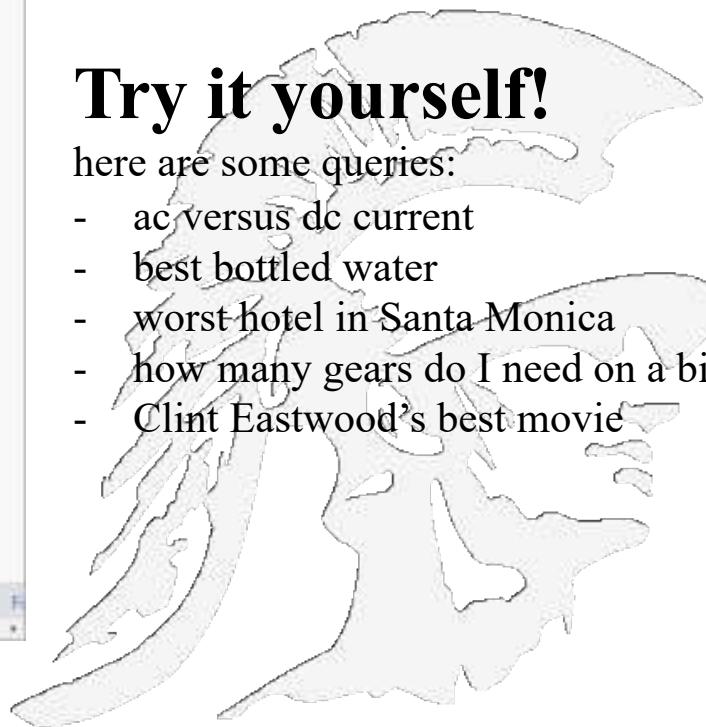
The screenshot shows a web browser window with the title "Bing It On - Take the Bing". The address bar contains "www.bingiton.com". The page itself features a large "Bing It On" logo with "Bing vs Google" below it. A sub-headline reads "A battle of two search engines side by side. Go through the test, and vote to see which one's better." There is a search bar with a magnifying glass icon and a link to "Or try these trending searches: Kevin Magnussen | Mt Fuji | WWE | Hillary Clinton | Weekend box office | Don't Breath | Juan Gabriel".

- This site is no longer active, but we can simulate the experiment

Try it yourself!

here are some queries:

- ac versus dc current
- best bottled water
- worst hotel in Santa Monica
- how many gears do I need on a bicycle
- Clint Eastwood's best movie



- How do we measure the quality of search engines?
- Precision = #(relevant items retrieved)
divided by
#(all retrieved items)
- Recall = #(relevant items retrieved)
divided by
#(all relevant items)



Formalizing Precision/Recall

	Relevant	Nonrelevant
Retrieved	True positive (tp)	False positive (fp)
Not retrieved	False negative (fn)	True negative (tn)

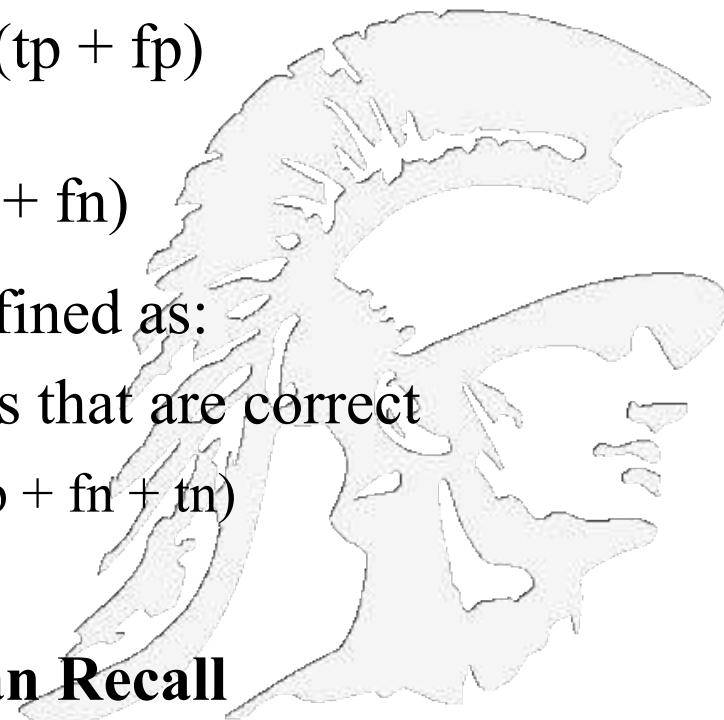
$$\text{Precision} = \text{tp}/(\text{tp} + \text{fp})$$

$$\text{Recall} = \text{tp}/(\text{tp} + \text{fn})$$

- The accuracy of an engine is defined as:
the fraction of these classifications that are correct

$$(\text{tp} + \text{tn}) / (\text{tp} + \text{fp} + \text{fn} + \text{tn})$$

**For web applications,
Precision is more important than Recall**



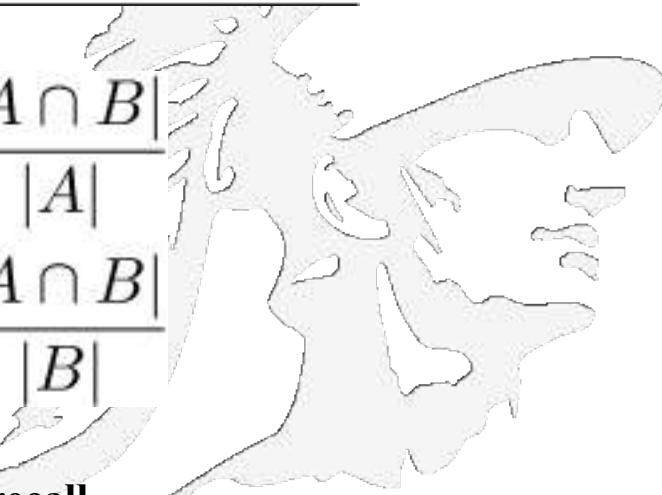
Precision/Recall Using Set Notation

A is set of relevant documents,
 B is set of retrieved documents

	Relevant	Non-Relevant
Retrieved	$A \cap B$	$\overline{A} \cap B$
Not Retrieved	$A \cap \overline{B}$	$\overline{A} \cap \overline{B}$

you may not be able
 to see them, but
 A and B have a bar
 over them and it
 denotes the
 complement set

$$\text{Recall} = \frac{|A \cap B|}{|A|}$$

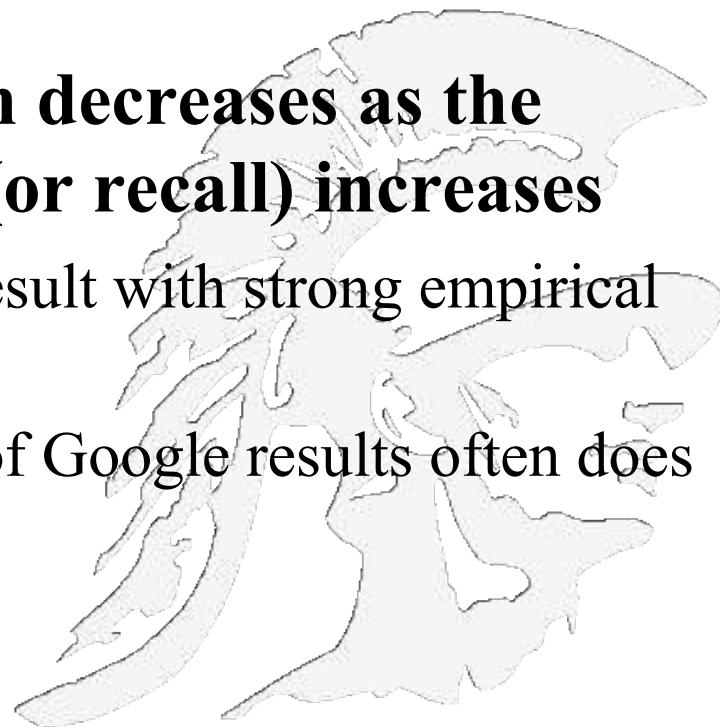
$$\text{Precision} = \frac{|A \cap B|}{|B|}$$


https://en.wikipedia.org/wiki/Precision_and_recall

Precision/Recall

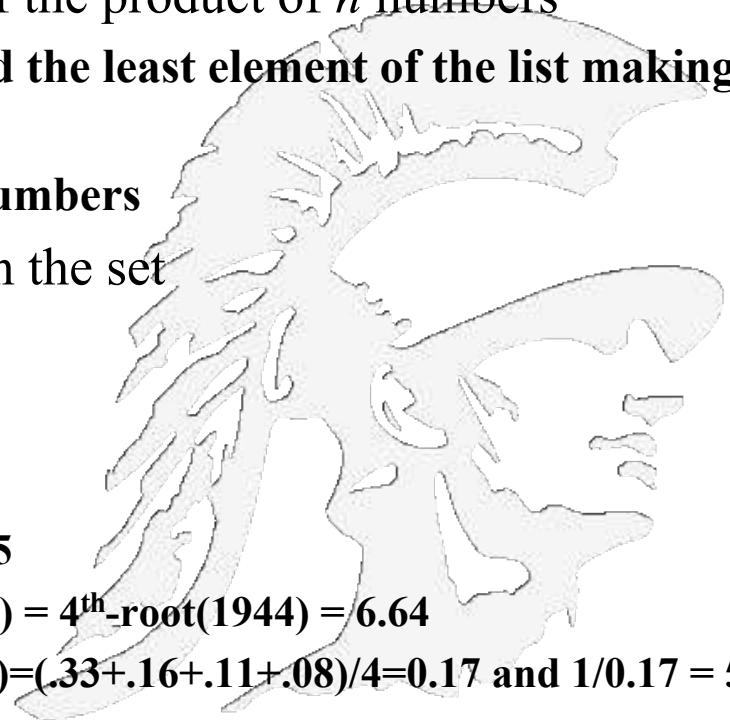
Two Observations

- You can get high recall (but low precision) by retrieving all docs for all queries!
 - a rather foolish strategy
- In a good system, precision decreases as the number of docs retrieved (or recall) increases
 - This is not a theorem, but a result with strong empirical confirmation
 - E.g. viewing multiple pages of Google results often does not improve precision at all



Harmonic Mean

- There are three Pythagorean means
 - 1. *arithmetic mean*, 2. *geometric mean*, 3. *harmonic mean*
 - of course we all know how to compute the arithmetic mean
 - the geometric mean is the n th root of the product of n numbers
- The harmonic mean tends strongly toward the least element of the list making it useful in analyzing search engine results
- To find the harmonic mean of a set of n numbers
 1. add the reciprocals of the numbers in the set
 2. divide the sum by n
 3. take the reciprocal of the result
- e.g. for the numbers 3, 6, 9, and 12
 - The arithmetic mean is: $(3+6+9+12)/4 = 7.5$
 - The geometric mean is: $\text{nth-root}(3*6*9*12) = 4^{\text{th-root}}(1944) = 6.64$
 - The harmonic mean is: $(1/3+1/6+1/9+1/12)=(.33+.16+.11+.08)/4=0.17 \text{ and } 1/0.17 = 5.88$



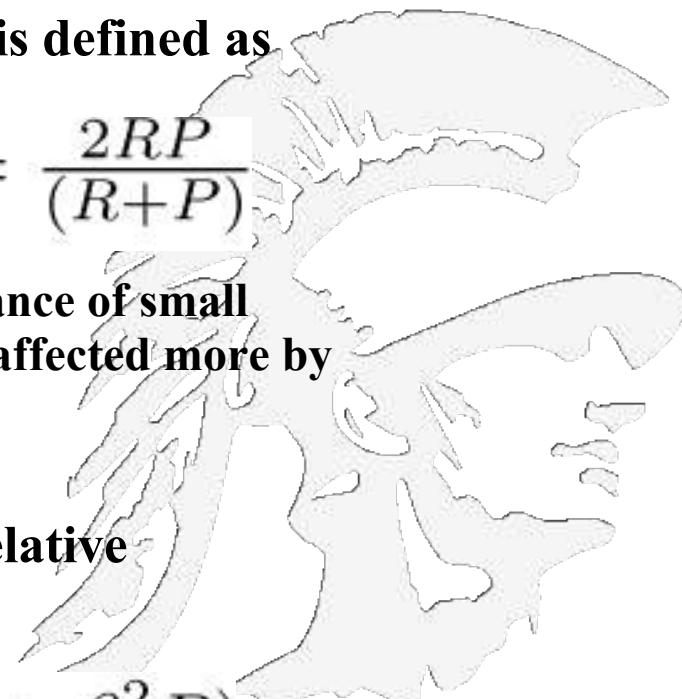
F Measure

- The harmonic mean of the precision and the recall is often used as an aggregated performance score for the evaluation of algorithms and systems: called the **F-score (or F-measure)**.
- *Harmonic mean* of recall and precision is defined as

$$F = \frac{1}{\frac{1}{2}(\frac{1}{R} + \frac{1}{P})} = \frac{2RP}{(R+P)}$$

- harmonic mean emphasizes the importance of small values, whereas the arithmetic mean is affected more by outliers that are unusually large
- More general form of F-Measure
 - β is a parameter that controls the relative importance of recall and precision

$$F_\beta = (\beta^2 + 1)RP / (R + \beta^2 P)$$



Calculating Recall/Precision at Fixed Positions



= the relevant documents

Steps

Recall:

$$1/6 = 0.17$$

Precision:

$$1/1$$

Ranking #1



e.g.
Google
Result

Recall:

$$1/6 = 0.17$$

Precision:

$$1/2 = 0.5$$

	Recall	0.17	0.17	0.33	0.5	0.67	0.83	0.83	0.83	1.0
	Precision	1.0	0.5	0.67	0.75	0.8	0.83	0.71	0.63	0.56



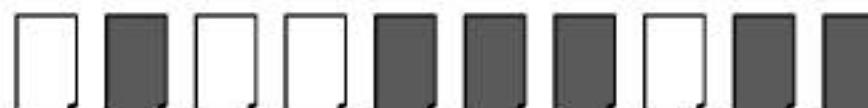
Recall:

$$2/6 = 0.33$$

Precision:

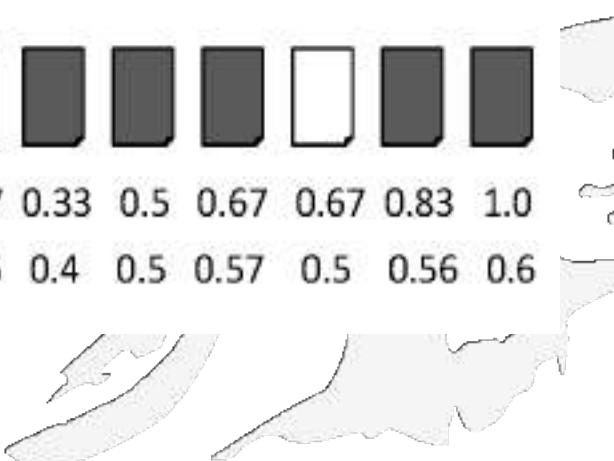
$$2/3 = 0.67$$

Ranking #2



e.g.
Bing
Result

	Recall	0.0	0.17	0.17	0.17	0.33	0.5	0.67	0.67	0.83	1.0
	Precision	0.0	0.5	0.33	0.25	0.4	0.5	0.57	0.5	0.56	0.6



Recall:

$$3/6 = 0.5$$

Precision:

$$3/4 = 0.75$$

$$\text{Recall} = \#RelevItemsRetr / \text{allRelevItems}$$

$$\text{Prec} = \#RelevItemsRetr / \text{allItemsRetr}$$

Average Precision of the Relevant Documents



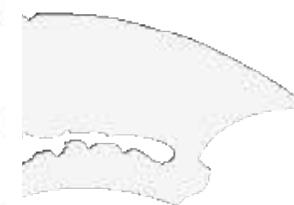
= the relevant documents

computes the sum of the precisions of the relevant documents

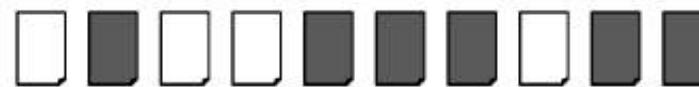
Ranking #1



	Recall	0.17	0.17	0.33	0.5	0.67	0.83	0.83	0.83	1.0
Precision	1.0	0.5	0.67	0.75	0.8	0.83	0.71	0.63	0.56	0.6



Ranking #2



	Recall	0.0	0.17	0.17	0.17	0.33	0.5	0.67	0.67	0.83	1.0
Precision	0.0	0.5	0.33	0.25	0.4	0.5	0.57	0.5	0.56	0.6	



$$\text{Ranking } \#1: (1.0 + 0.67 + 0.75 + 0.8 + 0.83 + 0.6) / 6 = 0.78$$

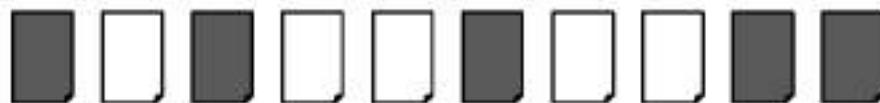
$$\text{Ranking } \#2: (0.5 + 0.4 + 0.5 + 0.57 + 0.56 + 0.6) / 6 = 0.52$$

Conclusion: Ranking #1 for this query is best

Averaging Across Queries

 = relevant documents for query 1

Ranking #1



Recall 0.2 0.2 0.4 0.4 0.4 0.6 0.6 0.6 0.8 1.0

Precision 1.0 0.5 0.67 0.5 0.4 0.5 0.43 0.38 0.44 0.5

 = relevant documents for query 2

Ranking #2

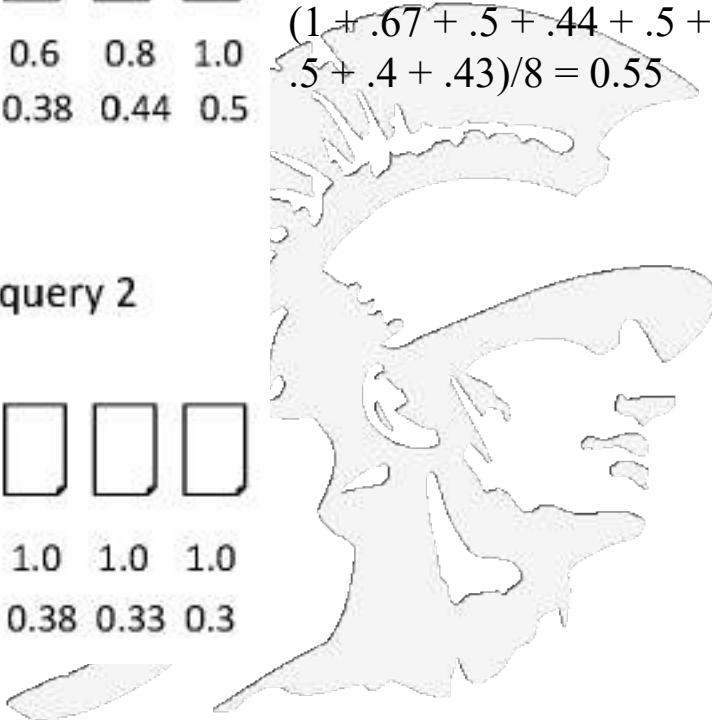


Recall 0.0 0.33 0.33 0.33 0.67 0.67 1.0 1.0 1.0 1.0

Precision 0.0 0.5 0.33 0.25 0.4 0.33 0.43 0.38 0.33 0.3

Average precision across the two queries for relevant docs is:

$$(1 + .67 + .5 + .44 + .5 + .5 + .4 + .43)/8 = 0.55$$



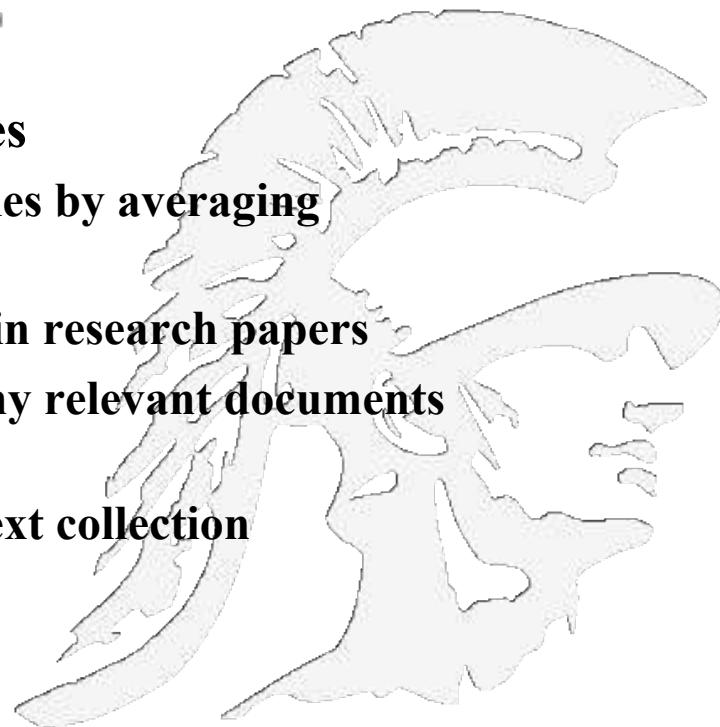
Averaging Across Queries

- ***Mean average precision (MAP)*** for a set of queries is the mean of the average precision scores for each query.

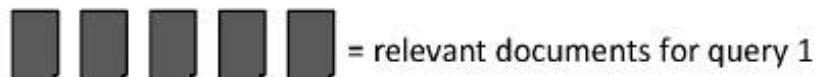
$$\text{MAP} = \frac{\sum_{q=1}^Q \text{AveP}(q)}{Q}$$

where Q is the number of queries

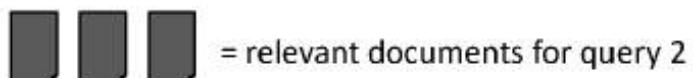
- Summarize rankings from multiple queries by averaging average precision
- This is the ***most commonly used*** measure in research papers
- Assumes user is interested in finding many relevant documents for each query
- Requires many relevance judgments in text collection



Mean Average Precision Example



Ranking #1										
Recall	0.2	0.2	0.4	0.4	0.4	0.6	0.6	0.8	1.0	
Precision	1.0	0.5	0.67	0.5	0.4	0.5	0.43	0.38	0.44	0.5



Ranking #2										
Recall	0.0	0.33	0.33	0.33	0.67	0.67	1.0	1.0	1.0	1.0
Precision	0.0	0.5	0.33	0.25	0.4	0.33	0.43	0.38	0.33	0.3

$$\text{average precision query 1} = (1.0 + 0.67 + 0.5 + 0.44 + 0.5) / 5 = 0.62$$

$$\text{average precision query 2} = (0.5 + 0.4 + 0.43) / 3 = 0.44$$

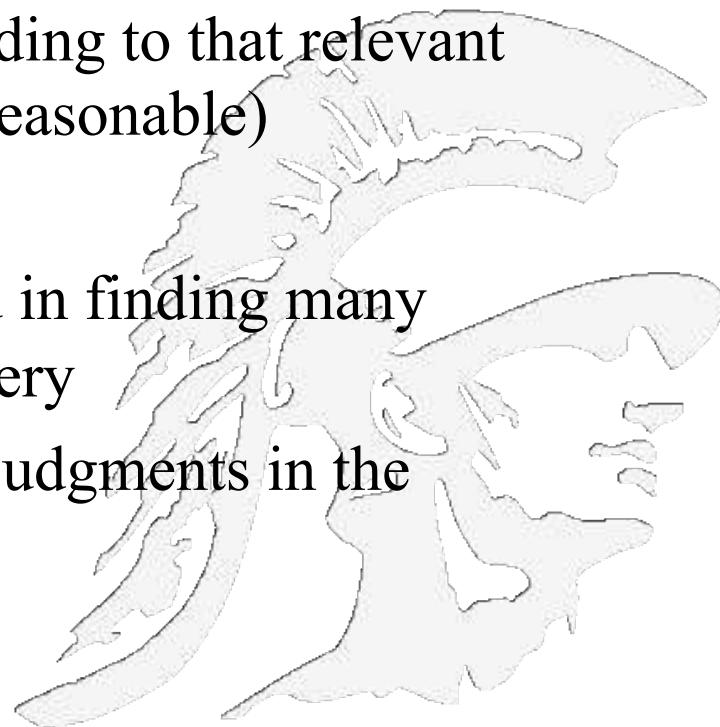
$$\text{mean average precision} = (0.62 + 0.44) / 2 = 0.53$$



More on Mean Average Precision Calculation

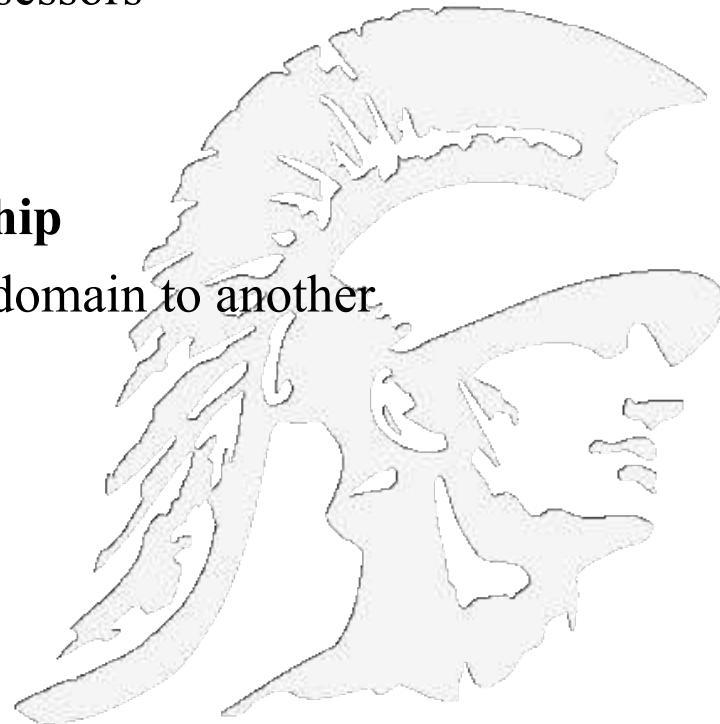
■ Mean Average Precision (MAP)

- Some negative aspects
 - If a relevant document never gets retrieved, we assume the precision corresponding to that relevant doc to be zero (this is actually reasonable)
 - Each query counts equally
 - MAP assumes user is interested in finding many relevant documents for each query
 - MAP requires many relevance judgments in the document collection



Difficulties in Using Precision/Recall

- Should average over large document collection and query ensembles
- Need human relevance assessments
 - But people aren't always reliable assessors
- Assessments have to be binary
 - Nuanced assessments?
- Heavily skewed by collection/authorship
 - Results may not translate from one domain to another



A Final Evaluation Measure: Discounted Cumulative Gain

- The premise of DCG is that highly relevant documents appearing lower in a search result list should be penalized as the graded relevance value is reduced logarithmically proportional to the position of the result.
- The discounted CG accumulated at a particular rank position p is defined as

$$\text{DCG}_p = \sum_{i=1}^p \frac{\text{rel}_i}{\log_2(i+1)} = \text{rel}_1 + \sum_{i=2}^p \frac{\text{rel}_i}{\log_2(i+1)}$$

where rel_i is the graded relevance of the result at position i

- Gain is accumulated starting at the top of the ranking and may be reduced, or *discounted*, at lower ranks
- Typical discount is $1/\log(\text{rank})$
 - With base 2, the discount at rank 4 is $1/2$, and at rank 8 it is $1/3$
- An alternative formulation of DCG places stronger emphasis on retrieving relevant documents:

$$\text{DCG}_p = \sum_{i=1}^p \frac{2^{\text{rel}_i} - 1}{\log_2(i+1)}$$

Discounted Cumulative Gain Example

we want high weights for high rank documents, because searchers are likely to inspect them, and low weights for low rank documents that searchers are unlikely to ever see.

The discount factor is commonly chosen as $\log_2(\text{rank} + 1)$ and is used to divide the relevance grade.

Using a logarithm for the position penalty makes the decay effect more gradual compared to using the position itself.

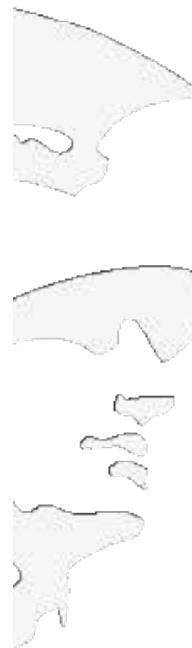
Discount examples

Rank	Grade	Discount1 [1/rank]	Discount2 [log2(rank + 1)]	Discount1 Grade	Discount2 Grade
1	4	1.000	1.000	4.000	4.000
2	3	0.500	0.631	1.500	1.893
3	2	0.333	0.500	0.667	1.000
4	1	0.250	0.431	0.250	0.431
5	1	0.200	0.387	0.200	0.387

Search Engine Evaluation Metrics

Metrics table

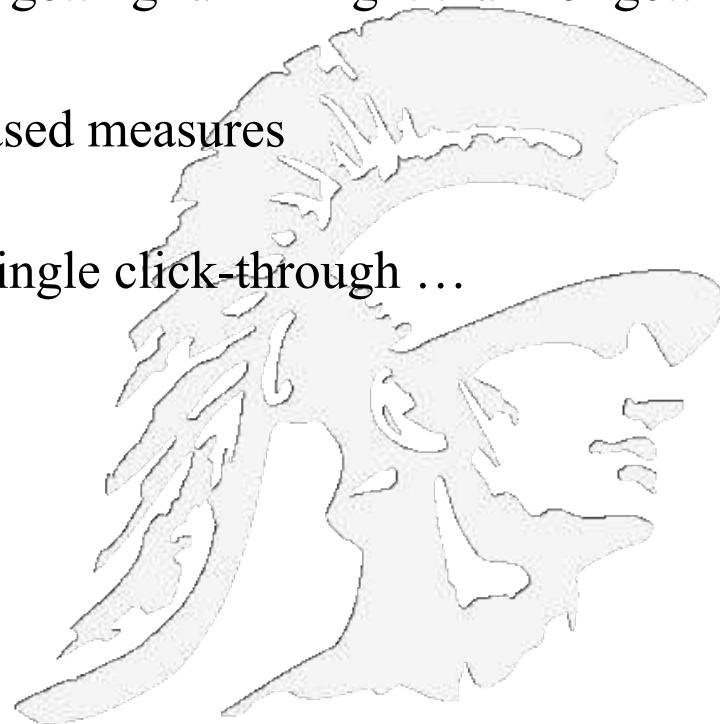
Scale	Metric	Measures	Drawbacks
Binary	Precision (P)	The relevance of the entire results set (gridded results display)	Doesn't account for position
Binary	Average Precision (AP)	Relevance to a user scanning results sequentially	Large impact of low-rank results
Graded	Cumulative Gain (CG)	Information gain from a results set	Same as Precision doesn't factor in position
Graded	Discount Cumulative Gain (DCG)	Information gain with positional weighting	Difficult to compare across queries
Graded	normalized DCG (nDCG)	How close the results are to the best possible	No longer shows information gain



Finally see [https://en.wikipedia.org/wiki/Evaluation_measures_\(information_retrieval\)](https://en.wikipedia.org/wiki/Evaluation_measures_(information_retrieval))
 Copyright Ellis Horowitz 2011-2022

How Evaluation is Done at Web Search Engines

- Search engines have test collections of queries and hand-ranked results
- Recall is difficult to measure on the web
- Search engines often use precision at top k positions, e.g., $k = 10$
- . . . or measures that reward you more for getting rank 1 right than for getting rank 10 right.
- Search engines also use non-relevance-based measures
 - Click-through on first result
 - Not very reliable if you look at a single click-through . . .
but pretty reliable in the aggregate.
 - Studies of user behavior in the lab
 - A/B testing



Google's Search Quality Rating Guidelines Document

- Google relies on raters, working in many countries and languages around the world
- The data they generate is rolled up statistically to give
 - a view of the quality of search results and search experience over time, and
 - an ability to measure the effect of proposed changes to Google's search algorithms

General Guidelines

October 14, 2020

General Guidelines Overview	5
Introduction to Search Quality Rating	6
3.1 The Search Experience	6
3.1 The Purpose of Search Quality Rating	6
3.2 Raters Must Represent People in their Rating Locale	6
3.3 Browser Requirements	7
3.4 Ad Blocking Extensions	7
3.5 Internet Safety Information	7
3.6 The Role of Examples in These Guidelines	7
Part 1: Page Quality Rating Guideline	8
1.1 Introduction to Page Quality Rating	8
2.0 Understanding Webpages and Websites	9
2.1 Important Definitions	9
2.2 What is the Purpose of a Webpage?	9
2.3 Your Money or Your Life (YMYL) Pages	10
2.4 Understanding Webpage Content	10
2.4.1 Identifying the Main Content (MC)	10
2.4.2 Identifying the Supplementary Content (SC)	11
2.4.3 Identifying Advertisements/Monetization (Ads)	11
2.4.4 Summary of the Parts of the Page	12
2.5 Understanding the Website	12
2.5.1 Finding the Homepage	12
2.5.2 Finding Who is Responsible for the Website and Who Created the Content on the Page	14
2.5.3 Finding About Us, Contact Information, and Customer Service Information	14
3.0 Reputation of the Website or Creator of the Main Content	15
3.6.1 Research on the Reputation of the Website or Creator of the Main Content	15
3.6.2 Sources of Reputation Information	16
3.6.3 Customer Reviews of Stores/Businesses	16
3.6.4 How Search for Reputation Information	16
3.6.5 What to Do When You Find No Reputation Information	18
3.8 Overall Page Quality Rating	19
3.1 Page Quality Rating: Most Important Factors	19
3.2 Expertise, Authoritativeness, and Trustworthiness (E-A-T)	19
4.0 High Quality Pages	20
4.1 Characteristics of High Quality Pages	20
4.2 A Satisfying Amount of High Quality Main Content	21
4.3 Clear and Satisfying Website Information: Who is Responsible and Customer Service	21
4.4 Positive Reputation	21
4.5 A High Level of Expertise/Authoritativeness/Trustworthiness (E-A-T)	22
4.6 Examples of High Quality Pages	22
3.8 Highest Quality Pages	25

http://csci572.com/papers/2020_10searchqualityevaluatorguidelines.pdf

Google's Search Quality Ratings Guidelines Document

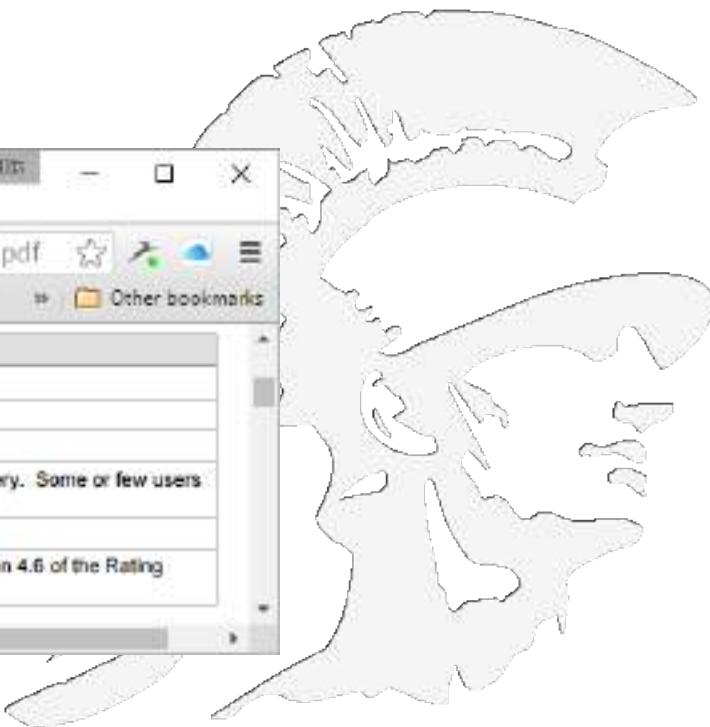
- This document gives evaluators examples and guidelines for appropriate ratings.
- the evaluator looks at a search query and a result that could be returned. They rate the relevance of the result for that query on a scale described within the document.

The six rating scale categories



A screenshot of a web browser window displaying a table titled "Rating Scale". The table has two columns: "Rating Scale" and "Description". The rows are: Vital, Useful, Relevant, Slightly Relevant, Off-Topic or Useless, and Unreliable. Each row contains a brief description of what that rating category means.

Rating Scale	Description
Vital	A special rating category. See Section 4.1 of the Rating Guidelines.
Useful	A page that is very helpful for most users.
Relevant	A page that is helpful for many or some users.
Slightly Relevant	A page that is not very helpful for most users, but is somewhat related to the query. Some or few users would find this page helpful.
Off-Topic or Useless	A page that is helpful for very few or no users.
Unreliable	A page that cannot be evaluated. A complete description can be found in Section 4.6 of the Rating Guidelines.



1. Precision Evaluations

People use the Guidelines to rate search results

2. Side-by-Side Experiments

people are shown two different sets of search results and asked which they prefer

3. Live Traffic Experiments

the search algorithm is altered for a small number of actual users

4. Full Launch

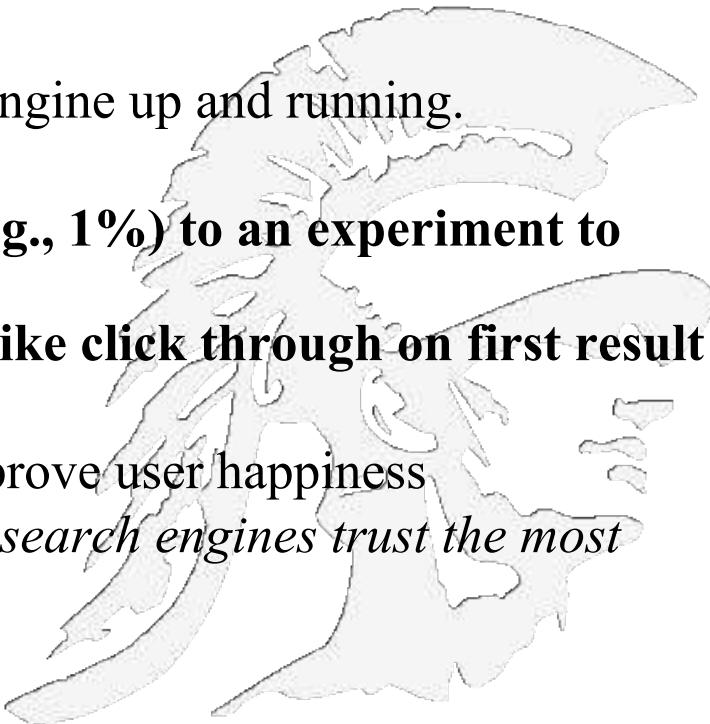
A final analysis by Google engineers and the improvement is released

Google's 4-Step Process for Changing Their Search Algorithm



A/B Testing at Web Search Engines

- **A/B testing** is comparing two versions of a web page to see which one performs better. You compare two web pages by showing the two variants (let's call them **A** and **B**) to similar visitors at the same time. The one that gives a better conversion rate, wins!
- 1. **Purpose:** Test a single innovation
- 2. **Prerequisite:** You have a large search engine up and running.
- 3. **Have most users use old system**
- 4. **Divert a small proportion of traffic (e.g., 1%) to an experiment to evaluate an innovation**
- 5. **Evaluate with an automatic measure like click through on first result**
 - we directly see if the innovation does improve user happiness
 - *This is the evaluation methodology large search engines trust the most*



USING USER CLICKS FOR EVALUATION



What Do Clicks Tell Us?



There is strong position bias, so absolute click rates unreliable

Relative vs Absolute Ratings

ALL RESULTS

RELATED SEARCHES
CIKM 2008

SEARCH HISTORY
Turn on search history to start remembering your searches.
Turn history on

ALL RESULTS

1-10 of 131,000 results · Advanced

CIKM 2008 | Home
Napa Valley Marriott Hotel & Spa: Napa Valley, California October 26-30, 2008
[cikm2008.org](#) · Cached page

- Papers
- Themes
- Important Dates
- Banquet
- Program Committee
- News
- Napa Valley
- Posters

Show more results from cikm2008.org

Conference on Information and Knowledge Management (CIKM)
Provides an international forum for presentation and discussion of research on information and knowledge management, as well as recent advances on data and knowledge bases...
[www.cikm.org](#) · Cached page

Conference on Information and Knowledge Management (CIKM'02)
SAIC Headquarters, McLean, Virginia, USA, 4-9 November 2002.
[www.cikm.org/2002](#) · Cached page

ACM CIKM 2007 - Lisbon, Portugal
News and announcements: 12/02 - Best interdisciplinary paper award at CIKM 2007 went to Fei Wu and Daniel Weld for Autonomously Semantifying Wikipedia.
[www.fc.ul.pt/cikm2007](#) · Cached page

CIKM 2009 | Home
CIKM 2009 (The 18th ACM Conference on Information and Knowledge Management) will be held on November 2-6, 2009, Hong Kong. Since 1992, CIKM has successfully brought together...
[www.comp.polyu.edu.hk/conference/cikm2009](#) · Cached page

Conference on Information and Knowledge Management (CIKM)
CIKM Conference on Information and Knowledge Management The Conference on Information and Knowledge Management (CIKM) provides an international forum for presentation and...
[cikmconference.org](#) · Cached page

User's click sequence

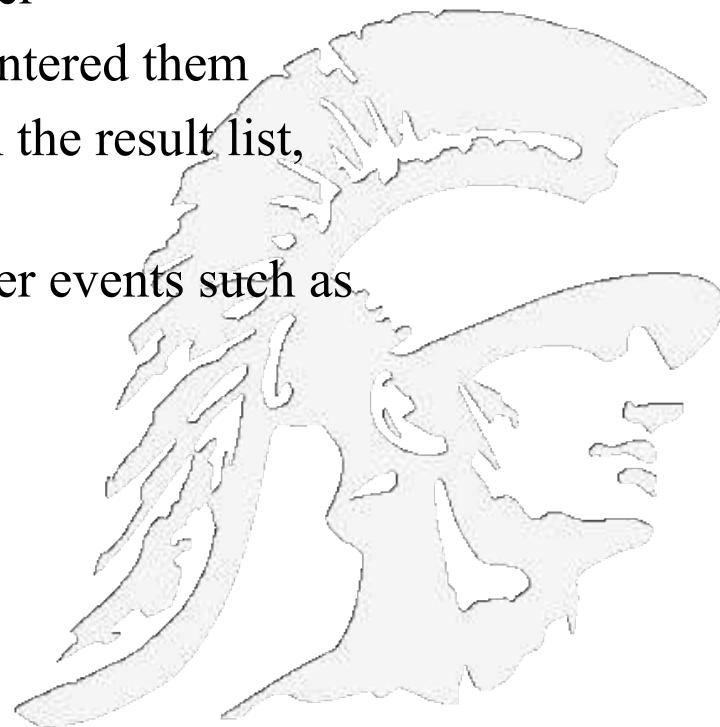


Hard to conclude Result1 > Result3

Probably can conclude Result3 > Result2

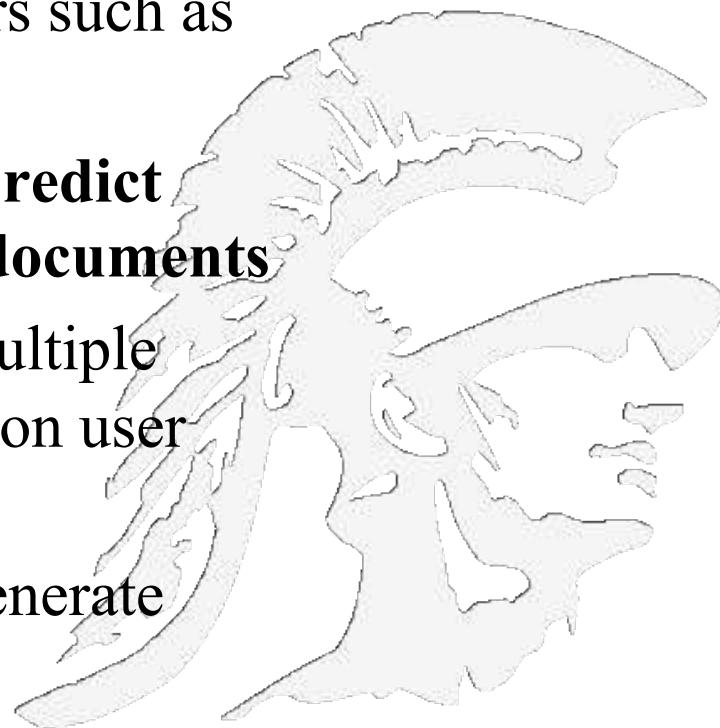
Query Logs

- **Used for both tuning and evaluating search engines**
 - also for various techniques such as query suggestion
- **Typical contents of the query log files**
 - User identifier or user session identifier
 - Query terms - stored exactly as user entered them
 - List of URLs of results, their ranks on the result list, and whether they were clicked on
 - Timestamp(s) - records the time of user events such as query submission, clicks



How Query Logs Can Be Used

- **Clicks are not relevance judgments**
 - although they are correlated
 - biased by a number of factors such as rank on result list
- **Can use clickthrough data to predict *preferences* between pairs of documents**
 - appropriate for tasks with multiple levels of relevance, focused on user relevance
 - various “policies” used to generate preferences



A Final Thought

Google's Enhancements of Search Results

Display improvements

- immediate answers
- autocomplete anticipations

Extensions to More Data

- results from books
- results from news
- results from images
- results from patents
- results from air schedules

New Input forms

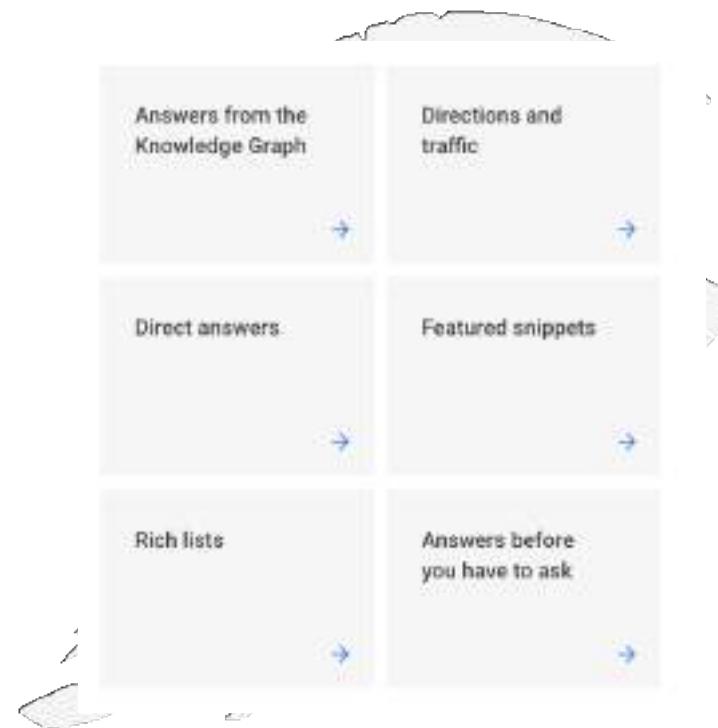
- search by voice
- search by image

information retrieval improvements

- snippets
- spelling correction
- translations
- People Also Ask boxes
- use of synonyms
- use of knowledge graph

The page below discusses the many aspects that go into producing search results at Google

<https://www.google.com/search/howsearchworks>



Final Thought

- See wikipedia on
- https://en.wikipedia.org/wiki/Comparison_of_web_search_engines

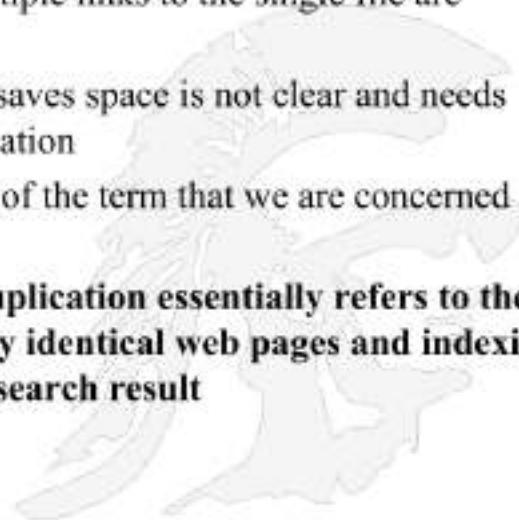


← 1/30 → *** 9:06:57

Deduplication

••

- *De-Duplication* – the process of identifying and avoiding essentially identical web pages
- The term is often used in connection with *locker storage* where only a single copy of a file is stored and multiple links to the single file are managed
 - Whether this strategy effectively saves space is not clear and needs analysis for each particular application
 - However, this is **not** the meaning of the term that we are concerned about in this class
- **With respect to *web crawling*, de-duplication essentially refers to the identification of identical and nearly identical web pages and indexing only a single version to return as a search result**



- One example is the same page, referenced by different URLs

<http://espn.go.com>

<http://www.espn.com>



- How can two URLs differ yet still point to the same page?

- the URL's host name can be distinct (virtual hosts) sharing the same document folder,
- the URL's protocol can be distinct (http, https), but still deliver the same document
- the URL's path and/or page name can be distinct

Copyright 2011-2022 Ellis Horowitz

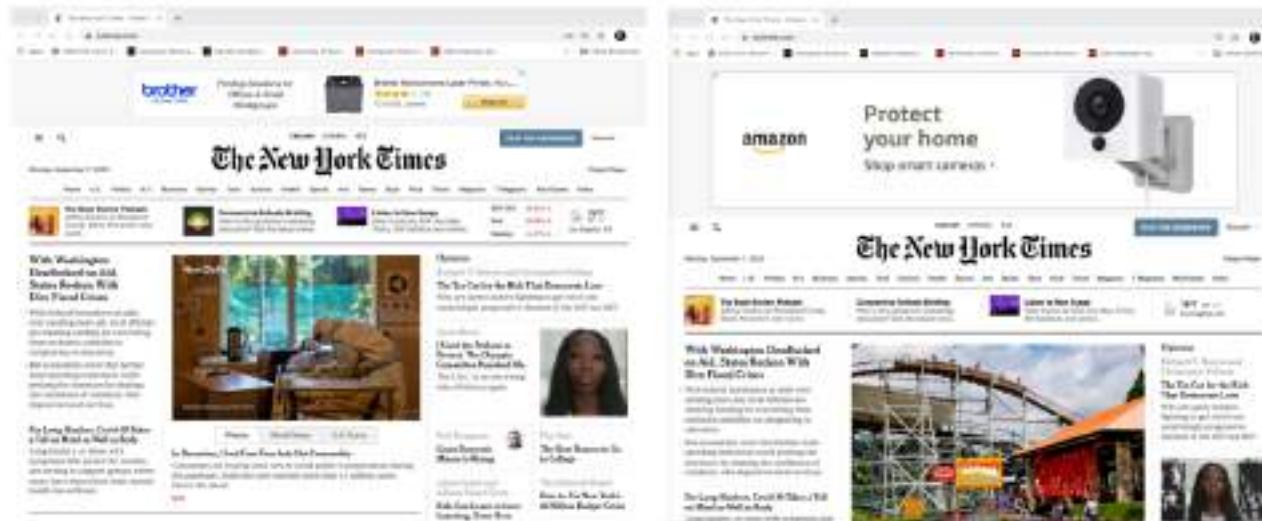
3

- At one time* all 3 URLs below pointed to the identical page
- Structural Classification of Proteins
 - <http://scop.mrc-lmb.cam.ac.uk/scop>
 - <http://scop.berkeley.edu/>
 - <http://scop.proteins.ru/>
- **The three URLs have distinct domain names, but all redirect to the same page**

* At least they did when I took this snapshot, no longer



Another example is two web pages whose content differs slightly



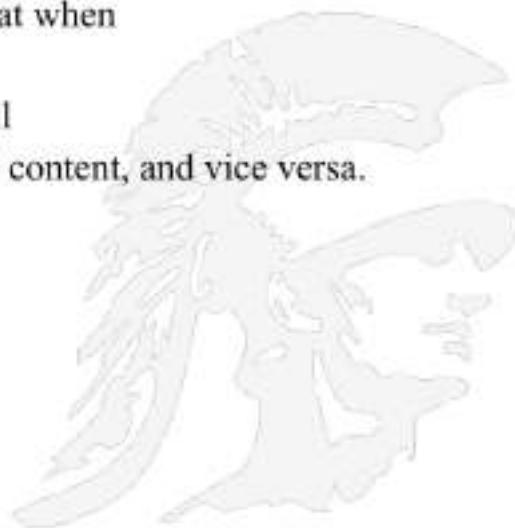
Two copies of www.nytimes.com snapshot within a few seconds of each other;
The pages are essentially identical except for the ads at the top and the photo in the middle

- In examining a web page a search engine may want to ignore ads, navigation links and other elements that do not specifically relate to the contents of the web page
- One way to do this is to delve into the structure of a web page and focus on content blocks
- E.g. the Document Object Model for HTML displays a web page as a tree hierarchy
 - Document
 - Head
 - Body
- However this is time consuming



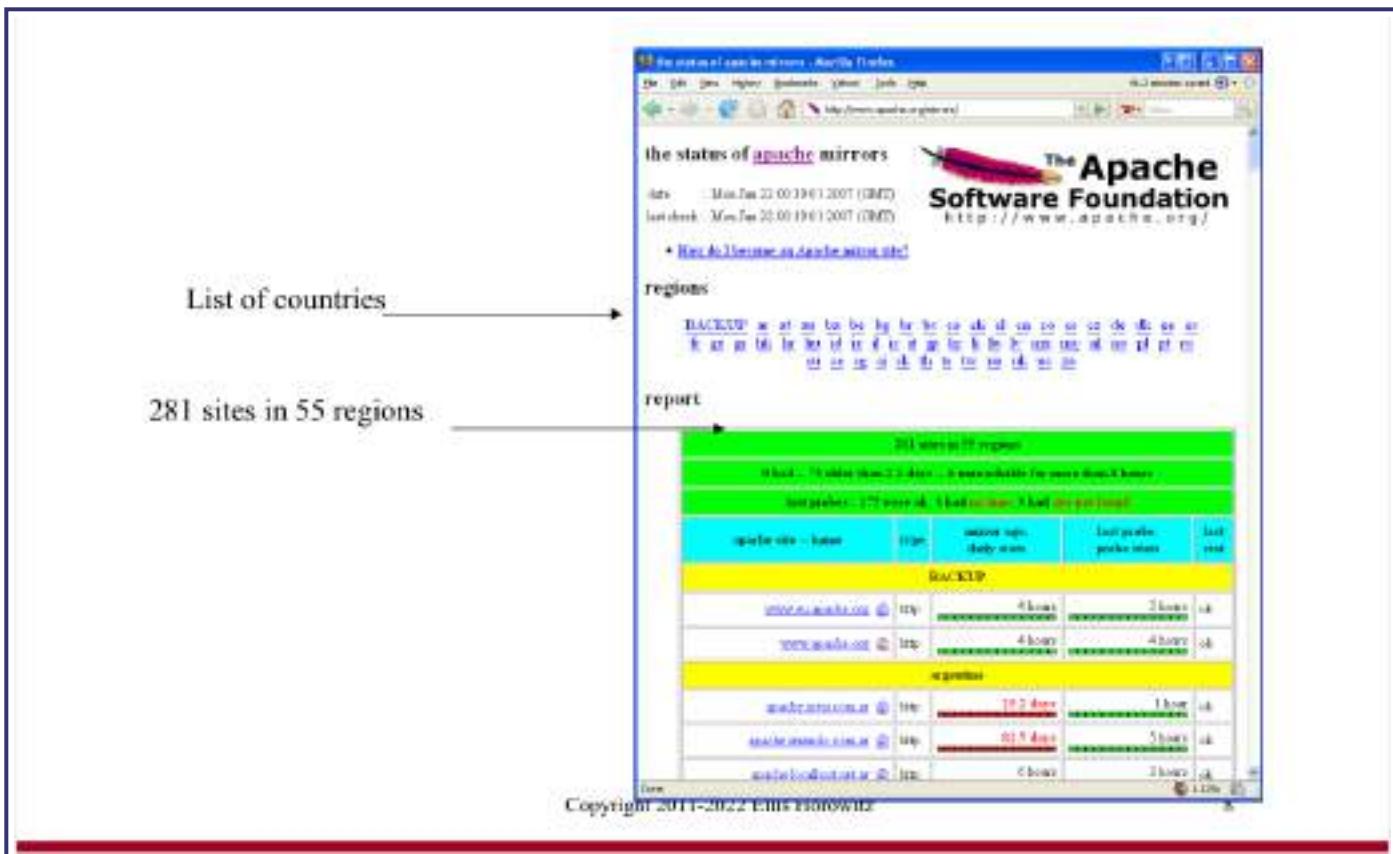
••

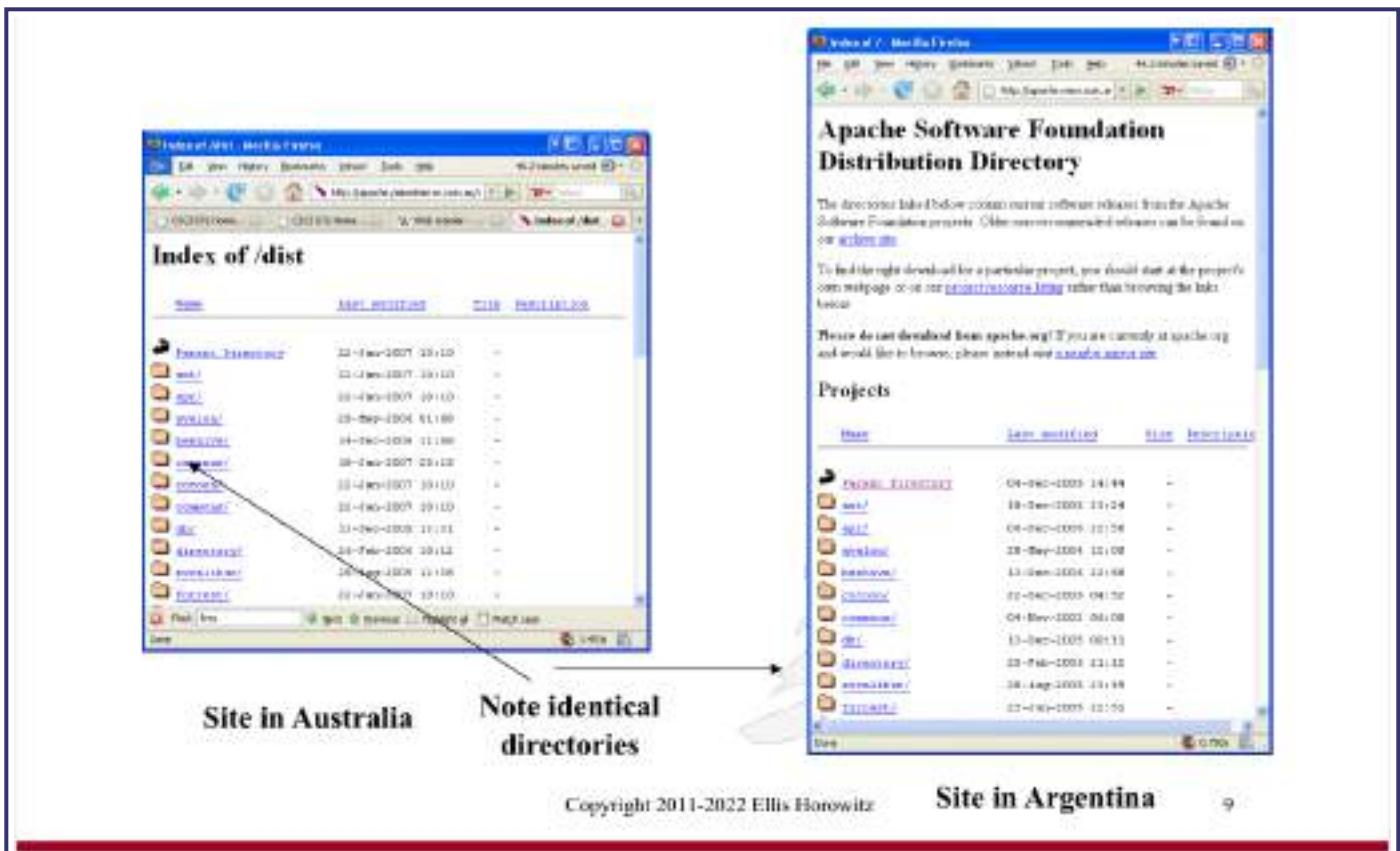
- Mirroring is the systematic replication of web pages across hosts.
 - Mirroring is the **single largest cause** of duplication on the web
- Host1/ α and Host2/ β are mirrors iff
 - For all (or most) paths p such that when
 $\text{http://Host1/ } \alpha / p$ exists
 - $\text{http://Host2/ } \beta / p$ exists as well
 - with identical (or near identical) content, and vice versa.



Copyright 2011-2022 Ellis Horowitz

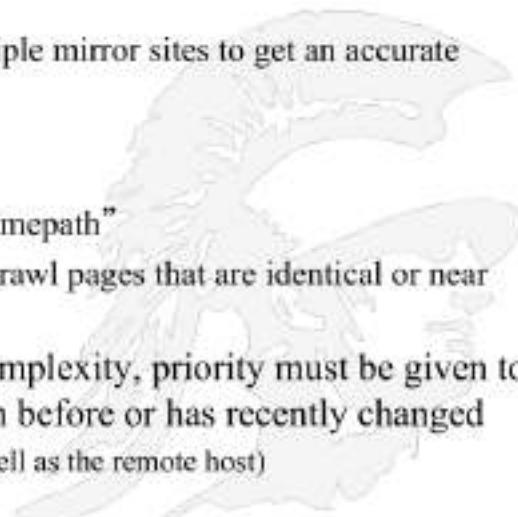
7





••

- ***Smarter crawling***
 - Avoid returning many duplicate results to a query
 - Allow fetching from the fastest or freshest server
- ***Better connectivity analysis***
 - By combining in-links from the multiple mirror sites to get an accurate PageRank (measure of importance)
 - Avoid double counting out-links
- ***Add redundancy in result listings***
 - “If that fails you can try: <mirror>/samepath”
- ***Reduce Crawl Time***: Crawlers need not crawl pages that are identical or near identical
- ***Ideally***: given the web's scale and complexity, priority must be given to content that has **not** already been seen before or has recently changed
 - Saves resources (on the crawler end, as well as the remote host)
 - Increases crawler politeness
 - Reduces the analysis that a crawler will have to do later

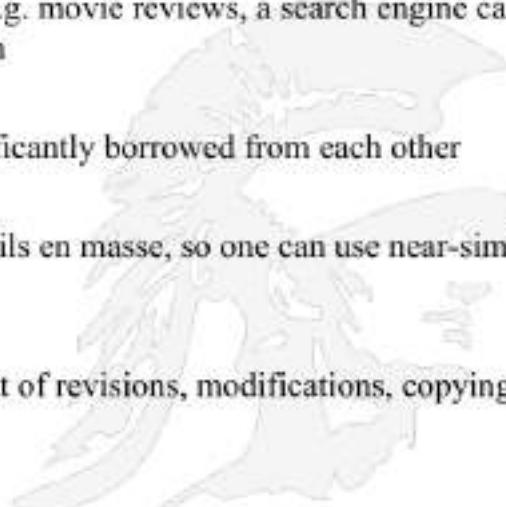


Copyright 2011-2022 Ellis Horowitz

10

••

- **Clustering**
 - Given a news article some people might wish to see “related articles” describing the same event
- **Data extraction**
 - Given a collection of similar pages, e.g. movie reviews, a search engine can extract and categorize the information
- **Plagiarism**
 - Identify pairs that seem to have significantly borrowed from each other
- **Spam detection**
 - Spammers typically send similar emails en masse, so one can use near-similarity techniques to identify the spam
- **Duplicates within a domain**
 - To identify near-duplicates arising out of revisions, modifications, copying or merging of documents



••

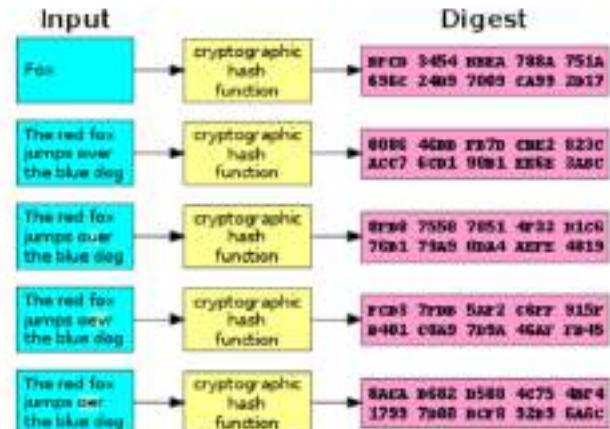
1. **Duplicate Problem: Exact match;**

- Solution: compute fingerprints using cryptographic hashing
- Useful for URL matching and also works for detecting identical web pages
- Hashes can be stored in sorted order for $\log N$ access

2. **Near-Duplicate Problem: Approximate match**

- Solution: compute the syntactic similarity with an edit-distance measure, and
- Use a similarity threshold to detect near-duplicates
 - e.g., Similarity > 80% => Documents are “near duplicates”
- The remaining slides are devoted to specific methods for duplicate and near duplicate detection

- A **cryptographic hash function** is a hash function which takes an input (or 'message') and returns a fixed-size alphanumeric string, which is called the **hash value** (sometimes called a **message digest**, **digital fingerprint**, **digest** or a **checksum**).
- The cryptographic hash function has four main properties:
 1. It is extremely easy (i.e. fast) to calculate a hash for any given data.
 2. It is extremely computationally difficult to calculate an alphanumeric text that has a given hash.
 3. A small change to the text yields a totally different hash value.
 4. It is extremely unlikely that two slightly different messages will have the same hash.



A small change in a single word, "over" produces a major Change in the output; see
https://en.wikipedia.org/wiki/Cryptographic_hash_function

• •

- The **MD5** (message-digest) hash function is a widely used cryptographic hash function producing a 128-bit (16-byte) hash value, typically expressed in text format as a 32 digit hexadecimal number.
 - Invented by Ron Rivest of MIT in 1991; replaced the earlier MD4
- The **SHA-1, SHA-2** hash functions are also quite popular (160 bit, 20 byte value)
 - SHA-1 was broken in 2005; using SHA-2 family of algorithms is now favored, see
 - <https://en.wikipedia.org/wiki/SHA-2>
- **SHA-3**, released in 2015; it produces digests of size 224, 256, 384 and 512 bits
- **RIPEMD-160** – a family of cryptographic hash functions and so far has not been broken; produces a 160 bit (20 byte) digest
- **E.g. See Chrome, Settings, Security and Privacy, Security, Manage certificates, certificates, Verisign**

••

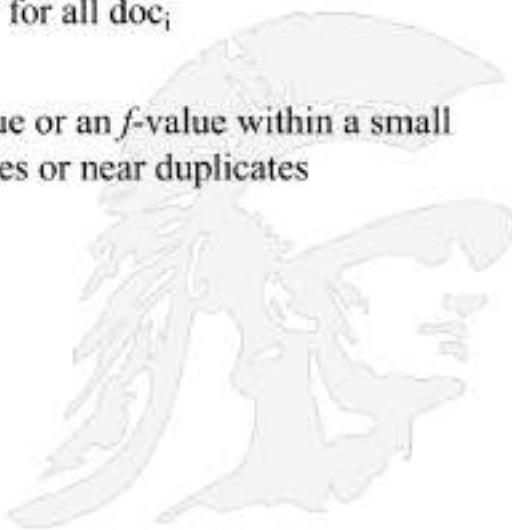
1. **Compare character by character** two documents to see if they are identical
 - very time consuming !!
2. **Hash just the first few characters and compare** only those documents that hash to the same bucket
 - But what about web pages where every page begins with <HTML> ??
3. **Use a hash function** that examines the entire document
 - But this requires lots of buckets
4. **Better approach** - pick some fixed random positions for all documents and make the hash function depend only on these;
 - This avoids the problem of a common prefix for all or most documents, yet we need not examine entire documents unless they fall into a bucket with another document
 - But we still need a lot of buckets
5. **Even better approach:** Compute the cryptographic hash (SHA-2 or MD5) of each web page and maintain in sorted order, $O(\log n)$ to search

••

1. ***Produce fingerprints and test for similarity*** - Treat web documents as defined by a set of features, constituting an n -dimensional vector, and transform this vector into an f -bit fingerprint of a small size
 - Use Simhash or Hamming Distance to compute the fingerprint
 - SimHash is an algorithm for testing how similar two sets are
 - Compare fingerprints and look for a difference in at most k bits
 - E.g. see Manku et al., WWW 2007, *Detecting Near-Duplicates for Web Crawling*, <http://www2007.org/papers/paper215.pdf>
2. ***Instead of documents defined by n-vector of features, compute subsets of words (called shingles) and test for similarity of the sets***
 - Broder et al., WWW 1997, *Finding Near Duplicate Documents*

••

1. Define a function f that captures the contents of each document in a number
 - E.g. hash function, signature, or a fingerprint
2. Create the pair $\langle f(doc_i), ID \text{ of } doc_i \rangle$ for all doc_i
3. Sort the pairs
4. Documents that have the same f -value or an f -value within a small threshold are believed to be duplicates or near duplicates



••

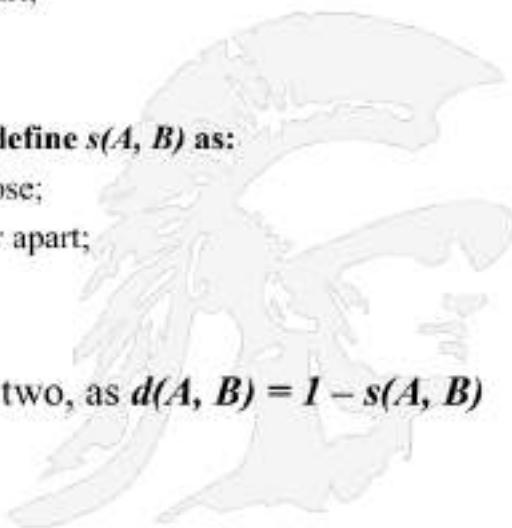
- To compute similarity, we need a distance measure
- A distance measure must satisfy 4 properties
 1. No negative distances
 2. $D(x,y) = 0$ iff $x=y$
 3. $D(x,y) = D(y,x)$ symmetric
 4. $D(x,y) \leq D(x,z) + D(z,y)$ triangle inequality
- There are several distance measures that can play a role in locating duplicate and near-duplicate documents
 - Euclidean distance - $D([x_1 \dots x_n], [y_1, \dots, y_n]) = \sqrt{\sum (x_i - y_i)^2}$ $i=1 \dots n$
 - Jaccard distance - $D(x,y) = 1 - \text{SIM}(x,y)$ or 1 minus the ratio of the sizes of the intersection and union of sets x and y
 - Cosine distance - the cosine distance between two points (two n element vectors) is the angle that the vectors to those points make; in the range 0 to 180 degrees
 - Edit distance - the distance between two strings is the smallest number of insertions and deletions of single characters that will convert one string into the other
 - Hamming distance - between two vectors is the number of components in which they differ (usually used on Boolean vectors)

Copyright 2011-2022 Ellis Horowitz

18

••

- A set is an **unordered collection of objects**, e.g. $\{a, b, c\}$
- Focusing on the notion of **distance** of two sets we define a distance $d(A, B)$ as
 - *small*, if objects in A and B are close;
 - *large*, if objects in A and B are far apart;
 - *0*, if they are the same, and finally
 - $d(A, B)$ is in the range $[0, \infty]$
- Focusing on the notion of **similarity** we define $s(A, B)$ as:
 - *large*, if the objects in A and B are close;
 - *small*, if the objects in A and B are far apart;
 - *1*, if they are the same, and finally
 - $s(A, B)$ is in the range $[0, 1]$
- Often we can convert between the two, as $d(A, B) = 1 - s(A, B)$



••

- Consider $A = \{0, 1, 2, 5, 6\}$ and $B = \{0, 2, 3, 5, 7, 9\}$
- $JS(A, B) = \text{size}(A \text{ intersection } B) / \text{size}(A \cup B)$
 - $= \text{size}(\{0, 2, 5\}) / \text{size}(\{0, 1, 2, 3, 5, 6, 7, 9\})$
 - $= 3 / 8 = 0.375$
- Suppose we divide our items into four clusters, e.g.
 - $C_1 = \{0, 1, 2\}$
 - $C_2 = \{3, 4\}$
 - $C_3 = \{5, 6\}$
 - $C_4 = \{7, 8, 9\}$
- perhaps
 C_1 represents action movies, C_2 comedies,
 C_3 documentaries, C_4 horror movies
- If $A_{\text{clu}} = \{C_1, C_3\}$ and $B_{\text{clu}} = \{C_1, C_2, C_3, C_4\}$, then
- $JS_{\text{clu}}(A, B) = JS(A_{\text{clu}}, B_{\text{clu}}) =$
 - $\text{size}(\{C_1, C_3\} \text{ intersect } \{C_1, C_2, C_3, C_4\}) / (\{C_1, C_3\} \cup \{C_1, C_2, C_3, C_4\})$
 - $= 5 / 10 = 0.5$
- If we are going to use Jaccard similarity to determine when two web pages are near duplicates; we need to say what are the elements of the sets we are comparing

••

- **Definition of Shingle:**

- a contiguous subsequence of words in a document is called a *shingle*;

The 4-shingling of the phrase below produces a bag of 5 items:

“a rose is a rose is a rose” => a set $S(D,w)$ is defined as

{ (a_rose_is_a), (rose_is_a_rose), (is_a_rose_is), (a_rose_is_a),
(rose_is_a_rose) }

- $S(D,w)$ is the set of shingles of a document D of width w

- **Similarity Measures**

- *Jaccard(A,B)* (also known as Resemblance) is defined as

size of ($S(A,w)$ intersect $S(B,w)$) / size of ($S(A,w)$ union $S(B,w)$)

- *Containment(A,B)* is defined as

size of ($S(A,w)$ intersect $S(B,w)$) / size of ($S(A,w)$)

- $0 \leq \text{Resemblance} \leq 1$

- $0 \leq \text{Containment} \leq 1$

- See *On the resemblance and containment of documents*, Conf. on Compression and Complexity, DEC Research Center, 1997

••

- **White space?**

- Should we include spaces and returns? Sometimes it makes sense, e.g.
“plane has touch down” versus “threw a touchdown”
(the space between “touch” and “down” is significant)

- **Capitalization?**

- Sam versus sam. Can help to distinguish proper nouns

- **Punctuation?**

- English is punctuated differently in the US and India; punctuation differs in articles, blogs, and tweets

- **How large should k be?**

- General rule: high enough so the probability of almost all shingles matching is low, so a collision is meaningful;

- **Count replicas?**

- Typically bag of words counts replicas, but shingling does not

- **Stop words?** Typically omitted as they are so common

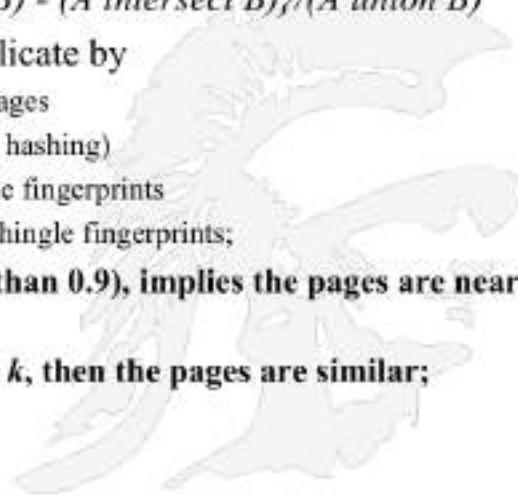


••

- **Original text**
 - “Tropical fish include fish found in tropical environments around the world, including both freshwater and salt water species”
- **All 3-shingles (there are 16 of them)**
 - (Tropical fish include), (fish include fish), (include fish found), (fish found in), (found in tropical), (in tropical environments), (tropical environments around), (environments around the), (around the world), (the world including), (world including both), (including both freshwater), (both freshwater and), (freshwater and salt), (and salt water), (salt water species)
- **Hash values for the 3-shingles (sets of shingles are large, so we hash them to make them more manageable, and we select a subset)**
 - 938, 664, 463, 822, 492, 798, 78, 969, 143, 236, 913, 908, 694, 553, 870, 779
- **Select only those hash values that are divisible by some number, e.g. here are selected hash values using $\theta \bmod 4$**
 - 664, 492, 236, 908; *these are considered the fingerprints*
- **Near duplicates are found by comparing fingerprints and finding pairs with a high overlap**

••

- Recall the Jaccard similarity of sets A and B, $J(A,B)$, is defined as
$$\frac{|A \text{ intersect } B|}{|A \cup B|}$$
- The Jaccard distance of sets A and B, measuring *dissimilarity* is defined as
$$1 - J(A,B), \text{ or equivalently } \{(A \cup B) - (A \cap B)\} / (A \cup B)$$
- We can test if two pages are near duplicate by
 1. First compute the k -shingles of the two pages
 2. Map the k -shingles into numbers (e.g. by hashing)
 3. Select a subset of the shingles to act as the fingerprints
 4. Compute the Jaccard similarity of the k -shingle fingerprints;
- A high Jaccard similarity (e.g. greater than 0.9), implies the pages are near duplicate; or
- if ($J(\text{fingerprint}(A), \text{fingerprint}(B)) > k$, then the pages are similar;



Copyright 2011-2022 Ellis Horowitz

24

FYI if you are interested: in p.475 of our text is a probabilistic approach to this that reduces the # of shingle comparisons we need to make.

SimHash

- There is another way to determine if two web pages are near duplicates
- The method is called SimHash
- It was developed by Moses Charikar and is described in his paper
Similarity Estimation Techniques from Rounding Algorithms, STOC May 2002
 - <https://www.cs.princeton.edu/courses/archive/spring04/cos598B/bib/CharikarEstim.pdf>
- The basic idea is the same as before
 - obtain an f -bit fingerprint for each document
 - A pair of documents are near duplicate if and only if fingerprints are at most k -bits apart
 - But in this case instead of using permutations and probability we use SimHash
- Documents D_1 and D_2 are near duplicates iff
 $\text{Hamming-Distance}(\text{Simhash}(D_1), \text{Simhash}(D_2)) \leq K$
- Typically $f = 64$ and $k = 3$

Copyright 2011-2022 Ellis Horowitz

30

SimHash (aka Charikar Similarity) is essentially a dimension reduction technique - it maps a set of weighted features (contents of a document) to a low dimensional fingerprint, eg. a 64-bit word.

And, **documents that are nearly identical have nearly similar fingerprints that differ only in a small # of bits.** In other words, similar inputs lead to similar outputs (hash values), hence 'Sim'Hash; other hashing techniques, eg. MD5, do not have this property (in other words, even a tiny change in the input leads to a huge change in the output). This similarity property is what makes SimHash, an excellent tool for similarity detection of documents.

Here is the SimHash paper.

••

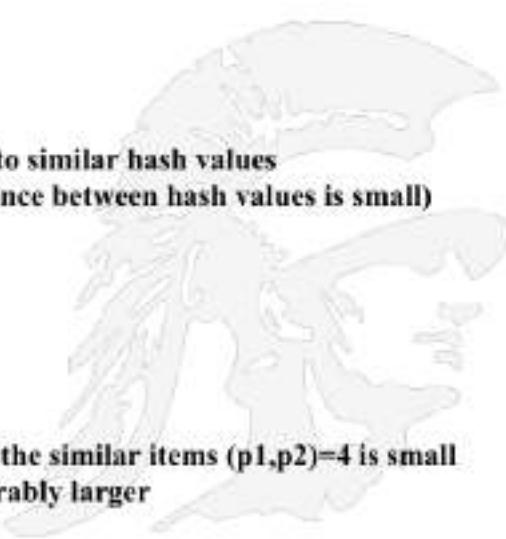
- A hash function usually hashes different values to totally different hash values; here is an example

```
p1 = 'the cat sat on the mat'  
p2 = 'the cat sat on a mat'  
p3 = 'we all scream for ice cream'  
p1.hash => 415542861  
p2.hash => 668720516  
p3.hash => 767429688
```

- Simhash is one where similar items are hashed to similar hash values
(by similar we mean the bitwise Hamming distance between hash values is small)

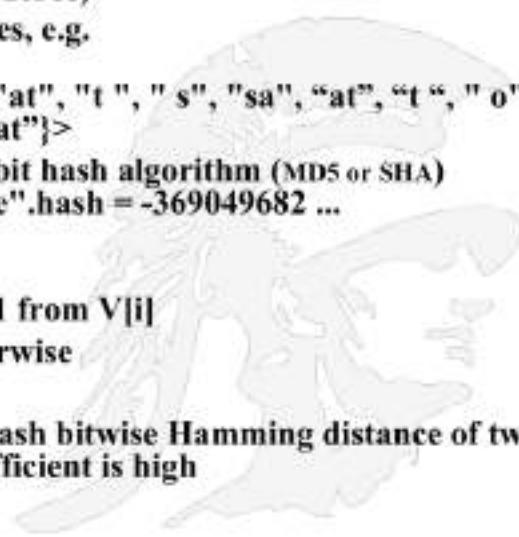
```
p1.simhash => 851459198  
001100101100000000111000111110  
p2.simhash => 847263864  
00110010100000000011100001111000  
p3.simhash => 984968088  
001110101011010110101110011000
```

- in this case we can see the hamming distance of the similar items ($p1, p2$)=4 is small whereas ($p1, p3$)=16 and ($p2, p3$)=12 are considerably larger



••

- The simhash of a phrase is calculated as follows:
 1. pick a hashsize, lets say 32 bits
 2. let $V = [0] * 32$ # (ie a vector of 32 zeros)
 3. break the input phrase up into shingles, e.g.
'the cat sat on a mat'.shingles(2) =>
#<Set: {"th", "he", "e ", " c", "ca", "at", "t ", " s", "sa", "at", "t ", " o",
"on", "n ", " a", "a ", " m", "ma", "at"}>
 4. hash each feature using a normal 32-bit hash algorithm (MD5 or SHA)
"th".hash = -502157718 "he".hash = -369049682 ...
 5. for each hash
 - if bit_i of hash is set then add 1 to V[i]
 - if bit_i of hash is not set then subtract 1 from V[i]
 6. simhash bit_i is 1 if $V[i] > 0$ and 0 otherwise
- Simhash is useful because if the Simhash bitwise Hamming distance of two phrases is low then their Jaccard coefficient is high



••

- In the case that two numbers have a low bitwise Hamming distance and the difference in their bits are in the lower order bits then it turns out that they will end up close to each other if the list is sorted.

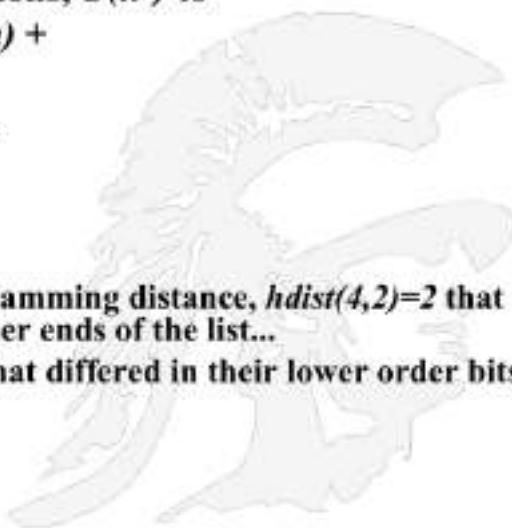
- consider the eight numbers and their bit representations

		if we sort them
• 1	37586 1001001011010010	4 934 0000001110100110
• 2	50086 1100001110100110 7 <--(this column lists hamming	3 2648 0000101001011000 9
• 3	2648 0000101001011000 11 distance to previous entry)	6 2650 0000101001011010 1
• 4	934 0000001110100110 9	1 37586 1001001011010010 5
• 5	40957 100111111111101 9	8 40955 1001111111111011 6
• 6	2650 0000101001011010 9	5 40957 100111111111101 2
• 7	64475 1111101111011011 7	2 50086 1100001110100110 9
• 8	40955 1001111111111011 4	7 64475 1111101111011011 9

notice that two pairs with very smallest hamming distance
 $\text{hdist}(3,6)=1$ and $\text{hdist}(8,5)=2$ have ended up adjacent to each other.

••

- Rather than check every combo we could just check the adjacent pairs of the list, each is a good candidate.
- This reduces the runtime from $n^*(n-1)/2$ coefficient calculations, $O(n^2)$ to
 - n fingerprints calculations $O(n)$ +
 - a sort $O(n \log n)$ +
 - n coefficient calculations $O(n)$,
- which is $O(n \log n)$ overall;
- A problem:
 - there is another pair with a low Hamming distance, $hdist(4,2)=2$ that have ended up totally apart at other ends of the list...
 - sorting only picked up the pairs that differed in their lower order bits.



••

- To get around this consider another convenient property of bitwise Hamming distance, *a permutation of the bits of two numbers preserves Hamming distance*
- If we permute by 'rotating' the bits, i.e. bit shift left and replace lowest order bit with the 'lost' highest order bit we get 'new' fingerprints that have the same Hamming distances

'rotate' bits left twice

4 3736 0000111010011000	3 10592 0010100101100000	6 10600 0010100101101000	1 19274 0100101101001010	8 32750 011111111101110 6	5 32758 0111111111101110 2	2 3739 0000111010011011	7 61295 1110111101101111 9
4 3736 0000111010011000	3 10592 0010100101100000	6 10600 0010100101101000	1 19274 0100101101001010	8 32750 011111111101110 6	5 32758 0111111111101110 2	2 3739 0000111010011011	7 61295 1110111101101111 9

if we sort again by fingerprint

4 3736 0000111010011000	2 3739 0000111010011011 2	3 10592 0010100101100000 11	6 10600 0010100101101000 1	1 19274 0100101101001010 5	8 32750 011111111101110 6	5 32758 0111111111101110 2	7 61295 1110111101101111 6
4 3736 0000111010011000	2 3739 0000111010011011	3 10592 0010100101100000	6 10600 0010100101101000	1 19274 0100101101001010	8 32750 011111111101110	5 32758 0111111111101110	7 61295 1110111101101111

this time the (2,4) pair ended up adjacent
we also identified the (3,6) and (5,8) pairs as candidates again

Copyright 2011-2022 Ellis Horowitz

35

So we can '**rotate, sort, check adjacent**' 'B' times (eg. 64 times; depending on how many bits we have), to discover (almost) all the near-duplicates.

In other words:

- comparing SimHash values (ie computing 'Charikar Similarity' values) is a great way to identify near-duplicates
- for 'n' documents, comparing them all pairwise would take a long time [O(n^2)]
- so as a shortcut, we can sort their decimal representations and only compare adjacents - this will identify similarities based on low-end bits; but this will miss similarities based on the higher-end bits; as an aside, we can look for one more possible low-bits near-duplicate by comparing the top-most and bottom-most values too, like in Gray Code
- so to fix the problem of missing finding high order bit similarities, we can rotate (spin) all the docs' bits identically to the right (so that the high order bits become a 'bit' (lol) lower) to produce 'new' hashes, sort *those*, compare for near-duplicates

- we can progressively spin right by 1 bit, 2 bits, 3 bits... to discover more and more similarities [we will rediscover existing similarities but ignore those]
- note that we can spin left as well
- doing the above is STILL faster than $O(n^2)$:)

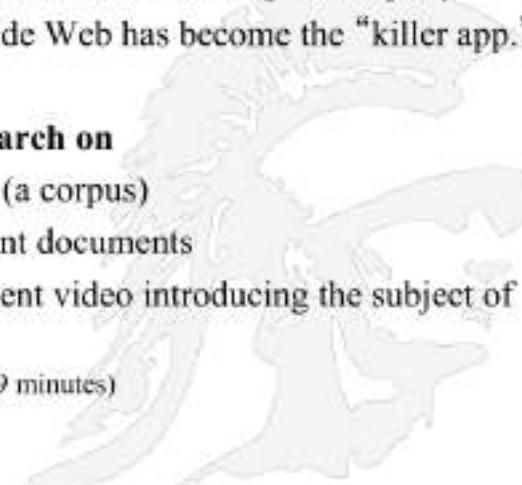
Here is a nice page on SimHash, and [this](#) is a Google paper that discusses using SimHash at scale.

← 1/46 → *** 9:07:24

Intro' to IR

••

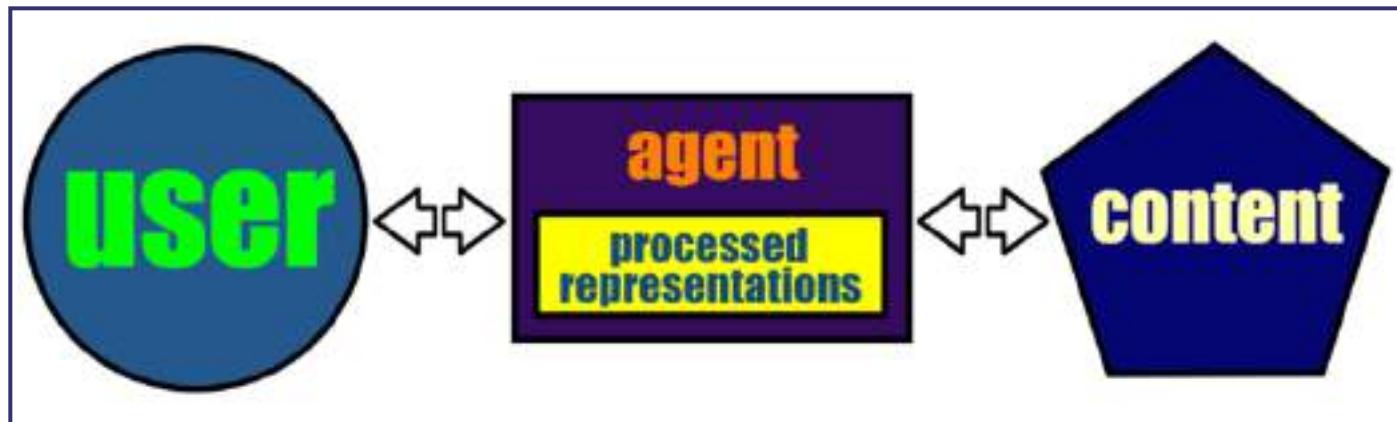
- **Information retrieval (IR) has been a computer science subject for many decades**
 - Traditionally it deals with the *indexing* of a given set of textual documents and the *retrieval* of relevant documents given a query
- Searching for pages on the World Wide Web has become the “killer app.”
- **There has been a great deal of research on**
 - How to index a set of documents (a corpus)
 - How to efficiently retrieve relevant documents
- Jurafsky and Manning have an excellent video introducing the subject of Information Retrieval;
- http://csci572.com/movies/01_IntroIR.mp4 (9 minutes)
then jump to slide 16



Copyright Ellis Horowitz, 2011-2022

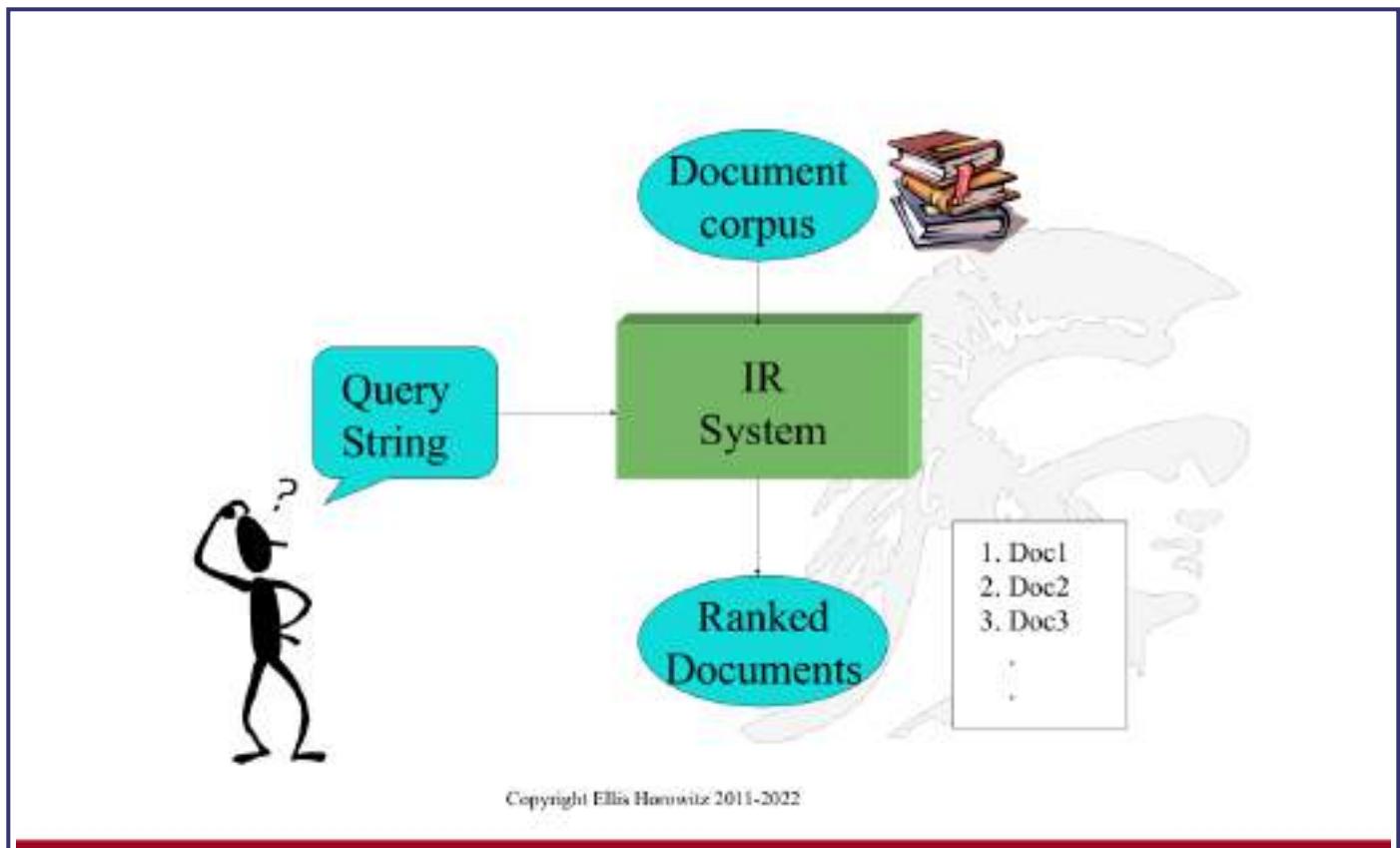
2

IR involves this (which we saw before):



This is a quick intro' to the 'search problem'...

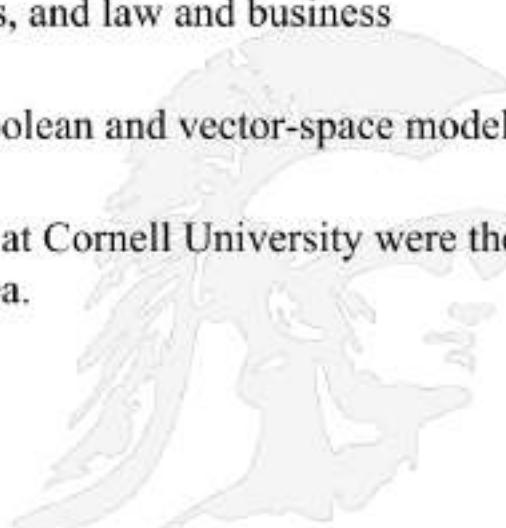
••



••

- **1960-70 s:**

- Initial exploration of text retrieval systems for “small” corpora of scientific abstracts, and law and business documents.
 - Development of the basic Boolean and vector-space models of retrieval.
 - Prof. Salton and his students at Cornell University were the leading researchers in the area.



••

- **1980's:**

- **Creation of large document database systems, many run by companies:**

- Lexis-Nexis, <http://www.lexisnexis.com/>
 - information to legal, corporate, government and academic markets, and publishes legal, tax and regulatory information
 - Dialog, <http://www.dialog.com/>
 - data from more than 1.4 billion unique records of key information.
 - MEDLINE, <http://www.medlineplus.gov/>
 - National Library of Medicine health information



Copyright Ellis Horowitz, 2011-2022

5

••

- **1990's:**
 - **Searching FTP'able documents on the Internet**
 - Archie
 - WAIS
 - **After the World Wide Web is invented, search engines appear**
 - Lycos
 - Yahoo
 - Altavista



Copyright Ellis Horowitz, 2011-2022

6

••

- 1990's continued:
 - Organized Competitions
 - NIST TREC (Text REtrieval Conferences, <http://trec.nist.gov/>)
 - Sponsored by National Institute of Standards and Technology, NIST
 - Several New Types of IR Systems are Developed
 1. *Recommender Systems*: computer programs which attempt to predict items (movies, music, books, news, web pages) that a user may be interested in, given some information about the user's profile.
 - Often implemented as a collaborative filtering algorithm, examples include:
 - » YouTube, perhaps the largest scale such system in existence
 - » <https://static.googleusercontent.com/media/research.google.com/en/pubs/archive/45530.pdf>
 - » Amazon's recommendation system, see <https://stackoverflow.com/questions/2323768/how-does-the-amazon-recommendation-feature-work>
 - » <http://rejoiner.com/resources/amazon-recommendations-secret-selling-online/>
 2. *Automated Text Categorization & Clustering Systems*
 - Useful for grouping news articles

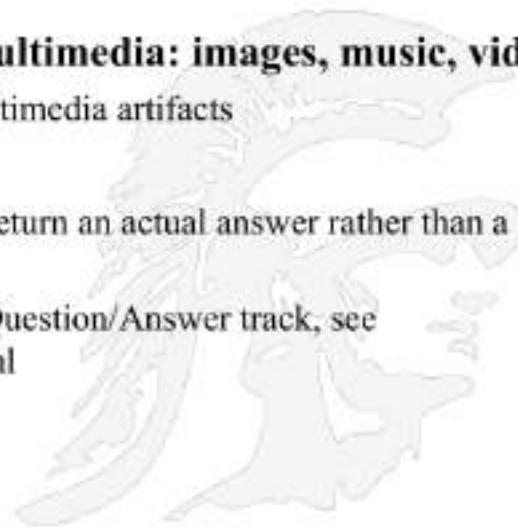
Copyright Ellis Horowitz, 2011-2022

7

TREC: <https://trec.nist.gov/> [an ongoing competition, eg. <https://fair-trec.github.io/how-to-trec>]...

••

- **2000 s**
 - **Link analysis for Web Search**
 - Google started this
 - **Extension to retrieval of multimedia: images, music, video**
 - It is much harder to index multimedia artifacts
 - **Question Answering**
 - Question answering systems return an actual answer rather than a ranked list of documents
 - Since 1999 TREC has had a Question/Answer track, see <http://trec.nist.gov/data/qa.html>



Copyright Ellis Horowitz, 2011-2022

8

••

- **Database Management**
- **Library and Information Science**
- **Artificial Intelligence**
- **Natural Language Processing**
- **Machine Learning**
- **Data Science**



Copyright Ellis Horowitz, 2011-2022

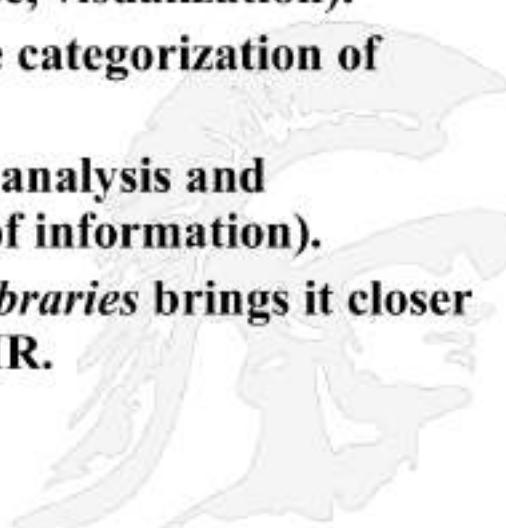
9

••

- Focused on *structured* data stored in relational tables rather than free-form text
- Focused on efficient processing of well-defined queries in a formal language (SQL)
- Clearer semantics for both data and queries
- Web pages are mostly unstructured, though the Document Object Model (DOM) can provide some clues

••

- **Focused on the human user aspects of information retrieval (human-computer interaction, user interface, visualization).**
- **Concerned with effective categorization of human knowledge.**
- **Concerned with citation analysis and *bibliometrics* (structure of information).**
- **Recent work on *digital libraries* brings it closer to Computer Science & IR.**



Copyright Ellis Horowitz, 2011-2022

11

••

- **Focused on the representation of knowledge, reasoning, and intelligent action.**
- **Formalisms for representing knowledge and queries:**
 - ***First-order Predicate Logic*** – a formal system that uses quantified variables over a specified domain of discourse
 - ***Bayesian Networks*** – a directed acyclic graph model that represents a set of random variables and their dependencies
 - E.g. A Bayesian Network that represents the probabilistic relationships between diseases and symptoms
- **Recent work on web ontologies and intelligent information agents brings it closer to IR**
 - Web Ontology Language OWL is a family of knowledge representation languages for authoring ontologies
 - See <https://www.w3.org/OWL/>

- Focused on the syntactic, semantic, and pragmatic analysis of natural language text and discourse.
- Ability to analyze syntax (phrase structure) and semantics allows retrieval based on *meaning* rather than keywords
- NLP now uses vast amounts of web pages as data to run statistical and machine learning models to infer meaning



Natural Language Understanding
Extractive Summarisation



Natural Language Processing
Entity Recognition



Natural Language Generation
Abstractive Text Summarization



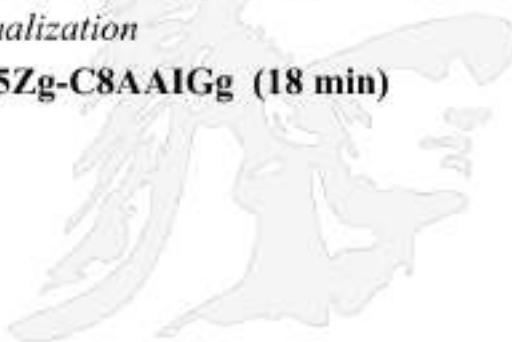
••

- A branch of Artificial Intelligence concerned with algorithms that allow computers to evolve their behavior based on empirical data
- Focused on the development of computational systems that improve their performance with experience
- Two major subtypes of machine learning are:
 - Automated classification of examples based on learning concepts from labeled training examples (*supervised learning*).
 - Automated methods for clustering unlabeled examples into meaningful groups (*unsupervised learning*)
- Machine learning is distinct from ***data mining***, which focuses on the discovery of previously unknown properties of the given data
 - Data mining is akin to query analysis and ranking



••

- “*Data science* is an inter-disciplinary field that uses scientific methods, processes, algorithms and systems to extract knowledge and insights from many structural and unstructured data. Data science is related to data mining, machine learning and big data.” wikipedia
- A *data scientist* is someone who knows how to extract meaning from and interpret data, which requires both tools and methods from **statistics** and **machine learning**, as well as human review. They spend a lot of time in the process of collecting and cleaning data, because data is never clean
- For fun watch *The Beauty of Data Visualization*
- <https://www.youtube.com/watch?v=5Zg-C8AAIGg> (18 min)

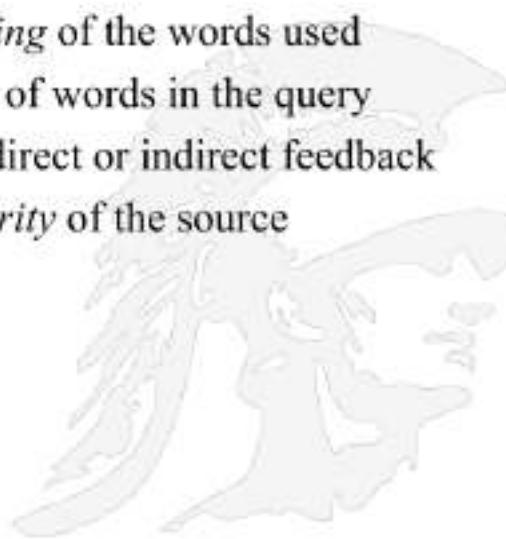


••

- **Simplest notion of relevance is that the query string appears verbatim in the document.**
 - Slightly less strict notion is that the words in the query appear frequently in the document, in any order (this is like viewing the document as a *bag of words*).
- **But that may not retrieve relevant documents that include synonymous terms.**
 - “restaurant” vs. “café”
 - “PRC” vs. “China”
- **And it may retrieve irrelevant documents that include ambiguous terms.**
 - “bat” (baseball vs. mammal)
 - “Apple” (company vs. fruit)
 - “bit” (unit of data vs. act of eating)

••

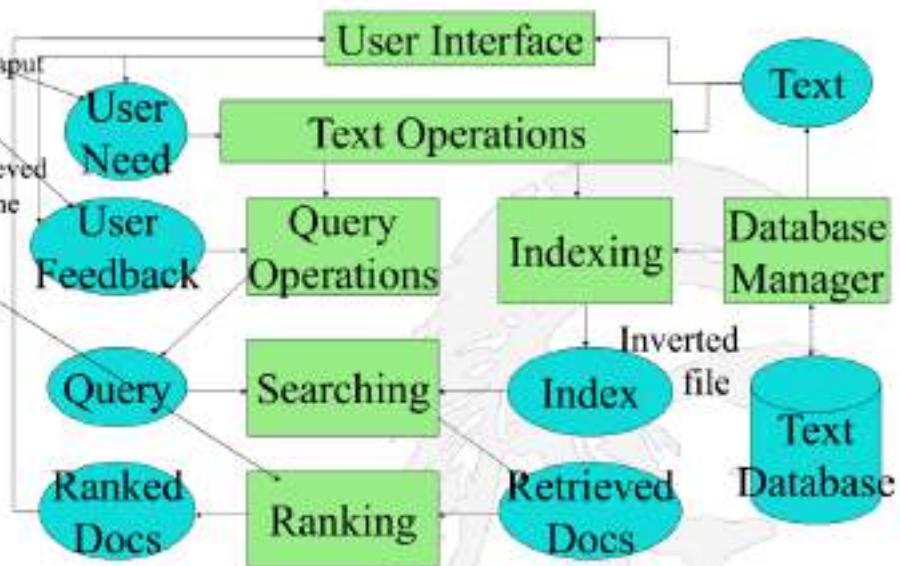
- **Goes beyond using just keyword matching, instead it**
 - Takes into account the *meaning* of the words used
 - Takes into account the *order* of words in the query
 - Adapts to the user based on direct or indirect feedback
 - Takes into account the *authority* of the source



••

Logical View

- User needs are part of the input
- User feedback is provided
- Queries initiate a search of the index and docs are retrieved
- A ranking function orders the results

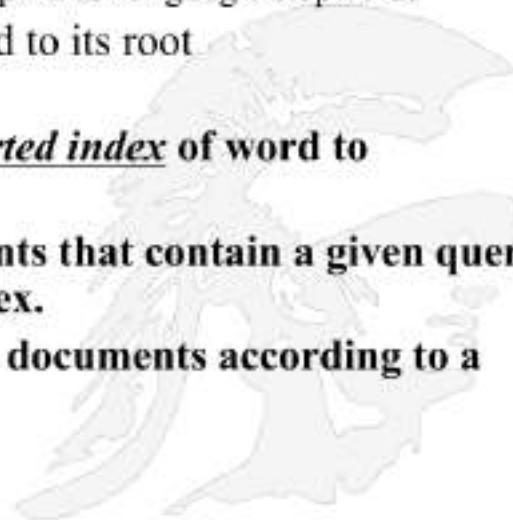


Start with a text database; it is indexed; a user interface permits query operations which cause a search on the Index; matched documents are retrieved and ranked

Copyright Ellis Horowitz 2011-2022

••

- **Parsing forms index words (tokens) and includes:**
 - *Stopword* removal
 - See <http://www.ranks.nl/stopwords> for google stopwords
 - *Stemming*: reducing a word to its root
 - More about this later
- ***Indexing* constructs an inverted index of word to document pointers.**
- ***Searching* retrieves documents that contain a given query token from the inverted index.**
- ***Ranking* scores all retrieved documents according to a relevance metric.**

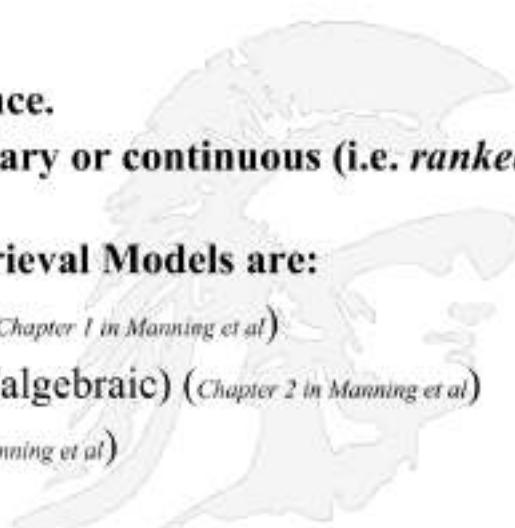


Copyright Ellis Horowitz, 2011-2022

19

••

- **A retrieval model specifies the details of:**
 - Document representation
 - Query representation
 - Retrieval function
- **Determines a notion of relevance.**
- **Notion of relevance can be binary or continuous (i.e. *ranked retrieval*)**
- **Three major Information Retrieval Models are:**
 1. Boolean models (set theoretic) (*Chapter 1 in Manning et al*)
 2. Vector space models (statistical/algebraic) (*Chapter 2 in Manning et al*)
 3. Probabilistic models (*Chapter 11 in Manning et al*)



••

1. Strip unwanted characters/markup (e.g. HTML tags, punctuation, page numbers, etc.).
2. Break into tokens (keywords) separating out whitespace.
3. Stem tokens to “root” words



4. Remove common stopwords (e.g. a, the, it, etc.).
5. Detect common phrases (possibly using a domain specific dictionary).
6. Build inverted index (keyword → list of docs containing it).



Copyright Ellis Horowitz, 2011-2022

22

•

- A document is represented as a **set** of keywords.
- Queries are Boolean expressions of keywords, connected by AND, OR, and NOT, including the use of brackets to indicate scope
- Here is a sample Boolean query with explicit AND, OR, NOT operators
 - [[Rio & Brazil] | [Hilo & Hawaii]] & hotel & !Hilton]

Google Advanced Search:

Note inclusion of AND, OR, NOT operators

The screenshot shows the Google Advanced Search interface. On the left, there is a sidebar with the text "Google Advanced Search:" and "Note inclusion of AND, OR, NOT operators". The main area has several search fields:

- "Find pages with..." dropdown: "all these words":
- "To do this in the search box": "Type the keyword 'book' or 'book book'."
- "Put exact words or phrase":
- "Put exact words or phrase like 'book, book book'."
- "any of these words":
- "Type in between all the words you want to see every site understand."
- "none of these words":
- "Put a minus sign just before words that you don't want to see, like '-book, book'."
- "Numbers ranging from": to
- "Put the full range between the numbers and add a unit of measurement: '10..50, 100..1000, 1000..10000'."

Copyright Ellis Horowitz, 2011-2022

23

••

- **Popular retrieval model because:**
 - Easy to understand for simple queries.
 - Clean formalism.
- **Boolean models can be extended to include ranking**
- **Reasonably efficient implementations possible for normal queries.**

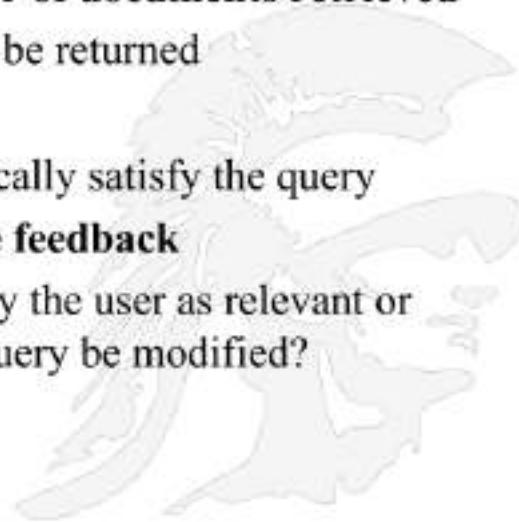


Copyright Ellis Horowitz, 2011-2022

24

••

- **Very rigid: AND means all; OR means any**
- **Difficult to express complex user requests**
- **Difficult to control the number of documents retrieved**
 - *All* matched documents will be returned
- **Difficult to rank output**
 - *All* matched documents logically satisfy the query
- **Difficult to perform relevance feedback**
 - If a document is identified by the user as relevant or irrelevant, how should the query be modified?



••

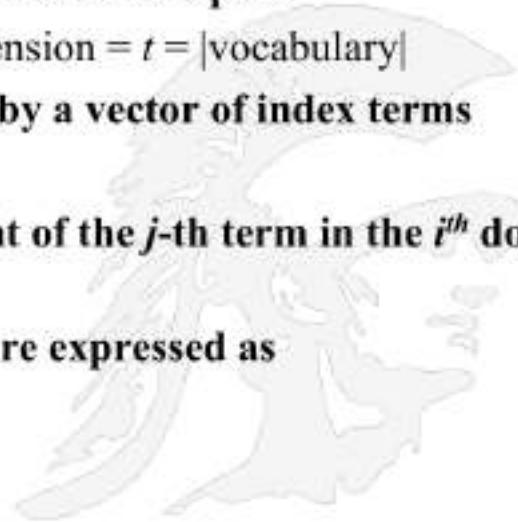
- **The simple query “Lincoln”**
 - Too many matches including Lincoln cars and places named Lincoln as well as Abraham Lincoln
- **More detailed query “President AND Lincoln”**
 - Returns documents that discuss the President of Ford Motor company that makes the Lincoln car
- **Even more detailed query “president AND Lincoln AND NOT (automobile OR car)”**
 - Better, but the use of NOT will remove a document about President Lincoln that says “Lincoln’s body departs Washington in a nine car funeral train”
- **Perhaps try**
 - President AND lincoln AND biography AND life AND birthplace AND gettysburg AND NOT (automobile OR car), but too many ANDs can lead to nothing, so
 - President AND lincoln AND (biography OR life OR birthplace OR gettysburg) AND NOT (automobile OR car)

Copyright Ellis Horowitz, 2011-2022

26

••

- Assume t distinct terms remain after preprocessing; call them index terms or the vocabulary
- These “orthogonal” terms form a vector space
size of the vocabulary = Dimension = t = |vocabulary|
- A document D_i is represented by a vector of index terms
$$D_i = (d_{i1}, d_{i2}, \dots, d_{it})$$
- Where d_{ij} represents the weight of the j -th term in the i^{th} doc
 - but how is the weight computed?
- Both documents and queries are expressed as t -dimensional vectors



••

Example:

$$D_1 = 2T_1 + 3T_2 + 5T_3$$

$$D_2 = 3T_1 + 7T_2 + T_3$$

$$Q = 0T_1 + 0T_2 + 2T_3$$

$$D_1 = 2T_1 + 3T_2 + 5T_3$$

Vocabulary consists of 3 terms
with weights the coefficients
There are two documents, D_1 and
 D_2 ; there is one query, Q

$$D_2 = 3T_1 + 7T_2 + T_3$$

$$7$$

$$Q = 0T_1 + 0T_2 + 2T_3$$

- Is D_1 or D_2 more similar to Q ?
- How to measure the degree of similarity? Distance? Angle? Projection?

••

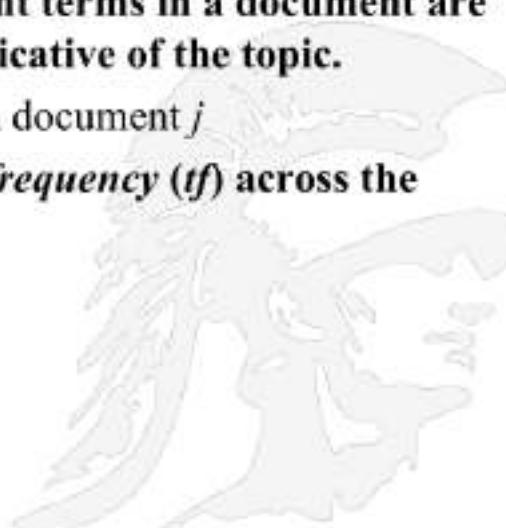
- A collection of n documents can be represented in the vector space model by a term-document matrix.
- An entry in the matrix corresponds to the “weight” of a term in the document; zero means the term has no significance in the document or it simply doesn’t exist in the document; but we still need a way to compute the weight

	T ₁	T ₂	T _t
D ₁	w ₁₁	w ₂₁	...	w _{t1}
D ₂	w ₁₂	w ₂₂	...	w _{t2}
:	:	:		:
D _n	w _{1n}	w _{2n}	...	w _{tn}

••

- One way to compute the weight is to use the term's frequency in the document
- Assumption: the more frequent terms in a document are more important, i.e. more indicative of the topic.
 f_{ij} = frequency of term i in document j
- May want to normalize *term frequency (tf)* across the entire corpus:

$$tf_{ij} = f_{ij} / \max\{f_{ij}\}$$



••

- Terms that appear in many *different* documents are *less* indicative of overall topic

df_i = document frequency of term i

= number of documents containing term i

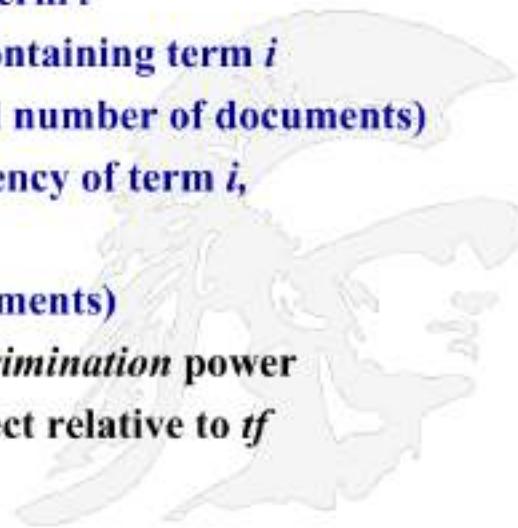
of course df_i is always $\leq N$ (total number of documents)

idf_i = inverse document frequency of term i ,

= $\log_2 (N / df_i)$

(N : total number of documents)

- An indication of a term's *discrimination* power
- Log is used to dampen the effect relative to tf



••

<i>term</i>	df_i	idf_i
Calpurnia	1	$\log(1,000,000/1)=6$
animal	100	4
Sunday	1,000	3
fly	10,000	2
under	100,000	1
the	1,000,000	0



- $idf_i = \log_{10}(N/df_i)$, $N = 1,000,000$
- there is one idf value for each term t in a collection

••

- A typical combined term importance indicator is ***tf-idf weighting*** (*note: it is often written with a hyphen, but the hyphen is NOT a minus sign; some people replace the hyphen with a dot*):

$$w_{ij} = \text{tf}_{ij} \cdot \text{idf}_i = (1 + \log \text{tf}_{ij}) * \log_2 (N / \text{df}_i)$$

- A term occurring frequently in the document but rarely in the rest of the collection is given high weight.
- Many other ways of determining term weights have been proposed.
- Experimentally, *tf.idf* has been found to work well
- Given a query q , then we score the query against a document d using the formula
- $\text{Score}(q, d) = \sum (\text{tf.idf}_{t,d})$ where t is in $q \cap d$

••

Given a document containing 3 terms with given frequencies:

A(3), B(2), C(1)

Assume collection contains 10,000 documents and document frequencies of these 3 terms are:

A(50), B(1300), C(250)

Then:

A: tf = 3/3; idf = log(10000/50) = 5.3; tf.idf = 5.3

B: tf = 2/3; idf = log(10000/1300) = 2.0; tf.idf = 1.3

C: tf = 1/3; idf = log(10000/250) = 3.7; tf.idf = 1.2

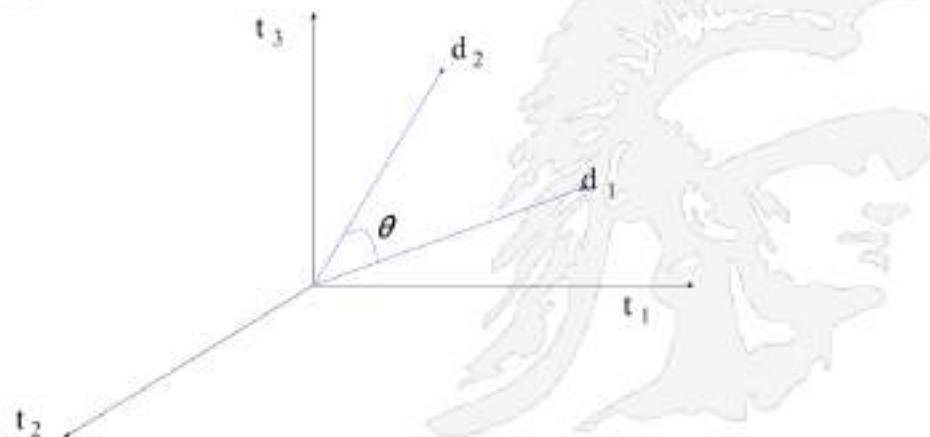
Copyright Ellis Horowitz, 2011-2022

34

Note - in some cases we normalize each tf.idf value using the L2 (sqrt of sum of squares), as opposed to L1 ((absolute)sum) norm.

••

- Distance between vectors d_1 and d_2 is *captured* by the cosine of the angle x between them.
- Note – this is a *similarity* measure, not a distance measure



Copyright Ellis Horowitz, 2011-2022

35

••

- A **similarity measure** is a function that computes the *degree of similarity* between two vectors
 - Look back at the previous lecture slides for the definition of similarity
- **Using a similarity measure between the query and each document has positive aspects:**
 - It is possible to rank the retrieved documents in the order of presumed relevance.
 - It is possible to enforce a certain threshold so that the size of the retrieved set can be controlled.

••

- A vector can be normalized (given a length of 1) by dividing each of its components by the vector's length
- This maps vectors onto the unit circle:
- Then, $|\vec{d}_j| = \sqrt{\sum_{i=1}^n w_{i,j}^2} = 1$
- Longer documents don't get more weight
- For normalized vectors, the cosine is simply the dot product:

$$\cos(\vec{d}_j, \vec{d}_k) = \vec{d}_j \cdot \vec{d}_k$$

Copyright Ellis Horowitz, 2011-2022

37

••

- Similarity between vectors for the document d_j and query q can be computed as the vector inner product:

$$\text{sim}(d_j, q) = d_j \cdot q = \sum_{i=1}^t w_{ij} \cdot w_{iq}$$

where w_{ij} is the weight of term i in document j and w_{iq} is the weight of term i in the query

- For binary vectors, the inner product is the number of matched query terms in the document (size of intersection) (Hamming distance)
- For weighted term vectors, it is the sum of the products of the weights of the matched terms.

••

Binary:

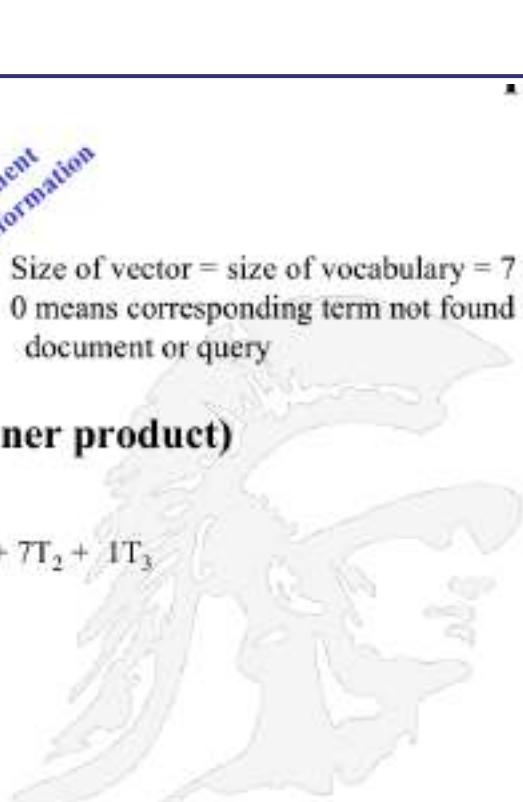
- $D = [1, 1, 1, 0, 1, 1, 0]$ Size of vector = size of vocabulary = 7
- $Q = [1, 0, 1, 0, 0, 1, 1]$ 0 means corresponding term not found in document or query

$$\text{similarity}(D, Q) = 3 \text{ (the inner product)}$$

Weighted:

$$D_1 = 2T_1 + 3T_2 + 5T_3 \quad D_2 = 3T_1 + 7T_2 + 1T_3$$
$$Q = 0T_1 + 0T_2 + 2T_3$$

$$\begin{aligned} \text{sim}(D_1, Q) &= 2*0 + 3*0 + 5*2 = 10 \\ \text{sim}(D_2, Q) &= 3*0 + 7*0 + 1*2 = 2 \end{aligned}$$



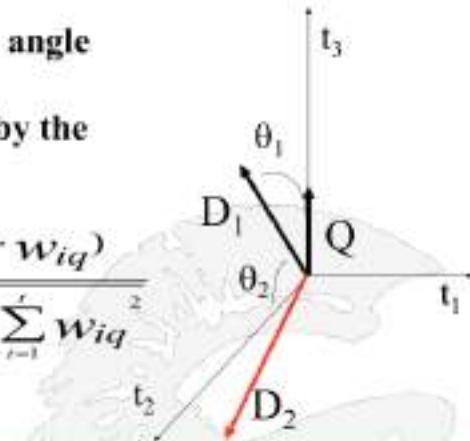
Copyright Ellis Horowitz, 2011-2022

40

••

- Cosine similarity measures the cosine of the angle between two vectors
- We compute the inner product normalized by the vector lengths

$$\text{CosSim}(d_j, q) = \frac{\vec{d}_j \cdot \vec{q}}{|\vec{d}_j| \cdot |\vec{q}|} = \frac{\sum_{i=1}^r (w_{ij} \cdot w_{iq})}{\sqrt{\sum_{i=1}^r w_{ij}^2} \cdot \sqrt{\sum_{i=1}^r w_{iq}^2}}$$



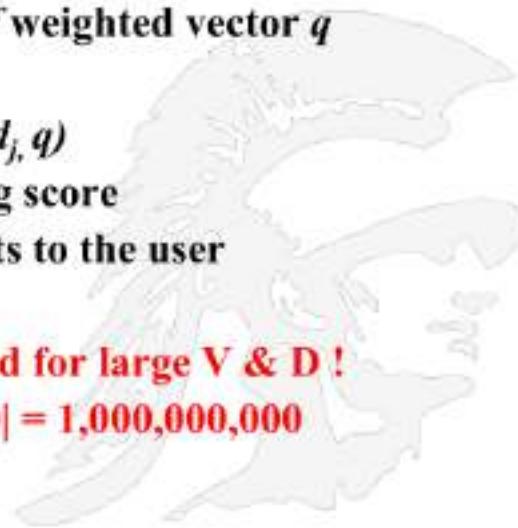
$D_1 = 2T_1 + 3T_2 + 5T_3 \quad \text{CosSim}(D_1, Q) = 10 / \sqrt{(4+9+25)(0+0+4)} = 0.81$
 $D_2 = 3T_1 + 7T_2 + 1T_3 \quad \text{CosSim}(D_2, Q) = 2 / \sqrt{(9+49+1)(0+0+4)} = 0.13$
 $Q = 0T_1 + 0T_2 + 2T_3$

D_1 is 6 times better than D_2 using cosine similarity but only 5 times better using inner product.

••

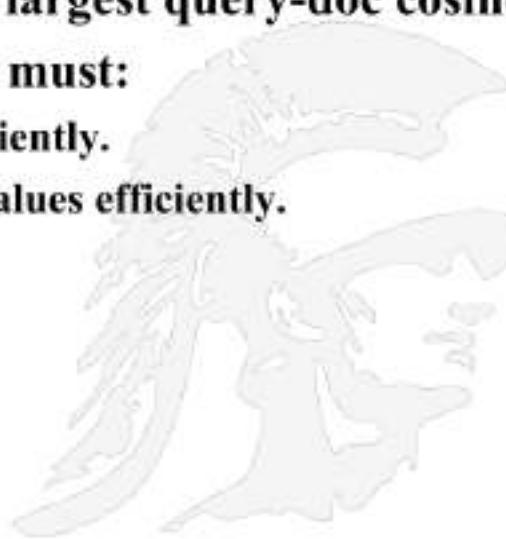
1. Convert all documents in collection D to $tf.idf$ weighted vectors, the j^{th} document denoted by d_j , for keywords in vocabulary V
2. Convert each query to a $tf.idf$ weighted vector q
3. For each d_j in D do
 - Compute score $s_j = \text{cosSim}(d_j, q)$
4. Sort documents by decreasing score
5. Present top ranked documents to the user

Time complexity: $O(|V| \cdot |D|)$ Bad for large V & D !
 $|V| = 10,000; |D| = 100,000; |V| \cdot |D| = 1,000,000,000$



••

- **Ranking consists of computing the k docs in the corpus “nearest” to the query $\Rightarrow k$ largest query-doc cosines.**
- **To do efficient ranking one must:**
 - Compute a single cosine efficiently.
 - Choose the k largest cosine values efficiently.



Copyright Ellis Horowitz, 2011-2022

43

••

cosineScore(q)

1. **float Scores[N] = 0;** //Scores array for all documents
2. **float Length[N]** //lengths of all documents
3. **for each query term t**
4. **do calculate $w_{t,q}$ and fetch postings list for t**
5. **for each pair $(d, tf_{t,d})$ in postings list**
6. **do $Scores[d] += w_{t,d} \times w_{t,q}$**
7. **Read the array Length**
8. **for each d do**
9. **$Scores[d] = Scores[d]/Length[d]$**
10. **return Top K components of Scores[]**

weight of query term is l ;
then,
for each document in the
postings list, the term t
occurs tf times;
then we take the dot product
of weight of term t in document
times weight of term t in query;

divide scores by length of each
document
ranking

in practice we only work with a subset of the documents

Copyright Ellis Horowitz, 2011-2022

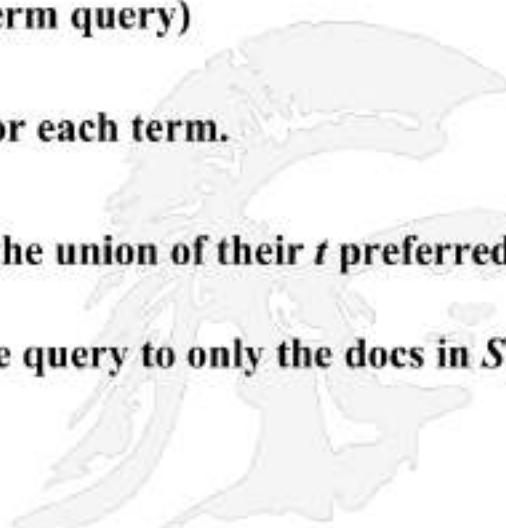
44

••

- Represent the query as a weighted $tf.idf$ vector
- represent each document as a weighted $tf.idf$ vector
- compute the cosine similarity score for the query vector and each document vector that contains the query term
- Rank documents with respect to the query by score
- Return the top k (e.g. $k=10$) to the user

••

- **Preprocess:** Pre-compute, for each term, its k nearest docs.
 - (Treat each term as a 1-term query)
 - lots of preprocessing.
 - Result: “preferred list” for each term.
- **Search:**
 - For a t -term query, take the union of their t preferred lists – call this set S .
 - Compute cosines from the query to only the docs in S , and choose top k .



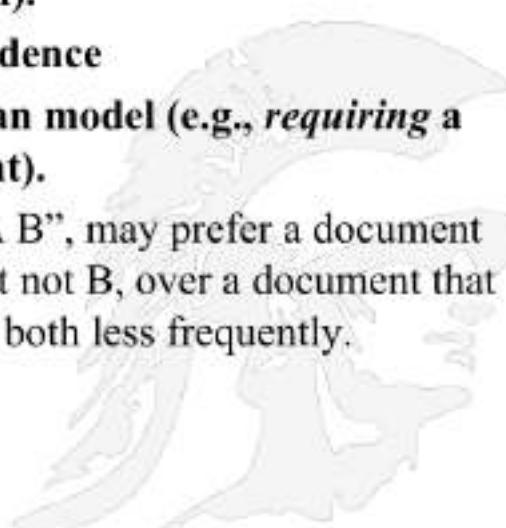
••

- **Simple, mathematically based approach.**
- **Considers both local (*tf*) and global (*idf*) word occurrence frequencies.**
- **Provides partial matching and ranked results.**
- **Tends to work quite well in practice despite obvious weaknesses.**
- **Allows efficient implementation for large document collections.**



••

- **Missing semantic information (e.g. word sense).**
- **Missing syntactic information (e.g. phrase structure, word order, proximity information).**
- **Assumption of term independence**
- **Lacks the control of a Boolean model (e.g., *requiring a term to appear in a document*).**
 - Given a two-term query “A B”, may prefer a document containing A frequently but not B, over a document that contains both A and B, but both less frequently.



1/40

9:07:42

Text processing

••



Standing Queries

- **The path from IR to text classification:**
 - You have an information need to monitor, say:
 - Unrest in the Niger delta region
 - You want to rerun an appropriate query periodically to find new news items on this topic
 - You will be sent new documents that are found
 - I.e., it's not ranking but classification (relevant vs. not relevant)
- **Such queries are called standing queries**
 - Long used by “information professionals”
 - A modern mass instantiation is Google Alerts
- **Standing queries are (hand-written) text classifiers**

Copyright Ellis Horowitz, 2011-2015.

2

••

From: Google Alerts
Subject: Google Alert - stanford -neuro-linguistic nlp OR "Natural Language Processing" OR parser OR tagger OR ner OR "named entity" OR segmenter OR classifier OR dependencies OR "core nlp" OR corenlp OR phrasal
Date: May 7, 2012 8:54:53 PM PDT
To: Christopher Manning

Web

3 new results for stanford -neuro-linguistic nlp OR "Natural Language Processing" OR parser OR tagger OR ner OR "named entity" OR segmenter OR classifier OR dependencies OR "core nlp" OR corenlp OR phrasal

[Twitter / Stanford NLP Group: @Robertoross If you only n ...](#)

@Robertoross If you only need tokenization, java -mx2m edu.stanford.nlp.process.PTBTokenizer file.txt runs in 2MB on a whole file for me.... 9:41 PM Apr 28th ...
twitter.com/stanfordnlp/status/196459102770171905

[\[Java\] LexicalizedParser lp = LexicalizedParser.loadModel\("edu ...](#)
loadModel("edu/stanford/nlp/models/lexparser/englishPCFG.ser.gz"); String[] sent = { "This", "is", "an", "easy", "sentence", "" }; Tree parse = lp.apply(Arrays.
pastebin.com/az14R9nd

[More Problems with Statistical NLP || kuro5hin.org](#)

Tags: nlp, ai, coursera, stanford, nlp-class, cky, ntk, reinventing the wheel ... Programming Assignment 6 for Stanford's nlp-class is to implement a CKY parser .
www.kuro5hin.org/story/2012/5/5/11011/68221

Tip: Use quotes ("like this") around a set of words in your query to match them exactly. [Learn more](#).

[Delete](#) this alert.
[Create](#) another alert.
[Manage](#) your alerts.

••

 **Spam filtering**
Another text classification task

From: "" <takworlld@hotmail.com>
Subject: real estate is the only way... gem_oalvgkay

Anyone can buy real estate with no money down

Stop paying rent TODAY !

There is no need to spend hundreds or even thousands for similar courses

I am 22 years old and I have already purchased 6 properties using the methods outlined in this truly INCREDIBLE ebook.

Change your life NOW !

Click Below to order:
[**http://www.wholesaledaily.com/sales/nmd.htm**](http://www.wholesaledaily.com/sales/nmd.htm)

Copyright Ellis Horowitz, 2011-2015.

4

••



USC **Viterbi**
School of Engineering

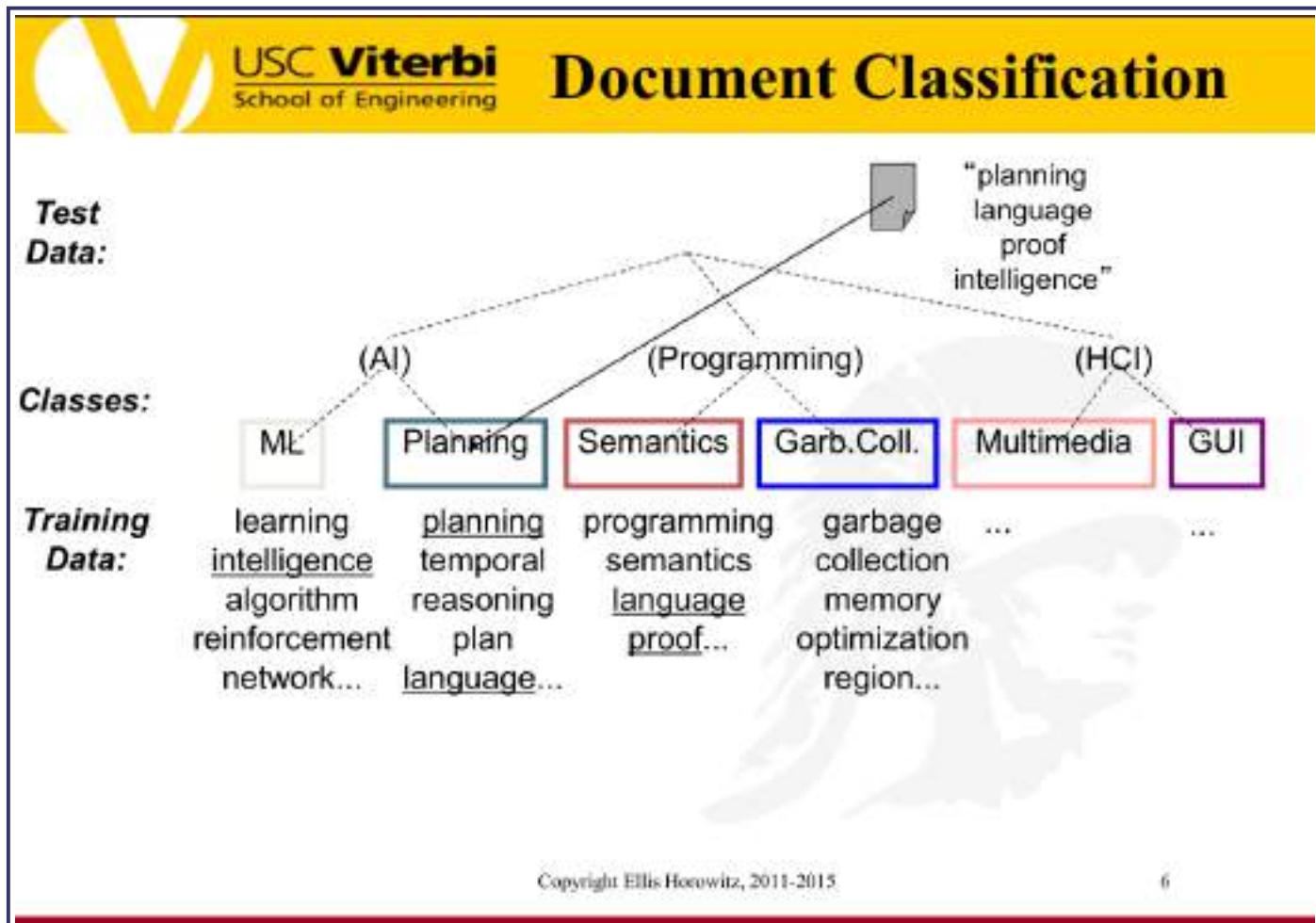
Categorization/Classification

- Given:
 - A representation of a document d
 - Issue: how to represent text documents.
 - Usually some type of high-dimensional space – bag of words
 - A fixed set of classes:
$$C = \{c_1, c_2, \dots, c_B\}$$
- Determine:
 - The category of d by generating a classification function, say $\gamma(d)$
 - We want to build classification functions (“classifiers”).

Copyright Ellis Horowitz, 2011-2015.

5

••



••



USC **Viterbi**
School of Engineering

Classification Methods (1)

- **Manual classification**
 - Used by the original Yahoo! Directory
 - Looksmart, about.com, ODP, PubMed
 - Accurate when job is done by experts
 - Consistent when the problem size and team is small
 - Difficult and expensive to scale
 - Means we need automatic classification methods for big problems

Copyright Ellis Horowitz, 2011-2015.

7

••



USC **Viterbi**
School of Engineering

Classification Methods (2)

- **Hand-coded rule-based classifiers**
 - One technique used by news agencies, intelligence agencies, etc.
 - Widely deployed in government and enterprises
 - Vendors provide “IDE” for writing such rules

Copyright Ellis Horowitz, 2011-2015.

8

••



USC **Viterbi**
School of Engineering

Classification Methods (2)

- **Hand-coded rule-based classifiers**
 - Commercial systems have complex query languages
 - Accuracy is can be high if a rule has been carefully refined over time by a subject expert
 - Building and maintaining these rules is expensive

Copyright Ellis Horowitz, 2011-2015.

9

••



Classification Methods (3): Supervised learning

- **Given:**
 - A document d
 - A fixed set of classes:
 $C = \{c_1, c_2, \dots, c_B\}$
 - A training set D of documents each with a label in C
- **Determine:**
 - A learning method or algorithm which will enable us to learn a classifier γ
 - For a test document d , we assign it the class
 $\gamma(d) \in C$

Copyright Ellis Horowitz, 2011-2015.

10

••



Classification Methods (3)

- **Supervised learning**
 - Naive Bayes (simple, common)
 - k-Nearest Neighbors (simple, powerful)
 - Support-vector machines (newer, generally more powerful)
 - ... plus many other methods
 - No free lunch: requires hand-classified training data
 - But data can be built up (and refined) by amateurs
- **Many commercial systems use a mixture of methods**

Copyright Ellis Horowitz, 2011-2015.

11

••

USC Viterbi
School of Engineering

The bag of words representation

I love this movie! It's sweet, but with satirical humor. The dialogue is great and the adventure scenes are fun... It manages to be whimsical and romantic while laughing at the conventions of the fairy tale genre. I would recommend it to just about anyone. I've seen it several times, and I'm always happy to see it again whenever I have a friend who hasn't seen it yet.

Copyright Ellis Horowitz, 2011-2015.

12

..

USC **Viterbi**
School of Engineering

The bag of words representation

great	2
love	2
recommend	1
laugh	1
happy	1
...	...

Y()=C

Copyright Ellis Horowitz, 2011-2015.

13

••



USC **Viterbi**
School of Engineering

Features

- **Supervised learning classifiers can use any sort of feature**
 - URL, email address, punctuation, capitalization, dictionaries, network features
- **In the simplest bag of words view of documents**
 - We use only word features
 - we use all of the words in the text (not a subset)

Copyright Ellis Horowitz, 2011-2015.

14

••



USC **Viterbi**
School of Engineering

Feature Selection: Why?

- **Text collections have a large number of features**
 - 10,000 – 1,000,000 unique words ... and more
- **Selection may make a particular classifier feasible**
 - Some classifiers can't deal with 1,000,000 features
- **Reduces training time**
 - Training time for some methods is quadratic or worse in the number of features
- **Makes runtime models smaller and faster**
- **Can improve generalization (performance)**
 - Eliminates noise features
 - Avoids overfitting

Copyright Ellis Horowitz, 2011-2015.

15

..



- The simplest feature selection method:
 - Just use the most common terms
 - No particular foundation
 - But it make sense why this works
 - They are the words that can be well-estimated and are most often available as evidence
 - In practice, this is often 90% as good as better methods
 - Smarter feature selection – future lecture

..



The slide features the USC Viterbi School of Engineering logo on the left, which includes a stylized 'V' icon and the text 'USC Viterbi School of Engineering'. On the right, the word 'SpamAssassin' is written in a large, bold, black font. In the background, there is a faint watermark of a person's face.

- **Naïve Bayes has found a home in spam filtering**
 - **Paul Graham's A Plan for Spam**
 - <http://www.paulgraham.com/spam.html>
 - **Widely used in spam filters**
 - **But many features beyond words:**
 - **black hole lists, etc.**
 - **particular hand-crafted text patterns**

Copyright Ellis Horowitz, 2011-2015. 17

Here is Paul's page, on his spam filtering expts.

Here is SpamAssassin, and this is an old page with results of tests performed.

••



Naive Bayes is Not So Naive

- Very fast learning and testing (basically just count words)
- Low storage requirements
- Very good in domains with many equally important features
- More robust to irrelevant features than many learning methods

Irrelevant features cancel each other without affecting results

Copyright Ellis Horowitz, 2011-2015.

19

..



USC **Viterbi**
School of Engineering

Evaluating Categorization

- **Measures:** precision, recall, F1, classification accuracy
- **Classification accuracy:** r/n where n is the total number of test docs and r is the number of test docs correctly classified

Copyright Ellis Horowitz, 2011-2015.

22

••



USC **Viterbi**
School of Engineering

WebKB Experiment (1998)

- Classify webpages from CS departments into:
 - student, faculty, course, project
- Train on ~5,000 hand-labeled web pages
 - Cornell, Washington, U.Texas, Wisconsin
- Crawl and classify a new site (CMU) using Naïve Bayes
- Results

	Student	Faculty	Person	Project	Course	Departmt
Extracted	180	66	246	99	28	1
Correct	130	28	194	72	25	1
Accuracy:	72%	42%	79%	73%	89%	100% <small>Copyright Ellis Horowitz, 2011-2015.</small>

..



USC **Viterbi**
School of Engineering

Recall: Vector Space Representation

- **Each document is a vector, one component for each term (= word).**
- **Normally normalize vectors to unit length.**
- **High-dimensional vector space:**
 - Terms are axes
 - 10,000+ dimensions, or even 100,000+
 - Docs are vectors in this space
- **How can we do classification in this space?**

Copyright Ellis Horowitz, 2011-2015.

25 25

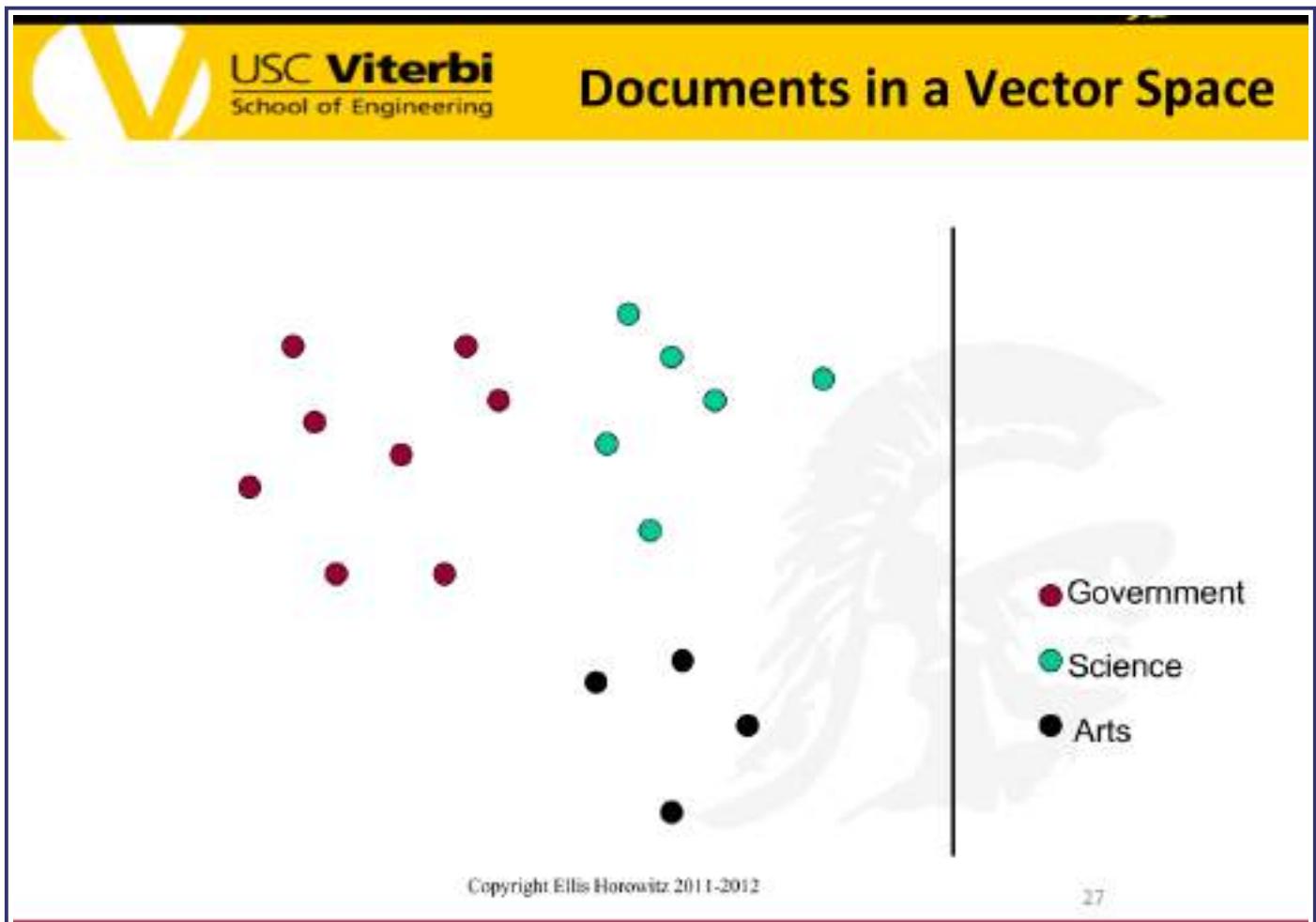
••



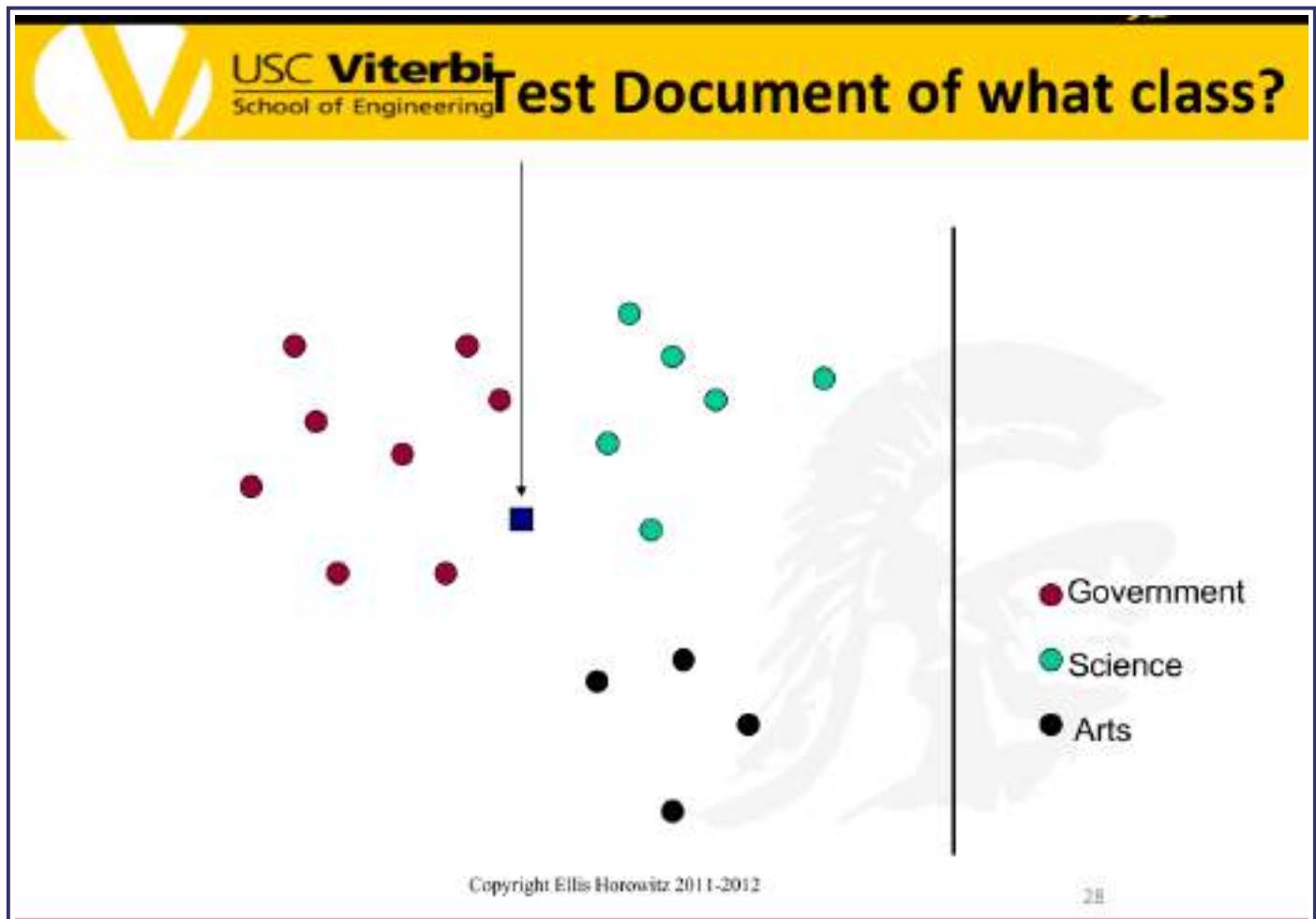
Classification Using Vector Spaces

- In vector space classification, training set corresponds to a labeled set of points (equivalently, vectors)
- Premise 1: Documents in the same class form a contiguous region of space
- Premise 2: Documents from different classes don't overlap (much)
- Learning a classifier: build surfaces to delineate classes in the space

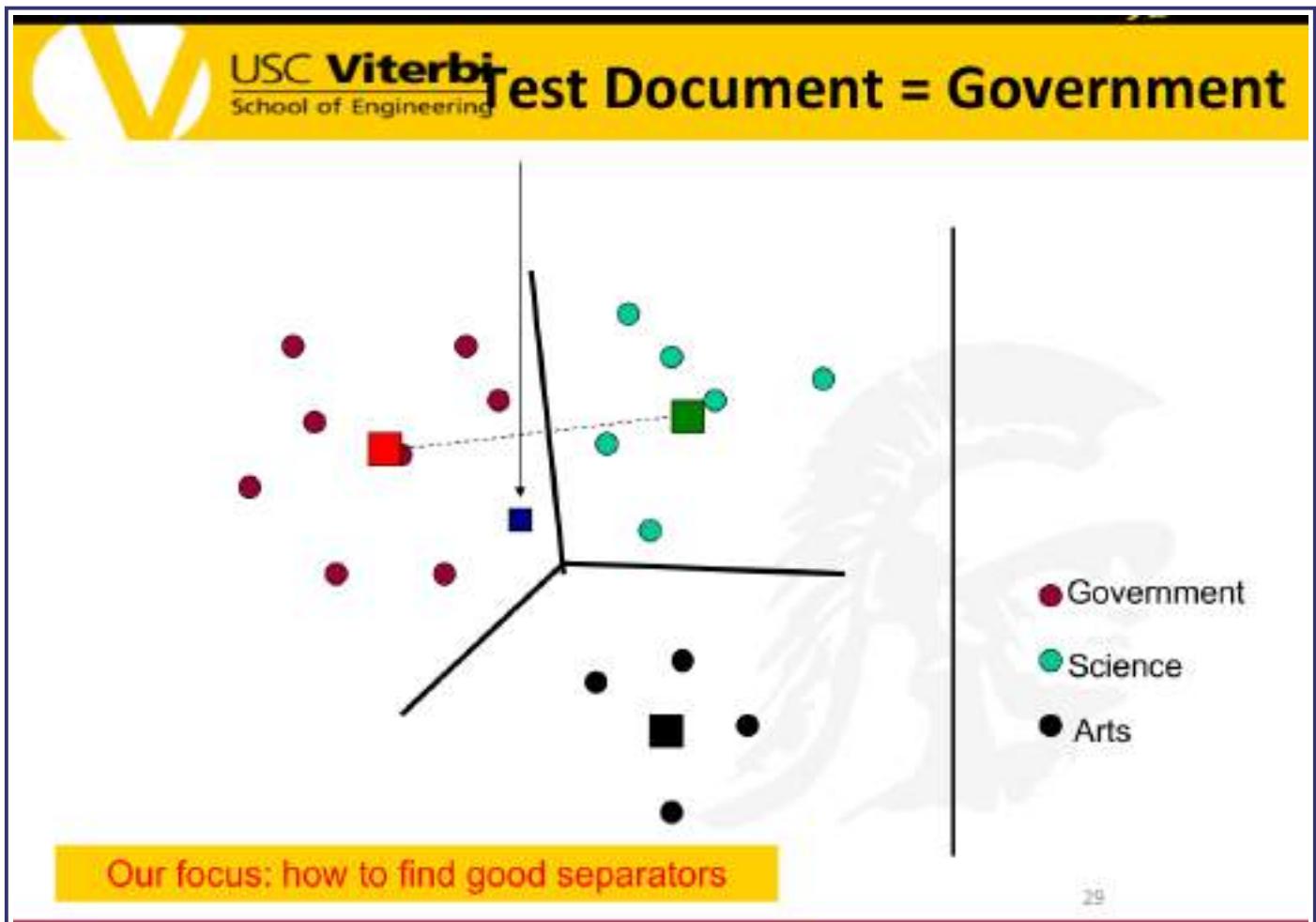
••



••



••



••



Definition of centroid

$$\vec{\mu}(c) = \frac{1}{|D_c|} \sum_{d \in D_c} \vec{v}(d)$$

- Where D_c is the set of all documents that belong to class c and $v(d)$ is the vector space representation of d .
- Note that centroid will in general not be a unit vector even when the inputs are unit vectors.

..



Rocchio classification

- Rocchio forms a simple representative for each class: the centroid/prototype
- Classification: nearest prototype/centroid
- It does not guarantee that classifications are consistent with the given training data

Why not?

Copyright: Ellis Horowitz, 2011-2015

31 31

In Rocchio classification, centroids (one for each term group) are used to specify regions; lines/planes/hyperplanes between centroids produce convex **Voronoi** regions. The new/incoming term's closest centroid is used to classify the term.

..



- A simple form of Fisher's linear discriminant
- Little used outside text classification
 - It has been used quite effectively for text classification
 - But in general worse than Naïve Bayes
- Again, cheap to train and test documents

••



USC **Viterbi**
School of Engineering

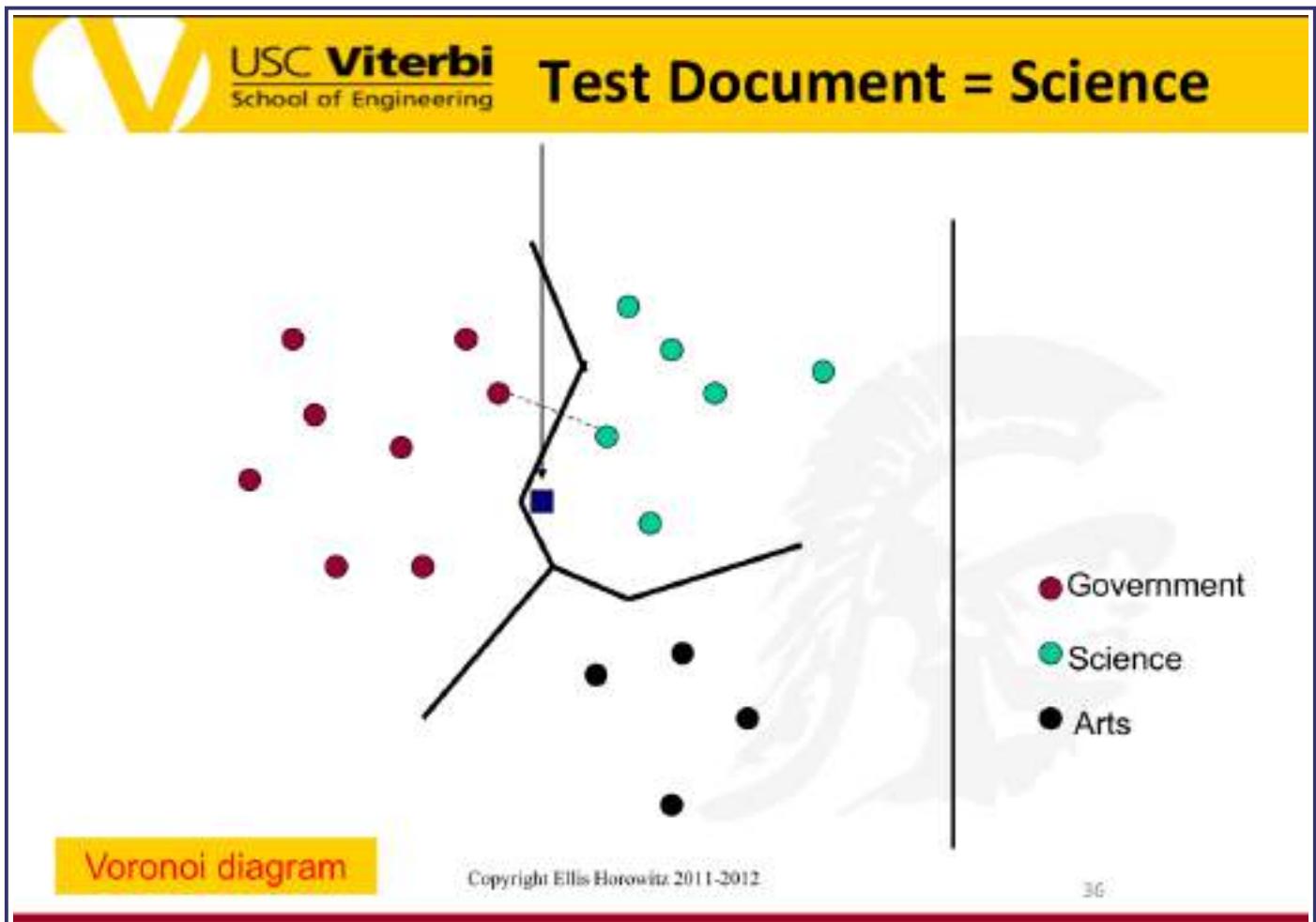
k Nearest Neighbor Classification

- **$kNN = k$ Nearest Neighbor**
- To classify a document d :
- Define k -neighborhood as the k nearest neighbors of d
- Pick the majority class label in the k -neighborhood
- For larger k can roughly estimate $P(c|d)$ as $\#(c)/k$

Copyright Ellis Horowitz, 2011-2015.

35 35

..



••



USC **Viterbi**
School of Engineering

Nearest-Neighbor Learning

- Learning: just store the labeled training examples D
- Testing instance x (*under 1NN*):
 - Compute similarity between x and all examples in D .
 - Assign x the category of the most similar example in D .
- Does not compute anything beyond storing the examples
- Also called:
 - Case-based learning
 - Memory-based learning
 - Lazy learning
- Rationale of kNN: contiguity hypothesis

Copyright Ellis Horowitz, 2011-2015.

37

••



k Nearest Neighbor

- Using only the closest example (1NN) subject to errors due to:
 - A single atypical example.
 - Noise (i.e., an error) in the category label of a single training example.
- More robust: find the k examples and return the majority category of these k
- k is typically odd to avoid ties; 3 and 5 are most common

..



Nearest Neighbor with Inverted Index

- Naively finding nearest neighbors requires a linear search through $|D|$ documents in collection
- But determining k nearest neighbors is the same as determining the k best retrievals using the test document as a query to a database of training documents.
- Use standard vector space inverted index methods to find the k nearest neighbors.
- Testing Time: $O(B|V_t|)$ where B is the average number of training documents in which a test-document word appears.
 - Typically $B \ll |D|$

Copyright Ellis Horowitz, 2011-2015.

39 39

••



- No feature selection necessary
- No training necessary
- Scales well with large number of classes
 - Don't need to train n classifiers for n classes
- Classes can influence each other
 - Small changes to one class can have ripple effect
- Done naively, very expensive at test time
- In most cases it's more accurate than NB or Rocchio

••



Rocchio Anomaly

- Prototype models have problems with polymorphic (disjunctive) categories.



Copyright Ellis Horowitz, 2011-2015.

42

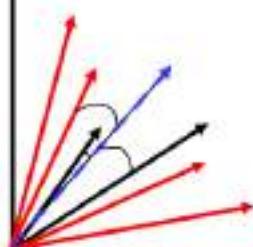
42

••



3 Nearest Neighbor vs. Rocchio

- Nearest Neighbor tends to handle polymorphic categories better than Rocchio/NB.



Copyright Ellis Horowitz, 2011-2015.

43 43

••



Bias vs. capacity – notions and terminology

- Consider asking a botanist: **Is an object a tree?**
 - Too much *capacity*, low *bias*
 - Botanist who memorizes
 - Will always say “no” to new object (e.g., different # of leaves)
 - Not enough capacity, high bias
 - Lazy botanist
 - Says “yes” if the object is green
 - You want the middle ground

Copyright Ellis Horowitz, 2011-2015. Example due to C. Burges) 44

••



kNN vs. Naive Bayes

- Bias/Variance tradeoff
 - Variance \approx Capacity
- kNN has high variance and low bias.
 - Infinite memory
- Rocchio/NB has low variance and high bias.
 - Linear decision surface between classes

••



Summary: Representation of Text Categorization Attributes

- Representations of text are usually very high dimensional
 - “The curse of dimensionality”
- High-bias algorithms should generally work best in high-dimensional space
 - They prevent overfitting
 - They generalize more
- For most text categorization tasks, there are many relevant features and many irrelevant ones

Copyright Ellis Horowitz, 2011-2015.

47 47

••



USC Viterbi
School of Engineering

Which classifier do I use for a given text classification problem?

- Is there a learning method that is optimal for all text classification problems?
- No, because there is a tradeoff between bias and variance.
- Factors to take into account:
 - How much training data is available?
 - How simple/complex is the problem? (linear vs. nonlinear decision boundary)
 - How noisy is the data?
 - How stable is the problem over time?
 - For an unstable problem, it's better to use a simple and robust classifier.

Copyright Ellis Horowitz, 2011-2015

48

1/40

9:08:01

Inverted indexing

••



Outline

- **Definition of an Inverted index**
- **Examples of Inverted Indices**
- **Representing an Inverted Index**
- **Processing a Query on a Linked Inverted Index**
- **Skip Pointers to Improve Merging**
- **Phrase Queries**
- **biwords**
- **Grammatical Tagging**
- **N-Grams**
- **Distributed Indexing**



Copyright Ellis Horowitz, 2011-2013

2

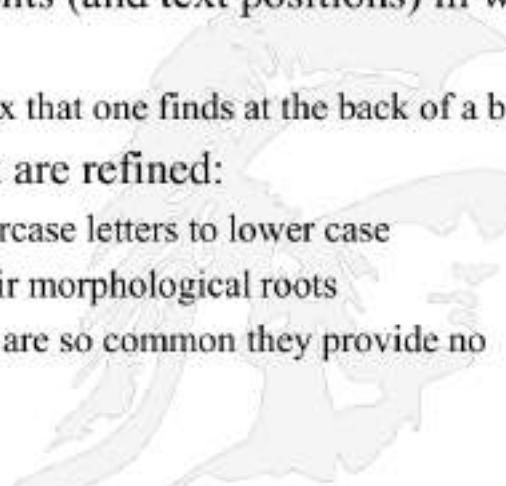
••



USC **Viterbi**
School of Engineering

Creating an Inverted Index

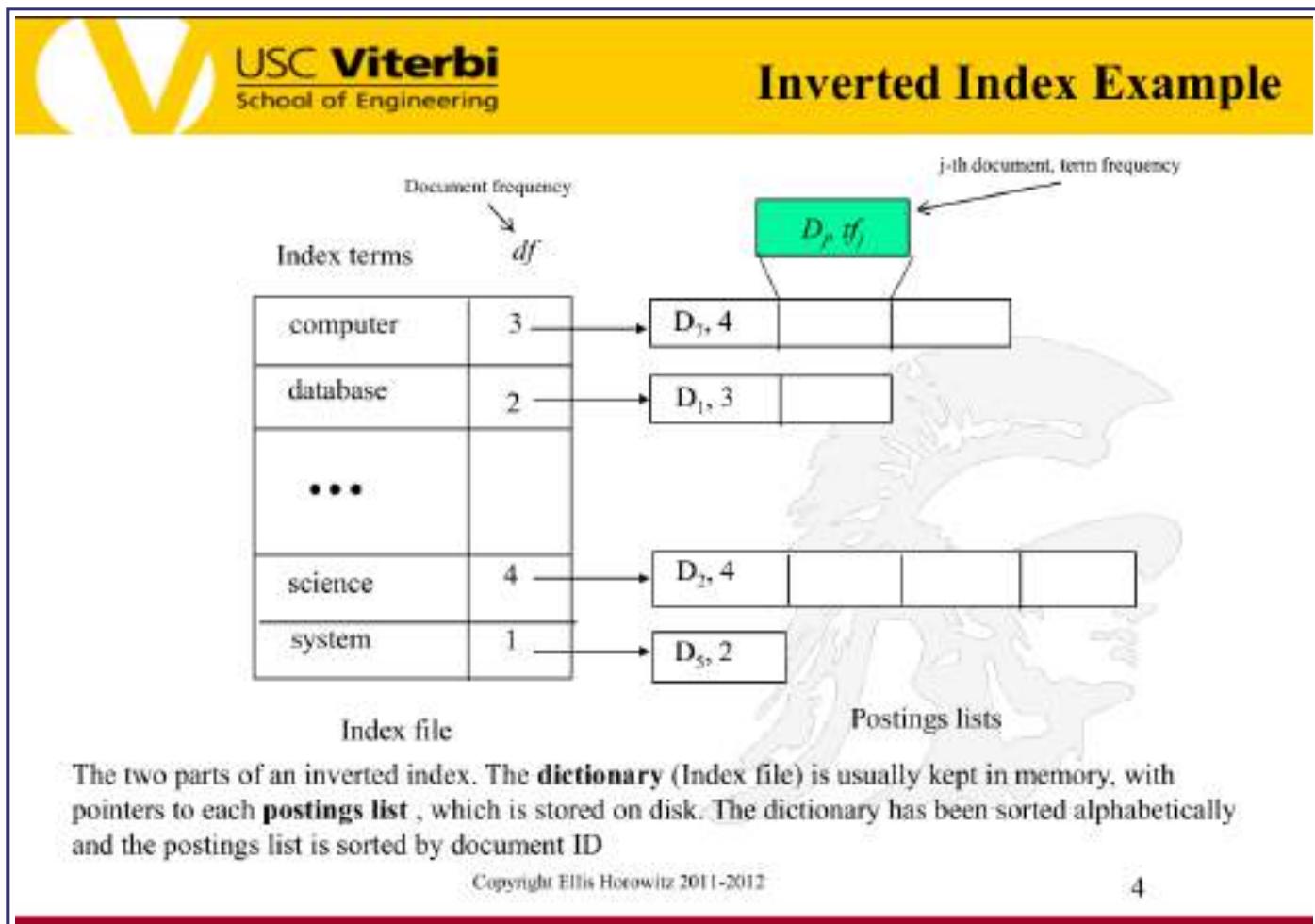
- An inverted index is typically composed of a vector containing all distinct words of the text collection in lexicographical order (which is called the **vocabulary**) and for each word in the vocabulary, a list of all documents (and text positions) in which that word occurs
 - This is nothing more than an index that one finds at the back of a book
- Terms in the inverted file index are refined:
 - **Case folding:** converting all uppercase letters to lower case
 - **Stemming:** reducing words to their morphological roots
 - **Stop words:** removing words that are so common they provide no information



Copyright Ellis Horowitz, 2011-2013

3

••



••

Another Example of an Inverted Index

POS 1 A file is a list of words by position
 10 First entry is the word in position 1 (first word) → FILE
 20 Entry 4562 is the word in position 4562 (4562nd word)
 30 Last entry is the last word
 36 An inverted file is a list of positions by word!

a (1, 4, 40)
 entry (11, 20, 31)
 file (2, 38)
 list (5, 41)
 position (9, 16, 26)
 positions (44)
 word (14, 19, 24, 29, 35, 45)
 words (7)
 4562 (21, 27)

The resulting INVERTED File



Copyright Ellis Horowitz, 2011-2013

5 5

Yet another example (from the book):

Draw the inverted index that would be built for the following document collection.
 (See Figure 1.3 for an example.)

- Doc 1 new home sales top forecasts
- Doc 2 home sales rise in july
- Doc 3 increase in home sales in july
- Doc 4 july new home sales rise

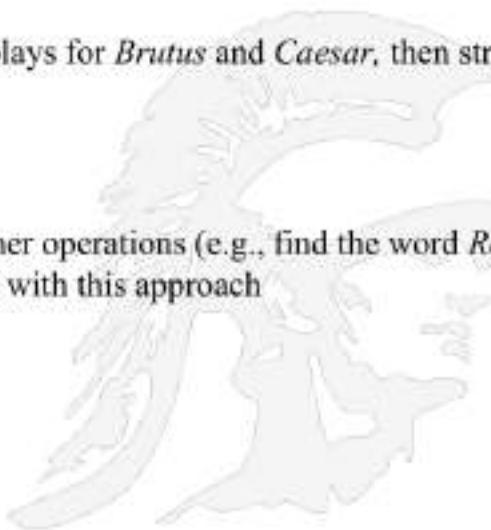
SOLUTION. Inverted Index: forecast->1 home->1->2->3->4 in->2->3
 increase->3 july->2->3 new->1->4 rise->2->4 sale->1->2->3->4 top->1

••



Processing a Query An Example

- The Query
 - Which plays of Shakespeare contain the words *Brutus* AND *Caesar* but NOT *Calpurnia*?
- One Possible Solution
 - One could grep all of Shakespeare's plays for *Brutus* and *Caesar*, then strip out lines containing *Calpurnia*?
 - Too Slow (for large corpora)
 - Requires lots of space
 - This method doesn't allow for other operations (e.g., find the word *Romans* near *countrymen*) are not feasible with this approach



Copyright Ellis Horowitz, 2011-2013

6

••



Term-Document Incidence Matrix

One way to think about an inverted index is to consider it as a sparse matrix where rows represent terms and columns represent documents

documents →	Antony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth
terms ↗	Antony	1	0	0	0	1
	Brutus	1	0	1	0	0
	Caesar	1	0	1	1	1
	Calpurnia	0	1	0	0	0
	Cleopatra	1	0	0	0	0
	mercy	1	0	1	1	1
	worser	1	0	1	1	0

Brutus AND Caesar but NOT Calpurnia

1 if play contains word, 0 otherwise

Copyright Ellis Horowitz, 2011-2013

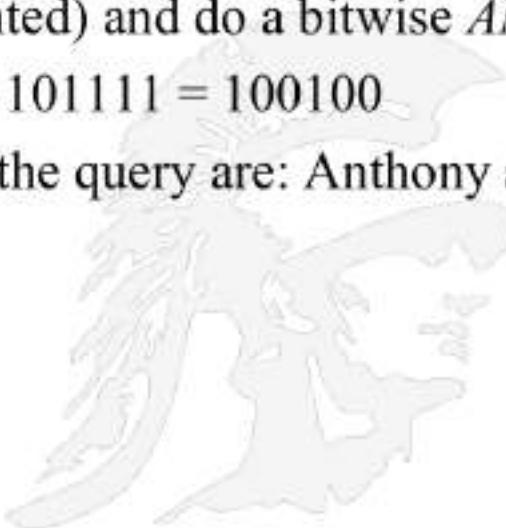
7

..



Incidence Vectors

- So we have a 0/1 vector for each term.
- To answer query: take the vectors for *Brutus*, *Caesar* and *Calpurnia* (complemented) and do a bitwise *AND*.
- $110100 \text{ AND } 110111 \text{ AND } 101111 = 100100$
- So the two plays matching the query are: Anthony and Cleopatra, Hamlet



Copyright Ellis Horowitz, 2011-2013

8

••



Actual Answers to the Query

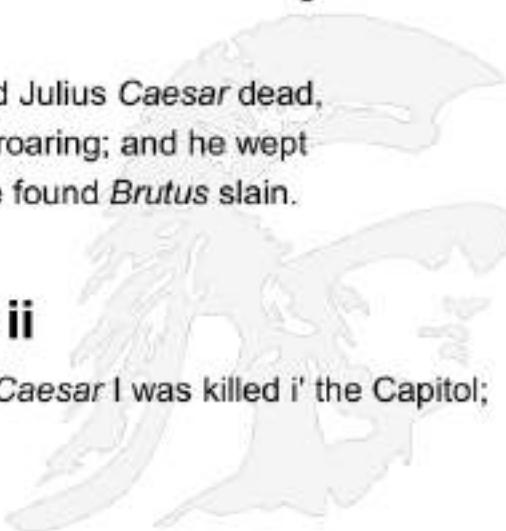
- **Antony and Cleopatra, Act III, Scene ii**

- ***Agrippa [Aside to DOMITIUS ENOBARBUS]:***

- Why, Enobarbus,
- When Antony found Julius Caesar dead,
- He cried almost to roaring; and he wept
- When at Philippi he found *Brutus* slain.

- **Hamlet, Act III, Scene ii**

- ***Lord Polonius:*** I did enact Julius Caesar. I was killed i' the Capitol;
Brutus killed me.

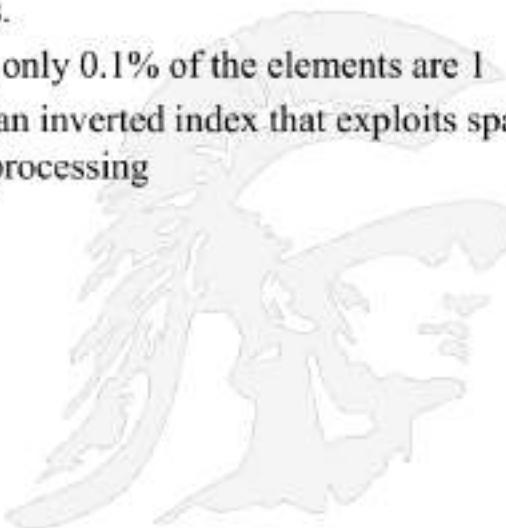


••



Inverted Indexes are Naturally Sparse

- Given 1 million documents and 500,000 terms
- The term x Document matrix in this case will have size 500K x 1M or half-a-trillion 0's and 1's.
- But it has no more than one billion 1's.
 - So the matrix is extremely sparse, only 0.1% of the elements are 1
- So instead we use a data structure for an inverted index that exploits sparsity and then devise algorithms for query processing



Copyright Ellis Horowitz, 2011-2013

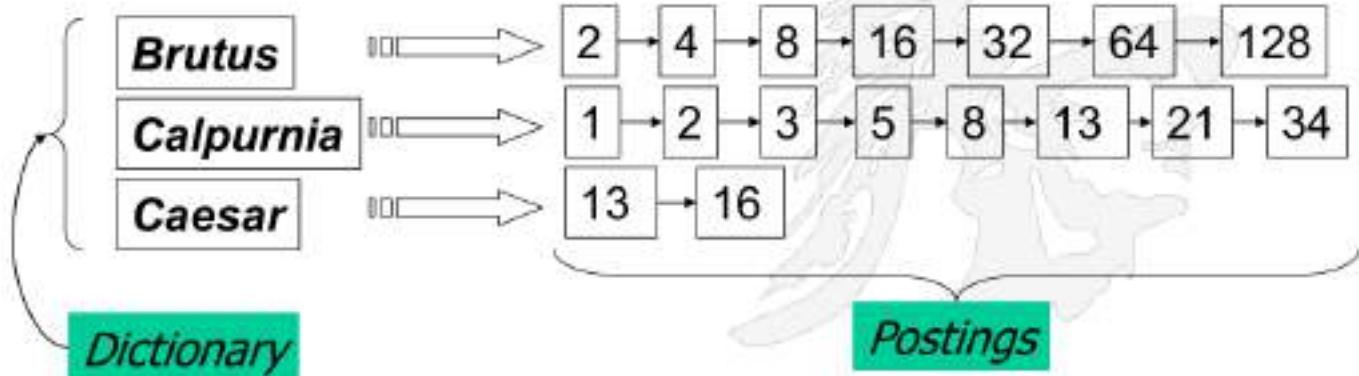
10

••



USC Viterbi
School of Engineering **Inverted Index Stored In Two Parts**

- For each term T , we must store a list of all documents that contain T .
- Linked lists are generally preferred to arrays
 - Dynamic space allocation
 - Insertion of terms into documents easy
 - However, there is space overhead of pointers, though this is not too serious



Copyright Ellis Horowitz, 2011-2013

11

••



USC **Viterbi**
School of Engineering

Inverted Index

- Documents are parsed to extract words and these are saved with the document ID i.e a sequence of (Modified token, Document ID) pairs

Doc 1 Doc 2

I did enact Julius Caesar I was killed i' the Capitol; Brutus killed me.

So let it be with Caesar. The noble Brutus hath told you Caesar was ambitious

Copyright Ellis Horowitz, 2011-2013

Term	Doc #
I	1
did	1
enact	1
julius	1
caesar	1
i	1
was	1
killed	1
'	1
the	1
capitol	1
brutus	1
killed	1
me	1
so	2
let	2
it	2
be	2
with	2
caesar	2
the	2
noble	2
brutus	2
hath	2
told	2
you	2
caesar	2
was	2
ambitious	2
	12

•

 USC **Viterbi**
School of Engineering

- If the corpus is known in advance, then after all documents have been parsed the inverted file is sorted by terms

Initial capture of terms →

Term	Doc #
I	1
did	1
enact	1
julius	1
caesar	1
I	1
was	1
killed	1
I	1
the	1
capitol	1
brutus	1
killed	1
me	1
so	2
let	2
it	2
be	2
with	2
caesar	2
the	2
noble	2
brutus	2
hath	2
told	2
you	2
caesar	2
was	2
ambitious	2

Refined list of terms

Term	Doc #
ambitious	2
be	2
brutus	1
brutus	2
capitol	1
caesar	1
caesar	2
caesar	2
did	1
enact	1
hath	1
I	1
I	1
I	1
it	2
julius	1
killed	1
killed	1
let	2
me	1
noble	2
so	2
the	1
the	2
told	2
you	2
was	1
was	2
with	2

Copyright Ellis Horowitz, 2011-2013

••

 USC **Viterbi**
School of Engineering

- Multiple term entries in a single document are merged.
- Frequency information is added.

Why frequency?
Will discuss later.

Term	Doc #
ambitious	2
be	2
brutus	1
brutus	2
capitol	1
caesar	1
caesar	2
caesar	2
did	1
enact	1
hath	1
I	1
I	1
it	1
it	2
julius	1
killed	1
killed	1
let	2
me	1
noble	2
so	2
the	1
the	2
told	2
you	2
was	1
was	2
with	2



Term	Doc #	Freq
ambitious	2	1
be	2	1
brutus	1	1
brutus	2	1
capitol	1	1
caesar	1	1
caesar	2	2
did	1	1
enact	1	1
hath	2	1
I	1	2
I	1	1
it	2	1
julius	1	1
killed	1	2
let	2	1
me	1	1
noble	2	1
so	2	1
the	1	1
the	2	1
told	2	1
you	2	1
was	1	1
was	2	1
with	2	1

Copyright Ellis Horowitz, 2011-2013

14

••

**USC Viterbi
School of Engineering**

- The file is commonly split into a *Dictionary* and a *Postings* file

Term	Doc #	Freq
ambitious	2	1
be	2	1
brutus	1	1
brutus	2	1
capitol	1	1
caesar	1	1
caesar	2	2
did	1	1
enact	1	1
hath	2	1
i	1	2
it	1	1
it	2	1
julius	1	1
killed	1	2
let	2	1
me	1	1
noble	2	1
so	2	1
the	1	1
the	2	1
told	2	1
you	2	1
was	1	1
was	2	1
with	2	1

→

Term	N docs	Tot Freq	Dec #	Freq
ambitious	1	1	2	1
be	1	1	1	1
brutus	2	2	2	1
capitol	1	1	1	1
caesar	2	3	1	1
did	1	1	2	2
enact	1	1	1	1
hath	1	1	1	1
i	1	2	2	1
it	1	1	1	2
julius	1	1	2	1
killed	1	2	1	1
let	1	1	1	2
me	1	1	2	1
noble	1	1	1	1
so	1	1	2	1
the	2	2	2	1
told	1	1	1	1
you	1	1	2	1
was	2	2	2	1
with	1	1	1	1
			2	1
			2	1

Copyright Ellis Horowitz, 2011-2013

15

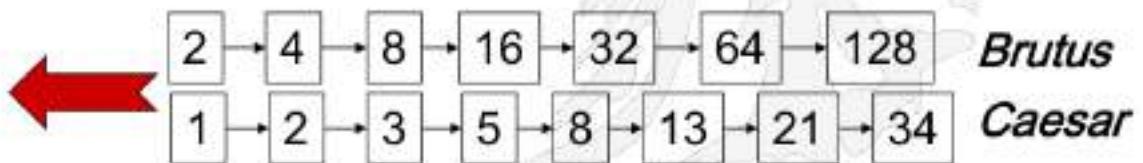
••



- Consider processing the query:

Brutus AND Caesar

- Locate *Brutus* in the Dictionary;
 - Retrieve its postings.
- Locate *Caesar* in the Dictionary;
 - Retrieve its postings.
- “Merge” the two postings (postings are document ids):

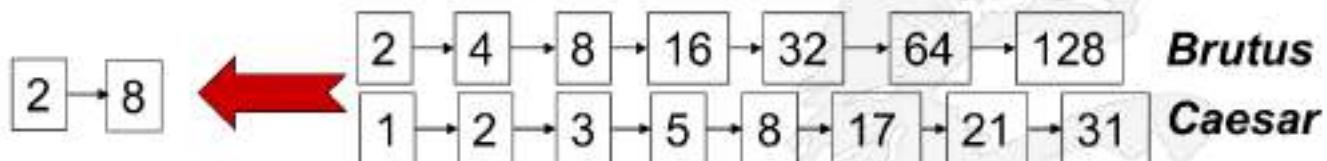


••



Basic Merge

- Walk through the two postings simultaneously, in time linear in the total number of postings entries



If the list lengths are m and n , the merge takes $O(m+n)$ operations.

••



Query Optimization

- What is the best order for query processing?
- Consider a query that is an *AND* of t terms.
- For each of the t terms, get its postings, then *AND* together.

Brutus	⇒	2 4 8 16 32 64 128
Calpurnia	⇒	1 2 3 5 8 16 21 34
Caesar	⇒	13 16

Query: Brutus AND Calpurnia AND Caesar

Copyright Ellis Horowitz, 2011-2013

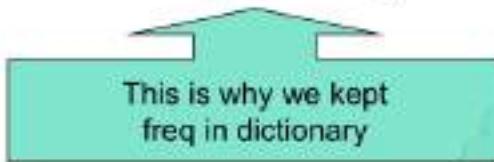
18

•

 USC **Viterbi**
School of Engineering

Query Optimization Example

- Process in order of increasing freq:
 - *start with smallest set, then keep cutting further.*



This is why we kept
freq in dictionary

Brutus	⇒	2 4 8 16 32 64 128
Calpurnia	⇒	1 2 3 5 8 13 21 34
Caesar	⇒	13 16

Execute the query as (*Caesar AND Brutus*) *AND Calpurnia*.

••



USC **Viterbi**
School of Engineering

To speed up the merging of postings we
use the technique of ***Skip Pointers***



Copyright Ellis Horowitz, 2011-2013

20

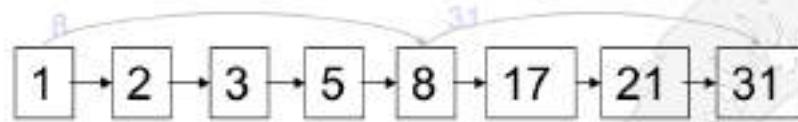
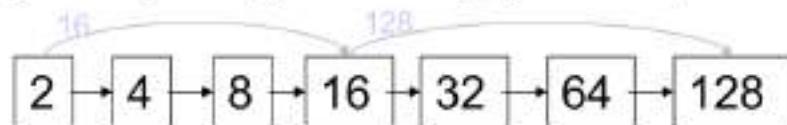
••



USC **Viterbi**
School of Engineering

The Technique of Skip Pointers

Augment postings with skip pointers (at indexing time)

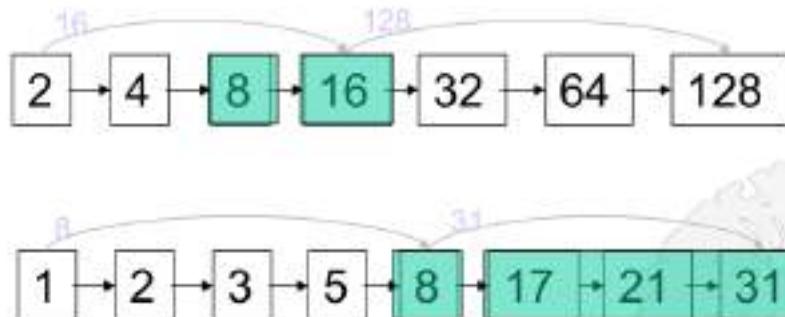


- Why?
- To skip postings that will not figure in the search results.
- How?
- Where do we place skip pointers?

••



Query processing with skip pointers



Suppose we've stepped through the lists until we process **8** on each list.

When we get to **16** on the top list, we see that its successor is **32**.

But the skip successor of **8** on the lower list is **31**, so we can skip ahead past the intervening postings.

22

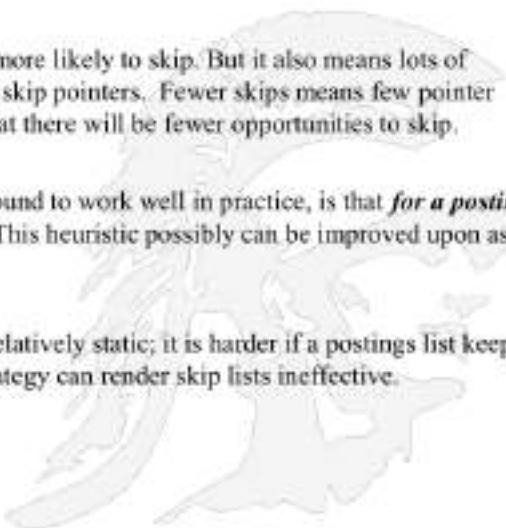
Aside: skip pointers (aka skip lists) can be used to search a linked list of sorted items faster than $O(n)$, ie. in $O(\sqrt{n})$...

••



Facts on Skip Pointers

- skip pointers are added at indexing time; they are shortcuts, and they only help for AND queries and they are useful when the corpus is relatively static
- there are two questions that must be answered:
 1. where should they be placed?
 2. how do the algorithms change?
- More skips means shorter skip spans, and that we are more likely to skip. But it also means lots of comparisons to skip pointers, and lots of space storing skip pointers. Fewer skips means few pointer comparisons, but then long skip spans which means that there will be fewer opportunities to skip.
- A simple heuristic for placing skips, which has been found to work well in practice, is that *for a postings list of length P , use \sqrt{P} evenly-spaced skip pointers*. This heuristic possibly can be improved upon as it ignores any details of the distribution of query terms.
- Building effective skip pointers is easy if an index is relatively static; it is harder if a postings list keeps changing because of updates. A malicious deletion strategy can render skip lists ineffective.
- See the YouTube video
- <http://www.youtube.com/watch?v=tPsCQOsa7j0>



••



Phrase Queries



Copyright Ellis Horowitz, 2011-2013

24

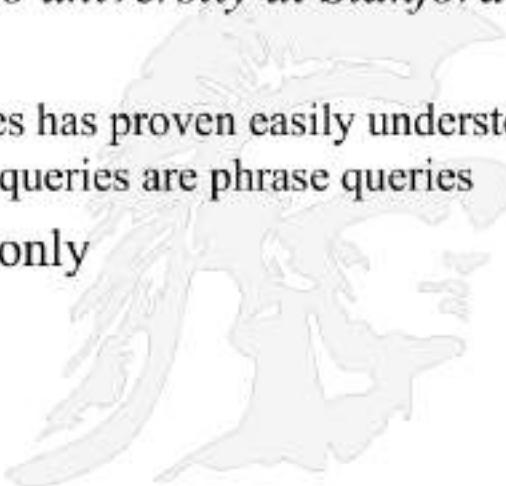
••



USC **Viterbi**
School of Engineering

Phrase queries

- We want to answer queries such as “*stanford university*” – as a phrase
- Thus the sentence “*I went to university at Stanford*” is not a match.
 - The concept of phrase queries has proven easily understood by users; about 10% of web queries are phrase queries
- No longer suffices to store only $\langle term : docs \rangle$ entries



Copyright Ellis Horowitz, 2011-2013

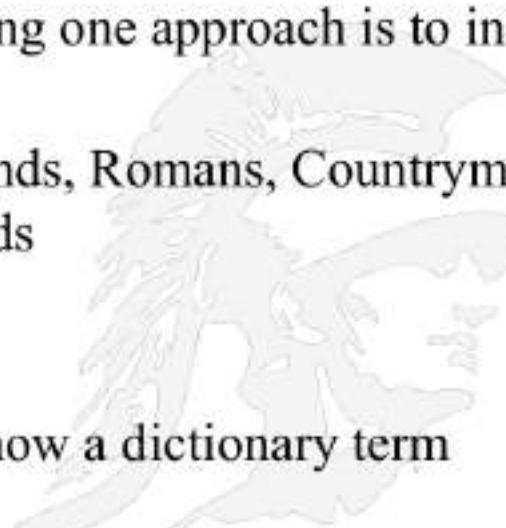
25

••



Using Biword Indexes for Phrase Searching

- A biword (or a 2-gram) is a consecutive pair of terms in some text
- To improve phrase searching one approach is to index every biword in the text
- For example the text “Friends, Romans, Countrymen” would generate the bi-words
 - *friends romans*
 - *romans countrymen*
- Each of these bi-words is now a dictionary term



••



- Consequences
 - Biwords will cause an explosion in the vocabulary database
 - Queries longer than 2 words will have to be broken into biword segments
- Example: suppose the query is the 4 word phrase

stanford university palo alto

The query can be broken into the Boolean query on biwords:

stanford university AND university palo AND palo alto

- Matching the query to terms in the index will work, but may also produce false positives (i.e. occurrences of the biwords, but not the full 4 word query)

••



Alternate Solution Using Positional Indexes

- Store, for each *term*, entries of the form:
*<number of docs containing term;
doc1: position1, position2 ... ;
doc2: position1, position2 ... ;
etc.>*



Copyright Ellis Horowitz, 2011-2013

28

••

USC **Viterbi**
School of Engineering

Positional Index Example

for each term in the vocabulary, we store postings of the form

docID: position1, position2, ...,

where each position is a token index in the document.

Each posting will also usually record the term frequency

Adopting a positional index expands required postings storage significantly,
even if we compress position values/offsets

<**be**: 993427;
1: 7, 18, 33, 72, 86, 231;
2: 3, 149;
4: 17, 191, 291, 430, 434;
5: 363, 367, ...>

- **Nevertheless, this expands postings storage substantially**

Copyright Ellis Horowitz, 2011-2013

29

••



Processing a Phrase Query

- Extract inverted index entries for each distinct term: *to, be, or, not*.
- Merge their *doc:position* lists to enumerate all positions with “*to be or not to be*”.
 - ***to:***
 - 2:1,17,74,222,551; 4:8,16,190,429,433; 7:13,23,191; ...
 - ***be:***
 - 1:17,19; 4:17,191,291,430,434; 5:14,19,101; ...
 - Same general method for proximity searches
 - In document 4 the word “*to*” appears in position 16 and the word “*be*” appears in position 17, so they are adjacent

••



Another Approach

- Among possible queries, **nouns and noun phrases** often appear, e.g. “abolition of slavery”, “renegotiation of the constitution”
- But as seen above, related nouns can often be divided from each other by various function words
- These needs can be incorporated into the biword indexing model in the following way:
 - First, we tokenize the text and perform **part-of-speech-tagging**. We can then group terms into nouns, including proper nouns, (N) and function words, including articles and prepositions, (X), among other classes.
 - Now deem any string of terms of the form NX*N to be an extended biword. Each such extended biword is made a term in the vocabulary.
 - E.g. “renegotiation of the constitution” is mapped to N X X N (two nouns and two others), so the others are ignored and the two word phrase “renegotiation constitution” is added to the index
- Programs that identify a word’s part-of-speech tag are based on statistical or rule-based approaches and are trained using large corpora

••



USC Viterbi
School of Engineering

Some High Frequency Noun Phrases

<i>TREC</i>		<i>Patent</i>	
<i>Frequency</i>	<i>Phrase</i>	<i>Frequency</i>	<i>Phrase</i>
65824	united states	975362	present invention
61327	article type	191625	u.s. pat
33864	los angeles	147352	preferred embodiment
18062	Hong kong	95097	carbon atoms
17788	North korea	87903	group consisting
17308	New York	81809	room temperature
15513	San diego	78458	seq id
15009	Orange county	75850	brief description

The phrases above were identified by POS tagging; The data above shows that common phrases are used more frequently in patent data as patents have a very formal style; many of the TREC phrases are proper nouns, whereas patent phrases are those that occur in all patents

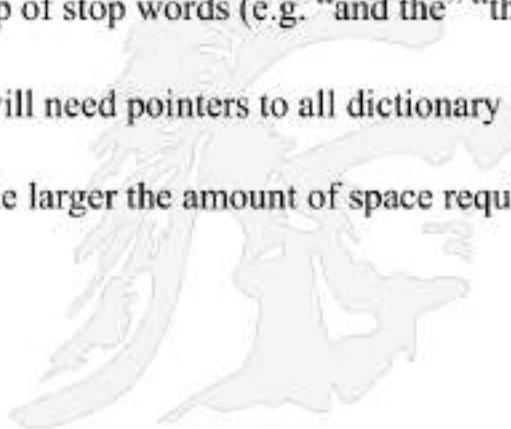
••



USC **Viterbi**
School of Engineering

Building *n*-gram Indexes

- Generalizing from bi-words, an *n-gram* is any sequence of *n* consecutive words
- N-grams can be identified at the time of parsing
- N-grams of all lengths form a Zipf distribution with a few common phrases occurring very frequently and a large number occurring with frequency 1
- Common n-grams are usually made up of stop words (e.g. “and the” “there is”)
- For each *n*-gram, the inverted index will need pointers to all dictionary terms containing it – the “postings”
- Therefore, the larger the value of *n*, the larger the amount of space required to hold all n-grams



Copyright Ellis Horowitz, 2011-2013

33

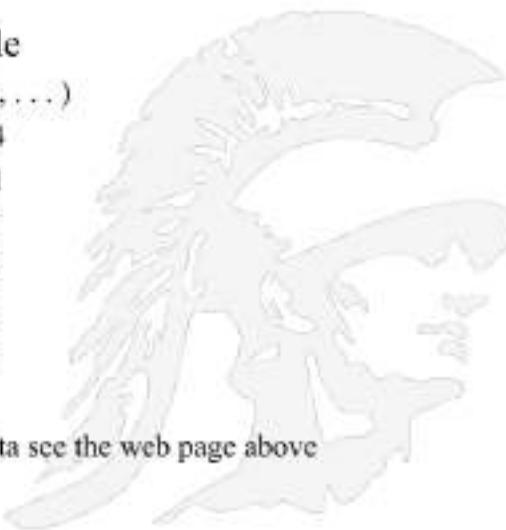
••



USC **Viterbi**
School of Engineering

Google's N-Gram Facts

- Google made available a file of n-grams derived from the web pages it indexed
- <http://googleresearch.blogspot.com/2006/08/all-our-n-gram-are-belong-to-you.html>
- Statistics for the Google n-gram sample
- Number of tokens 1,024,908,267,229 (1 trillion, . . .)
- Number of sentences 95,119,665,584
- Number of unigrams 13,588,391
- Number of bigrams 314,843,401
- Number of trigrams 977,069,902
- Number of four grams 1,313,818,354
- Number of five grams 1,176,470,663
- For specific examples of 3-gram and 4-gram data see the web page above



Copyright Ellis Horowitz, 2011-2013

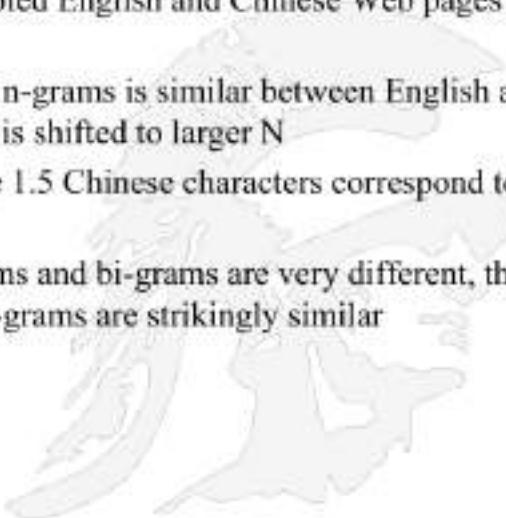
34

••



Comparing N-Grams Across Languages

- S. Yang et al, N-gram statistics in English and Chinese: Similarities and differences, ICSC, 2007, Int'l Conf. on semantic computing, 454-460
- http://static.googleusercontent.com/external_content/untrusted_dlcp/research.google.com/en/us/pubs/archive/33035.pdf
- They analyzed 200 million randomly sampled English and Chinese Web pages and concluded:
 1. The distribution of the unique number of n-grams is similar between English and Chinese, though the Chinese distribution is shifted to larger N
 2. The distribution indicates that on average 1.5 Chinese characters correspond to 1 English word
 3. While frequency distributions of uni-grams and bi-grams are very different, the frequency distribution for 3-grams and 4-grams are strikingly similar



Copyright Ellis Horowitz, 2011-2013

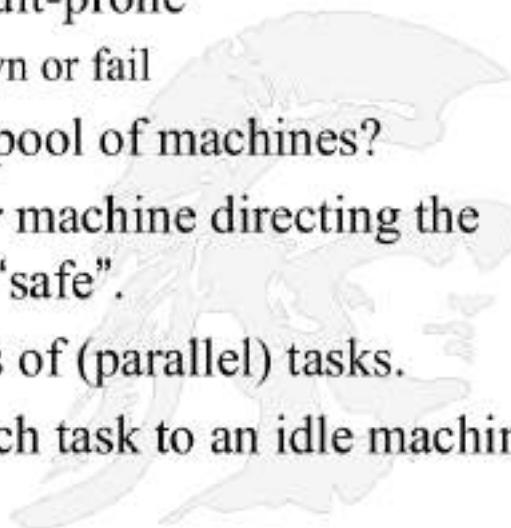
35

••



Distributed Indexing

- For web-scale indexing one must use a distributed computing cluster
- Individual machines are fault-prone
 - Can unpredictably slow down or fail
- How do we exploit such a pool of machines?
- Must we maintain a *master* machine directing the indexing job – considered “safe”.
- Break up indexing into sets of (parallel) tasks.
- Master machine assigns each task to an idle machine from a pool.

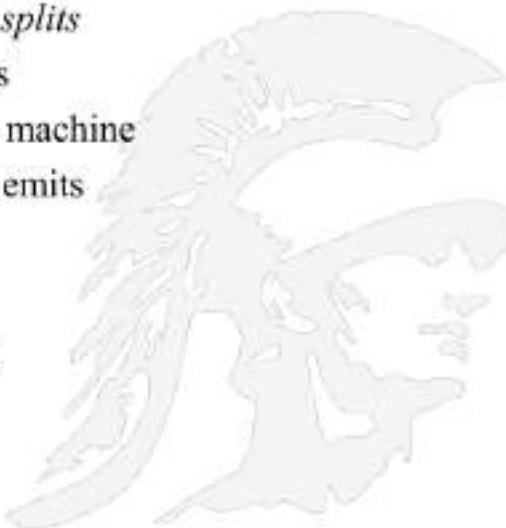


••



Parallel Tasks

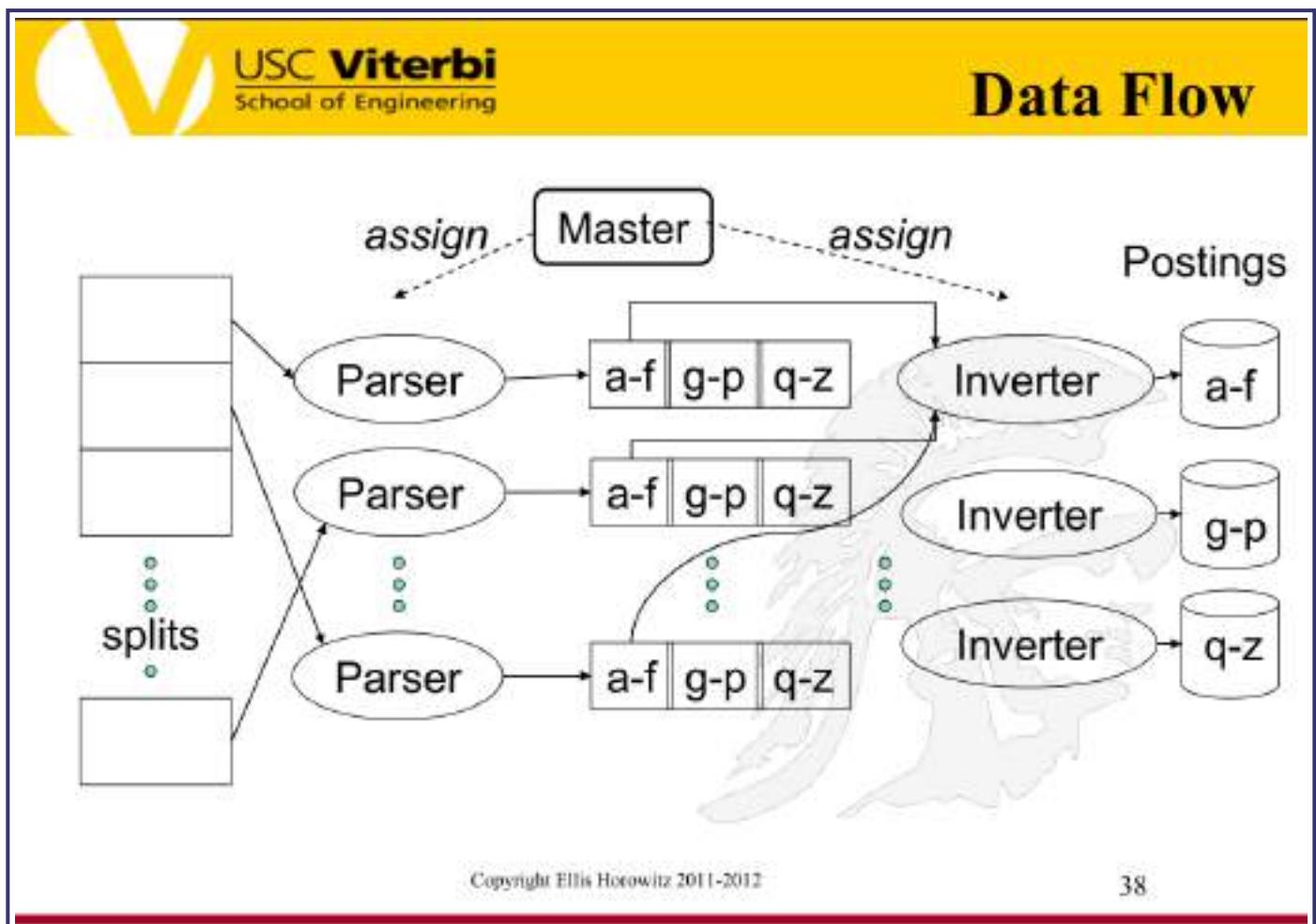
- One approach is to use two sets of parallel tasks
 - Parsers
 - Inverters
- Break the input document corpus into *splits*
 - Each split is a subset of documents
- Master assigns a split to an idle parser machine
- Parser reads a document at a time and emits (term, doc) pairs
- Parser writes pairs into j partitions
- Each for a range of terms' first letters
 - (e.g., $a-f, g-p, q-z$) – here $j=3$.
- Now to complete the index inversion



Copyright Ellis Horowitz, 2011-2013

37

••



••



- Collect all (term, doc) pairs for a partition
- Sorts and writes to postings list
- Each partition contains a set of postings

Above process flow a special case of *MapReduce*.

••



Simplest Approach to Dynamic Indexing

- Maintain “big” main index
- New docs go into “small” auxiliary index
- Search across both, merge results periodically
- Deletions
 - Invalidation bit-vector for deleted docs
 - Filter docs output on a search result by this invalidation bit-vector
- Periodically, re-index into one main index

Copyright Ellis Horowitz, 2011-2013

40

← 1/40 → * * * 9:08:20

Videos!

••

University of Southern California

 USC **Viterbi**
School of Engineering



Video Search Engines YouTube et al

Copyright Ellis Horowitz 2011-2022

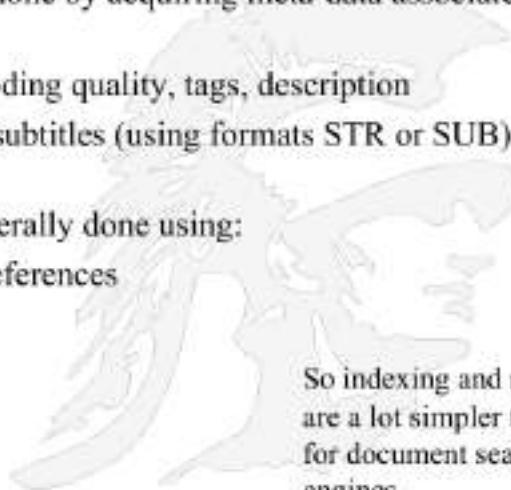
••

University of Southern California

USC **Viterbi**
School of Engineering

Video Search Engines – Quick Summary

- A **video search engine** is a web-based search engine which crawls the web primarily for video content.
 - YouTube is not strictly a video search engine as it does not crawl the web looking for video content
- The *indexing* of video content is normally done by acquiring meta-data associated with the video, e.g.
 - Author, title, creation date, duration, coding quality, tags, description
 - Other aspects of video recognition are subtitles (using formats STR or SUB) and transcription (using format TTXT)
- The *ranking* of videos under a query is generally done using:
 - Relevance: using metadata and user preferences
 - Ordered by date of upload
 - Ordered by number of views
 - Ordered by duration
 - Ordered by user rating



So indexing and ranking
are a lot simpler than
for document search
engines

••

University of Southern California 

 USC **Viterbi**
School of Engineering

Video Search Engines That *Crawl* for Content

- ***Those no longer existing***
 - ***CastTV*** was a Web-wide video search engine that was founded in 2006
 - ***No longer active***
 - ***Munax*** released their first version all-content search engine in 2005 and powers both nationwide and worldwide search engines with video search
 - <http://www.munax.com/> ***no longer active***
 - ***ScienceStage*** is an integrated universal search engine for science-oriented videos. All videos are also semantically matched to millions of research documents from open-access databases.
 - ***No longer active***
- ***A few remain***
 - ***Bing*** does crawl for videos, see <https://www.bing.com/videos/>
 - ***blinkx*** (renamed as RhythmOne) was launched in 2004 and uses speech recognition and visual analysis to process downloaded video rather than rely on metadata alone
 - <http://www.blinkxtv.com/> ***now redirects to 360Daily.com***

Copyright Ellis Horowitz, 2011-2022

3

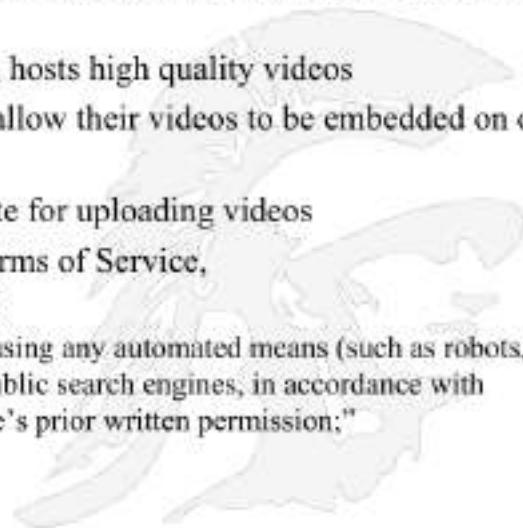
••

University of Southern California 

 USC **Viterbi**
School of Engineering

Video Search Engines That Host

- Largely because of the large file sizes involved, video hosting is highly concentrated on a fairly small number of websites
 - **vimeo.com**, first to support HD video, focuses on short, arty, films
 - **vevo.com**, a joint venture of Universal Music Group, Sony Music Entertainment and Warner Music Group
 - **dailymotion.com**, owned by Vivendi, hosts high quality videos
- Most of these websites which host video allow their videos to be embedded on other websites
- **YouTube.com** has become the defacto site for uploading videos
- It is legal to crawl YouTube, see their Terms of Service, www.youtube.com/static?template=terms
- “3. You are not allowed to access the Service using any automated means (such as robots, botnets or scrapers) except (a) in the case of public search engines, in accordance with YouTube’s robots.txt file; or (b) with YouTube’s prior written permission.”



Copyright Ellis Horowitz, 2011-2022

4

••

University of Southern California



Video Search Engines That Stream Entertainment

- **Hulu** is an American subscription video on demand service jointly owned by Walt Disney, 21st Century Fox, Comcast, and Time Warner
 - In December 2017, Disney acquired Fox's partial ownership, giving it a majority stake; other owners include Comcast
- **Netflix** is an American subscription video on demand service, that originally delivered DVDs;
 - They develop their own content as well as offering content from major film distributors
- **Amazon Prime** is an American subscription video on demand service offering television and file shows for rent or purchase
- **Disney+** a recent entry
- There are many others: XtremeHD, Sling TV, Apple TV+, HBO Max, Acorn TV, etc

• Entertainment

Copyright Ellis Horowitz, 2011-2022

5

••

University of Southern California 

 USC **Viterbi**
School of Engineering

Some Technologies Supporting Video Content

- **Subtitles:** there are two formats, one for subtitles and one for transcripts
 - There are three main types of video subtitling services:
 1. **open caption:** burned into the video
 2. **closed caption:** can be turned on/off, generally at the bottom of the screen
 3. **SDH (Subtitles for the Deaf and Hard of Hearing):** similar to closed caption, but includes words describing actions or moods
 - **SRT or SUB for subtitles**
 - SRT (.srt) stands for “SubRip Subtitle” file, and it’s the most common subtitle/caption file format. It is a **text format**
 - **TTXT for transcripts**
- **Speech Recognition,** used to extract phrases from audio transcripts for better indexing
 - **Gaudi, Google Audio Indexing** uses voice recognition to locate the exact spot where words are spoken
 - <https://www.searchenginejournal.com/google-audio-search-will-it-ever-be-possible/397129/>
 - **Text Recognition:** uses OCR on video slides to detect words,
 - e.g. **TalkMiner System**, see https://www.youtube.com/watch?v=7N6I_m9LywM

Copyright Ellis Horowitz, 2011-2022

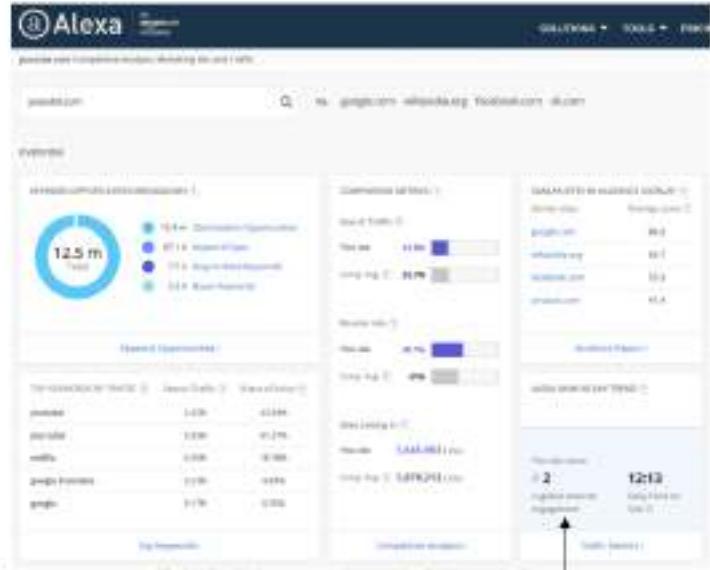
6

••

University of Southern California  USC 

YouTube Background

- YouTube is an American video hosting website headquartered in San Bruno, California, created by three former PayPal employees: Chad Hurley, Steve Chen, Jawed Karim in February 2005.
- In November 2006, it was bought by Google for US\$1.65 billion
- In 2020 Google announced that YouTube generated revenue of \$19.8 billion
- The site allows users to upload, view, rate, share, add to favorites, report and comment on videos
- In January 2022, the website was ranked as the second most popular site by Alexa Internet, a web traffic analysis company (now owned by Amazon)
 - See also https://en.wikipedia.org/wiki/List_of_most_popular_websites

 For details see Related Articles page, Mar 2020

Copyright Ellis Horowitz, 2011-2022

••

University of Southern California  USC 

YouTube as a Search Engine

- YouTube - The 2nd Largest Search Engine (cite:Infographic)
- YouTube processes more than 3 billion searches a month.
- It's bigger than Bing, Yahoo!, Ask and AOL combined!
- <http://www.mushroomnetworks.com/infographics/youtube---the-2nd-largest-search-engine-infographic>



Copyright Ellis Horowitz, 2011-2022

••

University of Southern California 

 USC **Viterbi**
School of Engineering

YouTube Traffic - Some Facts

- **As of 2021:**
 - **60 hours of video are uploaded every minute, or one hour of video is uploaded to YouTube every second.**
 - **Over 4 billion videos are viewed a day**
 - **Over 800 million unique users visit YouTube each month**
 - **Over 3 billion hours of video are watched each month on YouTube**
 - **More video is uploaded to YouTube in one month than the 3 major US networks created in 60 years**
 - **70% of YouTube traffic comes from outside the US**
 - **YouTube is localized in 39 countries and across 54 languages**
 - **It is estimated that YouTube holds 1 sextillion gigabytes of data**
 - <https://www.quora.com/What-is-the-total-size-storage-capacity-of-YouTube-and-at-what-rate-is-it-increasing-How-is-Google-keeping-up-with-the-increasing-demands-of-YouTube%E2%80%99s-capacity-given-that-thousands-of-videos-are-uploaded-every-day>

Copyright Ellis Horowitz, 2011-2022

9

••

University of Southern California 

 USC **Viterbi**
School of Engineering

YouTube Search Engine Issues to Consider

- Since crawling, indexing and ranking are not big challenges for YouTube, what are the major hurdles
- 1. What video formats are acceptable
 - For uploading
 - For downloading
- 2. How are videos to be displayed on: desktops, iPhones, iPads, Android devices, etc
- 3. How does YouTube distribute videos worldwide
 - A content distribution network (CDN)
- 4. How does YouTube monetize its website?
 - YouTube's ContentID system
- 5. How does YouTube keep users watching
 - The YouTube Recommendation System



Copyright Ellis Horowitz, 2011-2022

10

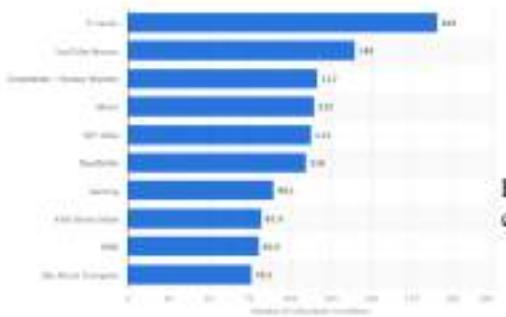
••

University of Southern California 

 USC **Viterbi**
School of Engineering

YouTube Channels

- In order to upload a video you must be a registered user
- In addition YouTube offers a special type of account called a *channel*; channels include
 - thumbnails of videos you've uploaded,
 - members to whom you've subscribed,
 - videos from other members you've picked as favorites,
 - lists of members who are your friends,
 - your subscribers, and
- Biggest YouTube Channels as of 2021



Rank	Channel	Subscribers (approx.)
1	PewDiePie	100M+
2	Logan Paul	80M+
3	Comedian Jimmy Kimmel	70M+
4	MrBeast	60M+
5	ASMR	50M+
6	David Dobrik	40M+
7	UnboxThangs	30M+
8	ASMR Stories	25M+
9	ASMR	20M+
10	Mr Beast Games	15M+

With 1 million subscribers, a YouTuber will make between \$300,000 – \$2 million. To be in the top 1000 YouTubers you must have ~1.8 million subscribers. As of 09/2020, there are more than 2000 YouTubers with over a million subscribers.

<https://www.statista.com/statistics/277758/most-popular-youtube-channels-ranked-by-subscribers/>

---PPT by Ellis Horowitz, 2011-2022

11

••

University of Southern California 

YouTube Gathers Information When Videos are Uploaded



YouTube captures:

- Name
- Description
- Tags
- Note: YouTube immediately assigns a URL
- Note: YouTube suggests possible thumbnails

Video on How to Upload a Video
<https://support.google.com/youtube/answer/57407>

Copyright Ellis Horowitz 2011-2022

The screenshot shows a web browser window with the following details:

- Title Bar:** "University of Southern California" and "USC Viterbi School of Engineering".
- Address Bar:** "Secure https://www.youtube.com/upload".
- Content Area:**
 - Section Header:** "Uploading to YouTube Second Input Screen".
 - Video Preview:** A thumbnail image of a video titled "Lemon Birthday".
 - Upload Status:** "Processing done".
 - Buttons:** "Click 'Publish' to make your video live.", "Basic info", "Translations" (which is selected), and "Advanced settings".
 - Language Options:** "Original language" and "Select language".
 - Translation Section:** "Translate into (1)" with "Selected language" dropdown.
 - Video Details:** "Lemon Birthday" and "Description".
 - Other Buttons:** "Get professional translation", "Buy translator (beta)", "Video Manager", and "Add more releases".

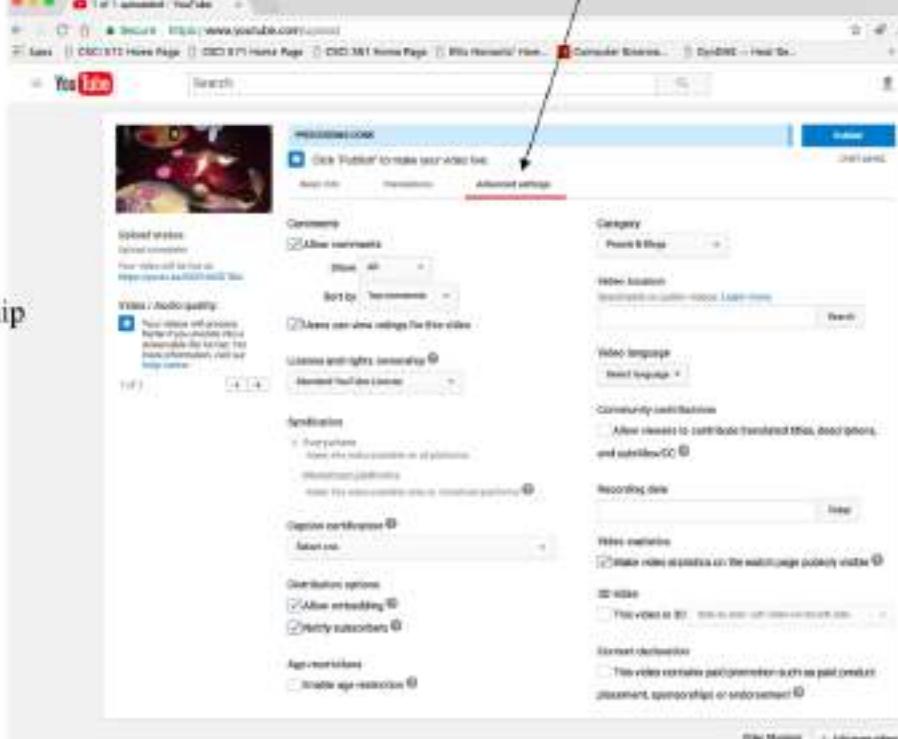
••

University of Southern California  USC

Uploading to YouTube Third Input Screen

YouTube allows the creator to specify:

- License and ownership
- Syndication
- Caption
- Embedding
- Age restrictions
- Categories



The screenshot shows the YouTube upload interface with a yellow header "Uploading to YouTube Third Input Screen". On the left, there's a thumbnail of a video titled "Introduction to Machine Learning". The main area contains several sections:

- Comments:** Options to allow or disallow comments, with a dropdown for comment sorting.
- License and rights:** Options to choose a license (Creative Commons Attribution-NonCommercial-ShareAlike) and add a copyright notice.
- Embedding:** Options to allow embedding and specify where it can be embedded.
- Age restrictions:** Options to enable age restriction and set a minimum age limit.
- Categories:** A dropdown menu for selecting categories.
- Community contributions:** Options to allow users to contribute to the video.
- Reporting data:** A dropdown menu for reporting data.
- Video metrics:** Options to make video statistics available to the audience.
- Custom description:** A text area for a custom video description.

At the bottom right, there are buttons for "Next Step" and "Upload video".

••

University of Southern California 

Business Model: Ads, Ads, Ads

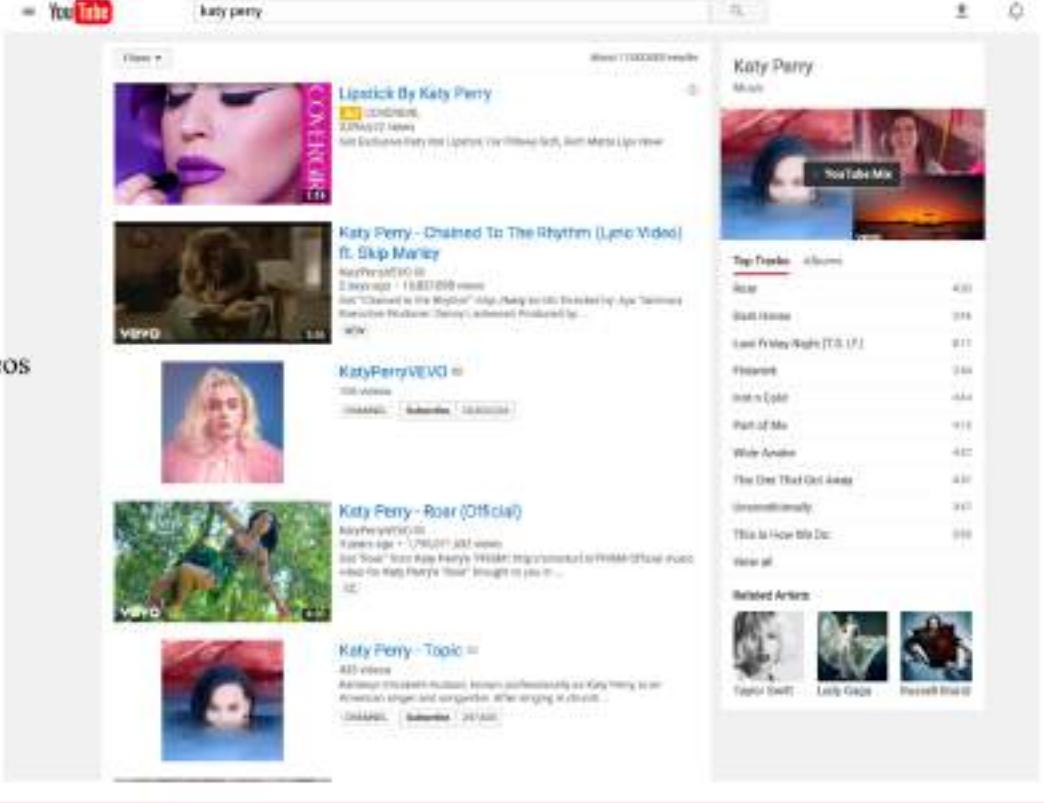
Sample YouTube Search Results for Katy Perry

First result is an Ad

2nd and 4th results are stored at Vevo

3rd and 5th results are links to a Katy Perry channel with 106 videos

To the right is a mix of Katy Perry songs and some “related” artists



Rank	Video Title	Views
1	Lipstick By Katy Perry	1,100,000,000
2	Katy Perry - Chained To The Rhythm (Lyric Video) ft. Skip Marley	2,300,000,000
3	Katy Perry - Roar (Official)	1,700,000,000
4	Katy Perry - Topic	1,000,000,000
5	Katy Perry Music	106

••

University of Southern California

USC Viterbi
School of Engineering

Ranking: Ads, Views, Age YouTube Search Results

Begins with an ad.

The next 4 results are ordered by the number of views: 420,004, 369,979, 228,004.

Subsequent listings are a mixture of highly viewed videos, but older, e.g. Lee 1 MIT has 3 million+ views but is 7 years old.

It is not obvious how the ranking was determined

Technology For Students
221,000 views
How to Be a Viterbi Student: Homecoming Program for Freshmen

Lecture 0 - Introduction to Computer Science I
3,000,000 views
This is a free lecture from the second-year "Introduction to Computer Science" (CS101) hypercourse at MIT. It is intended for students who have completed AP Computer Science A and are interested in learning more about computer science.

Computer Science is good insta?
369,979 views
How much will it cost people to obtain today's most popular tech products in the next decade? And their timelines of 2019 delivery continue...

Computer science is for everyone (Hadi Partovi | TEDxHofstra)
2,000,000 views
The only free open course (CS101) based around a fundamental set of 100+ concepts. Free assignments, free code...

Computer Science vs Self Taught vs Coding Bootcamp (R. Quarmy Lareen)
1,000,000 views
How do you learn computer science? Is it better to self-teach or take a bootcamp?

Computer science education: why does it suck so much and what if it didn't? (Ashley Germi...)
162,000 views
Surveys indicate a lack of job satisfaction among computer science degree holders in the U.S. (1.3 million new jobs per...

Computer science education: why does it suck so much and what if it didn't? (Ashley Germi...)
162,000 views
Computer science education: why does it suck so much and what if it didn't? (Ashley Germi...)

Computer Science Tube
21,000 views
Computer Programming is an interesting field of computation, environment, history, technology, for Computer Programming.

Vlog: What to expect in a Computer Science major
20,000 views
Computer Programming is an interesting field of computation, environment, history, technology, for Computer Programming.

•

University of Southern California

USC Viterbi
School of Engineering

YouTube Advanced Search Ranking Filters

- During a search YouTube provides filters for users to refine their search:
 - UPLOAD DATE
 - TYPE
 - DURATION
 - FEATURES
 - SORT BY

The screenshot shows a YouTube search results page for 'algorithms'. At the top, there's a navigation bar with the USC Viterbi logo and a search bar containing 'algorithms'. Below the search bar is a yellow header bar with the text 'YouTube Advanced Search Ranking Filters'. The main content area displays a table of search filters:

FILTER	DESCRIPTION	OPTIONS	SORT BY
UPLOAD DATE	00:00:00 - 10:00:00	4h	Relevance
Today	Longer (20 minutes)	10h	Upload date
This week	Shorter (10 minutes)	14h	View count
This month	Oldest (1 month)	18h	Created (oldest)
This year	Most recent (1 year)	22h	Rating
			Length
			Published
			Views

Below the filters, three video thumbnails are shown:

- Intro to Algorithms: Crash Course Computer Science #12**
Upload Date: 8/11/2016 | Duration: 10m
- MIT 6.S006 Introduction to Algorithms, Fall 2011**
Upload Date: 9/10/2011 | Duration: 10:00
- John MacCormick: Nine Algorithms That Changed the Future**
Upload Date: 10/10/2011 | Duration: 10m 45s

Copyright Ellis Horowitz, 2011-2022

17

••

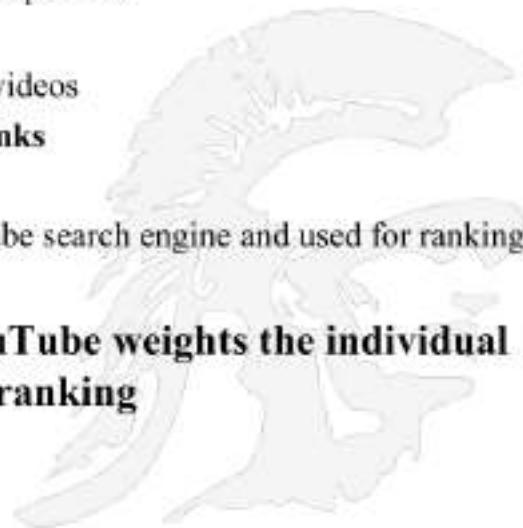
University of Southern California 

USC Viterbi School of Engineering YouTube Ranking Factors

- YouTube uses the following metrics for ranking search results:

1. **Meta Data**
 - video titles, descriptions and tags are core ranking factors
 - include links to a website and social profiles
2. **Video Quality**
 - HD ranks higher than low quality videos
3. **Number of views, likes, shares and links**
4. **Subtitles and Closed Captions**
 - captions are crawled by the YouTube search engine and used for ranking

- **What is not known is how YouTube weights the individual factors to make up their final ranking**



Copyright Ellis Horowitz, 2011-2022

18

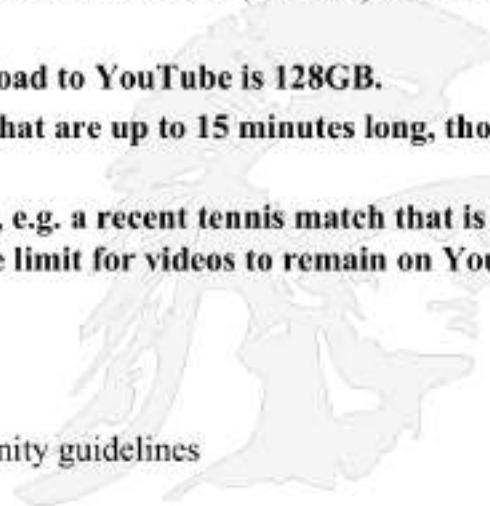
••

University of Southern California 

 USC **Viterbi**
School of Engineering

YouTube Upload Characteristics

- **YouTube Upload Characteristics**
 - YouTube supports 8 video formats for uploading: MOV, MP4 (MPEG4), AVI, WMV, FLV, 3GP, MPEGPS, WebM
 - **Aspect Ratio:** the standard aspect ratios are: 4:3 or 16:9. When the video is uploaded to the site, YouTube will either leave it as-is (for 16:9) or add vertical black bars (for 4:3)
 - **The maximum file size you can upload to YouTube is 128GB.**
 - **By default, you can upload videos that are up to 15 minutes long, though that can be extended**
 - **Many videos have a short life cycle, e.g. a recent tennis match that is soon forgotten, however, there is no time limit for videos to remain on YouTube, unless**
 - You delete the video.
 - You delete your account.
 - You violate copyright or community guidelines



Copyright Ellis Horowitz, 2011-2022

19

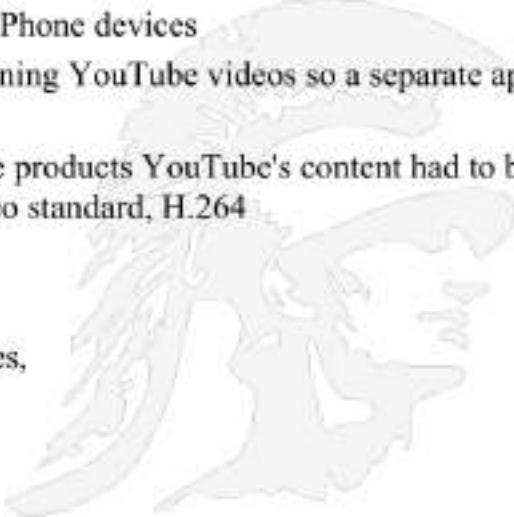
••

University of Southern California 

 USC **Viterbi**
School of Engineering

YouTube Videos Run On Multiple Platforms

- **Desktops/laptops**
 - Videos are played in your browser assuming it supports HTML5
 - This avoided the need to use Adobe Flash Player
- **Smartphones**
 - YouTube apps exist for Android and iPhone devices
 - There is no native support for running YouTube videos so a separate app is required
 - For YouTube's videos to run on Apple products YouTube's content had to be transcoded into Apple's preferred video standard, H.264
- **Other Devices**
 - Apple TV, Fire TV, iPod Touch,
 - TiVo, PlayStation, Wii Game consoles,
 - Xbox Live, Roku Players
 - Google Chromecast



Copyright Ellis Horowitz, 2011-2022

20

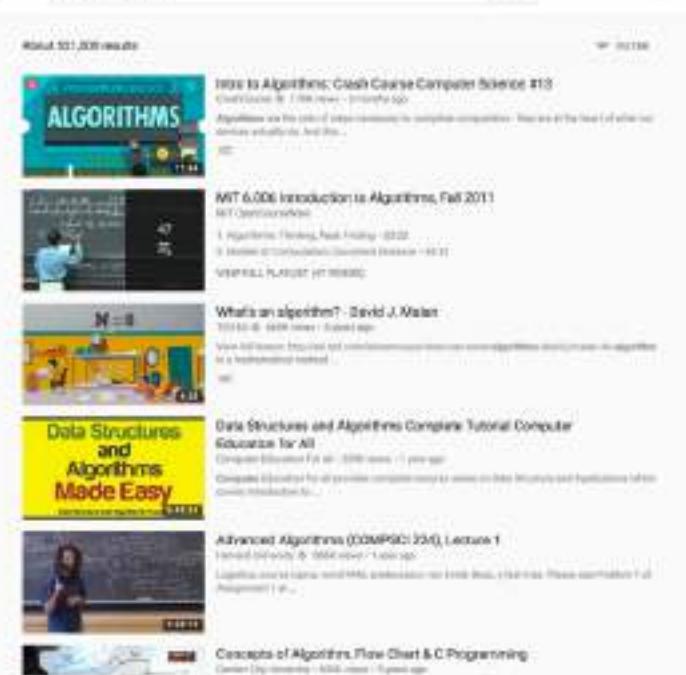
••

University of Southern California  USC 

YouTube Makes Recommendations to Retain Viewers

- YouTube Search Results Example for query "computer algorithms"
- Assume we choose the first result

Recommendations are made to maximize watch time



<https://www.pcnewsonline.com/tech/social-media/algorithms-take-over-youtube-s-recommendations-highlight-human-problem-n867596>

Copyright Ellis Horowitz, 2011-2022

•

University of Southern California  USC 

YouTube Recommendation Algorithm

- Given the query "computer algorithms" followed by a selection, YouTube makes recommendations for subsequent videos
- Recommendations account for 60% of all video clicks



Copyright Ellis Horowitz, 2011-2022

22

••

University of Southern California  USC 

YouTube Recommendation System Uses Graph Properties

- Association Rule Mining
 - For each pair of videos v_i, v_j compute co-visitation counts, i.e. they count how often they were co-watched; if c_{ij} is the co-visitation count, then relatedness is defined as

$$r(v_i, v_j) = \frac{c_{ij}}{f(v_i, v_j)}$$
 where c_i and c_j are the total occurrence counts across all sessions for videos v_i and v_j . $f(v_i, v_j)$ is a normalization function that takes the global popularity of both the seed video and the candidate video into account; e.g. $f(v_i, v_j) = c_i * c_j$

The set of related videos, R_i for a given seed video v_i is determined by taking the top N candidate videos ranked by their scores $r(v_i, v_j)$

Related videos induce a directed graph over the set of videos, namely:
 For each pair of videos (v_i, v_j) , there is an edge e_{ij} from v_i to v_j iff v_j is in R_i

For details see: *The YouTube Recommendation System*
<http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.434.9301&rep=rep1&type=pdf>
Copyright Ellis Horowitz, 2011-2022 23

Here is the paper referenced above.

••

University of Southern California 

 USC **Viterbi** School of Engineering **Media Sites (including YouTube)
Move Away from False Information**

- **YouTube's recommendation algorithm used to send people to misinformation, e.g. see**
 - <https://www.youtube.com/watch?v=Fl8tFmBIPak> (3 min)
 - <https://www.wsj.com/articles/how-youtube-drives-viewers-to-the-internets-darkest-corners-1518020478>
- **As a result YouTube changed its recommendation algorithm in 2020, eliminating so-called “fringe” sites**
 - <https://www.nytimes.com/2020/11/03/technology/youtube-misinformation-fox-news.html>
- **There is a website that tracks the recommendations of Youtube,**
- **<https://algotransparency.org/>**
- “We used a multi-step program to analyze videos recommended by its algorithm every day”
 - Step 1: We start from a list of 1000+ US channels (these channels are listed below)
 - Step 2: We gather all recommended videos from the last video uploaded by these channels
 - Step 3: We compute which channel was recommended the most from those videos
 - Step 4: We repeat step 2 and 3 until we have gathered recommendations from 2000 channels
 - Step 5: For each video that was observed, we count and display from how many channels it was recommended

Copyright Ellis Horowitz, 2011-2022

24

••

University of Southern California 

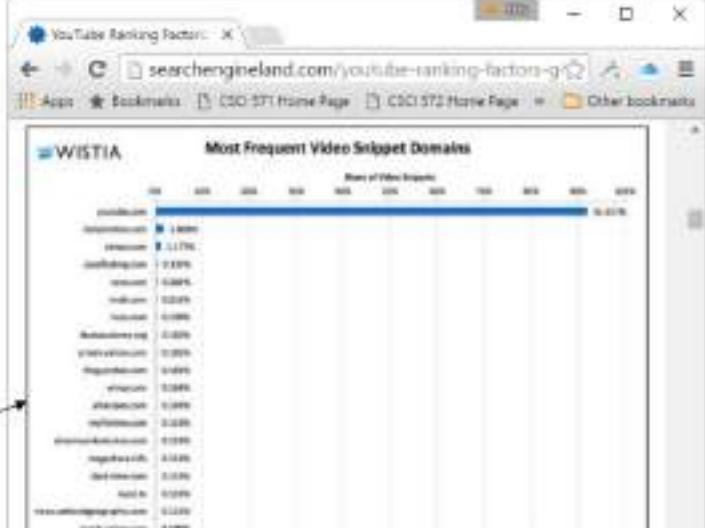
Google Search is Biased Towards YouTube Videos

A video rich snippet means that when someone searches for something on Google, you can have a small tiny video show up next to your result to let the user know that particular result (yours) has a video to help.

Google weeded out the video competition in Web search by predominantly displaying **only video-rich snippets** for YouTube videos back in 2014.

Here is a graph outlining the percentage share of video-rich snippets in Google; 91% are from YouTube

see
<https://wistia.com/blog/where-did-my-video-snippets-go>



Domain	Share of Video Snippets
YouTube	91.0%
Wistia	1.17%
vimeo	1.03%
viddler	1.02%
imovier.com	1.02%
imovier.net	1.02%
dailymotion	1.02%
videoblocks.net	1.02%
videoblocks.com	1.02%
imgur.com	1.02%
imgur.net	1.02%
imgurusercontent.com	1.02%
imgurusercontent.net	1.02%
imgurstatic.com	1.02%
imgurstatic.net	1.02%
imgurstaticusercontent.com	1.02%
imgurstaticusercontent.net	1.02%

Source: [Wistia](#)

e.g. try "tutorial on bitcoin"

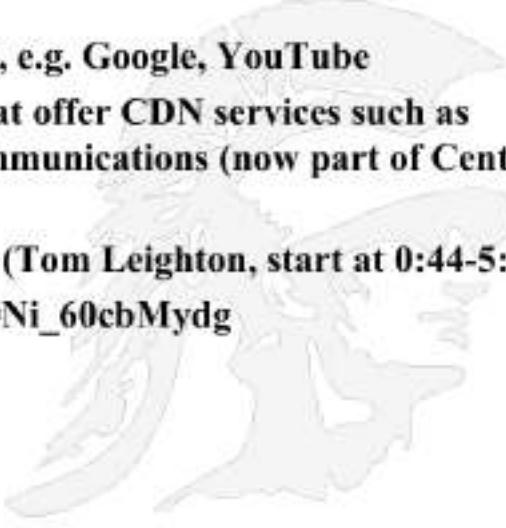
Copyright Ellis Horowitz 2011-2022

••

University of Southern California 

 USC **Viterbi**
School of Engineering **Content Delivery Networks**

- A content distribution network (CDN) consists of a large set of content servers and a means for dynamically selecting servers based on knowledge of the location of the user and possibly the content being requested
- Some sights operate their own CDN, e.g. Google, YouTube
- There are third party companies that offer CDN services such as Akamai, Limelight and Level 3 Communications (now part of Century Link)
- See the Akamai video for 5 minutes (Tom Leighton, start at 0:44-5:00),
- https://www.youtube.com/watch?v=Ni_60cbMydg



Copyright Ellis Horowitz, 2011-2022

26

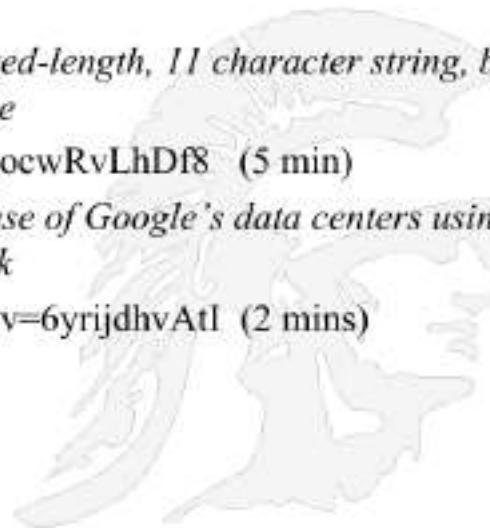
••

University of Southern California 

 USC **Viterbi**
School of Engineering

YouTube Video Delivery System

- **Two Critical Technology Challenges for YouTube:**
 - *how to identify billions of videos*
 - *How to efficiently deliver the video to the desktop/mobile device*
- **The Solutions:**
- **Identification:** *YouTube assigns a fixed-length, 11 character string, base 64, unique identifier to each video, see*
 - <https://www.youtube.com/watch?v=goewRvLhDf8> (5 min)
- **Efficient Delivery:** *YouTube makes use of Google's data centers using them as a content distribution network*
 - <https://www.youtube.com/watch?v=6yrijdhvAtI> (2 mins)



Copyright Ellis Horowitz, 2011-2022

27

••

University of Southern California 

 USC **Viterbi**
School of Engineering

YouTube (Google's) Content Delivery Datacenters

- A map of Google's data centers, see
- <https://www.google.com/about/datacenters/inside/locations/index.html>



Figure 4: Geographical distribution of YouTube Video Cache Locations.

Copyright Ellis Horowitz, 2011-2022

28

••

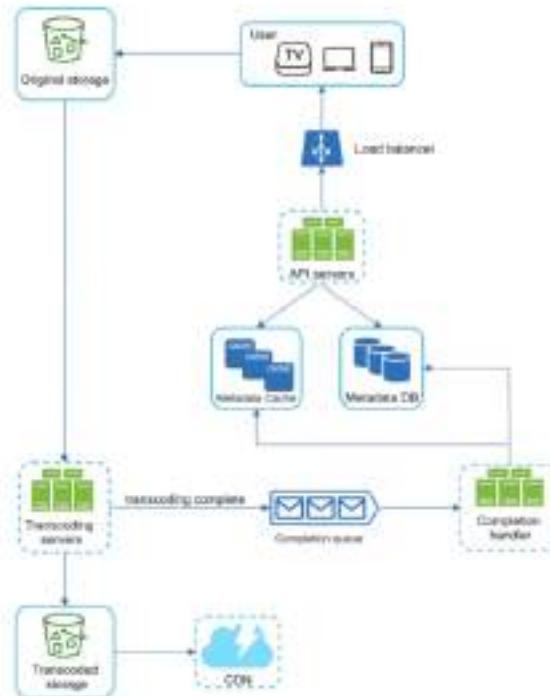
University of Southern California

USC **Viterbi**
School of Engineering

Uploading a YouTube Video

1. videos are uploaded from a desktop to a central Data Center
2. the video is then transcoded into multiple formats
3. transcoded copies are sent to the Content Distribution Network

Video transcoding is a technique of converting a video into multiple different formats and resolutions to make it playable across different devices and bandwidths. The technique is also known as *video encoding*. This enables YouTube to stream videos in different resolutions such as *144p, 240p, 360p, 480p, 720p, 1080p & 4K*.



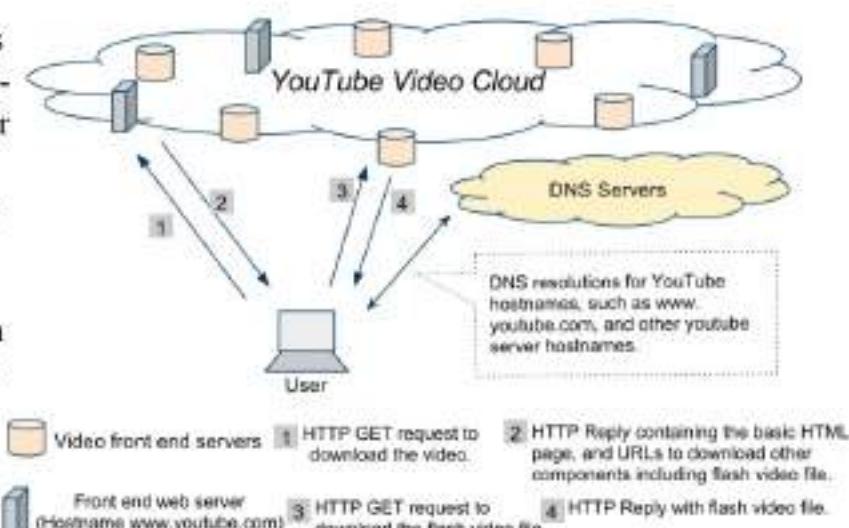
Copyright Ellis Horowitz 2011-2022

••

University of Southern California 

USC Viterbi YouTube's Content Distribution Network Downloading a YouTube Video

A local DNS server resolves www.youtube.com and is redirected to a YouTube server which downloads the page information and a pointer to a YouTube server that can deliver the video, e.g. v23.lscache5.c.youtube.com. The request to v23.lscache5 may be further resolved.



4 steps describing the delivery of a YouTube video

<http://www-users.cs.umn.edu/~zhzhang/Papers/youtube-tech-report.pdf>

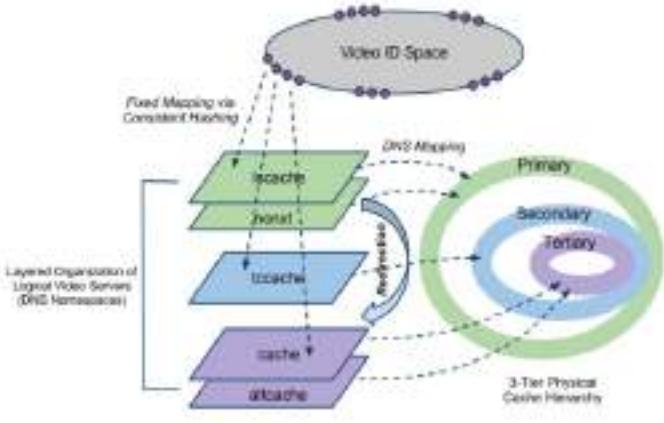
Copyright Ellis Horowitz 2011-2022

••

University of Southern California  USC 

YouTube Delivery System

- The design of the YouTube video delivery system consists of three components:
 - a “flat” video id space,
 - a multi-layered logical server organization consisting of five anycast namespaces (and two unicast namespaces), and
 - a 3-tiered physical cache hierarchy with (at least) 38 primary locations, 8 secondary and 5 tertiary locations.



The diagram illustrates the YouTube Architectural Design. It shows the layered organization of logical video servers (DNS Namespace) and the 3-tier physical cache hierarchy.

Layered Organization of Logical Video Servers (DNS Namespace):

- Video ID Space:** A large oval containing numerous small dots representing video IDs.
- Fixed Mapping via Consistent Hashing:** Dashed arrows point from specific video IDs in the Video ID Space to specific servers in the logical namespace.
- Logical Namespace:** A stack of four horizontal layers:
 - cache:** Top layer, green.
 - host1:** Second layer, green.
 - host2:** Third layer, blue.
 - host3:** Bottom layer, purple.
- DNS Mapping:** A circular arrow labeled “DNS Mapping” connects the logical namespace layers to the 3-Tier Physical Cache Hierarchy.

3-Tier Physical Cache Hierarchy:

- Primary:** The outermost green circle.
- Secondary:** The middle blue circle.
- Tertiary:** The innermost purple circle.
- 3-Tier Physical Cache Hierarchy:** Labels at the bottom right indicating the three levels of the cache hierarchy.

Figure 3: YouTube Architectural Design.

<https://www-users.cse.umn.edu/~zhang089/Papers/youtube-tech-report.pdf>

••

University of Southern California 

 USC **Viterbi**
School of Engineering

References to YouTube's CDN

- There are four research papers that investigated and discussed the YouTube CDN, they are:
 1. *Vivisecting YouTube: An Active Measurement Study*, 2012, cited by Jefay
 2. *Dissecting Video Server Selection Strategies in the YouTube CDN*, 2011, cited by Jefay
 3. *YouTube Traffic Dynamics and Its InterPlay with a Tier-1 ISP*, 2010
 4. <https://www-users.cse.umn.edu/~zhang089/Papers/youtube-tech-report.pdf>
- All of the papers describe a complicated re-direction scheme to find the nearest data center to serve the video; they attempt to minimize Round Trip Time or RTT
- For rarely-called-for videos the “*Dissecting*” paper did a study requesting in California a rare video and observed that the first request came from the Netherlands, but future requests were served from California
- - Conclusion: videos are constantly being moved around to be closer to the place that is requesting them

Copyright Ellis Horowitz, 2011-2022

32

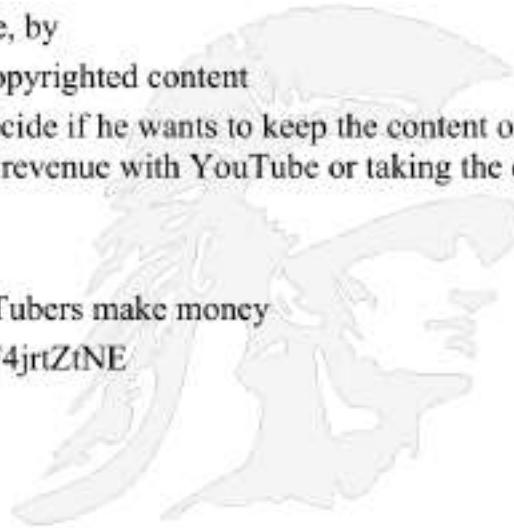
••

University of Southern California

USC **Viterbi**
School of Engineering

Monetizing YouTube

- **YouTube challenges in the early days**
 - YouTube had no way of making money and its infrastructure is very expensive
 - YouTube was being sued by content creators as many of YouTube's videos were uploaded illegally
 - YouTube **solved both problems** at once, by
 - Developing a system for spotting copyrighted content
 - Allowing the copyright owner to decide if he wants to keep the content on the site and let ads appear, splitting the revenue with YouTube or taking the content down
 - Here is a video that describes how YouTubers make money
 - <https://www.youtube.com/watch?v=v8F4jrtZtNE>
 - (8 min)



Copyright Ellis Horowitz, 2011-2022

33

••

University of Southern California 

 USC **Viterbi**
School of Engineering

ContentID

- YouTube's solution was to create a fingerprint database of copyrighted content, called Content ID
- YouTube solicited cooperation from content owners asking them to submit copies of their content so YouTube could fingerprint them
 - There are millions of reference files in YouTube's Content ID database.
- When a new video is uploaded, it is immediately checked against the database, and the video is flagged as a copyright violation if a match is found.
- When this occurs, the content owner has the choice of
 1. blocking the video to make it unviewable,
 2. tracking the viewing statistics of the video, or
 3. adding advertisements to the video



<https://arstechnica.com/tech-policy-policy/2014/10/youtube-has-paid-1-billion-to-rights-holders-via-content-id-since-2007/>

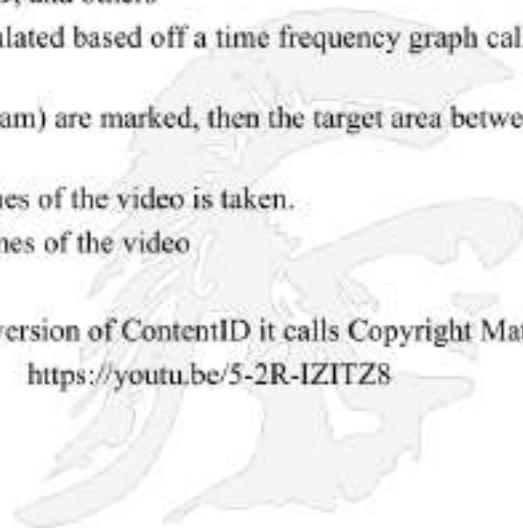
Copyright Ellis Horowitz, 2011-2022

34

••

University of Southern California 

USC Viterbi School of Engineering More Details on ContentID



1. Content ID is based off audio and video samples that rights holders have uploaded to YouTube
2. User uploads a video.
3. YouTube then queues up the video to be processed i.e. it is transcoded into multiple formats including:
 - HTML5, H.264, WebM VP8, HD, non-HD, and others
4. *If the video contains audio*, a hash is then calculated based off a time frequency graph called a spectrogram.
 - Target zones (peak points in the spectrogram) are marked, then the target area between them is also taken and hashed
5. *For the video portion*, a sample section of frames of the video is taken.
 - A hash is created from those sampled frames of the video

- Note recently YouTube has introduced a new version of ContentID it calls Copyright Match
- See the following videos for details, (2 min). <https://youtu.be/5-2R-IZITZ8>

Copyright Ellis Horowitz, 2011-2022

35

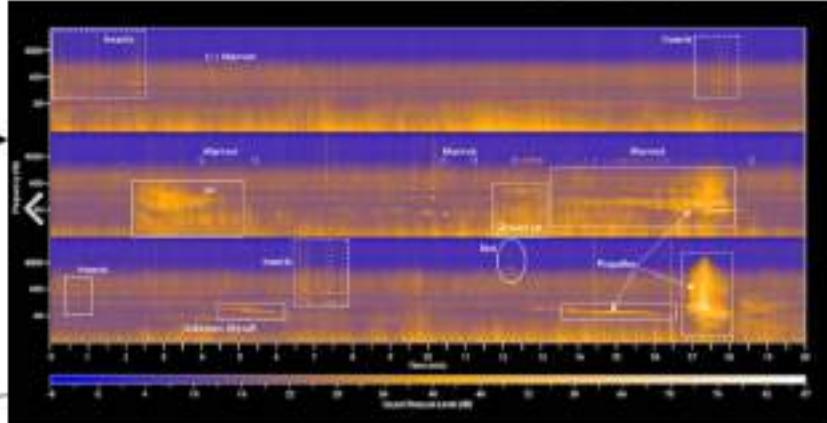
••

University of Southern California  USC

USC Viterbi
School of Engineering

Creating an Acoustic Fingerprint

- The audio signal is digitized and converted to a spectrogram – a time-frequency graph**
 - The graph below plots three dimensions of audio: frequency versus amplitude versus time
 - A common format is a graph with two dimensions: one axis represents time, and the other axis represents frequency; a third dimension indicating the amplitude of a particular frequency at a particular time is represented by the intensity or color of each point in the image.



frequency →

Time axis →

36

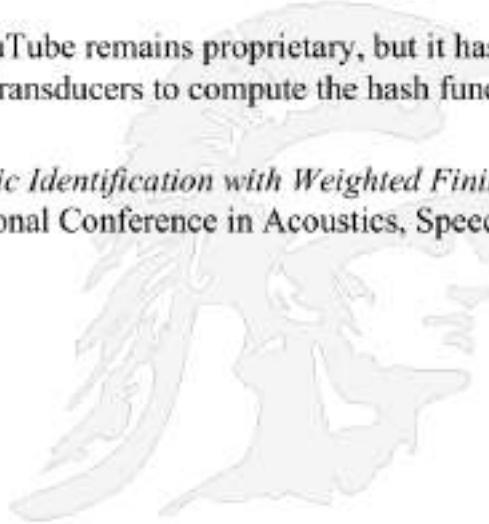
••

University of Southern California

USC **Viterbi**
School of Engineering

How Good is Content ID

- According to stats released by YouTube **99.5 percent** of all copyright issues specifically related to sound recordings are automatically resolved by Content ID
- In addition to music, Content ID also identifies 98% of copyright claims, including those tied to film, TV, gaming
- The actual hashing algorithm used by YouTube remains proprietary, but it has been suggested that YouTube uses finite-state transducers to compute the hash function, e.g. see
- Eugene Weinstein, Pedro J. Moreno; *Music Identification with Weighted Finite-State Transducers*, Proceedings of the International Conference in Acoustics, Speech and Signal Processing (ICASSP), 2007



Copyright Ellis Horowitz, 2011-2022

37

••

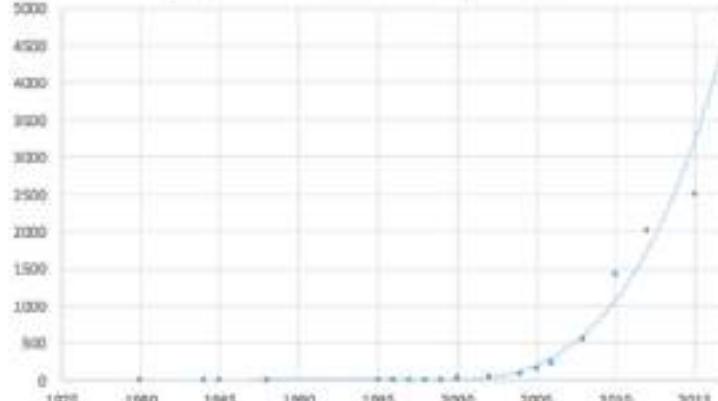
University of Southern California  USC

 USC **Viterbi**
School of Engineering

Will YouTube Ever Run Out of Storage

- The storage you can buy with \$100 has grown exponentially — or equivalently, the cost of storing 1GB of videos has decreased exponentially

Gigabytes of storage you can buy with \$100



Year	Gigabytes of storage (\$100)
1970	~10
1980	~10
1990	~10
2000	~100
2005	~200
2010	~1500
2011	~2500

Kryder's Law
https://en.wikipedia.org/wiki/Mark_Kryder#Kryder%27s_law_projection



Copyright Ellis Horowitz, 2011-2022

38

••

University of Southern California 

 USC Viterbi
School of Engineering <https://www.quora.com/What-is-the-total-capacity-of-YouTube-storage>

Total Capacity of YouTube Storage

• An answer by Rasty Turek from Quora	 1280x720	mp4	 download - 59.01 MB
• There is roughly 24TB of new videos uploaded daily	 640x360	mp4	 download - 15.34 MB
• Each video is re-encoded based on pre-selected profiles and each is stored as a separate file	 640x360	webm	 download - 19.07 MB
• Here is his computation:	 400x240	flv	 download - 8.51 MB
	 320x240	3gp	 download - 5.94 MB
	 176x144	3gp	 download - 2.12 MB
	 4k (no audio)	mp4	 download - 297.69 MB

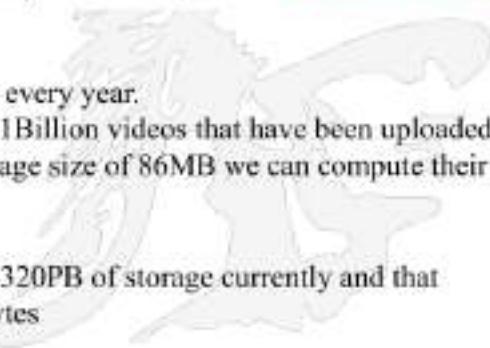
24TB * 4x (for profiles) * 365 days = 35PB/year

So YouTube needs to store roughly 35PB of new data every year.

From multiple sources we know that there is roughly 1Billion videos that have been uploaded to YouTube to date. Assuming each video has an average size of 86MB we can compute their total storage needs as:

86MB * 4 (for profiles) * 1,000,000,000 = 320PB

So it is estimated that YouTube needs to have at least 320PB of storage currently and that the storage needs are growing each year by 35 PetaBytes



Copyright Ellis Horowitz, 2011-2022

39

← 1/53 → * * * 9:08:42

Query formulation

how to ask 'properly'

••

University of Southern California

 USC **Viterbi**
School of Engineering



The Power of Google's Query Box

Copyright Ellis Horowitz 2011-2022

••

University of Southern California  

Traditional Boolean Queries in a Search Box Don't Work As Expected

Query: "apple AND orchard BUT NOT computer" still returns Apple Computer results



Google



Yahoo



Bing

Notice Related searches

Copyright Ellis Horowitz 2011-2022

••

University of Southern California

USC Viterbi
School of Engineering

But Google Advanced Search Permits Boolean Queries

Search engines typically offer an “Advanced Search” page where users **can enter**, in effect, a Boolean query; e.g. Here is Google’s “advanced search” screen

site:www.csie.edu.tw

Find web pages that have...

- all these words: ← ANDing a set of words
- the exact wording or phrase: ← Exact phrase
- one or more of these words: ← ORing a set of words

But don't show pages that have...

- any of these unwanted words: ← NOT words

Read more tools?

- reading level: ← reading level
- Results per page: ← language
- Language: ← file type
- File type: ← etc
- Search within a site or domain: ← See also
<https://help.bing.microsoft.com/#apex/bing/en-US/10002/-1>

Advanced Search

••

University of Southern California

USC Viterbi
School of Engineering

Advanced Search Works Properly

Find pages with...

all these words:

this exact sentence phrase:

any of these words:

none of these words:

near these words:

near these phrases:

Then narrow your results by...

distance:

rating:

last update:

site or domain:

terms & operators:

link from:

file type:

usage rights:

Google

apple orchard -computer

About 64,200,000 results (0.7 seconds)

Apple Orchard - Computer

Rating: Hours:

Little Orchard Music
3.0 (2) · Orchard
Sherman Oaks, CA · (310) 488-2989
"Great place! I enjoyed the Little Orchard Music!"

Apple Farm Collections
No reviews · Farm
Sherman Oaks, CA · (818) 769-1525

Whaley's Orchard
4.4 (50) · Produce Stand
Upland, CA · (909) 466-8454
Open: 8:00am-5:00pm
"They also have the best Pluots, grapes, nectarines, and peaches!"

Copyright Ellis Horowitz 2011-2022

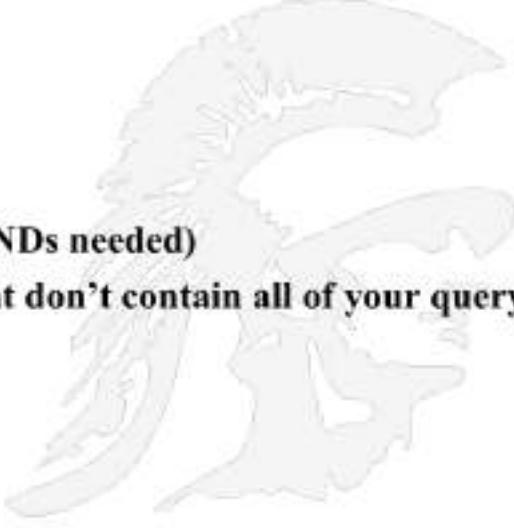
••

University of Southern California 

 USC **Viterbi**
School of Engineering

Query Box Default is AND

- If you search for more than one keyword at a time, Google/Bing will automatically search for pages that contain ALL of your keywords
 - This is called “implicit AND”
- A search for
disney disneyland pirates
is the same as searching for
disney AND disneyland AND pirates
(without the ANDs needed)
- Google sometimes returns pages that don't contain all of your query terms



Copyright Ellis Horowitz, 2011-2022

5

••

University of Southern California  USC

Put Exact Phrases in Quotes

- To search for phrases, just put your phrase in quotes.
- For example,

disney disneyland "pirates of the caribbean"

- This would show you all the pages in Google's index that contain the word **disney** AND the word **disneyland** AND the exact phrase "**pirates of the Caribbean**" (of course, without the quotes)



Copyright Ellis Horowitz, 2011-2022

••

University of Southern California



USC Viterbi
School of Engineering

Differences With/Without Quotes

Query: general principles of electricity

Google GLE "general principles of electricity"

10,100,000 results (0.05 seconds)

General principles of electricity supply systems
www.viterbi.usc.edu/~jdyte/.../General%20principles%20of%20electricity%20supply%20systems.pdf ✓
 The general consists of a prime mover and a magnetic field source. The required field is produced electromagnetically by passing a direct current (DC) through a winding or an iron core, which creates inside three-phase windings on the rotor of the machine. The magnetic field is created by means of a current whose value can ...

General principles of electricity supply systems | EEP | Transmissions
<https://www.jpmoney.com/electricity/transmissions/>
 electric power distribution ... valve, alternating, amperage, biomass, battery, biomass, communication, connection, converter, current, design, distribution, electrical, electricity, energy, equipment, factory, fuel, gas, generation, generator, high-voltage, insulation, insulation, low-volt, link, metal, multi, generator, substation, ...

articles - Mentor EBS
<https://mentor.ebs.schule.aau.at/ARTICLES/> ✓
 To give the answers to this question, we have to know about the general principles of electricity pricing in the electricity market. What we hope to find here is the market price of electricity (i.e. the price at which electricity is sold to consumers by suppliers, not the rate to the end user market). This topic naturally includes two ...

Electricity - Guides turnitin.com
<https://guides.turnitin.com/.../Physics/Library/secondary-education/> ✓
 See 18, 2017 - Preprint: Today you will research electricity and consider some of the methods used to measure it to support different answers. Then, you will read a passage that explains some general principles of electricity. Then you will read an article about what causes an open circuit. Finally, you will read an article ...

PDF C1 Phyton Dept History,T1968CumtouJm.wpd
<https://tinyurl.com/25003211/c1cumtouj1968.pdf> ✓
 PDFs, Images and Audio/Video Measurements (3.3.8 x [reference page]) ... A. A combination of work begun in Phythian-Deas, intended for those who wish to pursue further the theory and practice of precise electric and magnetic measurements. Topics of all the principal instruments used in modern electrical methods are ...

Google GLE "general principles of electricity"

1,100,000 results (0.05 seconds)

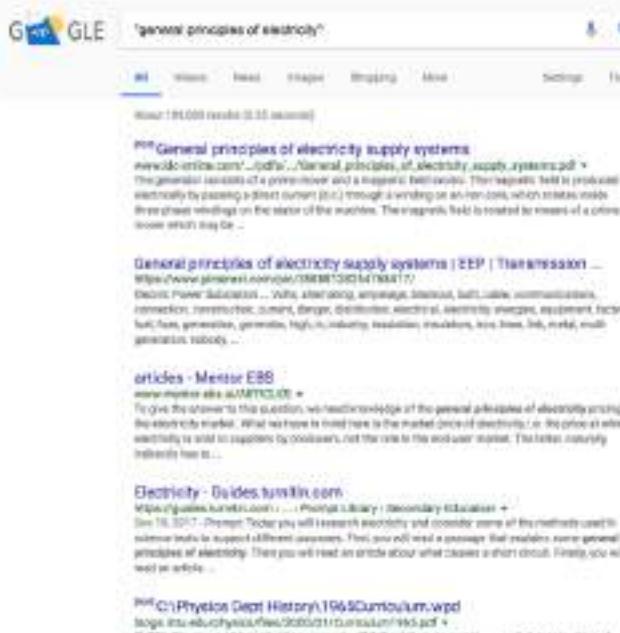
The Principles of Electricity - Energizer
www.energizer.com/learn-about-the-principles-of-electicity/ ✓
 Even materials that conduct electricity resist the flow of electrons. The unit of electrical resistance is an ohm. The pressure needed to move one coulomb per second (one ampere) flow through a conductor having a resistance of one ohm is one volt. ... The quantity of electric charge is measured in coulombs.

People also ask:

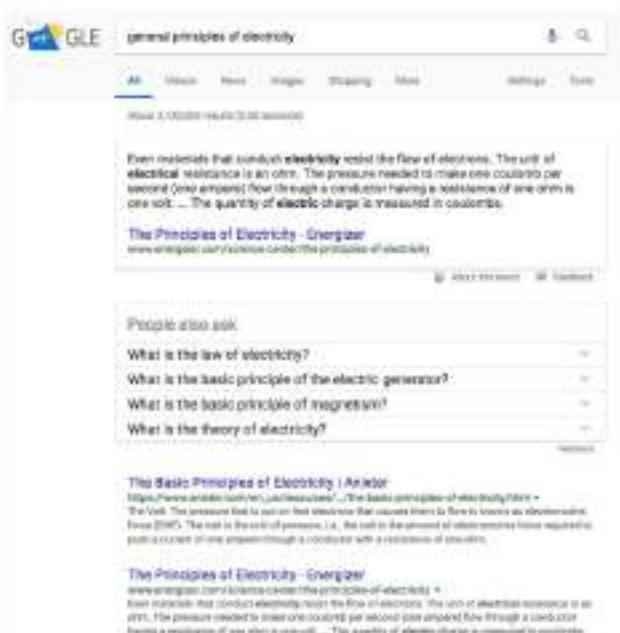
- What is the law of electricity?
- What is the basic principle of the electric generator?
- What is the basic principle of magnetism?
- What is the theory of electricity?

The Basic Principles of Electricity | Ativador
<https://www.viterbi.usc.edu/~jdyte/.../The%20basic%20principles%20of%20electricity.htm> ✓
 The Volt. The tension that is put on hot electrode that causes there to flow its excess as electrons. From [195]. The volt is the unit of pressure, i.e., the volt is the amount of voltage required to push a current of one ampere through a conductor with a resistance of one ohm.

The Principles of Electricity - Energizer
www.energizer.com/learn-about-the-principles-of-electicity/ ✓
 Even materials that conduct electricity resist the flow of electrons. The unit of electrical resistance is an ohm. The pressure needed to move one coulomb per second (one ampere) flow through a conductor having a resistance of one ohm is one volt. ... The quantity of electric charge is measured in coulombs.



With quotes



Without quotes

Copyright Ellis Horowitz 2011-2022

••

University of Southern California 

 USC **Viterbi**
School of Engineering

Google Stemming and Stop Words

- The query [**child bicycle helmet**] finds pages that contain words that are similar to some or all of your search terms,
 - e.g., “child,” “children,” or “children’s,” “bicycle,” “bicycles,” “bicycle’s,” “bicycling,” or “bicyclists,” and “helmet” or “helmets.”
 - Google calls this feature *word variations* or *automatic stemming*
- Google will often ignore Stop Words
 - However, a query with only Stop Words, e.g. [**the who**] gets treated as significant, returning pages for the Rock Group, the Who
- Google limits queries to 32 words
 - Google will return nonsense results for this nonsense query
 - aardvark aback abacus abalone abandon abashed abbey abbreviate abdicate abdomen abduct aberration abhor abide ability abject able abnormal aboard abode abolish abolitionist abort about above abrade abridge abroad abrupt abscond absent absinthe

Copyright Ellis Horowitz, 2011-2022

••



Other Google Query Rules

- **Google favors results that have your search terms near each other**
 - The query [snake grass] finds pages about plants;
 - The query [snake in the grass] finds pages about sneaky people
- **Google gives higher priority to pages that have the terms in the same order as in your query**
- **Google is NOT case sensitive; it shows both upper- and lowercase results**
 - [Red Cross], [red cross], and [RED CROSS] return the same results.
- **Google ignores some punctuation and special characters, including ! ? , . ; [] @ / # < >**
 - Exceptions: C++, or math symbols in Google calculator

••

University of Southern California

USC Viterbi
School of Engineering

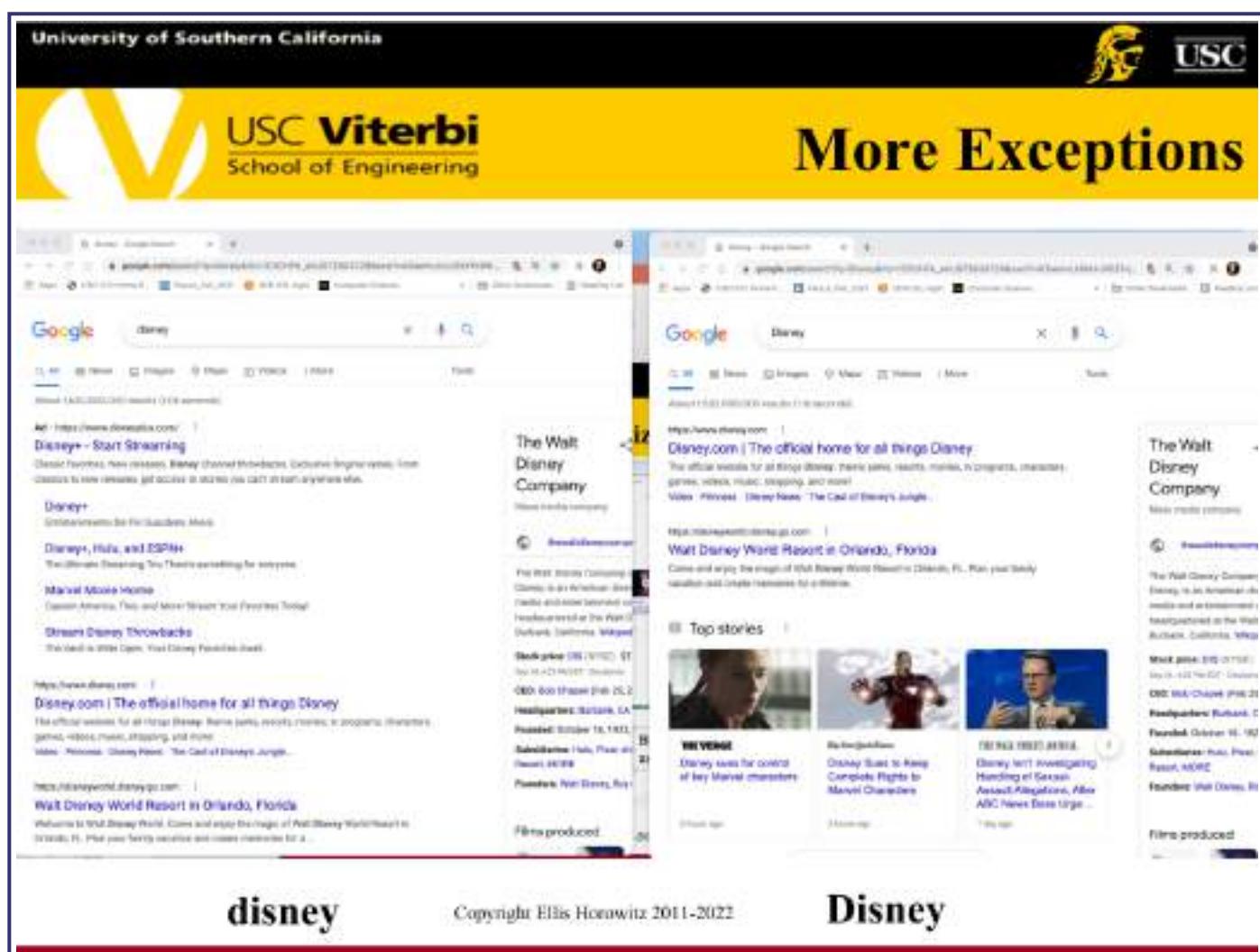
Capitalization Does NOT Matter (but there are exceptions)

bush

Bush

bush and Bush yield the same results as does “apple” and “Apple”

Copyright Ellis Horowitz 2011-2022



••

University of Southern California 

 USC Viterbi
School of Engineering

Boolean OR

- The Boolean OR operator is acceptable in Google queries, placed between keywords, and **OR** is *always* in all caps
- For example,
disney disneyland OR "pirates of the caribbean"
- This would show you pages in Google's index that contain the word **disney** AND the word **(disneyland OR the phrase pirates of the caribbean)**
- OR has higher precedence than AND



Copyright Ellis Horowitz, 2011-2022

12

••

University of Southern California 

 USC **Viterbi**
School of Engineering

How Google/Bing Treat Queries with Boolean Operators

- All query terms are implicitly ANDed
- OR has higher precedence than AND
- Three examples (a, b, c stand for query terms):
 1. a b OR c d is treated as a AND (b OR c) AND d
 2. a OR b c OR d is treated as (a OR b) AND (c OR d)
 3. a OR "b c" d is treated as (a OR ("b c")) AND d
- and see
<https://support.google.com/websearch/answer/2466433?hl=en>
- <https://help.bing.microsoft.com/#apex/bing/en-US/10002/-1>

Copyright Ellis Horowitz, 2011-2022

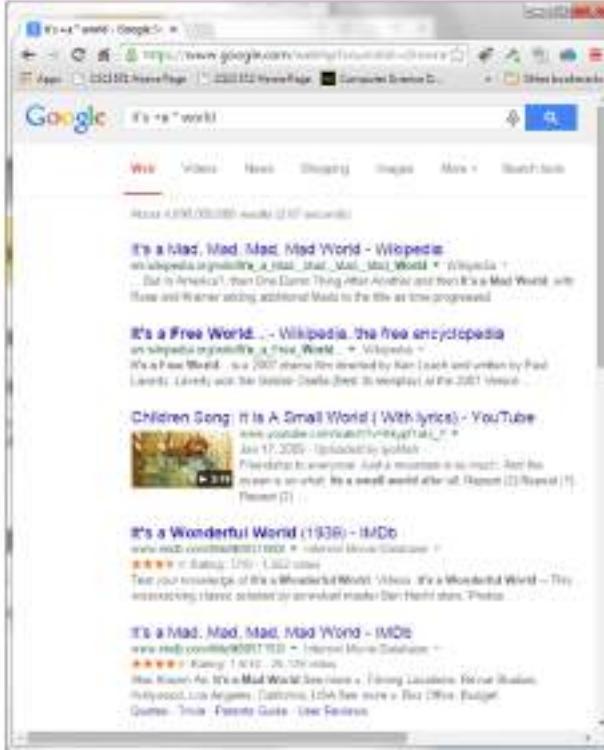
13

••

University of Southern California  USC

Google and Wildcards

- Google offers full-word wildcard queries
- For example, if you search Google for
- **it's +a * world,**
- Google shows you all of the pages in its database that contain the phrase, e.g. “it's a small world” ... and “it's a nano world” ... and “it's a Linux world” ... and so on
- The **+** before **a** is required because it is a stop word and would otherwise be ignored
- This query works the same in Bing
- There must be a space after the **+a**



Copyright Ellis Horowitz, 2011-2022

••

University of Southern California  USC

Google Advanced Operators

Search Service	Search Operators
Web Search	<u>allinanchor:</u> , <u>allintext:</u> , <u>allintitle:</u> , <u>allinurl:</u> , <u>cache:</u> , <u>define:</u> , <u>filetype:</u> , <u>id:</u> , <u>inanchor:</u> , <u>info:</u> , <u>intext:</u> , <u>intitle:</u> , <u>inurl:</u> , <u>links:</u> , <u>related:</u> , <u>site:</u>
Image Search	<u>allintitle:</u> , <u>allinurl:</u> , <u>filetype:</u> , <u>inurl:</u> , <u>intitle:</u> , <u>site:</u>
Groups	<u>allintext:</u> , <u>allintitle:</u> , <u>author:</u> , <u>groups:</u> , <u>insubject:</u> , <u>intext:</u> , <u>intitle:</u>
Directory	<u>allintext:</u> , <u>allintitle:</u> , <u>allinurl:</u> , <u>ext:</u> , <u>filetype:</u> , <u>intext:</u> , <u>intitle:</u> , <u>inurl:</u>
News	<u>allintext:</u> , <u>allintitle:</u> , <u>allinurl:</u> , <u>intext:</u> , <u>intitle:</u> , <u>inurl:</u> , <u>location:</u> , <u>source:</u>
Product Search	<u>allintext:</u> , <u>allintitle:</u>

Query modifiers

- **daterange:**
- **filetype:**
- **inanchor:**
- **intext:**
- **intitle:**
- **inurl:**
- **site:**

Alternative query types

- **cache:**
- **link:**
- **related:**
- **info:**

Other information needs

- **stocks:**

http://www.googleguide.com/advanced_operators_reference.html
 See also
<https://help.bing.microsoft.com/#apex/bing/en-US/10001/-1>

••

University of Southern California **USC**

Searching by Date

The screenshots illustrate the steps to search for 'html' and apply a date range filter:

- Google search for "html".
- Click on "Search Tools" and "Any Time" appears.
- Click on "Any Time" and select "Custom Range".
- A calendar appears, enter dates.
- Final result showing filtered search results.

••

University of Southern California  USC

filetype:



- **filetype:** restricts your results to files ending in a specific file suffix, e.g ".doc" (or .xls, .ppt, etc.), and shows you only files created with the corresponding program
- There can be no space between **filetype:** and the file extension
- The “dot” in the file extension – .doc – is optional
- filetype:doc will NOT return docx

•



inanchor:

- **inanchor:** will restrict the results to pages containing the query terms you specify in the anchor text or links to the page.
- For example,
 - [**restaurants inanchor:gourmet**]
- will return pages in which the anchor text on links to the pages contain the word “gourmet” and the page contains the word “restaurants.”
- **allinanchor:** restricts results to pages containing all query terms you specify in the anchor text on links to the page.

For example, [**allinanchor: best museums sydney**]
 will return only pages in which the anchor text on links to the pages contain the words “best,” “museums,” and “sydney.”

18

•

University of Southern California

USC Viterbi
School of Engineering

intext:

A screenshot of a web browser window showing search results for "pirates intext:"Disney.com". The results are as follows:

- Pirates of the Caribbean | Official Website | Disney pirates.disney.com**
Visit the Pirates of the Caribbean site to learn about the movies, watch video, play games, find activities, meet the characters, license images, and more!
- Movies**
On Stranger Tides - Dead Men's Chest - At World's End -...
- On Stranger Tides**
Visit the Pirates of the Caribbean: On Stranger Tides site to play...
- Games**
Play online games and find fun activities based on the Pirates of...
- Pirates vs. Mermaids - Disney Games**
games.disney.com/pirates-vs-mermaids Disney.com - Choose your battle in Pirates vs. Mermaids! Pirates of the Caribbean - Pirates vs. Mermaids: Choose your battle in Pirates vs. Mermaids! Share...
- Pirates of the Caribbean - Pintel's Conquest | Disney Games**
games.disney.com/pirates-conquest Disney.com - Watch out for pirate traps, and avoid colliding with whirlpools! If the Damage meter is depleted, the battle ends, and you'll have to start over... To conquer a city...

• intext: ignores link text, URLs, and titles, and only searches body text.

• intext: helps you avoid query words that are too common in URLs and links.

• pirates intext:"Disney.com" requires Disney.com to be within the body of the web page

19

•

intitle:

- **intitle:** restricts the results to documents containing a particular word in its title.
- You can also search for phrases. Just put your phrase in quotes
- **intitle:pirates**
- **Intitle:"pirates of the caribbean"**

20

•

University of Southern California

USC Viterbi
School of Engineering

inurl:



- **inurl:** restricts the results to documents containing a particular word in its URL.
- **inurl:disney**
- Results include
- **Disney.go.com**
- **www.disney.de**

••

University of Southern California

USC Viterbi
School of Engineering

site:

Google

masters site:cs.usc.edu

1, 2 3 4 5 6 7 8 9

Search News Images Videos Maps Images Settings Tools

www.cs.usc.edu - Academic Programs | M.S. Program - USC Viterbi | Department of Computer Science
The Master of Science in Computer Science provides intensive preparation in the basic concepts and techniques related to the design, programming, and application ...

www.cs.usc.edu - Academic Programs | M.S. Program | Computer Science (General) - USC Viterbi | Department of ...
The Master of Science in Computer Science provides intensive preparation in the basic concepts and techniques related to the design, programming and ...

www.cs.usc.edu - Academic Programs | M.S. Program | Intelligent Robotics - USC Viterbi | Department of Computer ...
The Master of Science in Computer Science (Intelligent Robotics) equips students in the design, implementation, operation, and application of robots, as well as ...

www.cs.usc.edu - Academic Programs | M.S. Program | Data Science - USC Viterbi | Department of Computer Science
The Master of Science in Computer Science (Data Science) trains students in data science with a solid background in computer science and statistical algorithms...
You've visited this page 2 times - Last visit: 10/10/20

www.cs.usc.edu - Academic Programs | M.S. Program | Software Engineering - USC Viterbi | Department of Computer ...
The Master of Science in Computer Science (Software Engineering) focuses on providing its graduates with skills in software development and design, but also equipping ...

www.cs.usc.edu - Academic Programs | M.S. Program | Game Development - USC Viterbi | Department of Computer ...
The Master of Science in Computer Science (Game Development) prepares students with a strong foundation in computer science.

- **site:** restricts the results to those websites in a domain.
- There can be no space between **site:** and the domain.
- Query is:
- **masters site:cs.usc.edu**

••

University of Southern California 

 USC Viterbi
School of Engineering

Using site:

- You can use **site:** in conjunction with another search term or phrase.
pirates site:disney.com
- You can also use **site:** and negation to exclude sites.
pirates -site:disney.com
- You can use **site:** to exclude or include entire top level domains (and, like with filetype, the dot is optional).
pirates -site:com
pirates site:edu



Copyright Ellis Horowitz, 2011-2022

23

••

University of Southern California 

 USC **Viterbi**
School of Engineering

cache:

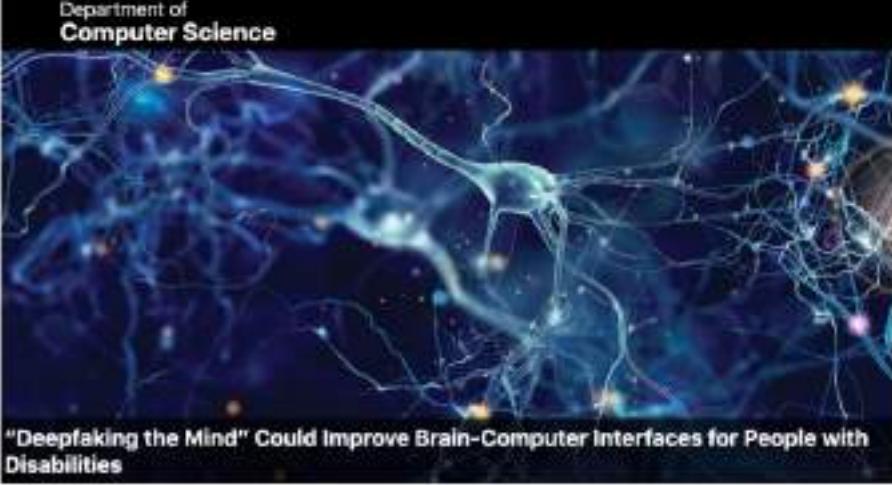
This is Google's cache of <https://www.cs.usc.edu/>. It is a snapshot of the page as it appeared on Feb 16, 2022 08:53:41 GMT. The current page could have changed in the meantime. [Learn more](#).

[Full version](#) [Text-only version](#) [View source](#)

Tip: To quickly find your search term on this page, press Ctrl+F or ⌘-F (Mac) and use the find bar.

≡ USC Viterbi

Department of Computer Science



"Deepfaking the Mind" Could Improve Brain-Computer Interfaces for People with Disabilities

24

•

University of Southern California  USC

link:



- **link:** restricts the results to those web pages that have links to the specified URL.
- **link:Disney.com**
- Note: apparently the link operator only returns a sample of web pages pointing to the link

•



The screenshot shows a Google search results page. The query entered is "related: https://www.cs.usc.edu". The results are displayed in a grid format:

- USC Viterbi School of Engineering**: A large yellow banner at the top features the USC logo and the text "related:".
- CS | Computer Science**: <https://www.cs.usc.edu/>
- Department of Computer Science, Columbia University | Home**: <http://www.cs.columbia.edu/>
- Donald Bren School of Information and Computer Sciences**: <http://www.cs.uci.edu/>
- Computer Science - Computer Science**: <http://www.cs.ubc.ca/~cs/>
- Stanford Computer Science**: <http://www-cs.stanford.edu/>
- Computer Science Department at Princeton University**: <http://cs.princeton.edu/>
- Computer Science and Engineering: Welcome to CSE @ UCR**: <http://www.cs.ucr.edu/>
- UCSB Computer Science**: <http://www.cs.ucsb.edu/>

To the right of the search results, there is a list of bullet points:

- **related: lists web pages that are "similar" to a specified web page.**
- **There can be no space between related: and the URL.**

A small number "26" is visible in the bottom right corner of the search results area.

••



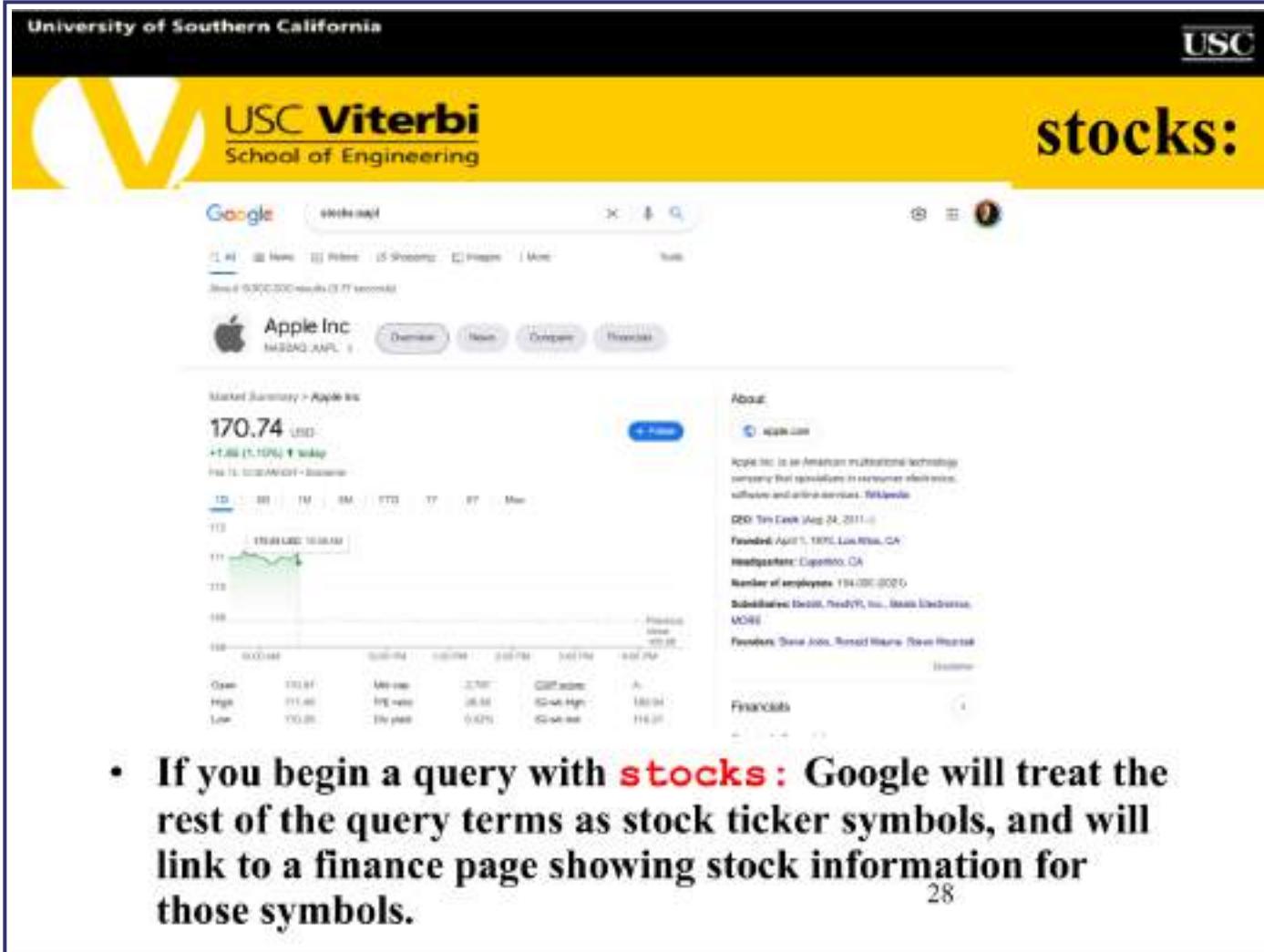
The screenshot shows a Microsoft Edge browser window. The address bar contains the URL "https://www.google.com/search?rlz=1C1GCEU_enUS882US882&q=disney.com". The search results page for "disney.com" is displayed, with the first result being the official Disney website. A red box highlights the word "info:" in the top right corner of the slide, which corresponds to the "Info" button in the Google search interface.

- **info:** presents some information that Google has about a particular web page.

••

University of Southern California  USC

stocks:



The screenshot shows a Google search results page with the query "stocks". The top result is a finance page for Apple Inc. on Google Finance. The page displays the stock price of \$170.74 USD, a chart showing price movement from January 1 to April 1, and financial data including Open, High, and Low values.

About:

- CEO: Tim Cook (Aug 24, 2011-)
- Founded: April 1, 1976, Los Altos, CA
- Headquarters: Cupertino, CA
- Number of employees: 134,000 (2020)
- Subsidiaries: Disney, Pixar, Inc., Beats Electronics, MORE
- Founders: Steve Jobs, Ronald Wayne, Steve Wozniak

28

- If you begin a query with **stocks**: Google will treat the rest of the query terms as stock ticker symbols, and will link to a finance page showing stock information for those symbols.

••

University of Southern California 

 USC **Viterbi**
School of Engineering

Even More Special Features of the Google Query Box

- **Math expressions are evaluated:** $12 + 34 + 10 * (150 / 7) = 260.285714$
- **Dictionary definitions:** define:antidisestablishmentarianism
- Put @ in front of a word to search social media. For example: @twitter.
- Put \$ in front of a number and search for a price. For example: camera \$400.
- Put .. between two numbers and search in a range.
For example, camera \$50..\$100.
- Put a valid tracking number from FedEx or UPS and it will take you to the tracking site
- Put a valid airline and flight number and it will give you its status
- Put a tilde, ~car repair and it queries on ALL synonyms of car, like auto
 - See the following links to further discussion of Google's operators
 - <http://www.google.com/support/websearch/bin/answer.py?hl=en&answer=136861>
 - http://www.googleguide.com/advanced_operators_reference.html
 - <http://searchengineland.com/google-power-user-tips-query-operators-48126>

Copyright Ellis Horowitz, 2011-2022

29

••

Even More Things One Can Do with Google

Google X

11 1M 12 News 12 Images 12 Video 12 Maps 12 Trends Settings Tools

10 results (2.01 seconds)

53,460,490,000 +
fifty-three billion four hundred ninety-three
million four hundred thirty-nine thousand
five hundred thirty-one

Speaking/writing out a number

Google X

Mortgage calculator

Home loan amount	Interest rate (%)	Mortgage term (years)
\$ 100,000	3.00	30

Estimated monthly payment
Mortgage costs

Estimated monthly payment
\$479

Mortgage calculator

Google X

11 1M 12 News 12 Images 12 Video 12 Maps 12 Trends Settings Tools

About 1,940,000,000 results (0.78 seconds)

What is my IP

172.248.32.30
Your public IP address

[Learn more about IP addresses](#)

What is my IP address

Google X

11 1M 12 News 12 Images 12 Video 12 Maps 12 Trends Settings Tools

About 14,400,000 results (0.78 seconds)

Spinners

Spinners for Games

•

University of Southern California

USC Viterbi
School of Engineering

A Failed Google Experiment - phonebook: operator



The screenshot shows a Google search results page for the query "phonebook". The results include various links related to phonebooks, such as "Phonebook", "Business Phonebook", "Residential Phonebook", and "Similar Pages (Google's Best)". The results are described as being from "Google News" and "Google Books".

- There were actually three different Google phonebook operators.
- **phonebook:** searches the entire Google phonebook.
- **rphonebook:** searches residential listings only.
- **bphonebook:** searches business listings only

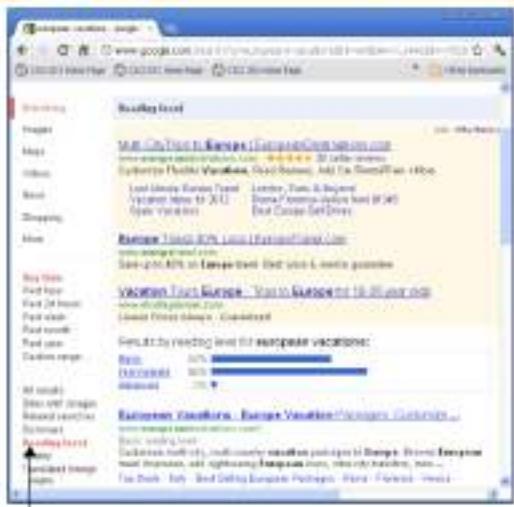
**As of 2010, Google's phone book feature has been officially retired.
Both the phonebook: and the rphonebook: search operator have both been dropped due to many complaints about privacy violations**

31

••

University of Southern California  

A Failed Google Experiment - Reading Level Examples



Query: European vacations

The feature is based primarily on statistical models built with the help of teachers. Google paid teachers to classify pages for different reading levels, and then took their classifications to build a statistical model. With this model, they can compare the words on any webpage with the words in the model to classify reading levels.

Google dropped this feature in 2015.



Query: ovarian cancer

Copyright Ellis Horowitz 2011-2022

••

University of Southern California

USC Viterbi
School of Engineering

A Failed Google Experiment - The Wonder Wheel

Google puts up a wheel of possible interpretations including the jaguar car and the animal; clicking on the spokes of the wheel bring up a refined wheel which eventually lead to a modified query

In 2009 Google introduced the Wonder Wheel, a flash-based interface

In 2011 Google removed the Wonder Wheel but provided no concrete explanation for why it did so;

In 2012 Google restored the wonder wheel, renaming it the Contextual Targeting Tool

In 2014 it was re-focused to help advertisers chose their keywords

In 2011 Google removed the Wonder Wheel

Copyright Ellis Horowitz, 2011-2022

33

••

University of Southern California  USC

A Failed Google Experiment - Google Code Search

Google Code Search

From Wikipedia, the free encyclopedia

Not to be confused with Google Code.

Google Code Search was a free beta product from Google which debuted in Google Labs on October 9, 2006, allowing web users to search for open-source code on the internet. Features included the ability to search using operators, namely language, package, licensee, and file.

The code available for searching was in various formats including tar.gz, Jar/Jar2, Jar, and zip, CVS, Subversion, git and Mercurial repositories.

Contents [view]

- 1 Regular expression engine
- 2 Discontinuation
- 3 See also
- 4 References
- 5 External links

Google Code Search

Developer:	Google
Initial release:	October 9, 2006
Development status:	Discontinued
Operating systems:	Any (web-based application)
Type:	Code search engine
Website:	www.google.com/codesearch

Regular expression engine [edit]

The site allowed the use of regular expressions in queries, which at the time was not offered by any other search engine for code.^[1] This makes it resemble grep, but over the world's public code. The methodology employed combines a program index with a custom-built, distributed-service-resistant regular expression engine.^[2]

In March 2010, the code of RE2, the regular expression engine used in Google Code Search, was made open source.^[3]

Google Code Search supported POSIX extended regular expression syntax, including back references, collating elements, and collection classes.

Languages not officially supported could be searched for using the file operator to match the common file extensions for the language.

Discontinuation [edit]

In October 2011, Google announced that Code Search was to be shut down along with the Code Search API.^[4] The service remained online until March 2012,^[5] and it saw roughly 40K.

In January 2012, Russ Cox published an overview of history and the technical aspects of the tool, and open-sourced a basic implementation of a similar functionality as a set of standalone programs that can run fast instead of regular expression searches over local code.^[6]

••

The slide shows two screenshots of a web browser. The top screenshot shows the initial search results for 'US 7917480B2' on Google Patents. The bottom screenshot shows the detailed patent document for 'Document compression system and method for use with cokernspace repository' (US 7917480 B2). A callout points to the patent number in the search bar with the text 'patent query using patent number'. Another callout points to the 'Download PDF' button on the right side of the patent document page with the text 'Sample patent results page; Note download'.

Initial Google Patents Page

**Special Content Search Engines
Google Patents**

patent query using patent number

Sample patent results page;
Note download

35

••



The screenshot shows a web browser window with the URL <https://books.google.com>. The page features the Google logo at the top left and a search bar below it. A banner at the top right reads "Special Content Search Engines" and "Google Books". The main content area displays a snippet of text from a scanned book, with the text "Search the world's most comprehensive index of full-text books." and a "My Library" link.

Special Content Search Engines

Google Books

Google Books is a service that searches the full text of books and magazines that Google has scanned, converted to text using optical character recognition (OCR), and stored in its digital database.

Books are provided either by

- publishers and authors, through the Google Books Partner Program, or by
- Google's library partners, through the Library Project.

Controversy:
Google has been criticized for potential copyright violations, and lack of editing to correct the many errors introduced into the scanned texts by the OCR process.

As of October 2015, the number of scanned book titles was over 25 million.

Google estimated in 2010 that there were about 130 million distinct titles in the world, and stated that it intended to scan all of them.

••

The screenshot displays three views of Google Books side-by-side:

- Full View:** Shows two pages of a book with text and images. An arrow points from the text "Full View" below it.
- Limited Preview:** Shows a single page with a large image at the top and text below. An arrow points from the text "Limited Preview" below it.
- Snippet View:** Shows a detailed search results page for "A history of psychology". An arrow points from the text "Snippet View" below it.

A faint illustration of a person reading a book is visible in the background.

<https://www.google.com/googlebooks/library/screenshots.html#books-fullview>

Copyright Ellis Horowitz 2011-2022

••

The screenshot shows a web browser window with the USC Viterbi School of Engineering logo at the top left. The main content area displays a Google Scholar search result for the query "A discrete particle swarm optimization algorithm for geographic map contour reconstruction". The result is from a 2010 paper by Li-Ping Tang, Ali Alirezai, and Saeid Mousavi, published in "Information and Communication Technology and its Applications". Below the search result, there is a quote from Isaac Newton: "Stand on the shoulders of giants".

- **Google Scholar** is a freely accessible search engine that indexes the full text or metadata of scholarly literature across an array of publishing formats and disciplines.
- The Google Scholar index includes most peer-reviewed online academic journals and books, conference papers, theses, dissertations, etc

••

University of Southern California

 USC **Viterbi**
School of Engineering



Relevance Feedback & Query Expansion

Copyright Ellis Horowitz 2011-2022

- After initial retrieval results are presented, allow the user to provide feedback on the relevance of one or more of the retrieved documents
 - Google offers this but not very prominently (see 10 related searches below).



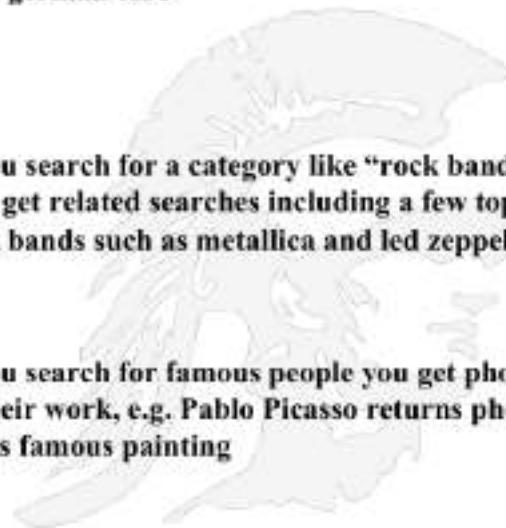
•

University of Southern California 

Google Related Searches



- Google has enhanced their related searches, e.g.
- If you search for the name of a category, Google will show the most popular members, e.g. "german cars"
- If you search for a category like "rock bands" You get related searches including a few top rock bands such as metallica and led zeppelin
- If you search for famous people you get photos of their work, e.g. Pablo Picasso returns photos of his famous painting



Copyright Ellis Horowitz, 2011-2022

••

University of Southern California  USC 

Search Engines and Relevance Feedback




On the query "jaguar" yahoo's 3rd result is the animal; Bing's 2nd result is the animal; all have extensive ads for the car at the top and side indicating that the query for the animal is far rarer; Bing, on the left, puts up alternatives, e.g. Jaguar Luggage, Jacksonville Jaguars; Yahoo also provides related searches but only for the car: bmw, lexus, etc

Bing and Yahoo clearly mark Related Searches

Copyright Ellis Horowitz, 2011-2022

••

University of Southern California

 USC **Viterbi**
School of Engineering



Auto-Completion

Copyright Ellis Horowitz 2011-2022

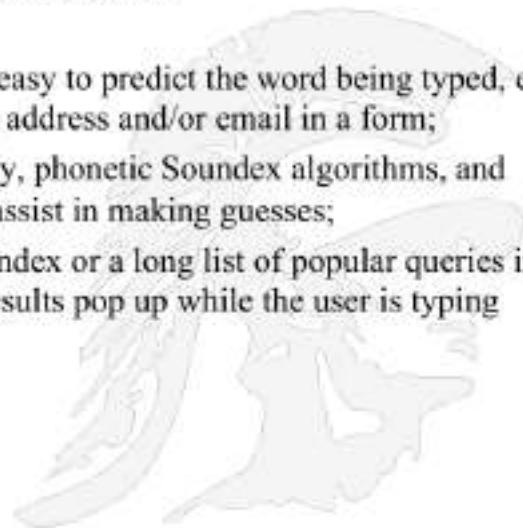
••

University of Southern California 

 USC **Viterbi**
School of Engineering

Auto-Completion

- **Auto-completion is the process of predicting a word or phrase that the user wants to type in without the user actually typing it in completely**
- **Auto-completion is a form of relevance feedback**
 - This feature is effective when it is easy to predict the word being typed, e.g. when a browser fills in your name, address and/or email in a form;
 - Search engines may use past history, phonetic Soundex algorithms, and spelling corrections algorithms to assist in making guesses;
 - The challenge is to search a large index or a long list of popular queries in a very short amount of time so the results pop up while the user is typing



Copyright Ellis Horowitz, 2011-2022

44

•

Google Auto-Completion

- Google has been offering auto-completion since 2008, though it was an experimental feature as far back as 2004.
- Google does automatic completion even after the user enters just the first character
- When the second character is entered a totally different set of possibilities may be offered

orowitz, 2011-2022

•

The screenshot shows a Google search results page. The search query is "anahelm ducks". The first result is a link to "Anahelm Ducks Tickets - Buy Ducks Hockey Tickets Today" from www.ticketmaster.com/Ducks. The second result is a link to "T14Tickets Ducks Tickets - Across from Honda Center" from www.T14Tickets.com. The third result is a link to "Anaheim Ducks" from www.nhl.com/ducks. A red arrow points from the word "anahelm" in the search bar to the first result.

Google Auto-Completion

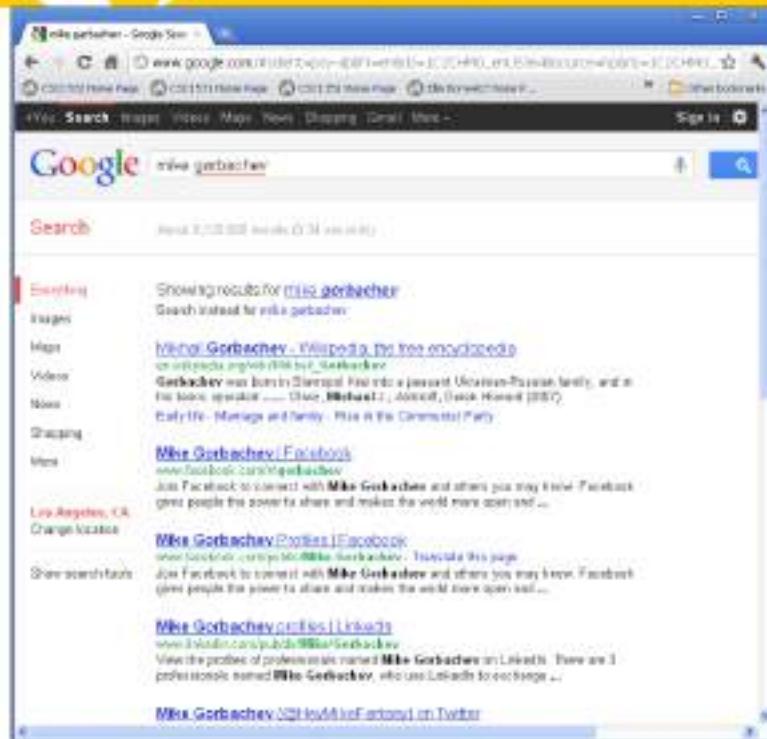
- By the time the fourth character is entered Google has already guessed that the word has been misspelled, and shows this by the squiggly red line; so auto-completion and spelling correction are intertwined

Copyright © 2011-2022 Horowitz, 46

••

University of Southern California  

Google Combines Spelling Correction and Auto Completion



The screenshot shows a Google search results page for the query "mike garbachev". The search bar at the top has "mike garbachev" typed into it. Below the search bar, the results are displayed under the heading "Showing results for **mike garbachev**". The first result is a link to "Mikhail Gorbachev - Wikipedia, the free encyclopedia", which includes a brief summary about Gorbachev's life and political career. The second result is a link to "Mike Gorbachev | Facebook", which is described as a Facebook page for Mike Gorbachev. The third result is a link to "Mike Gorbachev on LinkedIn", which is described as a LinkedIn profile for Mike Gorbachev. The fourth result is a link to "Mike Gorbachev (@lesmike) on Twitter", which is described as a Twitter account for Mike Gorbachev.

After hitting Enter, Google will display results for “Mikhail Gorbachev”, but also provide a link if instead the user actually wanted to search for “mike garbachev”



Copyright Ellis Horowitz, 2011-2022

47

••

University of Southern California  USC 

Yahoo Auto-Completion



- **Yahoo does auto-completion though it starts after typing the *third* character, whereas Google starts on the *first* character**
- **As the user types more characters in this example Yahoo runs out of alternatives and when one more character is added there will be no more auto-complete suggestions**



as Horowitz, 2011-2022

••

University of Southern California

USC Viterbi
School of Engineering

Yahoo Also Offers Spelling Correction

After hitting Enter, Yahoo does offer the correct spelling of “Gorbachev” and returns many results; in case the user actually wanted to search for “garbachev” it provides a link to produce those results, but after clicking Yahoo delivers more results for Gorbachev and none for Garbachev

Copyright Ellis Horowitz, 2011-2022

••

University of Southern California

USC Viterbi
School of Engineering

Bing Auto-Completion

- On the other hand, Bing does not even wait for the first character to be entered, as seen on the left they make use of previous queries and enter some possibilities before the user even types a single character
- After entering “mike garbac” bing finally comes up with the correct spelling, see below

Copyright Ellis Horowitz, 2011-2022

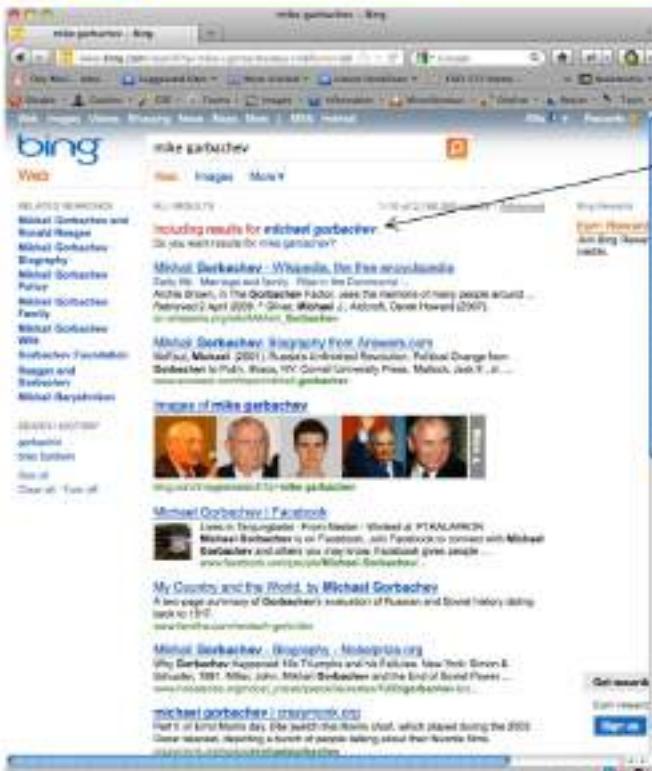
50

••



Bing Final Result with Spelling Corrected

As with Google and Yahoo, Bing will offer results using the corrected spelling and include a link for the user spelling



The screenshot shows the Bing search results for "mike gorbachev". The top result is for "mikhail gorbachev" with a link to "Wiki". Below it are results for "mikhail gorbachev biography", "mikhail gorbachev wikipedia", "mikhail gorbachev facebook", and "mikhail gorbachev biography book". A red arrow points from the text "including results for mikhail gorbachev" to the "mikhail gorbachev" link in the search results.



Copyright Ellis Horowitz, 2011-2022

51

••



Judging the Quality of Answers: Mean Reciprocal Rank (MRR) Scoring

- The mean reciprocal rank is a statistical measure for evaluating any process that produces a list of possible responses to a sample of queries, ordered by probability of correctness.
The **reciprocal rank** of a query response is the multiplicative inverse of the **rank** of the first correct answer
 - The mean reciprocal rank is the average of the reciprocal ranks of results for a sample of queries.
- $$\text{MRR} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{\text{rank}_i}.$$
- For example, suppose we have the following three sample queries for a system that tries to translate English words to their plurals. In each case, the system makes three guesses, with the first one being the one it thinks is most likely correct:

Query	Results	Correct response	Rank	Reciprocal rank
cat	catten, cati, cats	cats	3	1/3
tori	torii, tori , toruses	tori	2	1/2
virus	viruses , virus, viri	viruses	1	1

the mean reciprocal rank as $(1/3 + 1/2 + 1)/3 = 11/18$ or about 0.61.

Copyright Ellis Horowitz, 2011-2022

52

← 1/51 → * * * 9:09:03

MapReduce

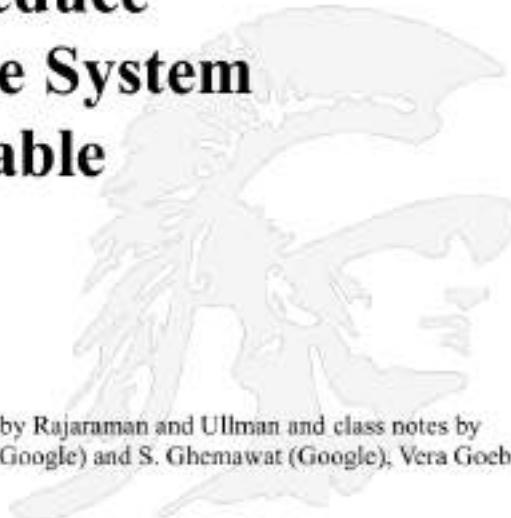
[MapCombineShuffleReduce]
data processing 'at scale'

••

University of Southern California



Google Building Blocks: MapReduce Google File System BigTable



These slides borrow material from Mining Massive Datasets by Rajaraman and Ullman and class notes by Rajaraman (Stanford) and Weld (U. Washington), D. Weld (Google) and S. Ghemawat (Google); Vera Goebel (Univ. of Oslo)

Copyright Ellis Horowitz 2011-2022

••

University of Southern California 

 USC **Viterbi**
School of Engineering

Google Specialized Software Systems

- **Google has built several major software systems for their internal processing**
 1. **MapReduce - an easy way to write and run large-scale jobs on clusters of machines**
 - generate production index data more quickly
 - perform ad-hoc experiments rapidly
 - Dean & Ghemawat. *MapReduce: Simplified Data Processing on Large Clusters*, OSDI, 2004
 2. **GFS (Google File System) a large-scale distributed file system**
 - Ghemawat, Gobioff, & Leung. *Google File System*, SOSP 2003
 3. **BigTable - a semi-structured storage system**
 - online, efficient access to per-document information at any time
 - multiple processes can update per-doc info asynchronously
 - critical for updating documents in minutes instead of hours
 - Chang, Dean, Ghemawat, Hsieh, Wallach, Burrows, Chandra, Fikes, & Gruber. *Bigtable: A Distributed Storage System for Structured Data*, OSDI 2006

Copyright Ellis Horowitz, 2011-2022

2

••

University of Southern California 

 USC **Viterbi**
School of Engineering **Introduction to MapReduce**

- **MapReduce is a methodology for exploiting parallelism in computing clouds (racks of interconnected processors)**
- **It has become a common way to analyze very large amounts of data**
- **MapReduce was initially developed at Google**
- **In 2004 Google was using MapReduce to process 100TB/day of data**
- **Today there are MapReduce Users Groups around the world, see <https://www.meetup.com/topics/mapreduce/>**



Copyright Ellis Horowitz, 2011-2022

3

••

University of Southern California



How is MapReduce Used by Search Engines?

- **At Google:**
 - Building Google's Search Index
 - Article clustering for Google News
 - Statistical machine translation
- **At Yahoo!:**
 - Building Yahoo!'s Search Index
 - Spam detection for Yahoo! Mail
- **At Facebook:**
 - Data mining
 - Ad optimization
 - Spam detection



Copyright Ellis Horowitz, 2011-2022

4

••

University of Southern California  USC

USC Viterbi
School of Engineering

Motivation Beyond Search Engines

- Modern Internet applications have created a need to manage immense amounts of data quickly.
- In many of these applications, the data is extremely regular, and there is ample opportunity to exploit parallelism.
- Some examples

1. Dish network collecting every click of the remote
 - Dish network supplies TV reception via satellite; they collect data on their set top box and send it back to headquarters
2. Tesla collecting every usage of the car
 - Tesla's are connected to the cellular network; the car reports back all of its actions to Tesla

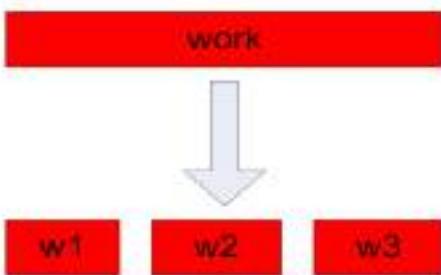
Copyright Ellis Horowitz, 2011-2022

5

••

University of Southern California 

Why Parallelization is Hard

- Parallelization is “easy” if processing can be cleanly split into n units:

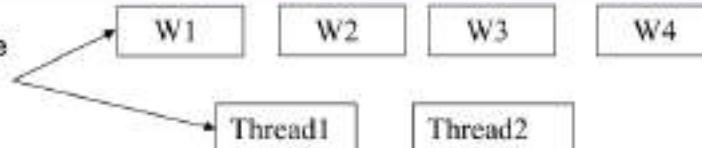
The diagram illustrates a process flow. At the top is a red horizontal bar labeled "work". A large grey arrow points downwards from "work" to three smaller red boxes below it, labeled "w1", "w2", and "w3" from left to right. To the right of this diagram is a faint watermark of a person's face with the text "Partition problem" overlaid.
- Assigned to n workers
- And there is an easy way to combine the outputs

••

University of Southern California  USC 

Why Parallelization is Hard

we would like to have as many threads as there are work units, but this may not be the case



```
graph LR; W1[W1] --- Thread1[Thread1]; W2[W2] --- Thread1; W3[W3] --- Thread2[Thread2]; W4[W4]
```

But there are complicated issues to deal with!

- What if we have more work units than threads?
- How do we assign work units to worker threads?
- How do we aggregate/combine the results at the end?
- How do we know all the workers have finished?
- What if the work cannot be divided into completely separate tasks?
- *MapReduce solves all of these problems so the programmer does not have to deal with them*

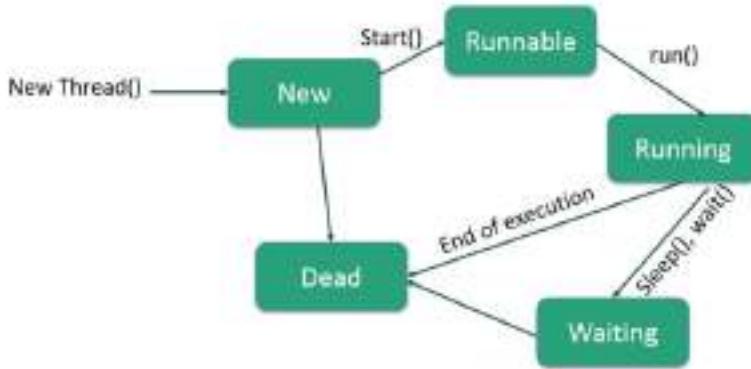
Copyright Ellis Horowitz, 2011-2022

7

••

University of Southern California  USC 

Programming with Multiple Threads Poses Challenges



Thread 1:

```
void foo() {  
    x++;  
    y = x;  
}
```

Thread 2:

```
void bar() {  
    y++;  
    x++;  
}
```

Life Cycle of a Thread

If the initial state is $x = 6$, $y = 0$, what are the final values of x and y after the threads finish running? Possible solutions include: (8,8) and (8,7)

Copyright Ellis Horowitz, 2011-2022

8

••



Multithreaded = Unpredictability

- Many things that look like “one step” operations actually take several steps under the hood:

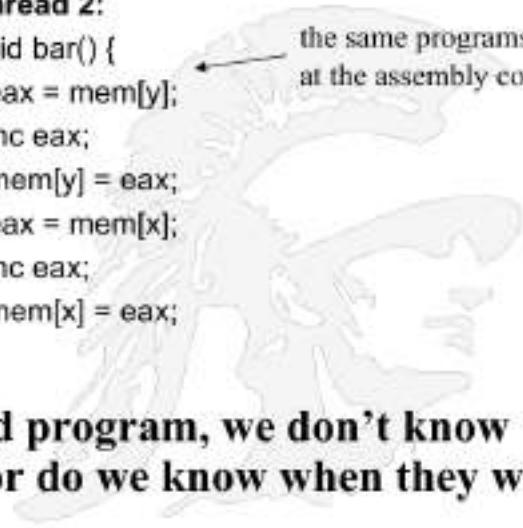
Thread 1:

```
void foo() {  
    eax = mem[x];  
    inc eax;  
    mem[x] = eax;  
    ebx = mem[x];  
    mem[y] = ebx;  
}
```

Thread 2:

```
void bar() {  
    eax = mem[y];  
    inc eax;  
    mem[y] = eax;  
    eax = mem[x];  
    inc eax;  
    mem[x] = eax;  
}
```

the same programs, but
at the assembly code level



- When we run a multithreaded program, we don't know what order threads run in, nor do we know when they will interrupt one another.

••



The “corrected” example

Thread 1:

```
void foo() {  
    sem.lock();  
    x++;  
    y = x;  
    sem.unlock();  
}
```

Thread 2:

```
void bar() {  
    sem.lock();  
    y++;  
    x++;  
    sem.unlock();  
}
```

The global variable `sem`, as defined by

```
Semaphore sem = new Semaphore();
```

guards access to `x` & `y`; semaphores are generally integer variables that are shared between threads; the variable protects the “critical section” from being simultaneously accessed

••

University of Southern California 

USC Viterbi
School of Engineering

Processing Across a Machine Cluster
Introduces Unpredictability on Many Levels

- **Synchronization problems apply to more than just low level operations within a critical section of code**
 - Other issues include:
 - Pulling work units from a queue
 - Assigning work units to an available thread
 - Work units reporting back to the master unit
 - Telling another thread that it can begin the “next phase” of processing

... All require synchronization!



Copyright Ellis Horowitz, 2011-2022

11

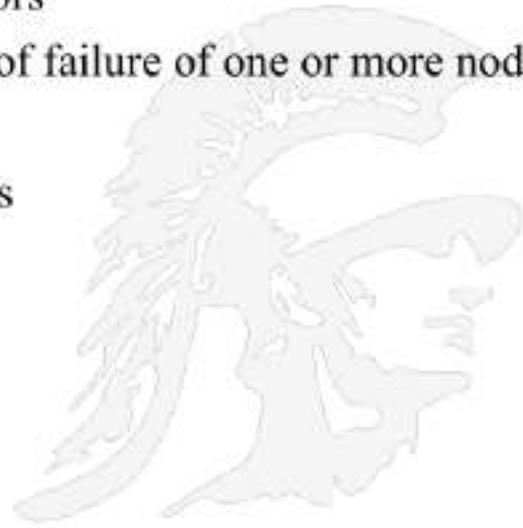
••

University of Southern California 

 USC **Viterbi**
School of Engineering

How MapReduce Solves the Parallelization Problems

- **So MapReduce provides**
 - Automatic parallelization of code across multiple threads and across multiple processors
 - Fault tolerance in the event of failure of one or more nodes
 - I/O scheduling
 - Monitoring & Status updates



Copyright Ellis Horowitz, 2011-2022

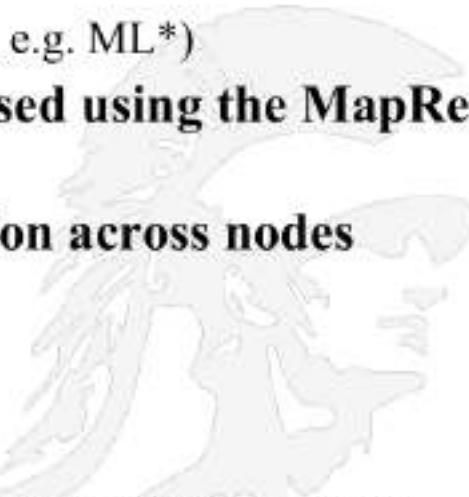
12

••



Map/Reduce - Beginnings

- **Map/Reduce**
 - Is a programming model borrowed from the programming language Lisp
 - (and other functional languages, e.g. ML*)
- **Many problems can be phrased using the MapReduce paradigm**
- **Easy to distribute computation across nodes**
- **Nice retry/failure semantics**



*ML programming language is a general purpose functional programming language, see
[https://en.wikipedia.org/wiki/ML_\(programming_language\)](https://en.wikipedia.org/wiki/ML_(programming_language))

Copyright Ellis Horowitz, 2011-2022

13

..

Here is a quick map(), reduce() example. Run the following, in
<https://www.mycompiler.io/new/python>:

```
Celsius = [39.2, 36.5, 37.3, 37.8]
Fahrenheit = map(lambda x: (float(9)/5)*x +
32, Celsius)
fa = list(Fahrenheit)
print(fa)

Centigrade = map(lambda x: (float(5)/9)*(x-3
2), fa)
print(list(Centigrade))

from functools import reduce
r = reduce(lambda x,y: x+y, [47,11,42,13])
print(r)
```

••

University of Southern California 

 USC **Viterbi**
School of Engineering

What is MapReduce?

- MapReduce is a programming model that generically works this way:
 - A *map function* extracts some intelligence from raw data
 - A *shuffle step* organizes the resulting output
 - A *reduce function* aggregates the data output from the shuffle step
 - Users specify the computation in terms of a *map* and a *reduce* function,
 - Underlying runtime system automatically parallelizes the computation across large-scale clusters of machines, and
 - Underlying system also handles machine failures, efficient communications, and performance issues.

-- Reference: Dean, J. and Ghemawat, S. 2008 "MapReduce: Simplified Data Processing on Large Clusters", *Communication of ACM* 51, 1 (Jan. 2008), 107-113.

Copyright Ellis Horowitz, 2011-2022

15

••

University of Southern California 

 USC **Viterbi**
School of Engineering

The Map/Reduce Paradigm

1. A large number of records are broken into segments
2. **Map:** extracts something of interest from each segment
3. **Group:** sorts the intermediate results from each segment (sometimes called **shuffle**)
4. **Reduce:** aggregates intermediate results
5. Generate final output

Key idea: to re-phrase problems in such a way that the input can be divided into parts and operated on in parallel and the results combined to produce a solution to the original problem

Copyright Ellis Horowitz, 2011-2022

16

••

University of Southern California 

 USC **Viterbi**
School of Engineering

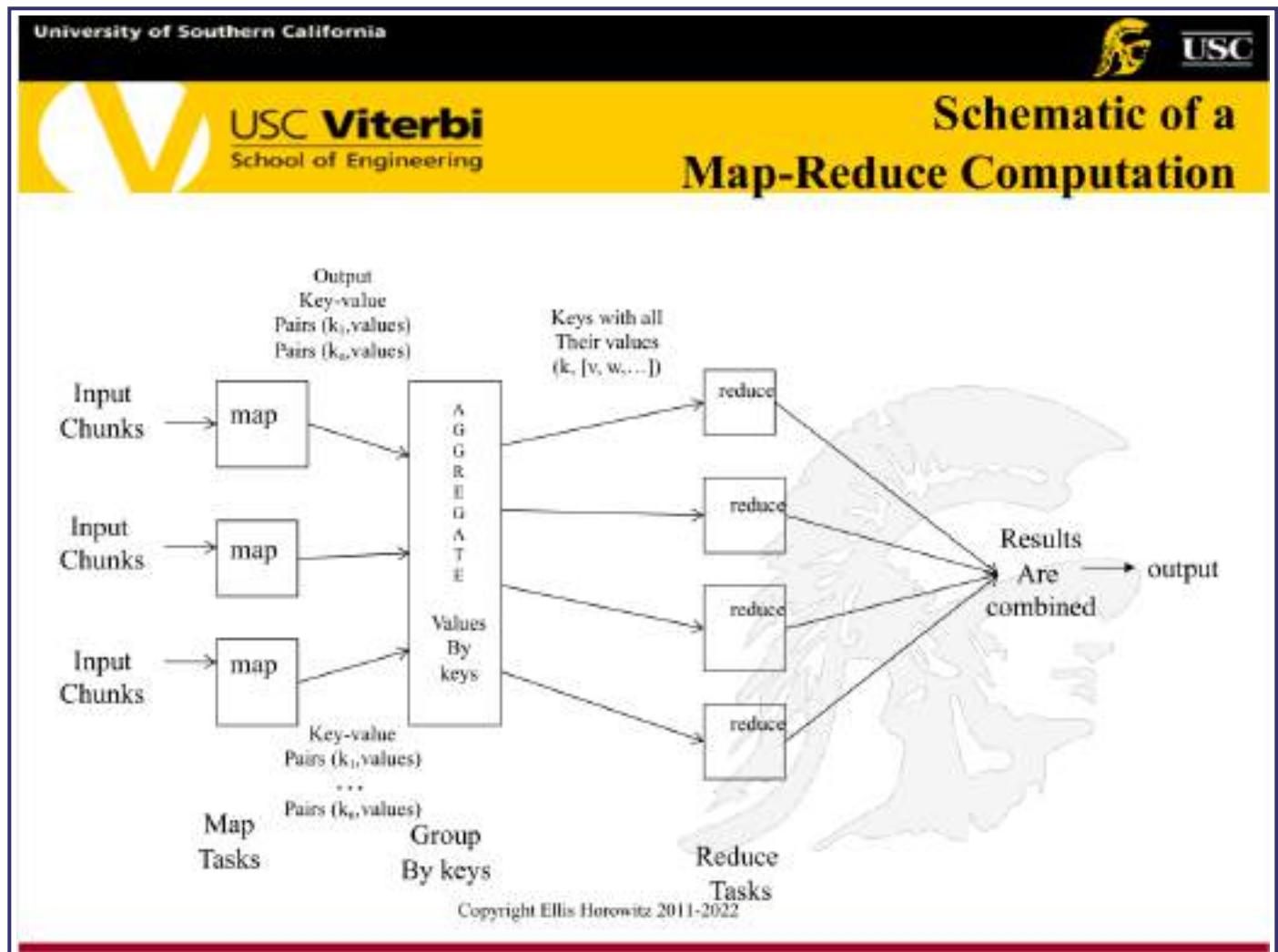
The Map&Reduce Routines

- Using map-reduce one must write 2 functions called *Map* and *Reduce*
- The system manages the parallel execution and coordination of tasks; it is all done automatically
- A map-reduce computation proceeds as follows:
 1. Some number of map tasks each are given one or more chunks to process
 2. These map tasks turn the chunk into a sequence of key-value pairs; the way the pairs are produced depends upon the code for the Map function
 3. Key-value pairs from each Map task are collected by a master controller and sorted by key; keys are divided among all the Reduce tasks, so all key-value pairs with the same key wind up at the same Reduce task
 4. Reduce tasks work on one key at a time, and combine all the values associated with that key in some way; the manner of combination depends upon the Reduce code

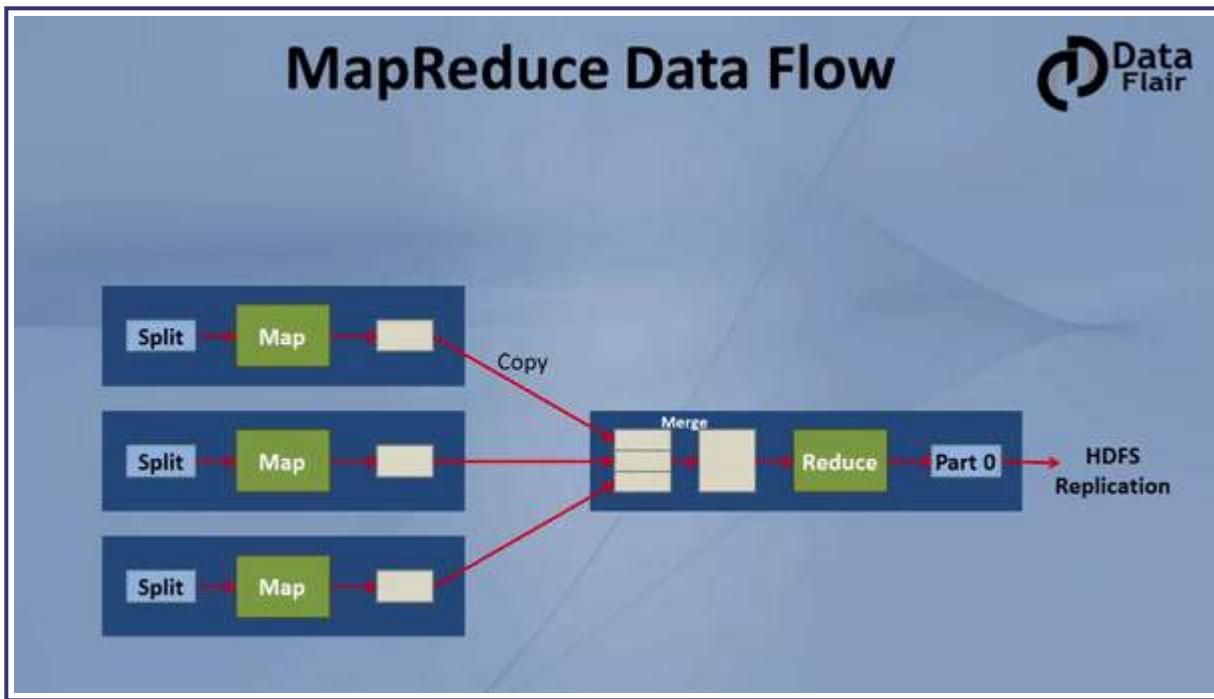
Copyright Ellis Horowitz, 2011-2022

17

••



Here is the dataflow:



••

University of Southern California 

 USC **Viterbi**
School of Engineering

A MapReduce Example – Counting Word Occurrences

- Counting the number of occurrences for each word in a collection of documents
- The input file is a repository of documents
- Each document is an element passed to a separate processor
- The Map function
 - Parses the document, extracts each word and uses each word as a key of type String (the words obtained by parsing), w_1, w_2, \dots
 - For each word it assigns an integer, 1;
 - Each processor outputs key-value pairs where the key is a word and the value is always 1, namely $(w_1, 1), (w_2, 1), \dots, (w_n, 1)$
- If a word w appears n times in a single document, then there will be n key-value pairs $(w, 1)$ in the output of the processor handling that document
- If a word w appears m times among all documents, then there will be m key-value pairs $(w, 1)$ in the output

Copyright Ellis Horowitz, 2011-2022

19

••



- The code below is similar to what a programmer would write to process multiple documents on a cluster of machines using map/reduce

Map(String input_key, String input_value):

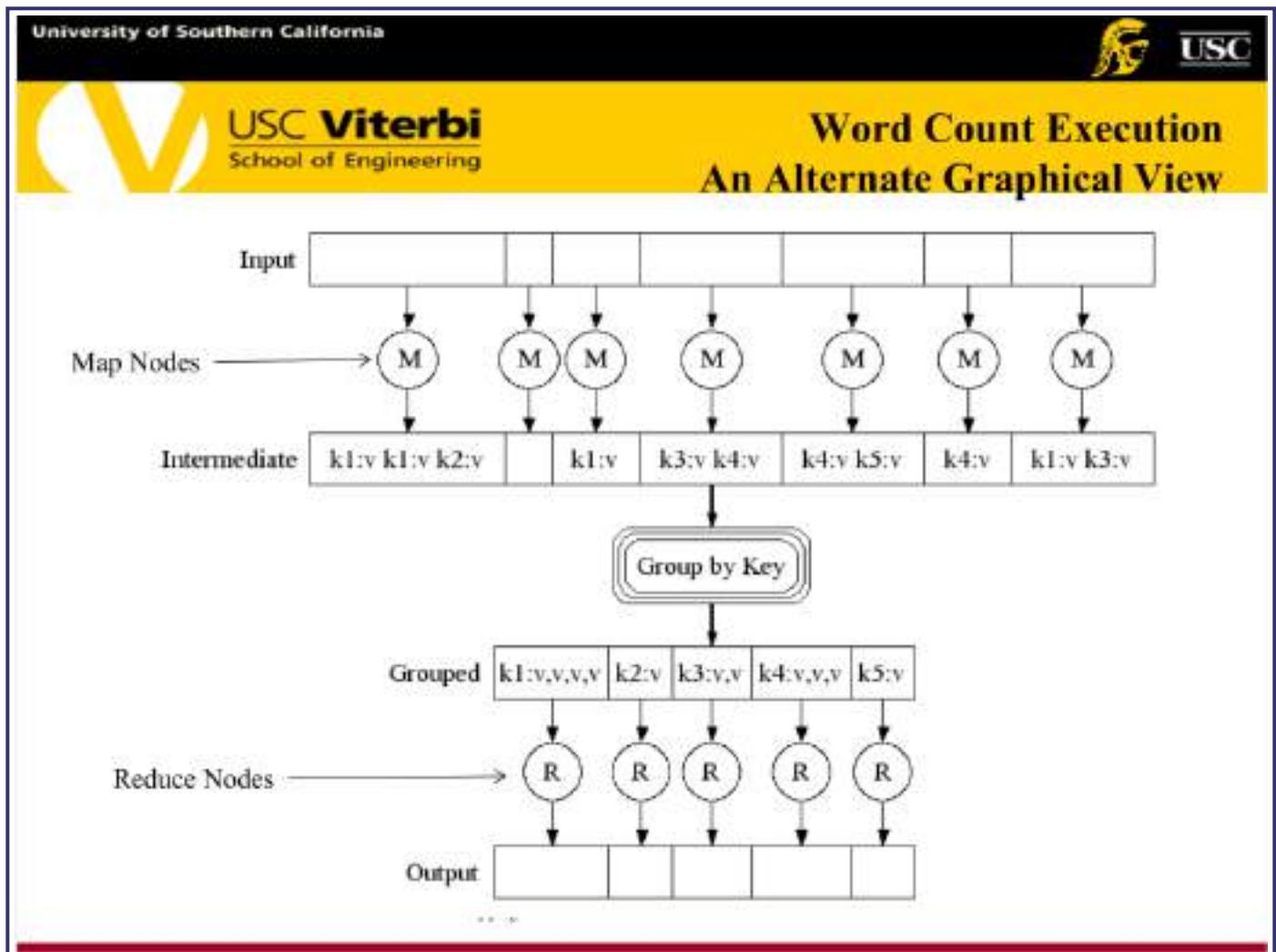
```
// input_key: document name  
// input_value: document contents  
for each word w in input_value:  
    EmitIntermediate(w, "1");
```

**reduce(String output_key, Iterator intermediate_values):**

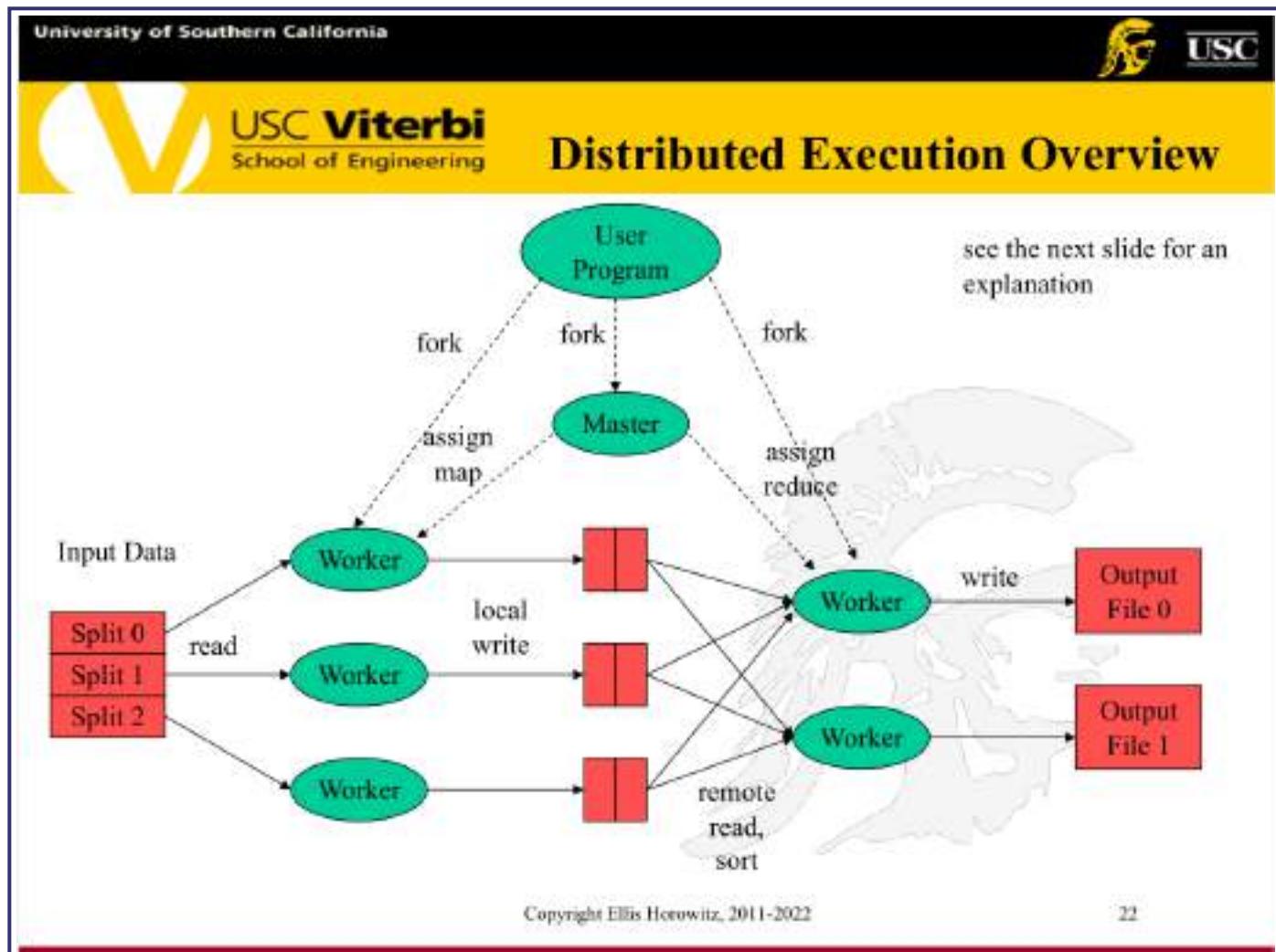
```
// output_key: a word  
// output_values: a list of counts  
int result = 0;  
for each v in intermediate_values:  
    result += ParseInt(v);  
EmitAsString(result);
```

Copyright Ellis Horowitz 2011-2022

••



••



••

University of Southern California 

USC Viterbi
School of Engineering

Looking Under the Hood at the Map Task

- The user program forks a *Master* controller process and some number of *Worker* processes at different compute nodes;
- A *Worker* handles either Map tasks or Reduce tasks, but not both
- The *Master* must
 - Create some number of Map and Reduce tasks
 - These tasks are assigned to *Worker* processes by the *Master*
 - Typically there is one Map task for every chunk of the input
 - Keeps track of the status of each Map and Reduce task (states are: idle, executing on a Worker, completed)
- Each Map task is assigned one or more chunks of the input file(s) and executes on it the code
- The Map task creates a file for each Reduce task on the local disk of the Worker that executes the Map task
- The *Master* is told of the location and sizes of each of these files and the Reduce task for which each is destined

Copyright Ellis Horowitz, 2011-2022

23

Here is the paper referenced above.

••



University of Southern California
USC Viterbi
School of Engineering

Looking Under the Hood at the Reduce Task

- The *master controller* process knows how many Reduce tasks there will be, say r
- The user defines r
- The master controller picks a hash function that applies to keys and produces a bucket number from 0 to $r-1$
- Each key output by a Map task is hashed and its key-value pair is put in one of r local files
 - Each file will be processed by a Reduce task
- After all Map tasks have completed successfully, the master controller merges the file from each Map task that are destined for a particular Reduce task and feeds the merged file to that process
- For each key k , the input to the Reduce task that handles key k is a pair $(k, [v_1, \dots, v_n])$ where $(k, v_1), (k, v_2), \dots, (k, v_n)$ are all the key-value pairs with key k coming from all the Map tasks

Copyright Ellis Horowitz, 2011-2022

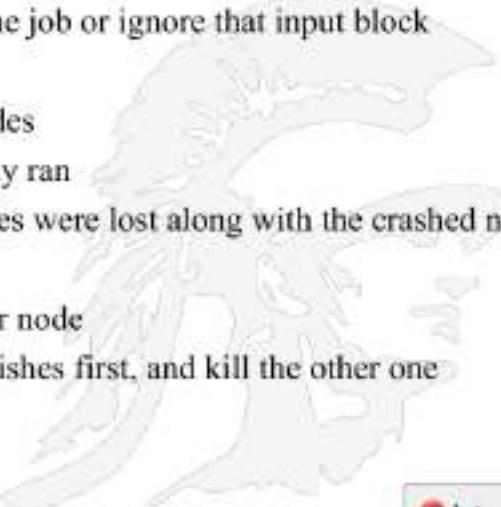
24

••

Explanation of the Reduce Task

- The *Reduce function* is written to take pairs consisting of a key and a list of associated values, and combines them in some way
- The *Reduce function* output is a sequence of key-value pairs consisting of each input key k paired with the combined value
- Outputs from all Reduce tasks are merged into a single file
- *Reduce function* adds up all the values and outputs a sequence of (w, m) pairs where w is a word that appears at least once in the documents and m is the total number of occurrences
- The *Reduce function* is generally associative and commutative implying values can be combined in any order yielding the same result

••



University of Southern California  USC

USC Viterbi
School of Engineering

Fault Tolerance in MapReduce

1. **If a task crashes:**
 - Retry on another node
 - OK for a map because it had no dependencies
 - OK for reduce because map outputs are on disk
 - If the same task repeatedly fails, fail the job or ignore that input block
2. **If a node crashes:**
 - Relaunch its current tasks on other nodes
 - Relaunch any maps the node previously ran
 - Necessary because their output files were lost along with the crashed node
3. **If a task is going slowly (straggler):**
 - Launch second copy of task on another node
 - Take the output of whichever copy finishes first, and kill the other one

Copyright Ellis Horowitz, 2011-2022

 bytes tip - saty@byte
The SSH2 session has terminated.
Error receiving bytes. Window aborted by the software in you.

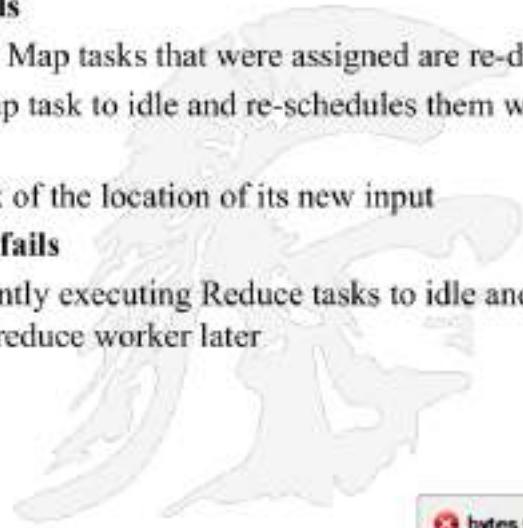
••

University of Southern California  USC

 USC Viterbi
School of Engineering

Coping with Node Failure

- Worst case: **the compute node where the Master is executing fails**
 - **Result:** the entire map-reduce job must be restarted
- Other failures are less severe and are handled by the Master
- **The compute node of a Map worker fails**
 - This is detected by the Master and all Map tasks that were assigned are re-done
 - The Master sets the status of each Map task to idle and re-schedules them when a worker becomes available
 - The Master informs each Reduce task of the location of its new input
- **The compute node of a Reduce worker fails**
 - The Master sets the status of its currently executing Reduce tasks to idle and they will be re-scheduled on another reduce worker later



Copyright Ellis Horowitz, 2011-2022

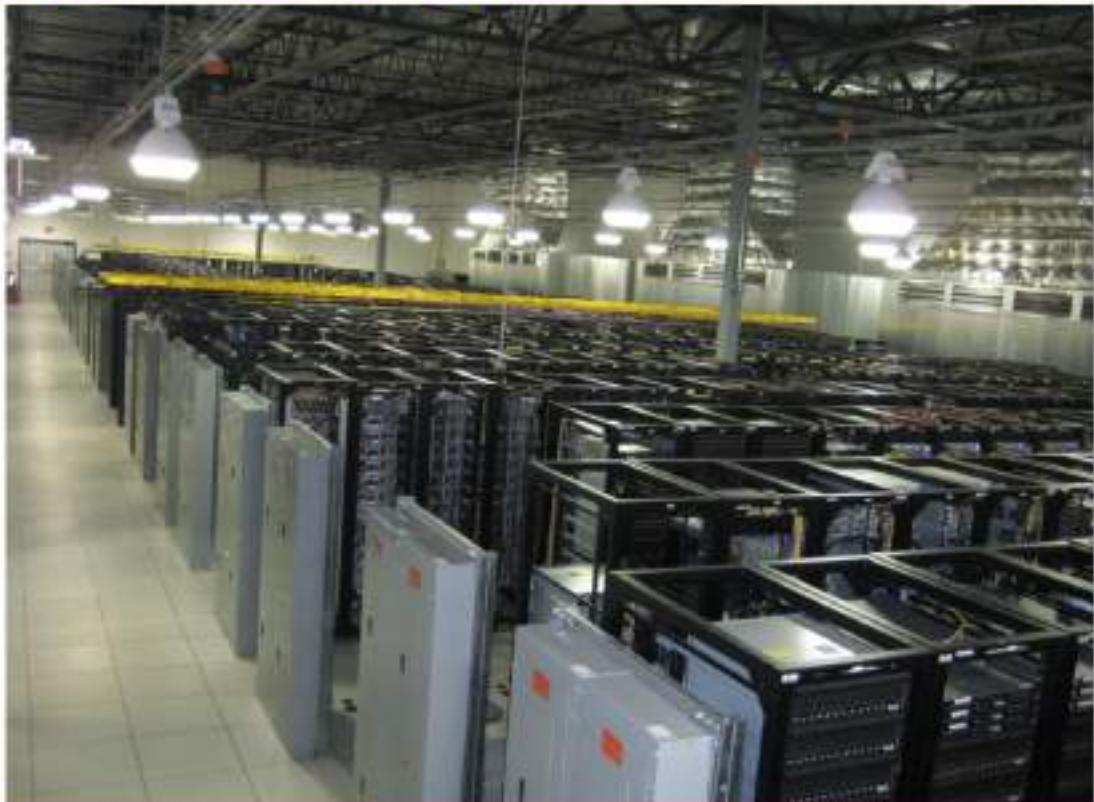
 bytes.tip - saty@byte
The SSH2 session has terminated.
Error receiving bytes. Window aborted by the software in you.

••

University of Southern California  USC

 USC **Viterbi**
School of Engineering

Typical Data Center Cluster



••

University of Southern California 

USC Viterbi
School of Engineering

Characteristics of a Google DataCenter

1. **Google data centers** (approx. two dozen): They come online and automatically, under the direction of the Google File System, start getting work from other data centers. These facilities, sometimes filled with 10,000 or more Google computers, find one another and configure themselves with minimal human intervention.
2. **Standard desktop PCs:** The hardware in a Google data center can be bought at a local computer store.
3. Each Google server comes in a standard case called a *pizza box* with one important change: the plugs and ports are at the front of the box to make access faster and easier.
4. **Google racks** are assembled for Google to hold servers on their front and back sides. This effectively allows a standard rack, normally holding 40 pizza box servers, to hold 80 servers.
5. A Google data center can go from a stack of parts to **online operation** in as little as **72 hours**, unlike more typical data centers that can require a week or even a month to get additional resources online.
6. Each server, rack and data center works in a way that is similar to what is called "**plug and play**." Like a mouse plugged into the USB port on a laptop, Google's network of data centers knows when more resources have been connected. These resources, for the most part, go into operation without human intervention.

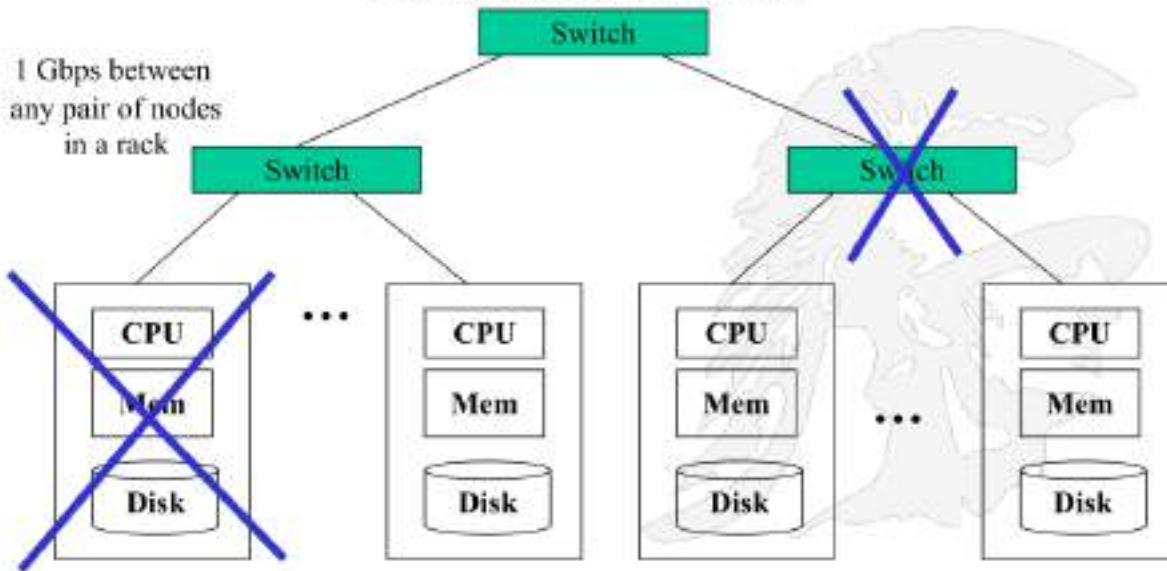
••

University of Southern California  USC 

Typical Cluster Architecture

- Each rack of cpu's contains between 16-64 nodes
- Nodes within a single rack are connected by gigabyte Ethernet
- Each rack is connected to another rack by a switch with speeds of 2-10 Gbps
- Individual cpu's can fail; switches between racks can fail

2-10 Gbps backbone between racks

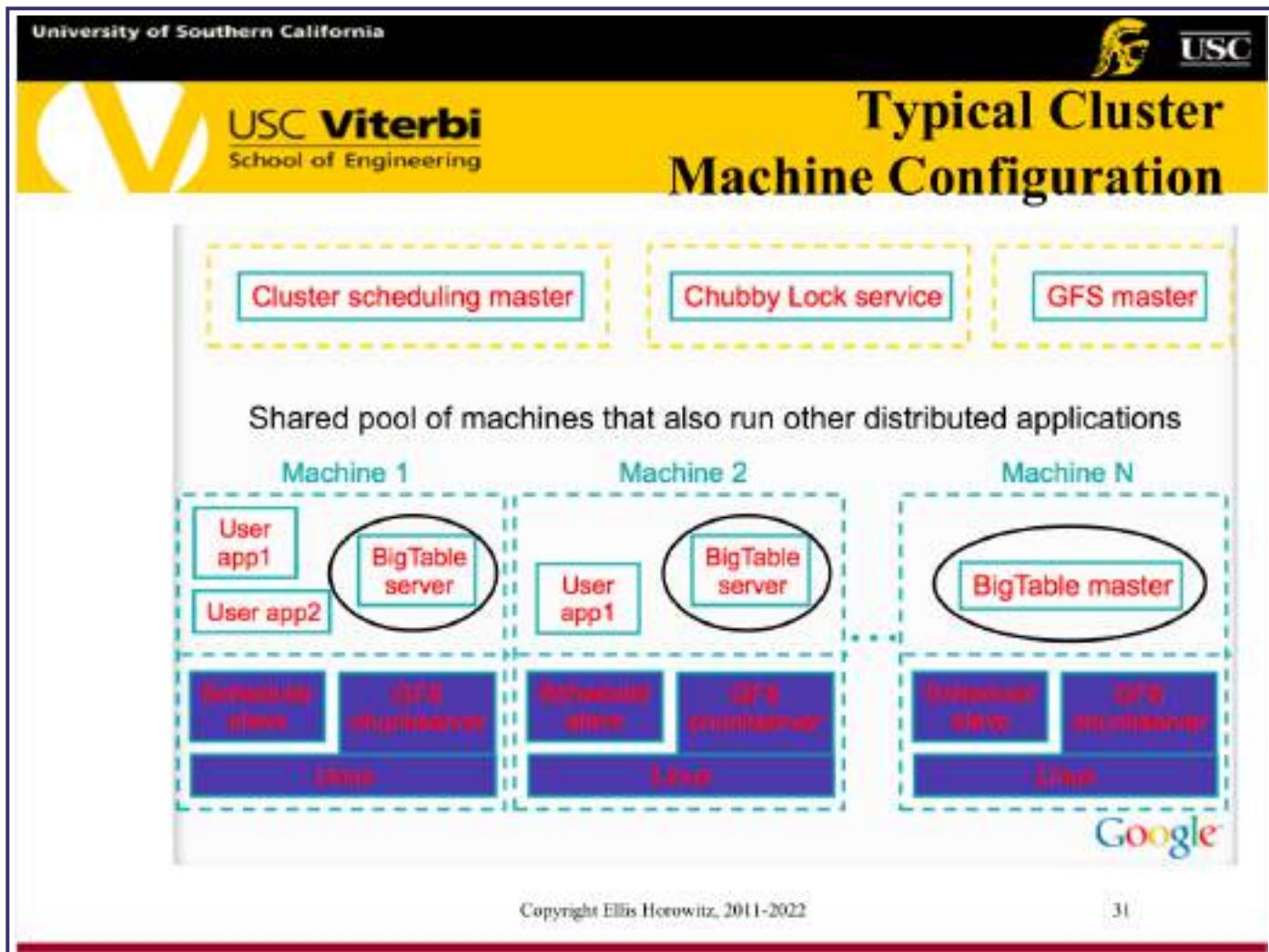


The diagram illustrates a cluster architecture with two racks of nodes. Each rack contains multiple nodes, each represented by a box divided into three horizontal sections: CPU, Mem, and Disk. The nodes within a single rack are connected to a central switch, indicated by lines connecting the nodes to a green box labeled "Switch". A label above the first rack states "1 Gbps between any pair of nodes in a rack". Between the two racks, there is a backbone switch represented by a green box labeled "Switch" with lines connecting it to the switches of both racks. A large blue "X" is drawn over the backbone switch, indicating that it is a potential failure point. A watermark of a person's head is visible in the background of the diagram area.

Copyright Ellis Horowitz, 2011-2022

30

••



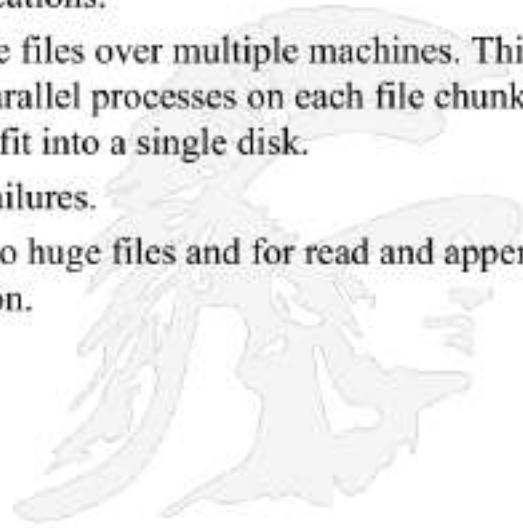
••

University of Southern California 

 USC **Viterbi**
School of Engineering

The Problems Google Tried to Solve with a New File System

- Google needed a large-scale and high-performance unified storage system that must:
 1. **Be global.** Any client can access (read/write) any file. This allows for sharing of data among different applications.
 2. **Support automatic sharding** of large files over multiple machines. This improves performance by allowing parallel processes on each file chunk and also deals with large files that cannot fit into a single disk.
 3. **Support automatic recovery** from failures.
 4. **Be optimized for sequential access** to huge files and for read and append operations which are the most common.



Copyright Ellis Horowitz, 2011-2022

32

••

University of Southern California 

 USC **Viterbi**
School of Engineering

Google File System General Goals

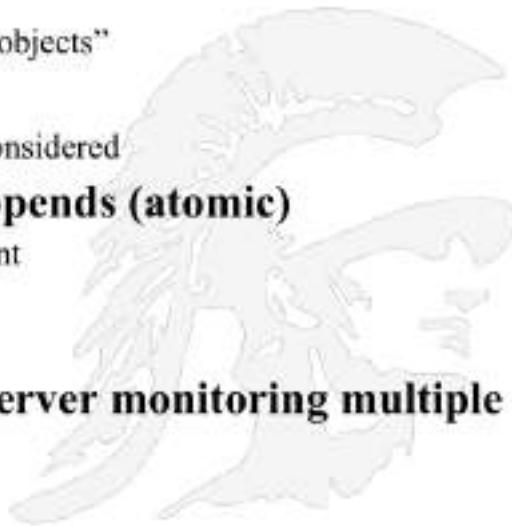
- **A scalable, distributed file system for large distributed data-intensive applications**
 - Provides fault tolerance
 - Runs on cheap, commodity hardware
 - Delivers high aggregate performance to large number of clients
- **GFS: not your typical file system**
 - Lacks typical per-directory data structure to list each file in the directory
 - Does not support aliases (i.e. hard or sym links)
 - Namespace: lookup table that maps full pathnames to metadata
 - Lookup table fits in memory (**prefix compression**)
 - Also known as incremental encoding is a type of delta encoding **compression** algorithm whereby common **prefixes** and their lengths are recorded so that they need not be duplicated
 - https://en.wikipedia.org/wiki/Incremental_encoding

Copyright Ellis Horowitz, 2011-2022

33

••

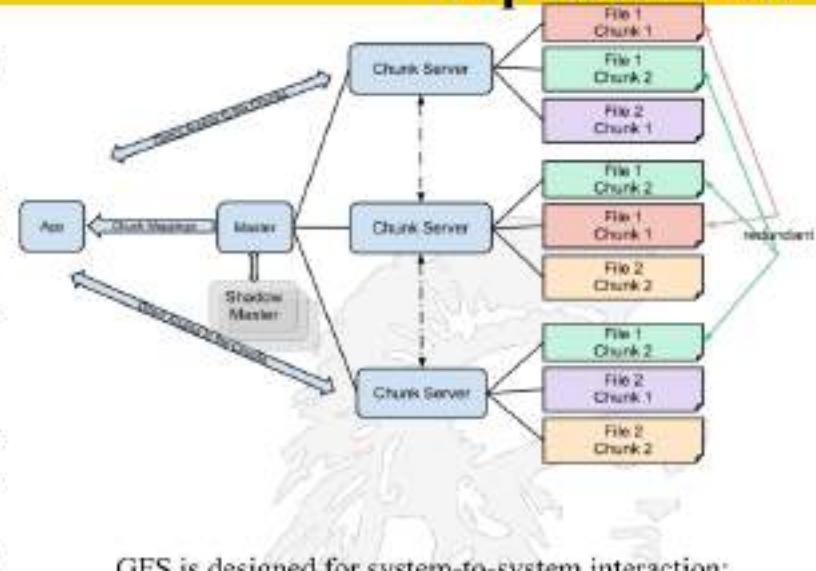
- **Files will be HUGE**
- **Multi-gigabyte files are common**
 - Not practical to have ~8 billion files
 - Each file contains many “application objects”
- **Multi-terabyte datasets**
 - I/O operations, block sizes must be considered
- **Most file modifications are appends (atomic)**
 - Random writes practically non-existent
 - Once written... sequential reads
 - Caching not terribly important
- **there will be a single master server monitoring multiple chunk servers**



•

University of Southern California  USC 

Google File System Top Level View



- Google File System (GFS),** is a proprietary distributed file system for efficient, reliable access to data using large clusters of commodity hardware
- Files are divided into fixed-size *chunks* of 64 megabytes, similar to clusters or sectors in regular file systems, which are only extremely rarely overwritten, or shrunk; files are usually appended to or read

GFS is designed for system-to-system interaction:
 Chunk servers replicate the data automatically
 See https://en.wikipedia.org/wiki/Google_File_System

Copyright Ellis Horowitz, 2011-2022

35

•



University of Southern California
USC Viterbi
School of Engineering

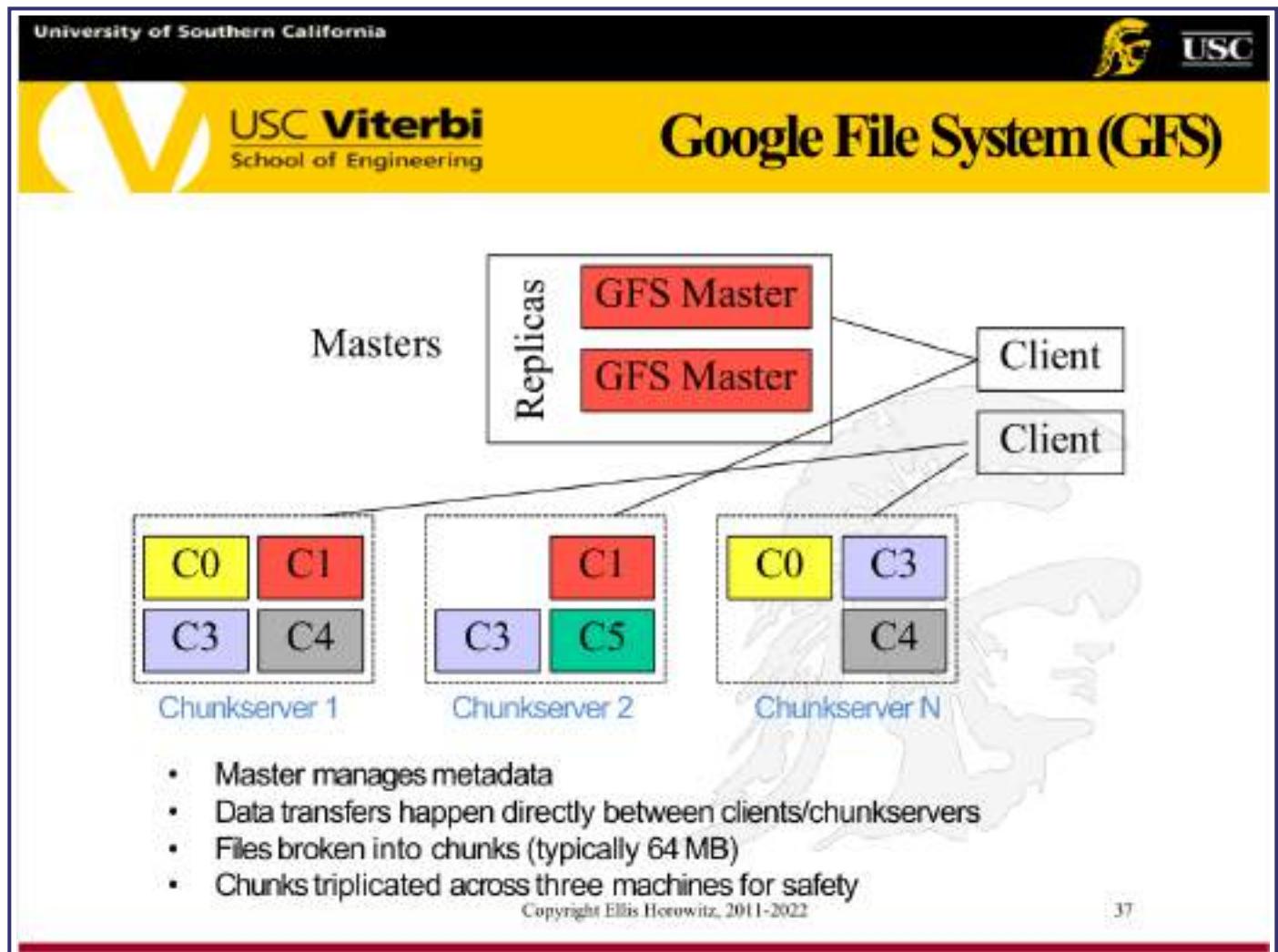
Master Server and Chunk Servers

- Master Server holds all metadata:
 - Namespace (directory hierarchy)
 - Accesscontrol information (per-file)
 - Mapping from files to chunks
 - Current locations of chunks (chunkservers)
- Delegates consistency management
- Garbage collects orphaned chunks
- Migrates chunks between chunk servers
- Chunk Server
 - Stores 64 MB file chunks on local disk using standard Linux filesystem, each with version number and checksum
 - Read/write requests specify chunk handle and byte range
 - Chunks replicated on configurable number of chunkservers (default: 3)
 - No caching of file data (beyond standard Linux buffer cache)

Copyright Ellis Horowitz, 2011-2022

36

••



••



GFS: Major Aspects

- **Append vs. Rewrite**

- GFS is optimized for appended files rather than rewrites. That's because clients within Google rarely need to overwrite files -- they add data onto the end of files instead. While it's still possible to overwrite data on a file in the GFS, the system doesn't handle those processes very efficiently

- **Which Replica Does GFS use?**

- The GFS separates replicas into two categories: **primary replicas** and **secondary replicas**. A primary replica is the chunk that a chunkserver sends to a client. Secondary replicas serve as backups on other chunkservers.
 - The master server decides which chunks will act as primary or secondary. If the client makes changes to the data in the chunk, then the master server lets the chunkservers with secondary replicas know they have to copy the new chunk off the primary chunkserver to stay current.

- **What About Big Files?**

- If a client creates a write request that affects multiple chunks of a particularly large file, the GFS breaks the overall write request up into an individual request for each chunk. The rest of the process is the same as a normal write request.

- **Heartbeats and Handshakes**

- The GFS components give system updates through electronic messages called **heartbeats** and **handshakes**. These short messages allow the master server to stay current with each chunkserver's status.

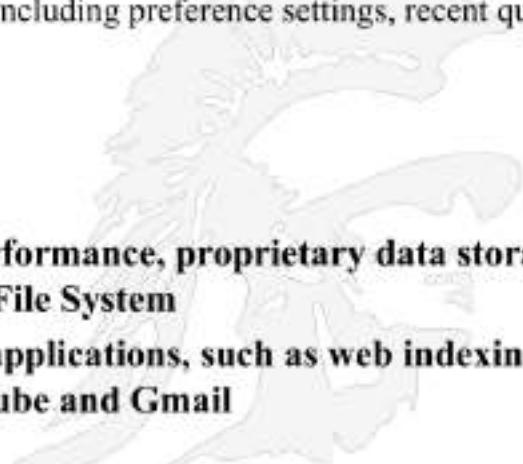
••

University of Southern California 

USC Viterbi
School of Engineering

Google File System vs. BigTable

- **GFS provides raw data storage**
- **But Google needs a system for handling:**
 - Trillions of URLs
 - Geographic locations such as physical entities, roads, satellite image data, etc
 - Per user data for billions of people including preference settings, recent queries and searches
 - And it must be capable of
 - storing semi-structured data
 - Reliable, scalable, etc
- **Bigtable is a compressed, high performance, proprietary data storage system built on top of the Google File System**
- **It is used by a number of Google applications, such as web indexing, MapReduce, Google Maps, YouTube and Gmail**



Copyright Ellis Horowitz, 2011-2022

39

••

University of Southern California 

 USC **Viterbi**
School of Engineering

Big Table Data Model

- Not a Full Relational Data Model
- Provides a simple data model
 - Supports dynamic control over data layout
 - Allows clients to reason about the locality properties
- A Table in Bigtable is a:
 - Sparse
 - Distributed
 - Persistent
 - Multidimensional
 - Sorted map



Copyright Ellis Horowitz, 2011-2022

40

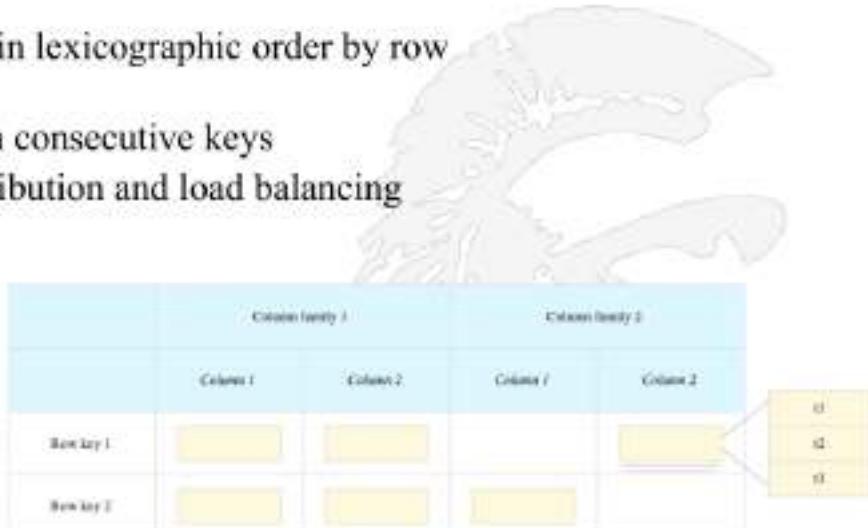
••

University of Southern California 

USC Viterbi
School of Engineering

Bigtable Storage Model

- Data is indexed using row and column names
- Data is treated as uninterpreted strings
 - (row:string, column:string, time:int64) → string
- Rows
 - Data maintained in lexicographic order by row key
 - Tablet: rows with consecutive keys
 - Units of distribution and load balancing
- Columns
 - Column families
- Cells
- Timestamps



	Column family 1	Column family 2		
	Column 1	Column 2	Column 1	Column 2
Row key 1				
Row key 2				v1 v2 v3

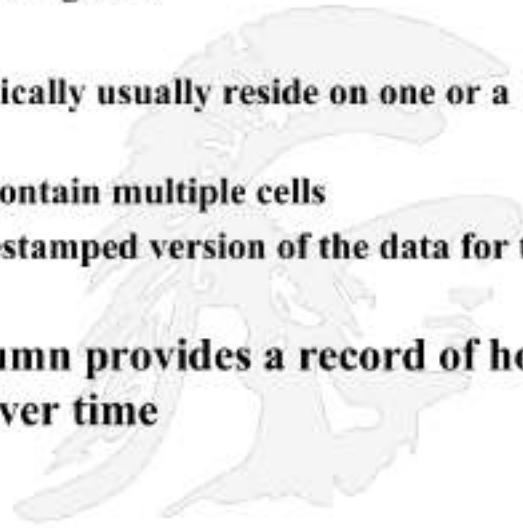
••

University of Southern California 

 USC **Viterbi**
School of Engineering

Rows

- Name is an arbitrary string
 - Access to data in a row is atomic
 - Row creation is implicit upon storing data
- Rows ordered lexicographically
 - Rows close together lexicographically usually reside on one or a small number of machines
- Each row/column intersection can contain multiple cells
 - Each cell contains a unique timestamped version of the data for that row and column
- Storing multiple cells in a column provides a record of how the stored data has changed over time



Copyright Ellis Horowitz, 2011-2022

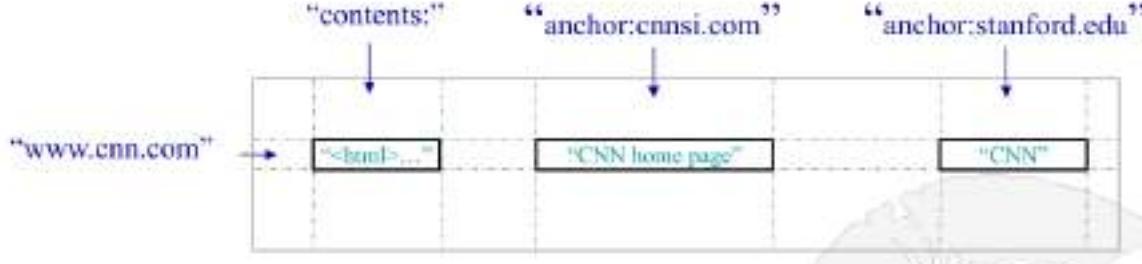
42

••

University of Southern California  USC

USC Viterbi School of Engineering

Columns



- **Columns have two-level name structure:**
 - family:optional_qualifier
- **Column family**
 - Unit of access control
 - Has associated type information
- **Qualifier gives unbounded columns**
 - Additional level of indexing, if desired

Copyright Ellis Horowitz, 2011-2022

43

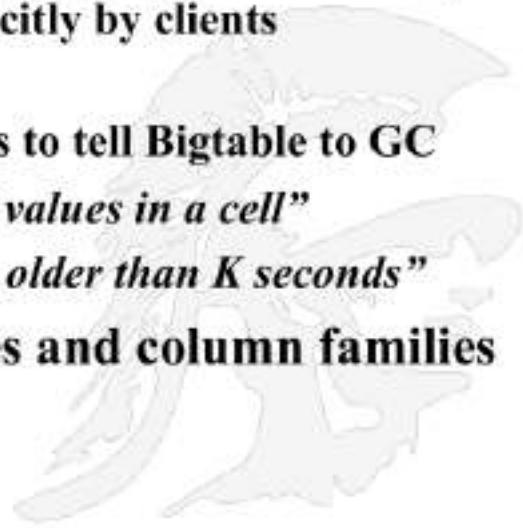
••



University of Southern California
USC Viterbi
School of Engineering

Timestamps

- Used to store different versions of data in a cell
 - New writes default to current time, but timestamps for writes can also be set explicitly by clients
- Garbage Collection
 - Per-column-family settings to tell Bigtable to GC
 - “*Only retain most recent K values in a cell*”
 - “*Keep values until they are older than K seconds*”
- API: Create / delete tables and column families



Copyright Ellis Horowitz, 2011-2022

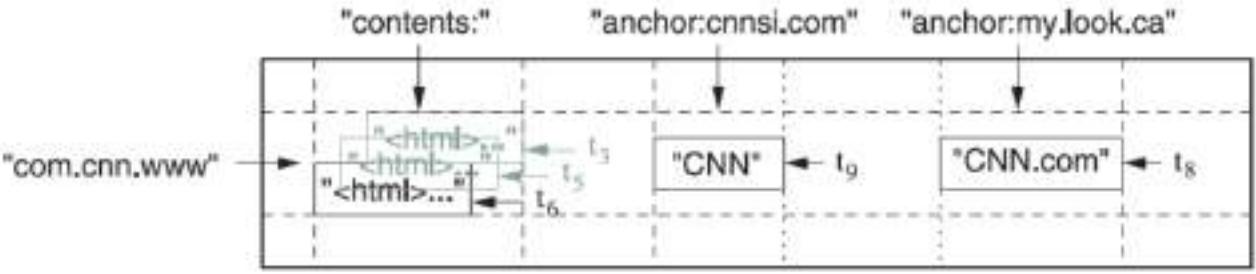
44

••

University of Southern California  USC

USC Viterbi School of Engineering

Data Model – WebTable Example (1 of 7)



The diagram illustrates a row in a WebTable. The first column, labeled "contents:", contains a complex nested structure representing an HTML document. An arrow points from the label "com.cnn.www" to this column. The second and third columns, both labeled "anchor:", contain URLs: "cnnsi.com" and "mylook.ca". The fourth column, labeled "anchor text:", contains the words "CNN" and "CNN.com". Arrows labeled t_1 through t_9 point from the anchor URLs to their respective anchor text boxes. A watermark of a person's face is visible in the background.

"com.cnn.www"

"contents:" "anchor:cnnsi.com" "anchor:mylook.ca"

"<html>" "CNN" "CNN.com"

"<html>" "CNN.com"

"<html>..."

t_1 t_2 t_3

t_4 t_5 t_6

t_7 t_8 t_9

A large collection of web pages and related information

Copyright Ellis Horowitz, 2011-2022

45

••

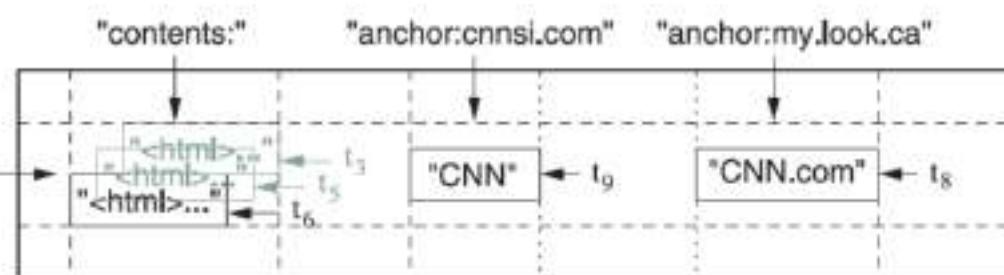
University of Southern California  USC 

Data Model – WebTable Example (2)

Row Key

"com.cnn.www"

"contents:" "anchor:cnnsi.com" "anchor:mylook.ca"



Tablet - Group of rows with consecutive keys.
Unit of Distribution
Bigtable maintains data in lexicographic order by row key

Copyright Ellis Horowitz, 2011-2022

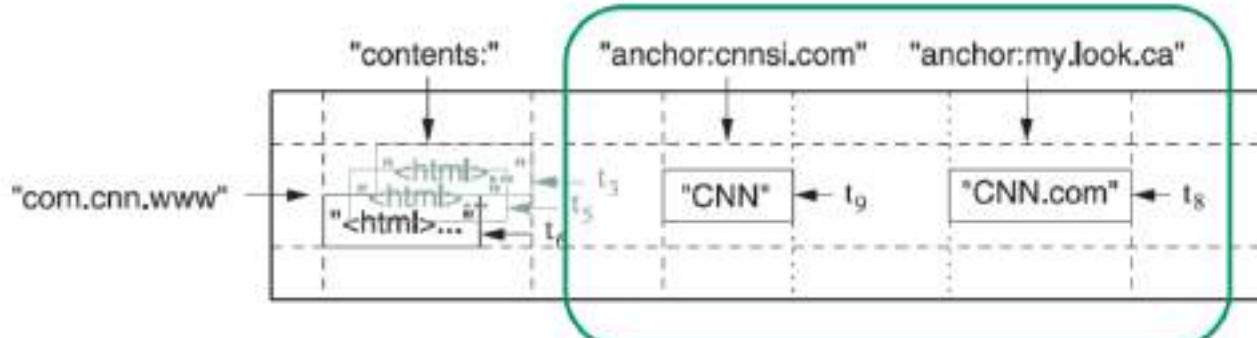
46

••

University of Southern California  USC

USC Viterbi School of Engineering

Data Model – WebTable Example (3)



The diagram illustrates a row in a WebTable. The row key is "com.cnn.www". The first column family is labeled "contents:" and contains several nested HTML elements. The second column family is labeled "anchor:cnnsi.com" and contains the value "CNN". The third column family is labeled "anchor:mylook.ca" and contains the value "CNN.com". A green oval highlights the second and third column families. Arrows point from the column names to their respective values. Below the diagram, a green speech bubble contains the text "Column Family".

"com.cnn.www"

"contents:"

"anchor:cnnsi.com" "anchor:mylook.ca"

"<html>" "CNN" "CNN.com"

Column Family

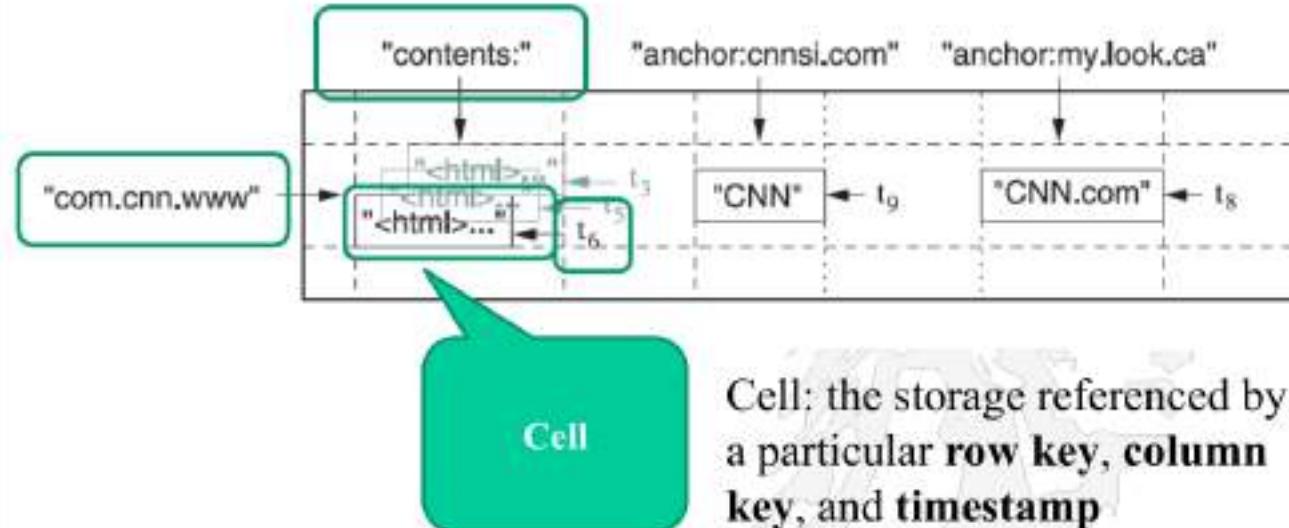
Copyright Ellis Horowitz, 2011-2022

47

••

University of Southern California  USC

Data Model – WebTable Example (6)



The diagram illustrates a row in a WebTable. The row key is "com.cnn.www". The row contains several cells:

- A cell for "contents:" with value "<html><thumb><html>..." and timestamp t_6 .
- Cells for "anchor:cnnsi.com" and "anchor:mylook.ca" both with timestamp t_3 .
- Cells for "CNN" and "CNN.com" both with timestamp t_9 .

A green callout bubble points to one of the timestamped cells with the text "Cell".

Cell: the storage referenced by a particular **row key**, **column key**, and **timestamp**

Copyright Ellis Horowitz, 2011-2022

50

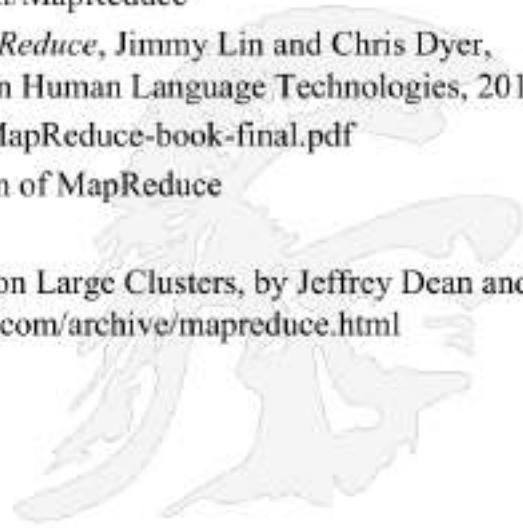
••

University of Southern California 

 USC **Viterbi**
School of Engineering

References

- **Google Videos on map/reduce**
<https://www.youtube.com/watch?v=yjPBkvYh-ss> (Lecture 1, 46 min)
<https://www.youtube.com/watch?v=vD6PUdf3Js> (Lecture 2, 52 min)
- **Wikipedia**, <http://en.wikipedia.org/wiki/MapReduce>
- *Data-Intensive Text Processing with MapReduce*, Jimmy Lin and Chris Dyer, Morgan & Claypool Synthesis Lectures on Human Language Technologies, 2010
<http://www.umiacs.umd.edu/~jimmylin/MapReduce-book-final.pdf>
- **Hadoop** is an open source implementation of MapReduce
<http://hadoop.apache.org/>
- **MapReduce: Simplified Data Processing on Large Clusters**, by Jeffrey Dean and Sanjay Ghemawat, <http://research.google.com/archive/mapreduce.html>



Copyright Ellis Horowitz, 2011-2022

53

And...

The past:

- <https://www.dataversity.net/a-brief-history-of-the-hadoop-ecosystem>
- <https://data-flair.training/blogs/hadoop-history/>

The present:

- <https://www.datacenterknowledge.com/archives/2014/06/25/google-dumps-mapreduce-favor-new-hyper-scale-analytics-system>