Web Science: Searching the Web

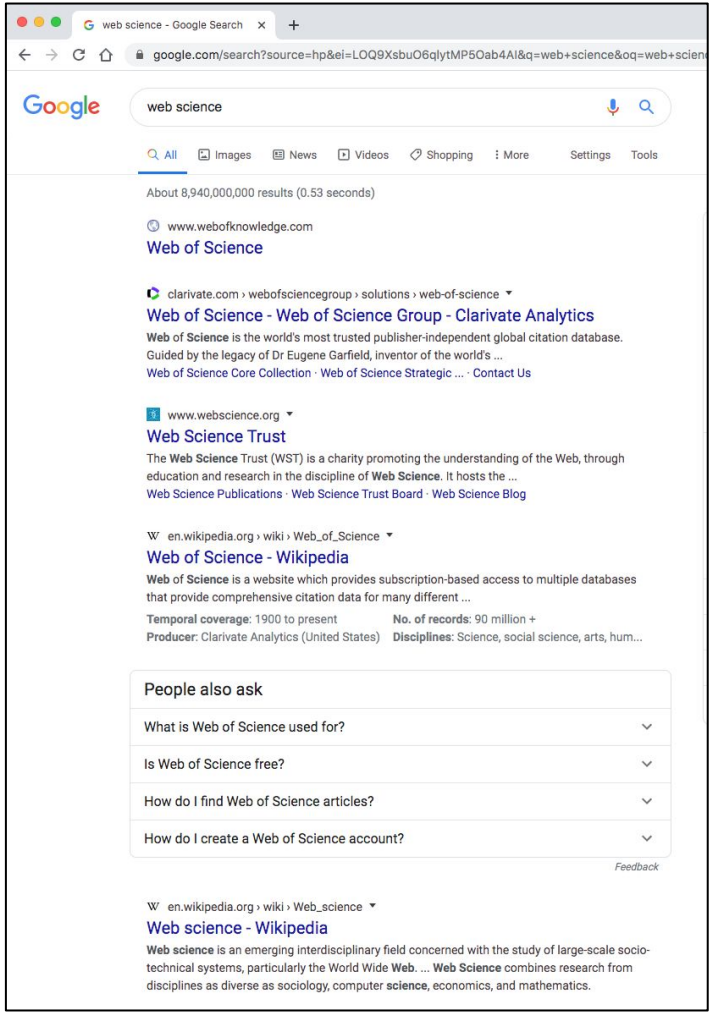
(Part 1 - Crawling the Web)

CS 432/532

Old Dominion University

Permission has been granted to use these slides from Frank McCown, Michael L. Nelson, Alexander Nwala, Michele C. Weigle

Search for “web science”



Overview of Web Search

- Must collect the documents
  - **crawl** the web
- Then **index** the documents
  - what words are in each document?
- Finally, return a ranked list of documents from a query
  - **ranking** is where the hard part comes in

What we'll examine

- Web crawling
- Building an index
- Querying the index
- Term frequency and inverse document frequency
- Other methods to increase relevance
- How links between web pages can be used to improve ranking search engine results
- Link spam and how to overcome it

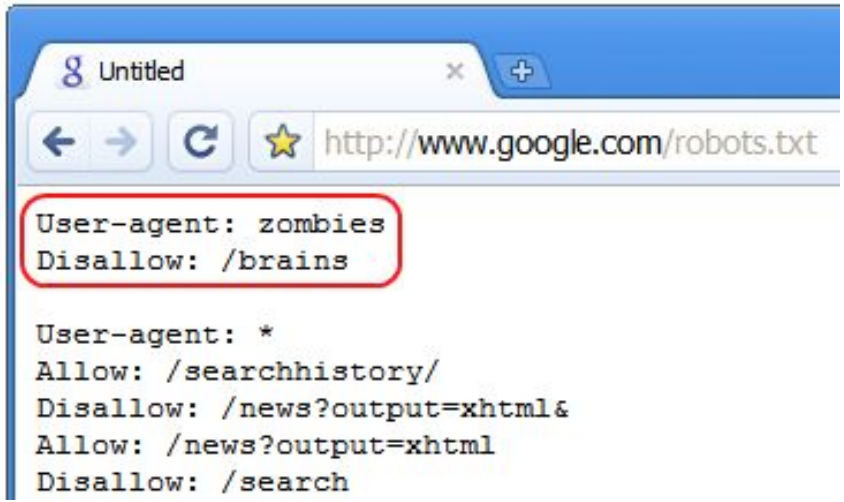
Web Crawling

- Large search engines use thousands of continually running web crawlers to discover web content
- Web crawlers fetch a page, place all the page's links in a queue, fetch the next link from the queue, and repeat
- Web crawlers are usually polite
  - Identify themselves through the HTTP User-Agent request header (e.g., googlebot)
  - Throttle requests to a web server, crawl at off-peak times
  - Honor robots exclusion protocol (robots.txt). Example:

```
User-agent: *
Disallow: /private
```

more about robots.txt: [The Web Robots Pages]

Robots.txt Humor



Halloween 2008

"Halloween easter egg: Google protects itself from zombies", which links to live version of robots.txt (archived version of robots.txt)

Web Crawler Components

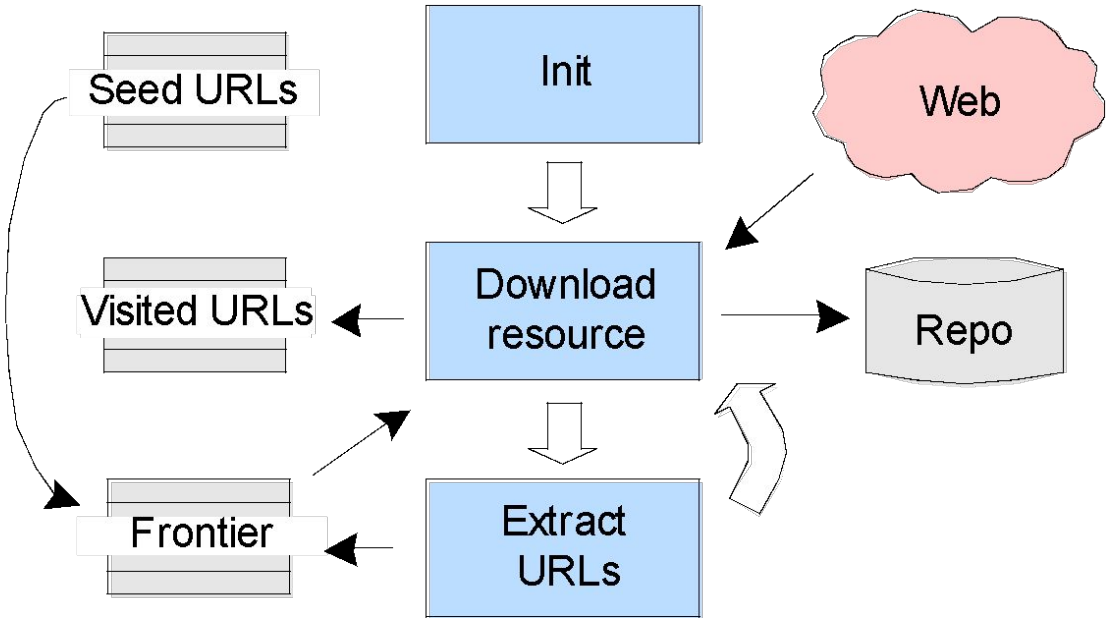
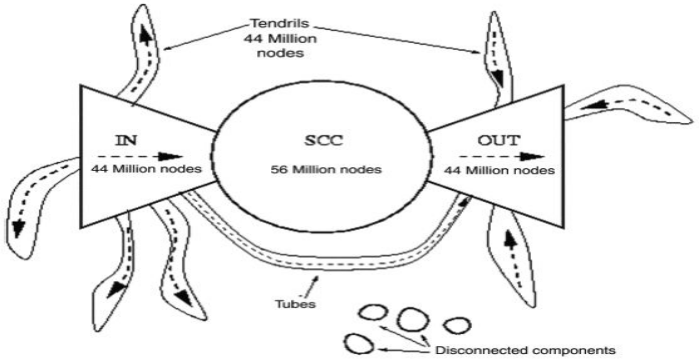


Figure: McCown, *Lazy Preservation: Reconstructing Websites from the Web Infrastructure*, Dissertation, 2007

Crawling Issues

- Good source for seed URLs:
  - DMOZ - The Directory of the Web
    - archived version of dmoz.org
  - Previously crawled URLs

remember the bow-tie?



- Search engine competing goals:
  - Keep index fresh (crawl often)
  - Comprehensive index (crawl as much as possible)
- Which URLs should be visited first or more often?
  - Breadth-first (FIFO)
  - Pages which change frequently & significantly
  - Popular or highly-linked pages

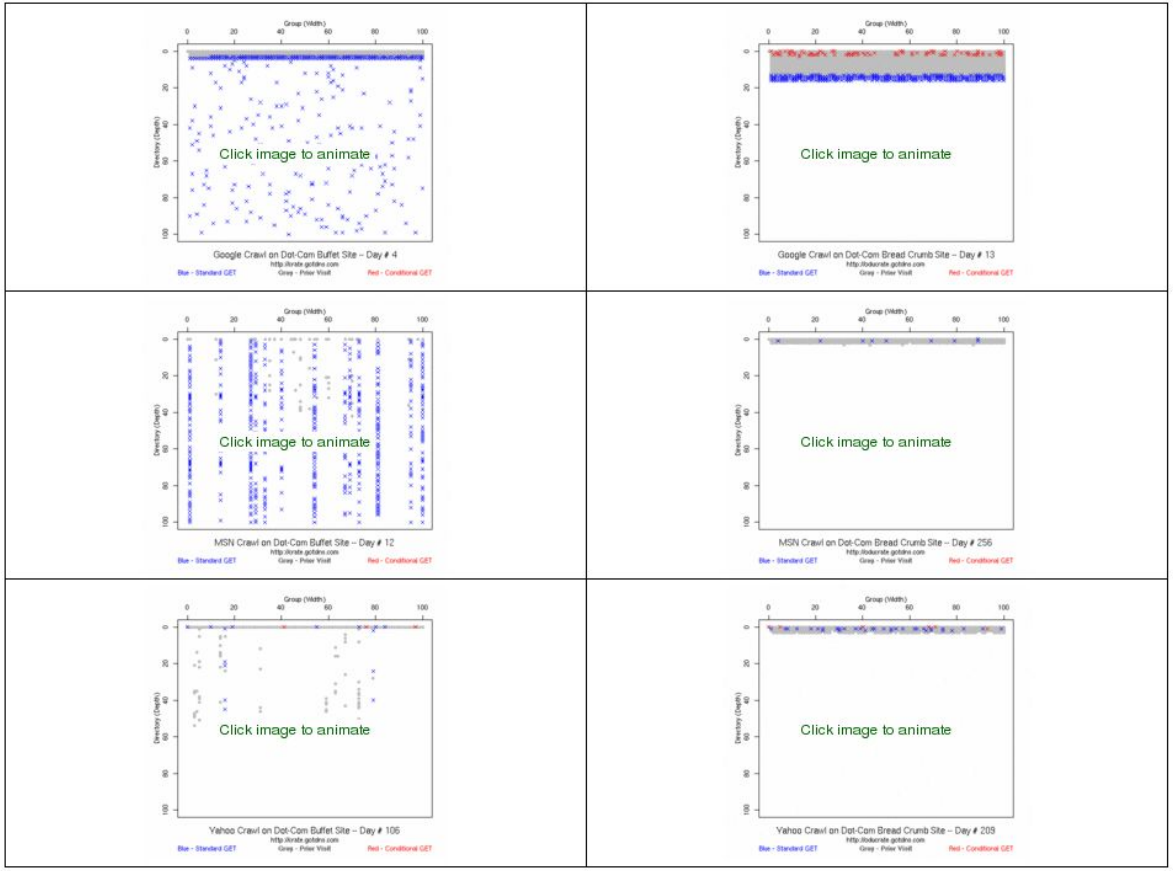


Table 3 Links to Animations of Crawler Activity at the Dot-Com Sites

web crawler animations - see Table 3 of Smith and Nelson, Site Design Impact on Robots: An Examination of Search Engine Crawler Behavior at Deep and Wide Websites, 2008

Crawling Issues - Part 2

- Should avoid crawling duplicate content
  - Convert page content to compact string (*fingerprint*) and compare to previously crawled fingerprints
- Should avoid crawling spam
  - Content analysis of page could make crawler ignore it while crawling or in post-crawl processing
- Robot traps
  - Deliberate or accidental trail of infinite links (e.g., calendar)
  - Solution: limit depth of crawl
- Deep Web
  - Throw search terms at interface to discover pages[1]
  - Sitemaps allow websites to publish URLs that might not be discovered in regular web crawling

[1]Madhavan et al., Google's Deep Web crawl, *Proc. VLDB 2008*

Example Sitemap

```
<?xml version="1.0" encoding="UTF-8"?>
<urlset xmlns="http://www.sitemaps.org/schemas/sitemap/0.9">
   <url>
      <loc>http://www.example.com/</loc>
      <lastmod>2009-10-22</lastmod>
      <changefreq>weekly</changefreq>
      <priority>0.8</priority>
   </url>
   <url>
      <loc>http://www.example.com/specials.html</loc>
      <changefreq>daily</changefreq>
      <priority>0.9</priority>
   </url>
   <url>
      <loc>http://www.example.com/about.html</loc>
      <lastmod>2009-11-4</lastmod>
      <changefreq>monthly</changefreq>
   </url>
</urlset>
```

Find Sitemaps in robots.txt

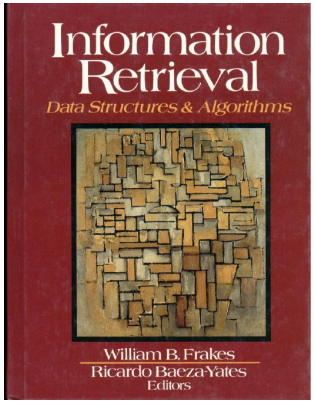
```
$ curl -i www.cnn.com/robots.txt
HTTP/1.1 200 OK
Date: Wed, 05 Feb 2020 20:29:35 GMT
Content-Type: text/plain; charset=utf-8
Content-Length: 1061
Accept-Ranges: bytes
Date: Wed, 05 Feb 2020 20:29:35 GMT
Transfer-Encoding: chunked
Connection: keep-alive

Sitemap: https://www.cnn.com/sitemaps/cnn/index.xml
Sitemap: https://www.cnn.com/sitemaps/cnn/news.xml
Sitemap: https://www.cnn.com/sitemaps/sitemap-section.xml
Sitemap: https://www.cnn.com/sitemaps/sitemap-interactive.xml
Sitemap: https://www.cnn.com/ampstories/sitemap.xml
Sitemap: https://edition.cnn.com/sitemaps/news.xml
User-agent: *
Allow: /audio/podcasts/
Allow: /partners/ipad/live-video.json
Disallow: /*.jsx$
Disallow: /ads/
Disallow: /aol/
[deletia…]
```

Focused Crawling

- A *vertical search engine* focuses on a subset of the Web
  - Google Scholar – scholarly literature
  - Shopzilla – Internet shopping
- A topical, or *focused, web crawler* attempts to download only pages about a specific topic
  - Has to analyze page content to determine if it's on topic and if links should be followed
  - Usually analyzes anchor text as well

Is What I've Found On Topic?

- Precision
  - "ratio of the number of relevant documents retrieved over the total number of documents retrieved" (p. 10)
  - *how much extra stuff did you get?*
- Recall
  - "ratio of relevant documents retrieved for a given query over the number of relevant documents for that query in the database" (p. 10)
    - note: assumes a priori knowledge of the denominator!
  - *how much did you miss?*



[FBY]

Precision and Recall

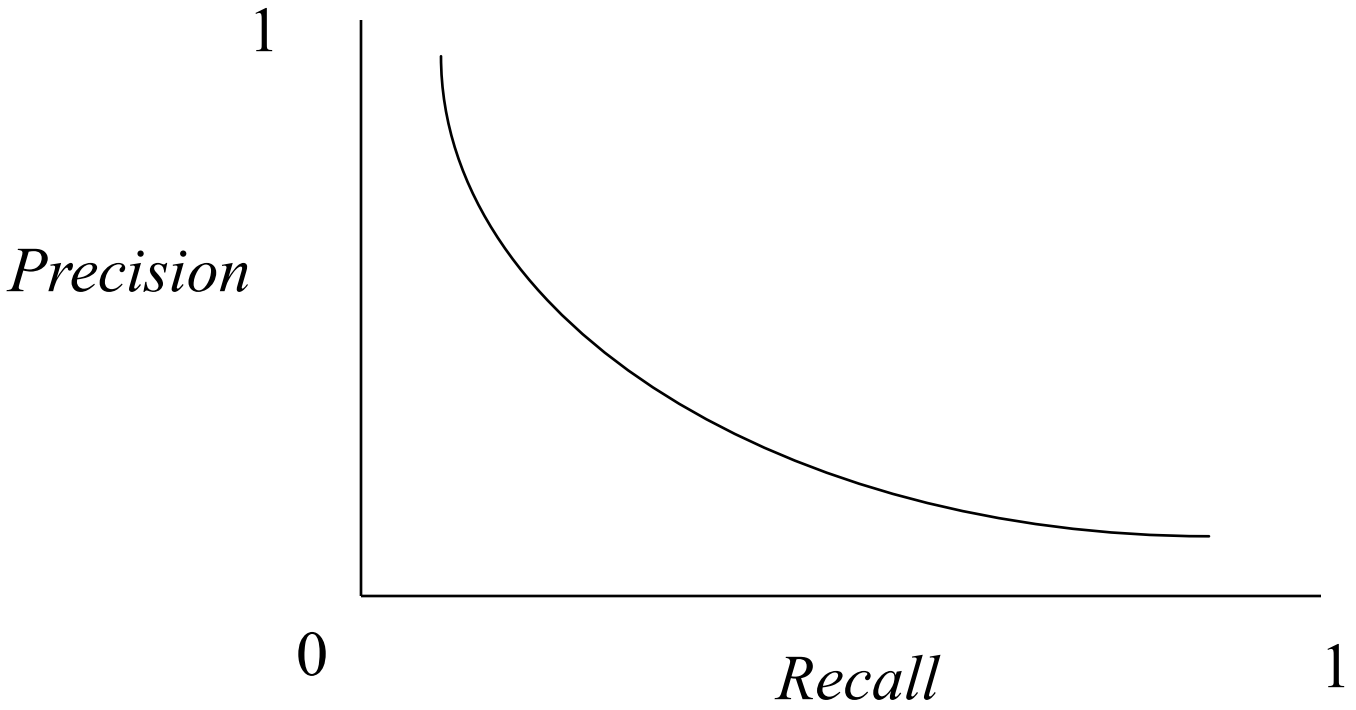
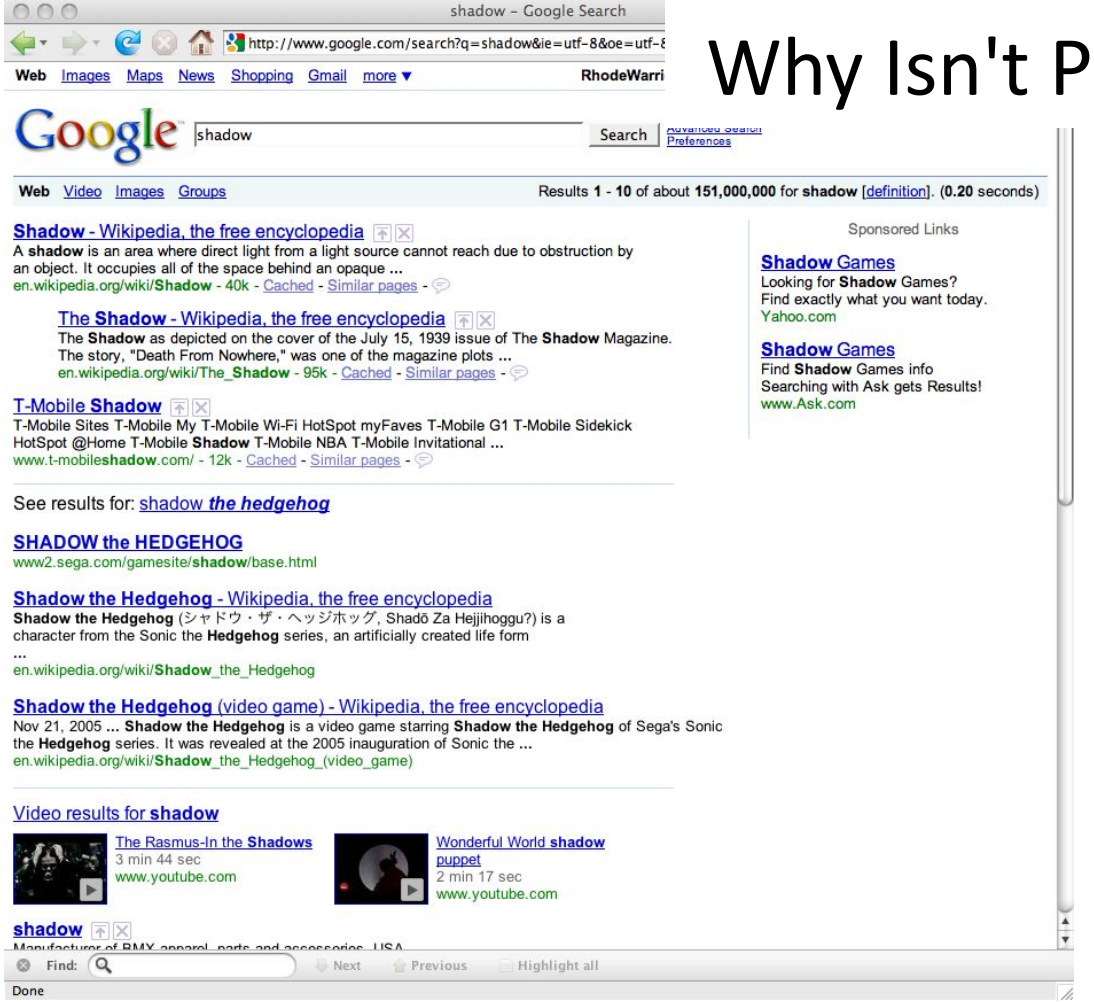


Figure 1.2
in FBY

Why Isn't Precision Always 100%?



What were we really searching for?
Science? Games? Music?

Why Isn't Recall Always 100%?

Virginia Agricultural and Mechanical College?

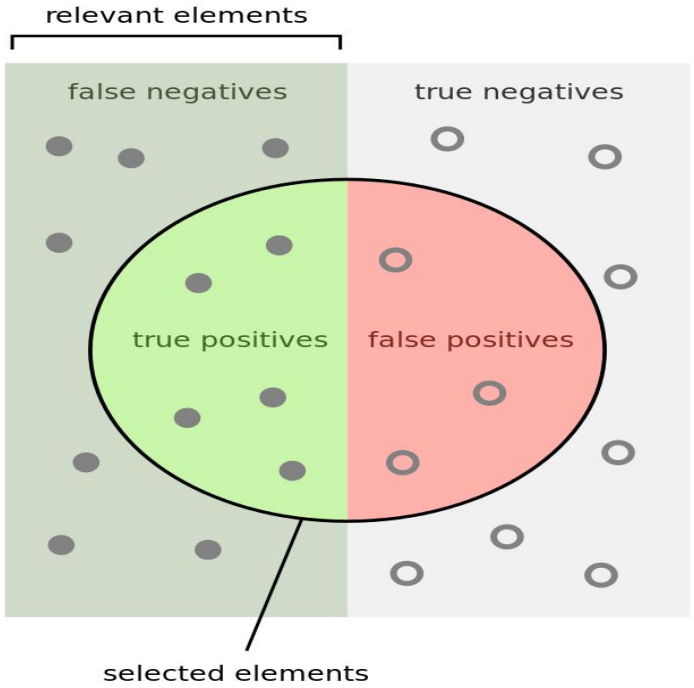
Virginia Agricultural and Mechanical College and Polytechnic Institute?

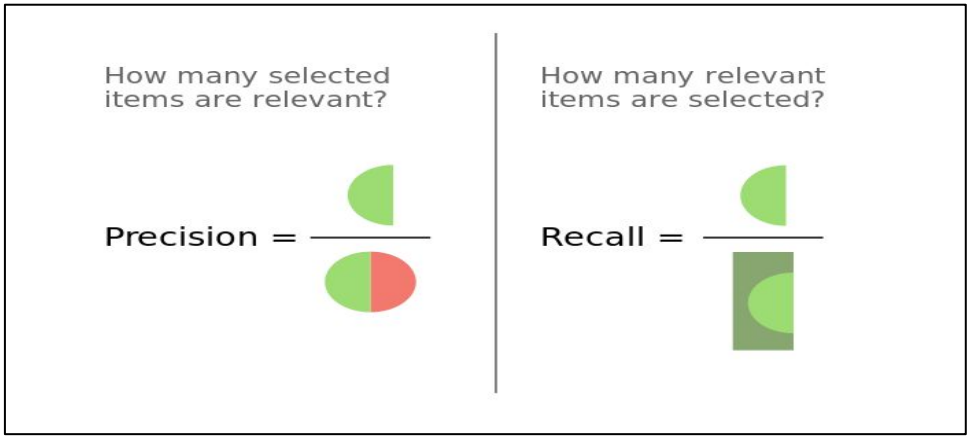
Virginia Polytechnic Institute?

Virginia Polytechnic Institute and State University?

Virginia Tech?

Precision and Recall in Diagrams

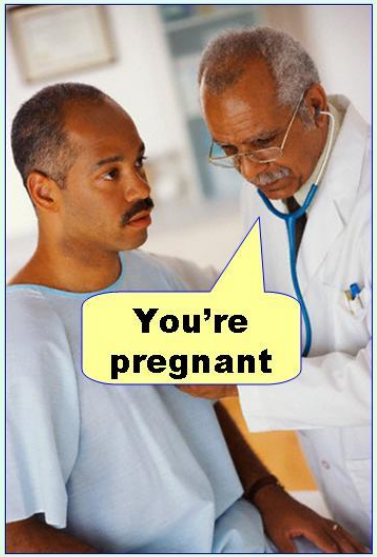




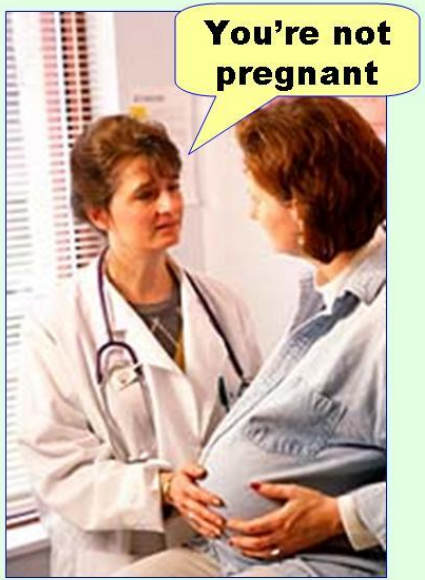
From: [Precision and recall](#) (Wikipedia)

Mnemonic for I vs. II

Type I error
(false positive)



Type II error
(false negative)



From: Posted in Type II error

More than just P&R…

		True condition		
	Total population	Condition positive	Condition negative	Prevalence = $\frac{\Sigma \text{ Condition positive}}{\Sigma \text{ Total population}}$	
Predicted condition	Predicted condition positive	**True positive**	**False positive** (Type I error)	Positive predictive value (PPV), Precision = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Test outcome positive}}$	False discovery rate (FDR) = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Test outcome positive}}$
	Predicted condition negative	**False negative** (Type II error)	**True negative**	False omission rate (FOR) = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Test outcome negative}}$	Negative predictive value (NPV) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Test outcome negative}}$
	Accuracy (ACC) = $\frac{\Sigma \text{ True positive} + \Sigma \text{ True negative}}{\Sigma \text{ Total population}}$	True positive rate (TPR), Sensitivity, Recall = $\frac{\Sigma \text{ True positive}}{\Sigma \text{ Condition positive}}$	False positive rate (FPR), Fall-out = $\frac{\Sigma \text{ False positive}}{\Sigma \text{ Condition negative}}$	Positive likelihood ratio (LR+) = $\frac{\text{TPR}}{\text{FPR}}$	Diagnostic odds ratio (DOR) = $\frac{\text{LR+}}{\text{LR−}}$
		False negative rate (FNR), Miss rate = $\frac{\Sigma \text{ False negative}}{\Sigma \text{ Condition positive}}$	True negative rate (TNR), Specificity (SPC) = $\frac{\Sigma \text{ True negative}}{\Sigma \text{ Condition negative}}$	Negative likelihood ratio (LR−) = $\frac{\text{FNR}}{\text{TNR}}$	

From: Precision and recall (Wikipedia)

Adapting Precision & Recall for Web Scale

- Precision @ n (often n=10)
  - previous query for shadow had ~151M hits!
  - no one will check more than just a few
  - "ten blue links"
    - 10 blue links: are they dead or alive in search? – Econsultancy

- Ex: 
  - Prec@3 of 2/3
  - Prec@4 of 2/4
  - Prec@5 of 3/5

- Relative recall
  - recall was hard enough on bounded collections; all but impossible for Web
  - relative recall is a way of judging SE overlap
    - (recall of system A) / (recall of A + B … + N)

Web Science: Searching the Web

(Part 2 - Indexing the Documents)

CS 432/532

Old Dominion University

Permission has been granted to use these slides from Frank McCown, Michael L. Nelson, Alexander Nwala, Michele C. Weigle

Processing Pages

- After crawling, content is *indexed* and links stored in link database for later analysis
- Text from text-based files (HTML, PDF, MS Word, etc.) are converted into *tokens*
- Stop words may be removed
  - Frequently occurring words like *a*, *the*, *and*, *to*, etc.
  - Stop word (Wikipedia)
  - Most traditional IR systems remove them, but most search engines do not ("to be or not to be")
- Special rules to handle punctuation
  - e-mail → email?
  - Treat "O'Connor" like "boy's"?
  - 123-4567 as one token or two?

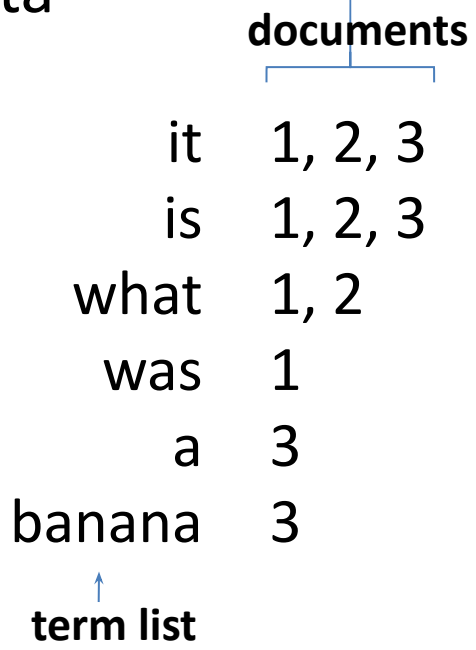
Processing Pages

- *Stemming* may be applied to tokens
  - Technique to remove suffixes from words (e.g., *gamer*, *gaming*, *games* → *gam*)
  - Porter Stemmer very popular algorithmic stemmer
  - Can reduce size of index and improve recall, but precision is often reduced
  - Google and Yahoo use partial stemming
- Tokens may be converted to lowercase
  - Most web search engines are case insensitive

Inverted Index

- *Inverted index,* or *inverted file,* is the data structure used to hold tokens and the pages they are located in

- Example:
  - Doc 1: It is what it was.
  - Doc 2: What is it?
  - Doc 3: It is a banana.



Example Search

- Search for *what is it* is interpreted by search engines as *what* AND *is* AND *it*
  - what: {1, 2} is: {1, 2, 3} it: {1, 2, 3}
  - {1, 2} ∩ {1, 2, 3} ∩ {1, 2, 3} = **{1, 2}**
  - **Answer:** Docs 1 and 2

it	1, 2, 3
is	1, 2, 3
what	1, 2
was	1
a	3
banana	3

- What if we want the phrase "*what is it*"?

Phrase Search

- Phrase search requires *position* of words to be added to inverted index

Doc 1: It is what it was.

Doc 2: What is it?

Doc 3: It is a banana.

it	(1,1) (1,4) (2,3) (3,1)
is	(1,2) (2,2) (3,2)
what	(1,3) (2,1)
was	(1,5)
a	(3,3)
banana	(3,4)

Example Phrase Search

- Search for "*what is it*"
- All items must be in same doc with position in increasing order
- what: (1,3) (2,1) is: (1,2) (2,2) (3,2) it: (1,1) (1,4) (2,3) (3,1)

- **Answer:** Document 2
- Position can be used to give higher scores to terms that are closer
  - "red cars" scores higher than "red bright cars"

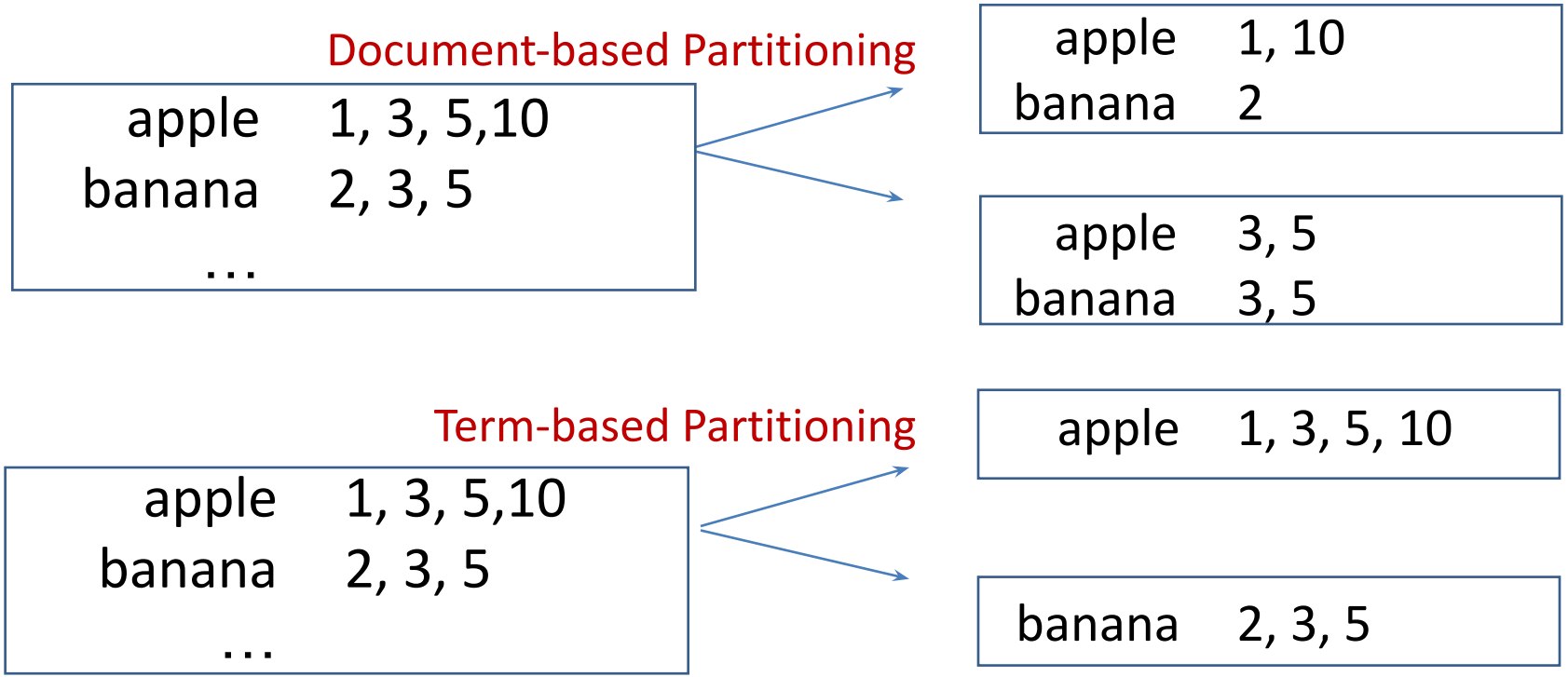
What About *Large* Indexes?

- When indexing the entire Web, the inverted index will be too large for a single computer
- Solution: Break up index onto separate machines/clusters
- Two general methods:
  - Document-based partitioning
  - Term-based partitioning

Google's Data Centers

Image source: [10 of the coolest photos from inside Google's secret data centres]

Partitioning Schemes



Ref: Distributing indexes, *Introduction to Information Retrieval*

Comparing the Two Schemes

Event	Document-Based	Term-Based
Fetch query results	All computers fetch local results and merge	Single machine fetches result for each term
Machine goes down	Some docs not in results	Some terms cannot be processed
Index new doc	Add new terms/docs to single machine	Rebuild index

Guess which scheme Google uses?

Web Science: Searching the Web

(Part 3 - Ranking: TF-IDF)

CS 432/532

Old Dominion University

Permission has been granted to use these slides from Frank McCown, Michael L. Nelson, Alexander Nwala, Michele C. Weigle

If two documents contain the same query terms, how do we determine which one is *more* relevant?

?

Term Frequency

Query for "*dogs*" results in two pages:

TF = 3

Dogs, **dogs**, I love them **dogs**!

Doc A

Dogs are wonderful animals.

Doc B

TF = 1

- Which page should be ranked higher?
- Simple method called **term frequency**: Count number of times the term occurs in the document
- Pages with higher TF get ranked higher

Normalizing Term Frequency

Query for "*dogs*" results in two pages:

TF = 3/6 = 0.5

Dogs, **dogs**, I love them **dogs**!

Doc A

Dogs are wonderful animals.

Doc B

TF = 1/4 = 0.25

- To avoid penalizing shorter documents, TF should be normalized
  - Divide by total number of words in the document
  - Other divisors possible

TF Can Be Spammed!

Watch out!

Dogs, **dogs**, I love them **dogs**! **dogs** **dogs** **dogs** **dogs** **dogs**

TF is susceptible to spamming, so SEs look for unusually high TF values when looking for spam

TF = 8/11 = 0.72

Inverse Document Frequency

- Problem: Some terms are frequently used throughout the corpus and therefore aren't useful when discriminating docs from each other
- Less frequently used terms are more helpful
- IDF(term) = total docs in corpus / docs with term
- Low frequency terms will have high IDF

Inverse Document Frequency

- To keep IDF from growing too large as corpus grows:

  $$IDF(term) = \log_2(\text{total docs in corpus} / \text{docs with term})$$

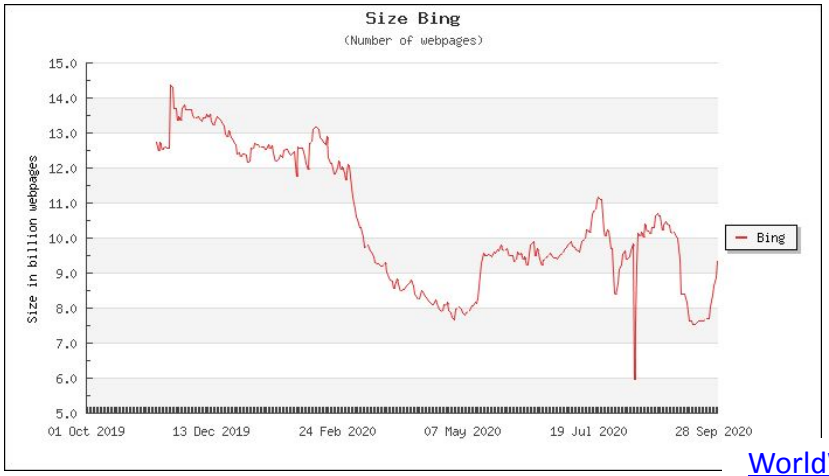
- IDF is not as easy to spam since it involves all docs in corpus
  - Could stuff rare words in your pages to raise IDF for those terms, but people don't often search for rare terms

TF-IDF

- TF and IDF are usually combined into a single score
- TF-IDF = TF × IDF
  = occurrence in doc / words in doc ×
  $\log_2$(total docs in corpus / docs with term)
- When computing TF-IDF score of a doc for *n* terms:
  - Score = TF-IDF($term_1$) + TF-IDF($term_2$) + … + TF-IDF($term_n$)

TF-IDF Example

- Using Bing, compute the TF-IDF scores for 2 documents that contain the words *harding* AND *university*
- Assume Bing has **10 billion documents** indexed

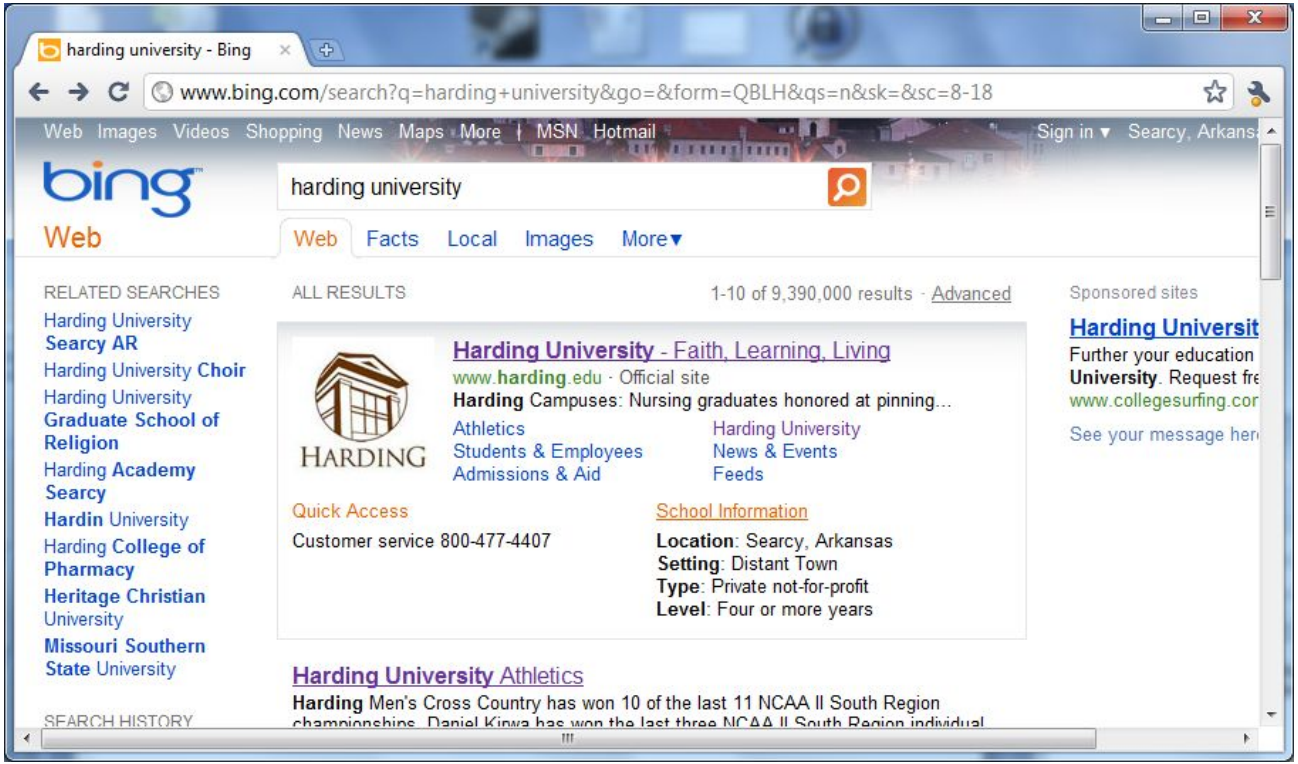


WorldWideWebSize.com (accessed Sep 2020)

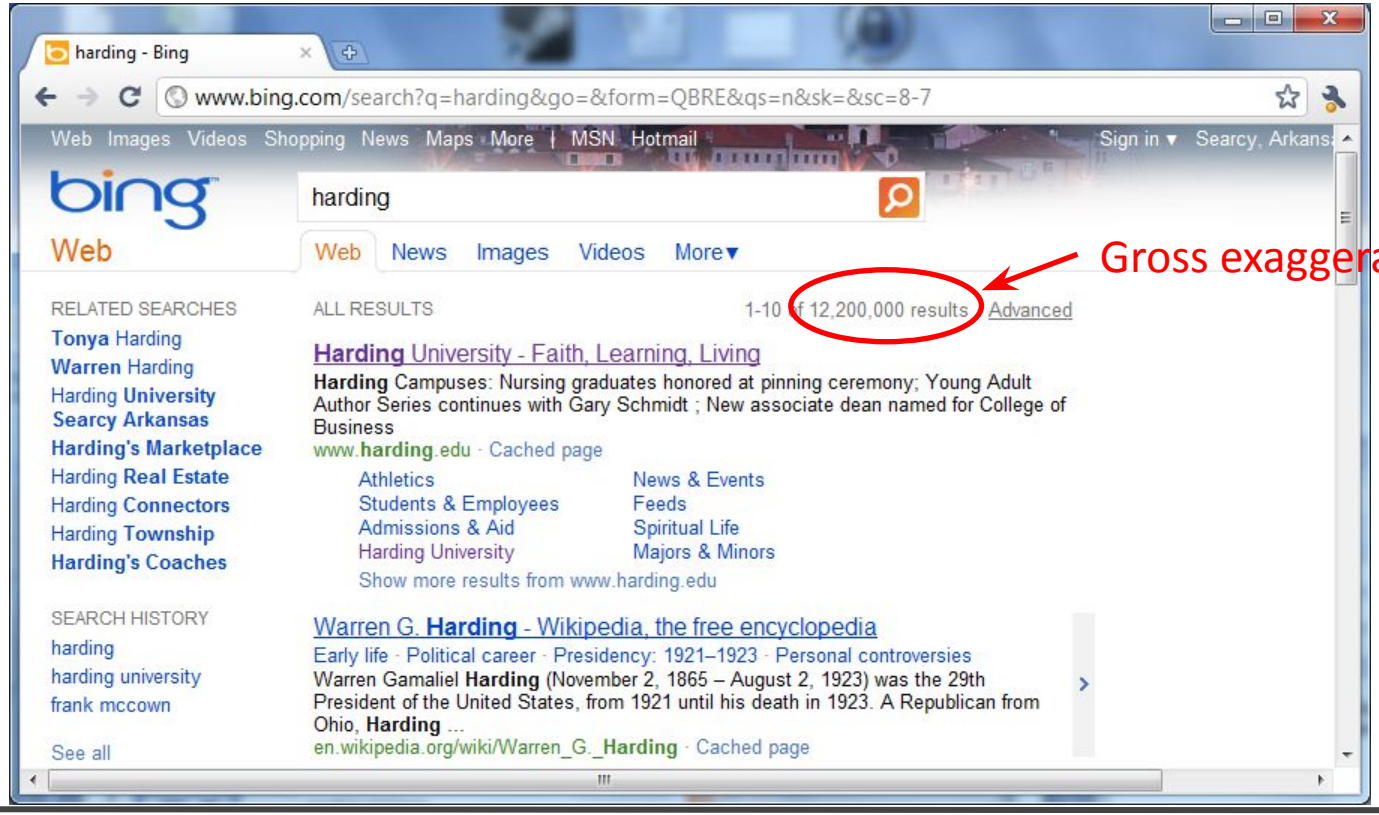
TF-IDF Example

- Actions to perform:
  1. Query Bing with *harding university* to pick 2 docs
  2. Query Bing with just *harding* to determine how many docs contain *harding*
  3. Query Bing with just *university* to determine how many docs contain *university*

1) Search for *harding university* and choose two results

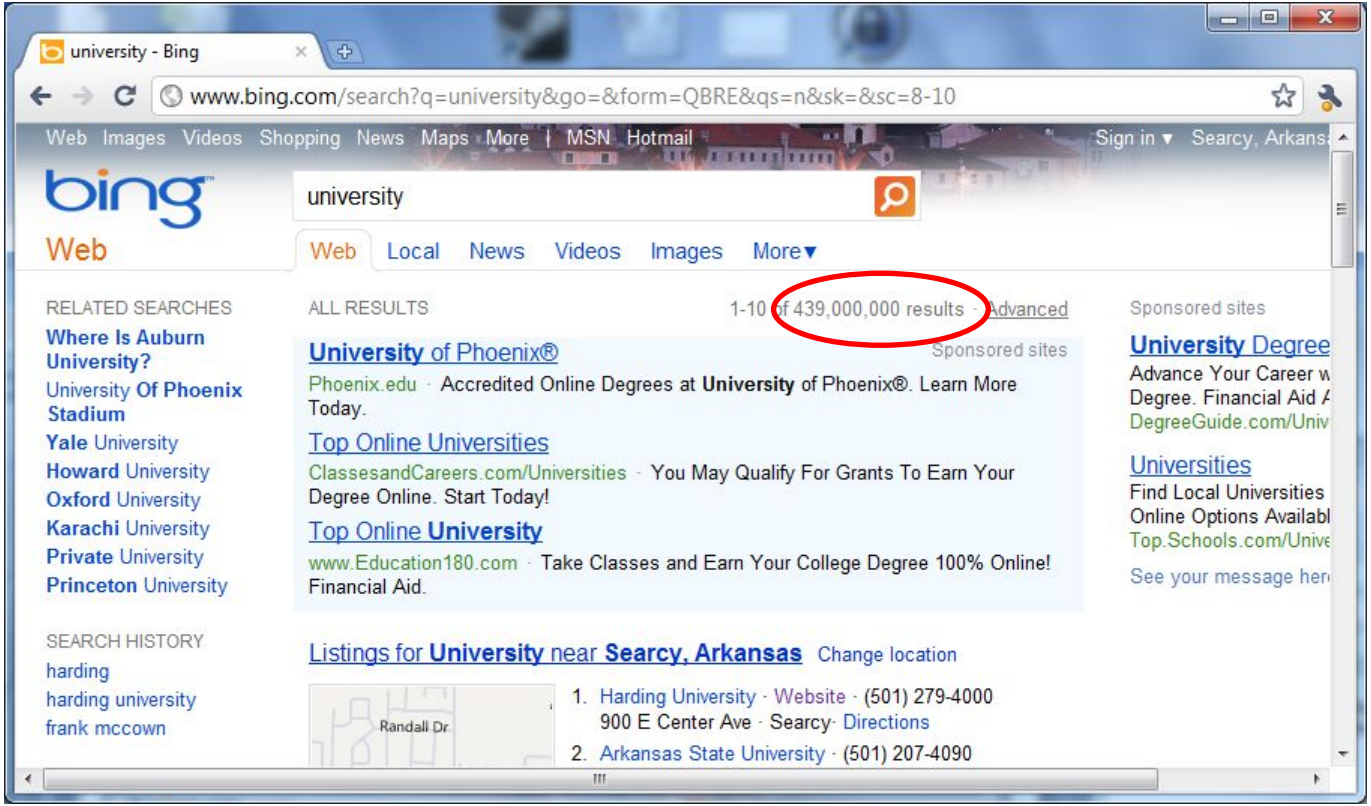


2) Search for *harding*



Gross exaggeration

2) Search for *university*

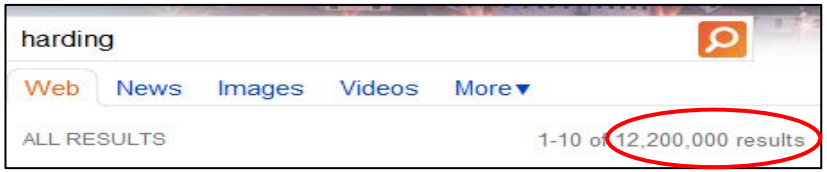


Doc 1: Harding - College of Pharmacy

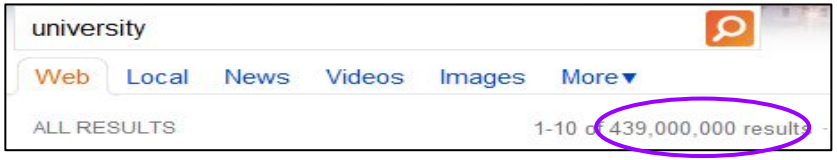
- webpage contains 967 words
  - "harding" appears 19 times
  - "university" appears 13 times

Assume Bing has **10 billion** documents indexed

- TF(harding) = 19 / 967
- IDF(harding) = $\log_2$(**10B** / 12.2M)



- TF(university) = 13 / 967
- IDF(university) = $\log_2$(**10B** / 439M)



Doc 1: Harding - College of Pharmacy

- TF(harding) = 19 / 967 = 0.020
- IDF(harding) = $\log_2$(10B / 12.2M) = 9.679

- TF(university) = 13 / 967 = 0.013
- IDF(university) = $\log_2$(10B / 439M) = 4.510

- TF-IDF(harding) + TF-IDF(university) =
  0.020 × 9.679 + 0.013 × 4.510 = **0.252**

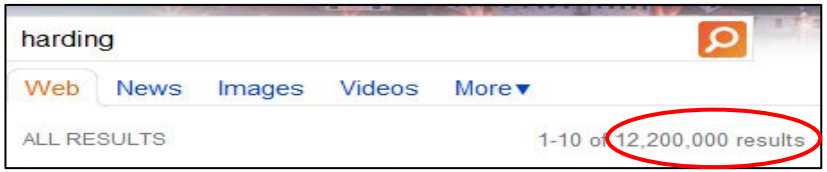
$\log_2 x$ is
$\log(x, 2)$ in R

Doc 2: Harding University (Wikipedia)

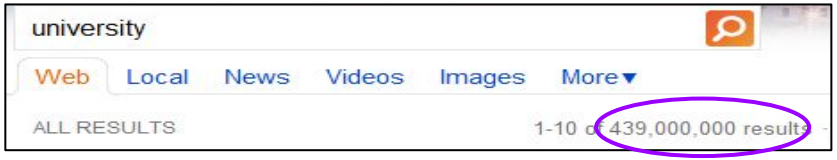
- webpage contains 3135 words
  - "harding" appears 44 times
  - "university" appears 25 times

Assume Bing has **10 billion** documents indexed

- TF(harding) = 44 / 3135
- IDF(harding) = $\log_2$(**10B** / 12.2M)



- TF(university) = 25 / 3135
- IDF(university) = $\log_2$(**10B** / 439M)



Doc 2: Harding University (Wikipedia)

- TF(harding) = 44 / 3135 = 0.014
- IDF(harding) = $\log_2$(10B / 12.2M) = 9.679

- TF(university) = 25 / 3135 = 0.008
- IDF(university) = $\log_2$(10B / 439M) = 4.510

- TF-IDF(harding) + TF-IDF(university) =
  0.014 × 9.679 + 0.008 × 4.510 = **0.172**

Doc 1 = **0.252**

Doc 2 = 0.172

Web Science: Searching the Web

(Part 4 - Ranking: Link-based Metrics, PageRank, and HITS)

CS 432/532

Old Dominion University

Permission has been granted to use these slides from Frank McCown, Michael L. Nelson, Alexander Nwala, Michele C. Weigle

Increasing Relevance

- Index words in URL
- **Weigh** *importance* of terms based on HTML or CSS styles
- Website responsiveness[1]
- Account for last modification date
- Allow for misspellings
- Link-based metrics
- Popularity-based metrics

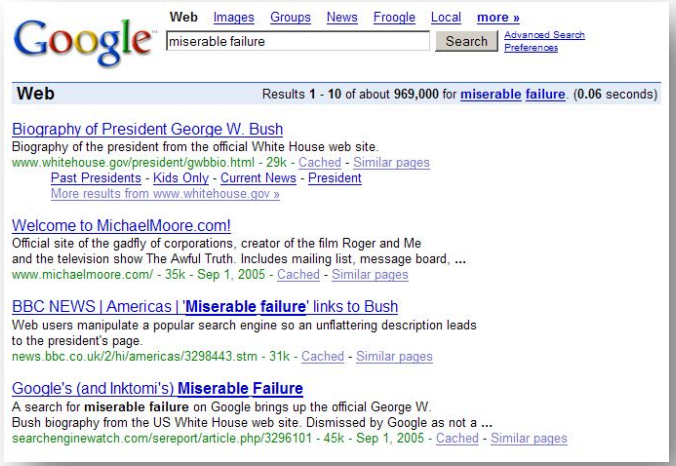
[1] Using site speed in web search ranking

Increasing Relevance

- Index a link's *anchor text* along with the page it points to

```
<a href="skill.html">Ninja skills</a>
```

  - Watch out: Google bombs (Google bombing (Wikipedia))



More info: Google Kills Bush's Miserable Failure Search & Other Google Bombs

Increasing Relevance

- Index words in URL
- **Weigh** *importance* of terms based on HTML or CSS styles
- Website responsiveness[1]
- Account for last modification date
- Allow for misspellings
- **Link-based metrics**
- Popularity-based metrics

[1] Using site speed in web search ranking

Link Analysis

- Content analysis is useful, but combining with link analysis allow us to rank pages much more successfully

- 2 popular methods
  - Sergey Brin and Larry Page's *PageRank*
  - Jon Kleinberg's Hyperlink-Induced Topic Search (*HITS*)

What Does a Link Mean?

- A recommends B
- A specifically does **not** recommend B
- B is an authoritative reference for something in A
- A & B are about the same thing (topic locality)

PageRank

- Developed by Brin and Page (Google) while Ph.D. students at Stanford
- Links are a recommendation system
  - The more links that point to you, the more important you are
  - Inlinks from important pages are weightier than inlinks from unimportant pages
  - The more outlinks you have, the less weight your links carry

Page et al., The PageRank citation ranking: Bringing order to the web, 1998.

Image credit: http://scrapetv.com/News/News%20Pages/Technology/images/sergey-brin-larry-page.jpg

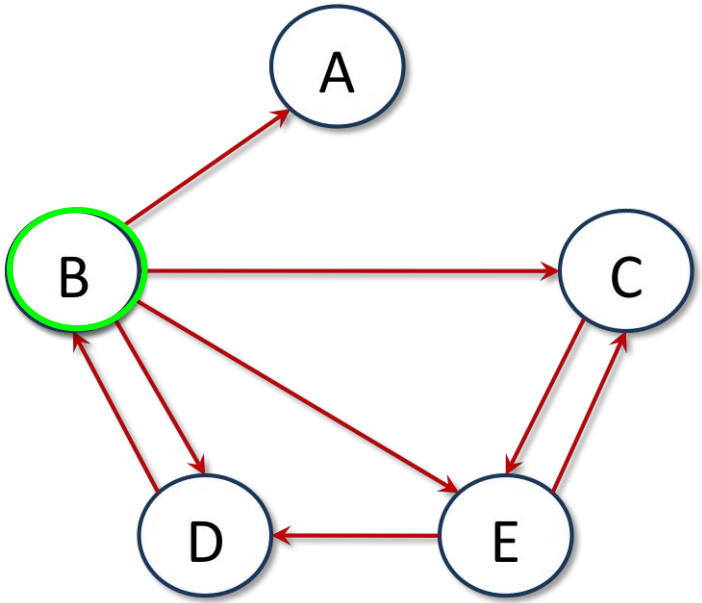
Random Surfer Model

- Model helpful for understanding PageRank
- The Random Surfer starts at a randomly chosen page and selects a link at random to follow

- PageRank of a page reflects the probability that the surfer lands on that page after clicking any number of links

Image credit: http://missloki84.deviantart.com/art/Random-Surfer-at-Huntington-Beach-319287873

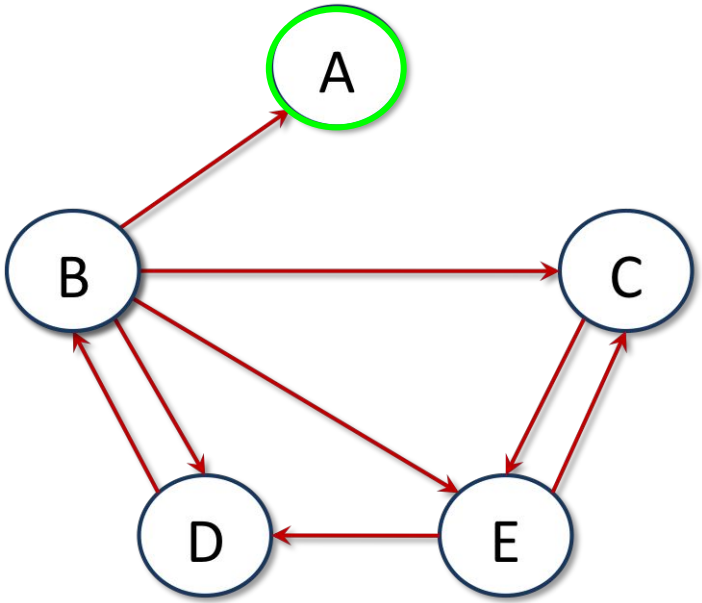
Example of Random Surfer



Start at: B
¼ probability of going to A
¼ probability of going to C
¼ probability of going to D
¼ probability of going to E

Choose: E
½ probability of going to C
½ probability of going to D

Problem 1: Dangling Node



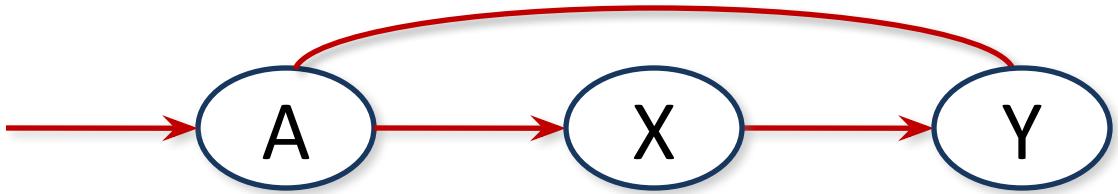
What if we go to A? We're stuck at a dead-end!

Solution: **Teleport** to any other page at random

Problem 2: Infinite Loop

What if we get stuck in an a **cycle**?

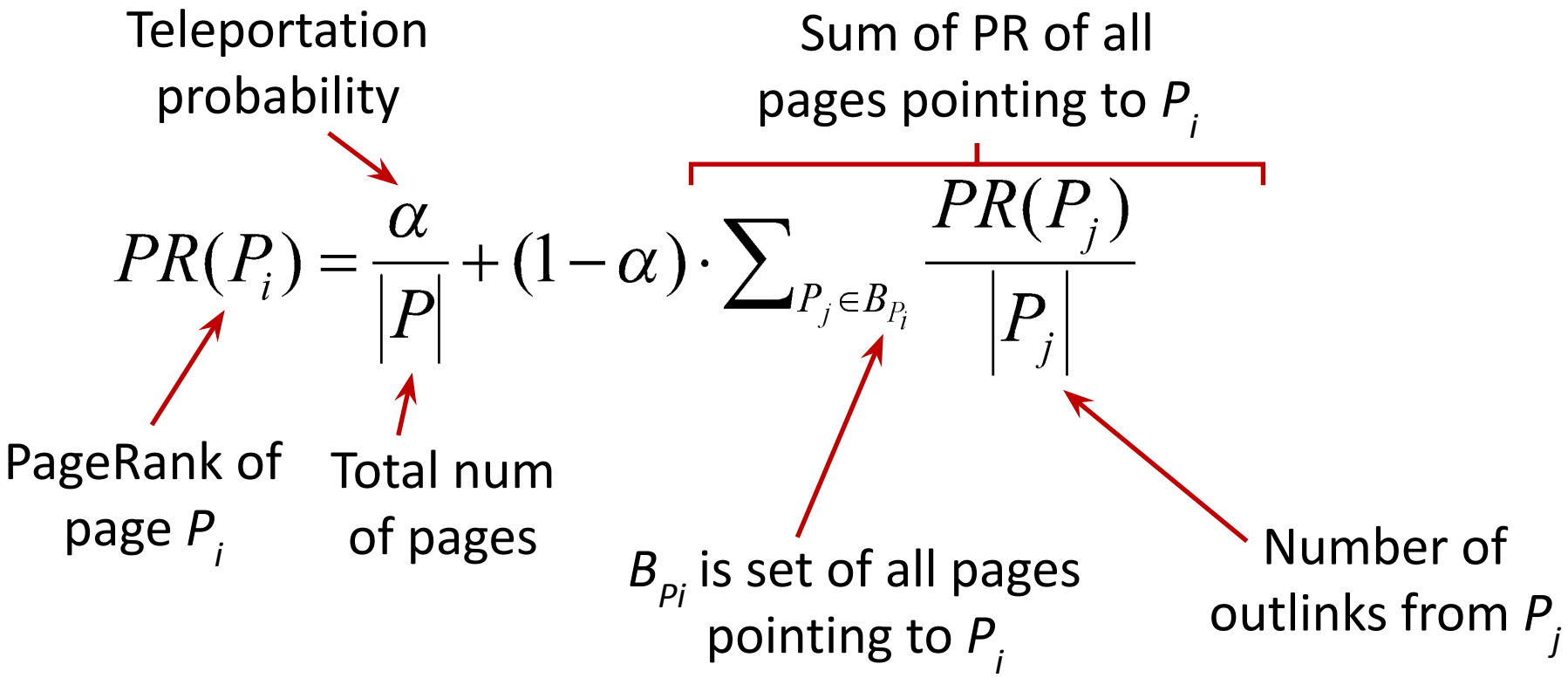
Solution: Teleport to any other page at random



Rank Sinks

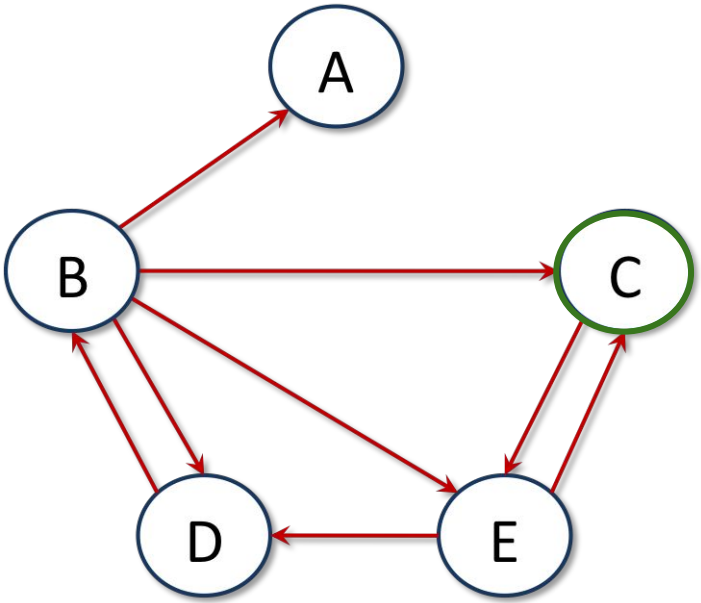
- Dangling nodes and cycles are called **rank sinks**
- Solution is to add a **teleportation probability α** to every decision
- α% chance of getting bored and jumping somewhere else, (1-α)% chance of choosing one of the available links
- $\alpha = 0.15$ is typical

PageRank Definition



PageRank Example

$$PR(P_i) = \frac{\alpha}{|P|} + (1-\alpha) \cdot \sum_{P_j \in B_{P_i}} \frac{PR(P_j)}{|P_j|}$$



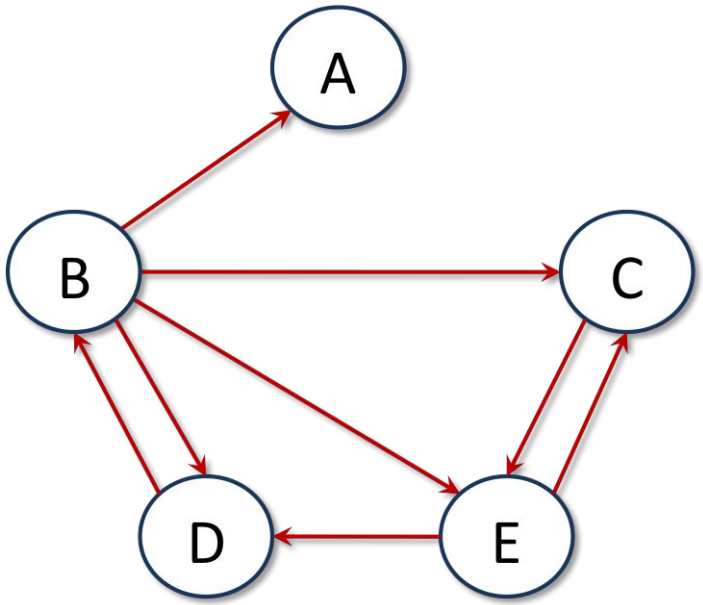
PR(C) = .15/5 + .85 × (PR(B)/4 + PR(E)/2)

Problem: What is PR(B) and PR(E)?

Solution: Give all pages same PR to start (1/|*P*|) & iteratively calculate new PR

PageRank Example

$$PR(P_i) = \frac{\alpha}{|P|} + (1-\alpha) \cdot \sum_{P_j \in B_{P_i}} \frac{PR(P_j)}{|P_j|}$$



PR(A) = .03 + .85 × PR(B)/4
= .03 + .85 x **.2**/4 = .0725

PR(B) = .03 + . 85 × PR(D)/1
= .03 + .85 x **.2**/1 = .2

α/|P| = .15 / 5 = **.03**

initial PR = 1/|P|= 1/5 = **.2**

PR(C) = .03 + .85(PR(B)/4 + PR(E)/2)
= .03 + .85(**.2**/4 + **.2**/2) = .1575

PR(D) = .03 + .85(PR(B)/4 + PR(E)/2) = .1575

PR(E) = .03 + .85(PR(B)/4 + PR(C)/1) = .2425

Calculating PageRank

- PageRank is computed over and over until it converges, around 20 times

- Can also be calculated efficiently using matrix multiplication
  - we'll see this a little later

Computing PageRank

original paper version, sums to *N* (number of pages in graph)

$$PR(A) = 1 - d + d\left(\frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)} + \cdots\right)$$

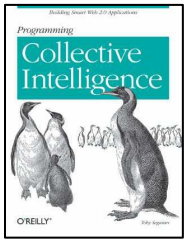
more common version, sums to 1

$$PR(A) = \frac{1-d}{N} + d\left(\frac{PR(B)}{L(B)} + \frac{PR(C)}{L(C)} + \frac{PR(D)}{L(D)} + \cdots\right).$$

d = damping factor
L() = out-degree of a page
PR() = PageRank of a page (all nodes start with *PR()* = 1 or 1/*N*)

Calculating PageRank for a Page, One Iteration

Ref: *Programming Collective Intelligence*, Ch 4



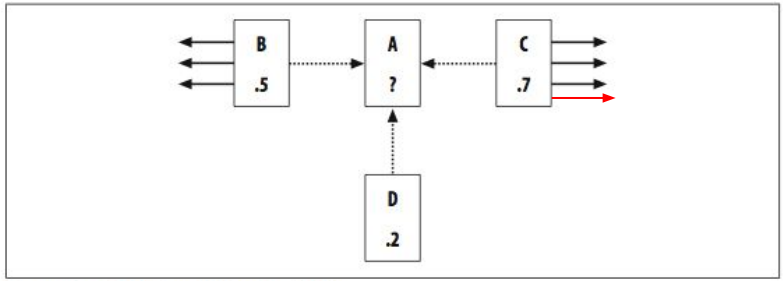


Figure 4-3. Calculating the PageRank of A

Fig 4-3 needs an extra link from C to match the text

```
PR(A) = (1-0.85) + 0.85 * (PR(B)/links(B) + PR(C)/links(C) + PR(D)/links(D))
      =   0.15   + 0.85 * (0.5/4          + 0.7/5          + 0.2/1)
      =   0.15   + 0.85 * (0.125          + 0.14           + 0.2)
      =   0.15   + 0.85 * 0.465
      =   0.54525
```

damping factor (*d*) = 0.85 (probability surfer landed on page by following a link)
1-*d* = 0.15 (probability surfer landed on page at "random")

since this is the original version where PR sums to *N*, and we've only accounted for ~1.95 of total PR, pages not shown must be holding PR

PageRank Definition as Matrix

Stated as a matrix equation where **R** is the vector of PageRank values and **T** the matrix for transition probabilities:

$$R = TR$$

where T_{ij} is the probability of going from page *j* to i*:*

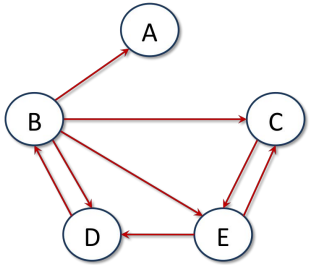
$$T_{ij} = \frac{\alpha}{|P|} + (1-\alpha)\frac{1}{|P_j|}$$

Total num of pages

Total outlinks from page *j*

if a link from page *j* to *i* exists, otherwise

$$T_{ij} = \frac{\alpha}{|P|}$$



PageRank Matrix Example

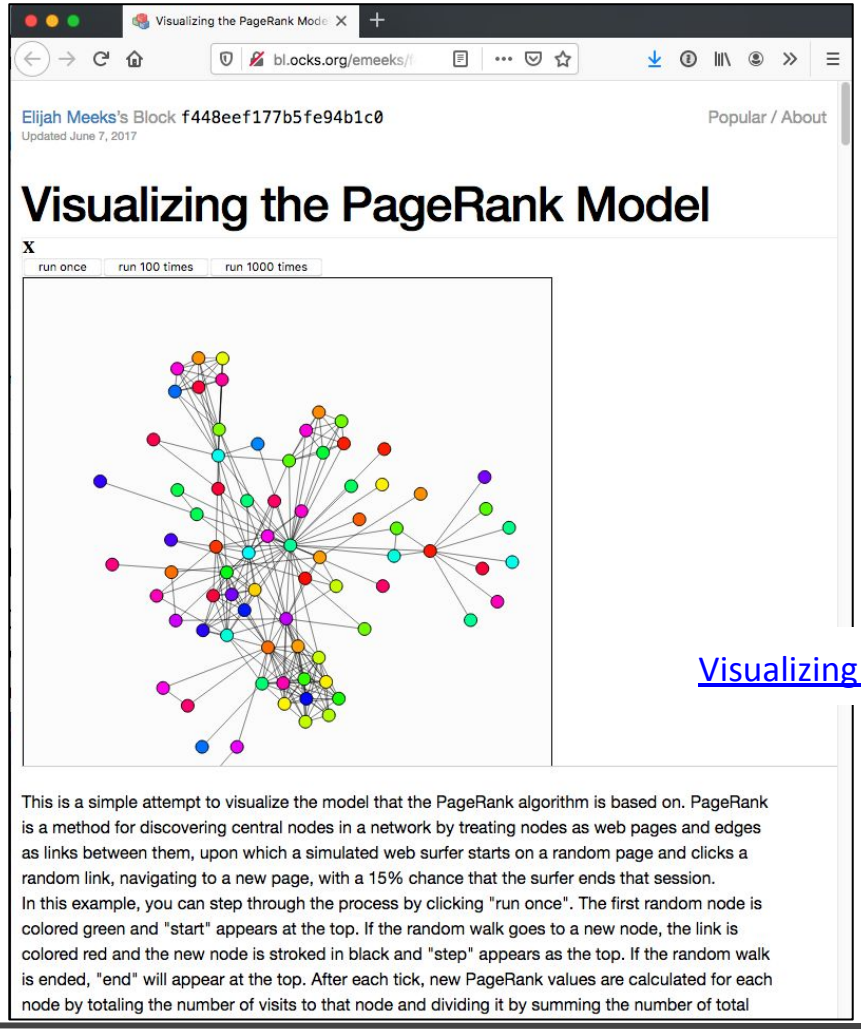
No link from C to A, so value = .15/5

$$\begin{bmatrix} PR_A \\ PR_B \\ PR_C \\ PR_D \\ PR_E \end{bmatrix} = \begin{bmatrix} . & . & .03 & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ . & . & . & . & . \\ . & .243 & . & . & . \end{bmatrix} \begin{bmatrix} .2 \\ .2 \\ .2 \\ .2 \\ .2 \end{bmatrix}$$

$$T_{EB} = \frac{\alpha}{|P|} + (1-\alpha)\frac{1}{|P_B|}$$

= .15/5 + (1 – .15)/4 = **.243**

Init PR values
1/|P|



[Visualizing the PageRank Model]

Check Your PageRank…

- Check Google page rank instantly

- 9/10: google.com, cnn.com
- 7/10: www.cs.odu.edu
- 5/10: ws-dl.blogspot.com
- 2/10: lsufootball.net

updated Feb 2020

PageRank Issues

- Rich-get-richer phenomenon
  - May be difficult for new pages with few inlinks to compete with older, highly linked pages with high PageRank
  - Could promote small fraction of new pages at random[1] or add decay factor to links
- Study[2] showed just counting number of inlinks gives similar ranking as PageRank
  - Study was on small scale and pages were not necessarily "typical"
  - Counting inlinks more susceptible to spamming

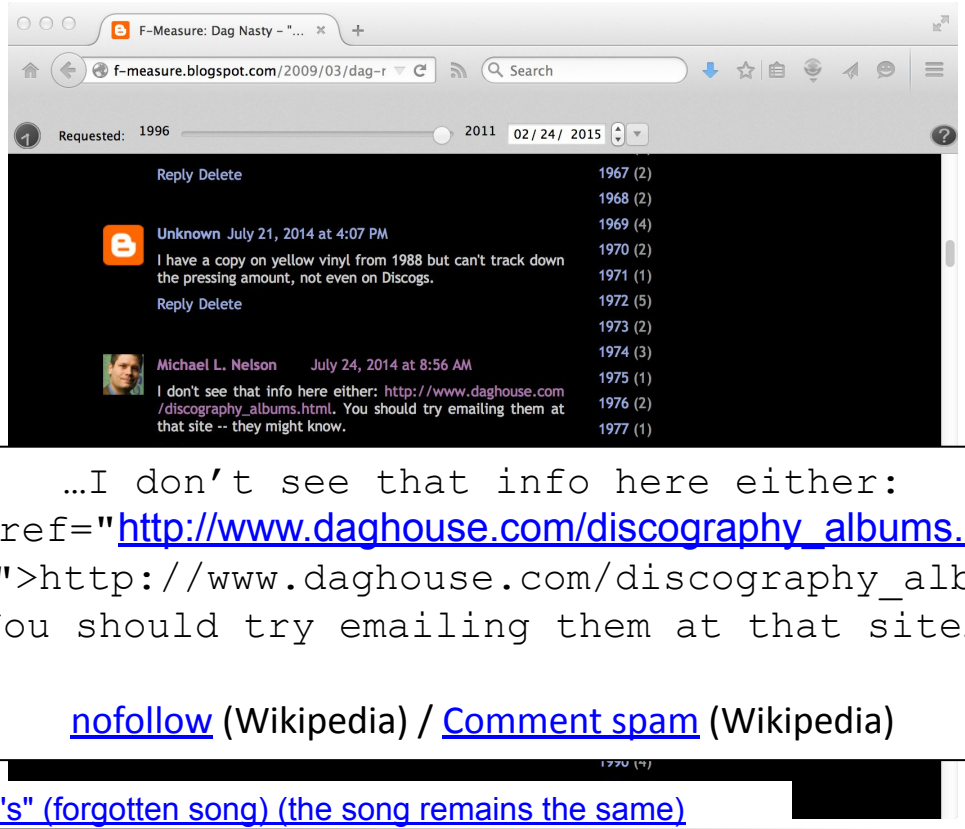
[1]Pandey et al., Shuffling a stacked deck, VLDB 2005,
[2]Amento et al., Does "authority" mean quality?, SIGIR 2000

Link Spam

- There is a strong economic incentive to rank highly in a SERP
- "White hat" SEO firms follow published guidelines to improve customer rankings[1]
- To boost PageRank, "black hat" SEO practices include:
  - Building elaborate link farms
  - Exchanging reciprocal links
  - Posting links on blogs and forums

[1] Google's Webmaster Guidelines

Trying to Stealing PageRank…



…I don't see that info here either:
<a href="http://www.daghouse.com/discography_albums.html"
rel="nofollow">http://www.daghouse.com/discography_albums.html.
You should try emailing them at that site…

nofollow (Wikipedia) / Comment spam (Wikipedia)

Dag Nasty - "Wig Out at Denko's" (forgotten song) (the song remains the same)

Combating Link Spam

- Sites like Wikipedia can discourage links that only promote PageRank by using "nofollow"

```
<a href="http://somesite.com/" rel="nofollow">Go here!</a>
```

- Davison[1] identified 75 features for comparing source and destination pages
  - Overlap, identical page titles, same links, etc.
- TrustRank[2]
  - Bias teleportation in PageRank to set of trusted web pages

[1]Davison, Recognizing nepotistic links on the web, 2000
[2]Gyöngyi et al., Combating web spam with TrustRank, VLDB 2004

Combating Link Spam

- SpamRank[1]
  - PageRank for whole Web has power-law distribution
  - Penalize pages whose supporting pages do not approximate power-law distribution
- Anti-TrustRank[2]
  - Give high weight to known spam pages and propagate values using PageRank
  - New pages can be classified spam if large contribution of PageRank from known spam pages, or if high Anti-TrustRank

[1]Benczúr et al., SpamRank – fully automated link spam detection, AIRWeb 2005
[2]Krishnan & Raj, Web spam detection with Anti-TrustRank, AIRWeb 2006

HITS

Img credit: http://scgp.stonybrook.edu/archives/6084

- Hyperlink-induced topic search (HITS) by Jon Kleinberg[1]
- **Hub**: page with outlinks to informative web pages
- **Authority**: informative/authoritative page with many inlinks

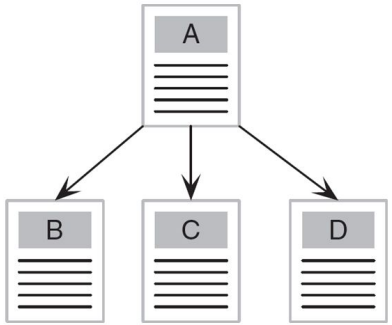


Figure 5.6. Page A is a hub.

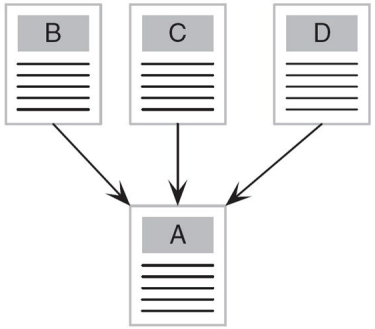


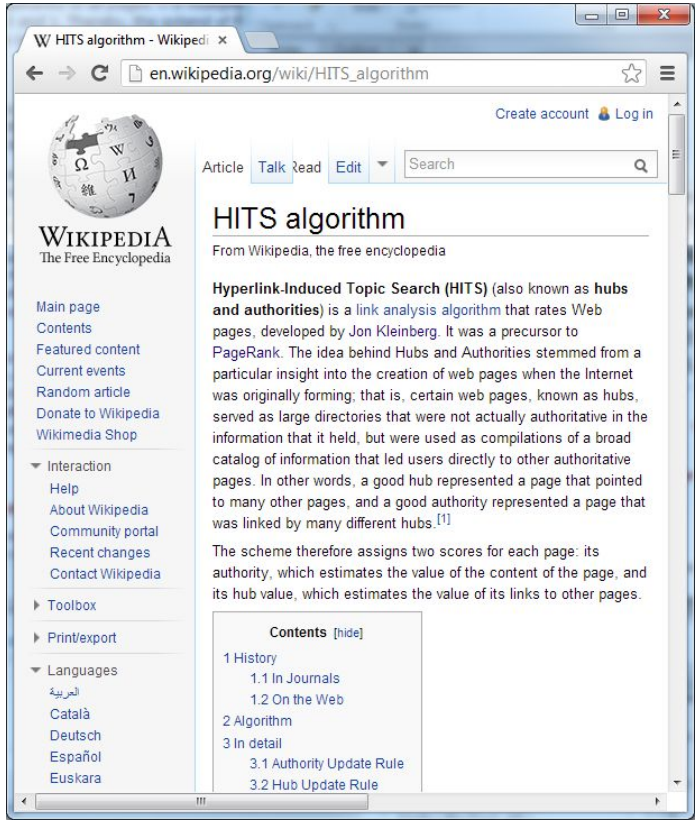
Figure 5.5. Page A is an authority.

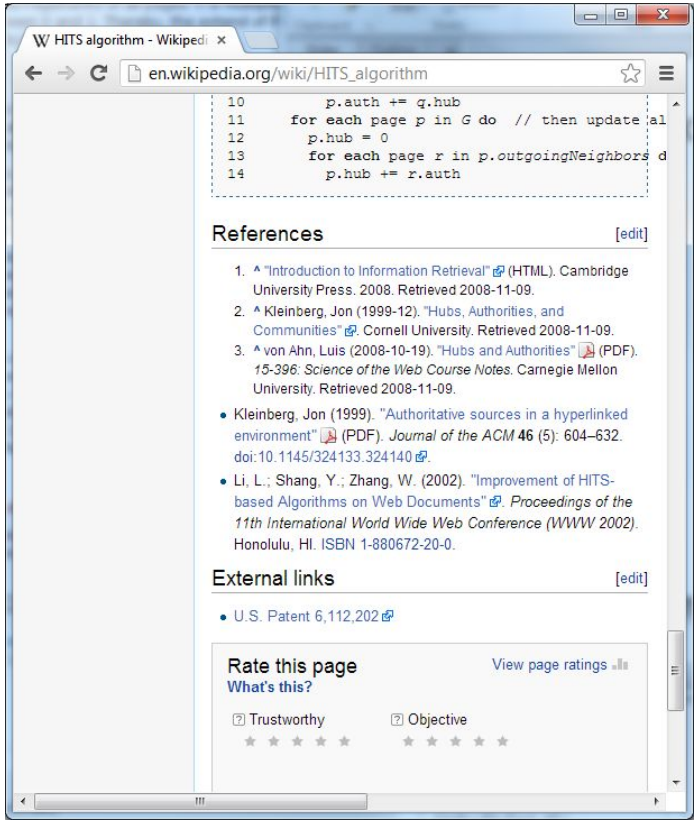
Figures from Levene (2010), *An Introduction to Search Engines and Web Page Navigation*

- Recursive definition:
  - Good *hubs* point to good *authorities*
  - Good *authorities* are pointed to by good *hubs*

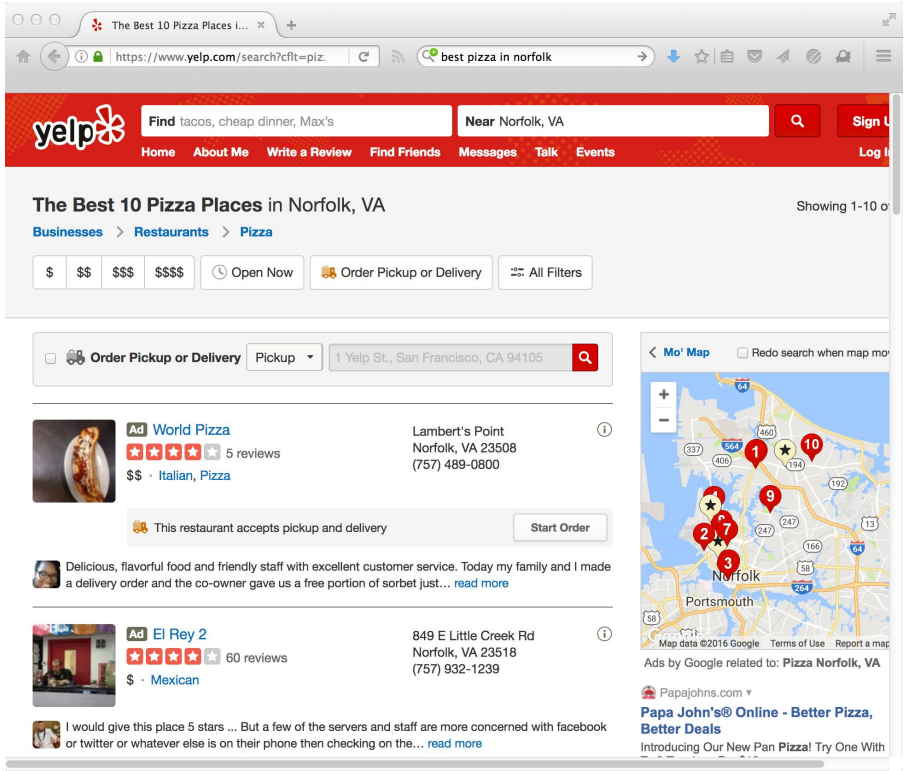
[1]Kleinberg, Authoritative sources in a hyperlinked environment, *J. ACM*, 1999

Good Authority & Hub?

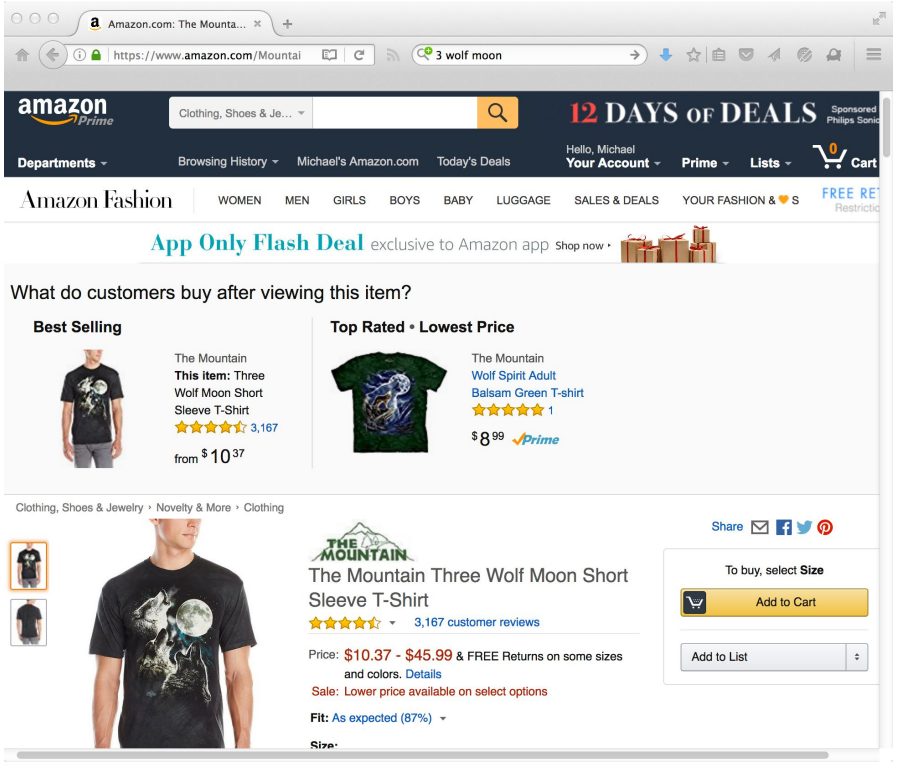




A Good Hub


The Best 10 Pizza Places i...
https://www.yelp.com/search?cflt=piz
best pizza in norfolk
yelp
Find tacos, cheap dinner, Max's
Near Norfolk, VA
Sign U
Home
About Me
Write a Review
Find Friends
Messages
Talk
Events
Log I
The Best 10 Pizza Places in Norfolk, VA
Showing 1-10 o
Businesses > Restaurants > Pizza
$
$$
$$$
$$$$
Open Now
Order Pickup or Delivery
All Filters
Order Pickup or Delivery
Pickup
1 Yelp St., San Francisco, CA 94105
Ad World Pizza
5 reviews
$$ · Italian, Pizza
Lambert's Point
Norfolk, VA 23508
(757) 489-0800
This restaurant accepts pickup and delivery
Start Order
Delicious, flavorful food and friendly staff with excellent customer service. Today my family and I made a delivery order and the co-owner gave us a free portion of sorbet just… read more
Ad El Rey 2
60 reviews
$ · Mexican
849 E Little Creek Rd
Norfolk, VA 23518
(757) 932-1239
I would give this place 5 stars ... But a few of the servers and staff are more concerned with facebook or twitter or whatever else is on their phone then checking on the… read more
Mo' Map
Redo search when map mo
Norfolk
Portsmouth
Map data ©2016 Google
Terms of Use
Report a map
Ads by Google related to: Pizza Norfolk, VA
Papajohns.com
Papa John's® Online - Better Pizza, Better Deals
Introducing Our New Pan Pizza! Try One With

A Good Authority



In the Bad Old Days, Discovery Was Difficult…

- ford.com, toyota.com, etc. don't describe themselves as "automobile manufacturers", though a query for those terms arguably should return those companies

```
<title>Ford Motor Company: Cars, Trucks, SUVs, Hybrids, Parts - Ford </title>

<meta http-equiv="Content-Type" content="charset=utf-8" />
<meta name="KEYWORDS" content="ford, ford motor company, ford vehicles, ford dealers, ford motors, f
<meta name="DESCRIPTION" content="Ford Motor Company maker of cars, trucks, SUVs and other vehicles.
```

- harvard.edu is clearly canonical for a query of "Harvard", even though it uses the term less frequently than many other pages
- Many search engines of the day (ca. late 90s) could not "find themselves"

HITS Algorithm

1. Retrieve pages most relevant to search query[1] → *root set*
2. Retrieve all pages linked to/from the root set → *base set*
3. Perform authority and hub calculations iteratively on all nodes in the subgraph
4. When finished, every node has an authority score and hub score

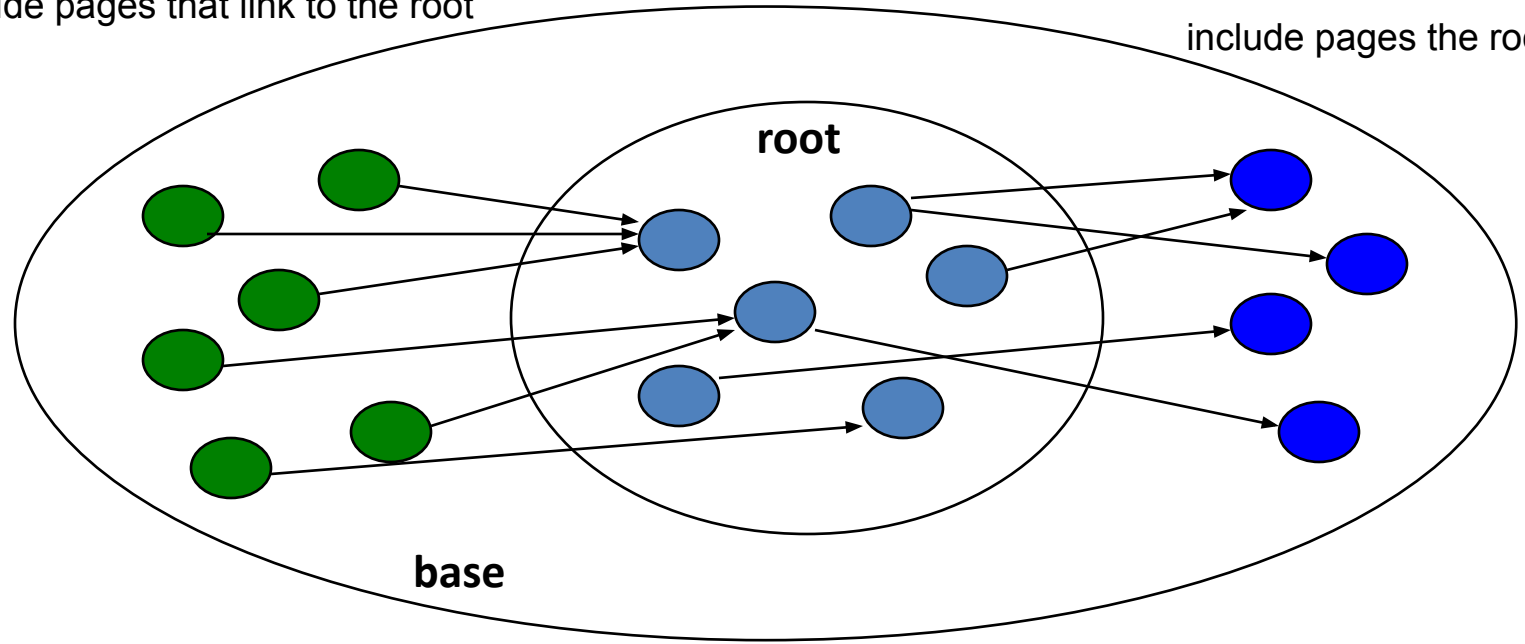
[1] note: you apply HITS to other search engines

Levene (2010), An Introduction to Search Engines and Web Page Navigation

Idea: Use Initial Search Results as "Root"

include pages that link to the root

include pages the root links to



now you have a subgraph to work with

Calculate H and A

E is set of all directed edges in subgraph

e_{qp} is edge from page *q* to *p*

$$H(p) = \sum_{q:e_{pq} \in E} A(q)$$

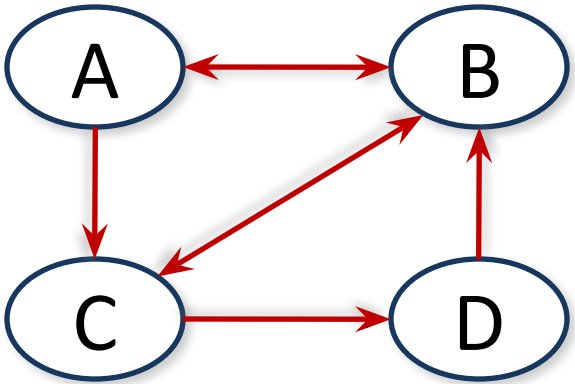
outlinks

$$A(p) = \sum_{q:e_{qp} \in E} H(q)$$

inlinks

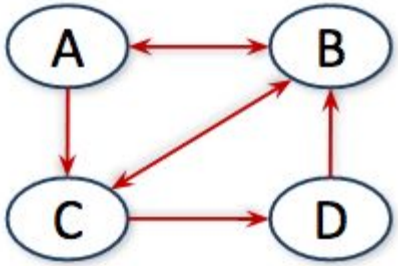
- *H* and *A* scores computed repetitively until they converge, about 10-15 iterations
- Can also be calculated efficiently using matrix multiplication

Example Subgraph



	A	B	C	D
A	0	1	1	0
B	1	0	1	0
C	0	1	0	1
D	0	1	0	0

Adjacency matrix



Authority Scores

Best authority

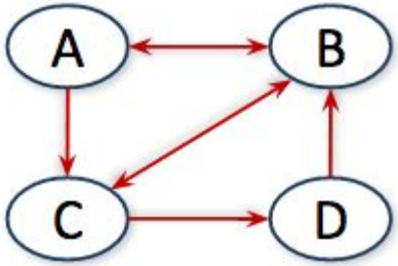
$$\begin{bmatrix} A_A \\ A_B \\ A_C \\ A_D \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \times \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 3 \\ 2 \\ 1 \end{bmatrix}$$

Authority scores

Transposed adjacency matrix

Initial hub scores

Resulting auth scores



Hub Scores

Best hub

$$\begin{bmatrix} H_A \\ H_B \\ H_C \\ H_D \end{bmatrix} = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 \end{bmatrix} \times \begin{bmatrix} 1 \\ 3 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 3 \\ 4 \\ 3 \end{bmatrix}$$

Hub scores

Adjacency matrix

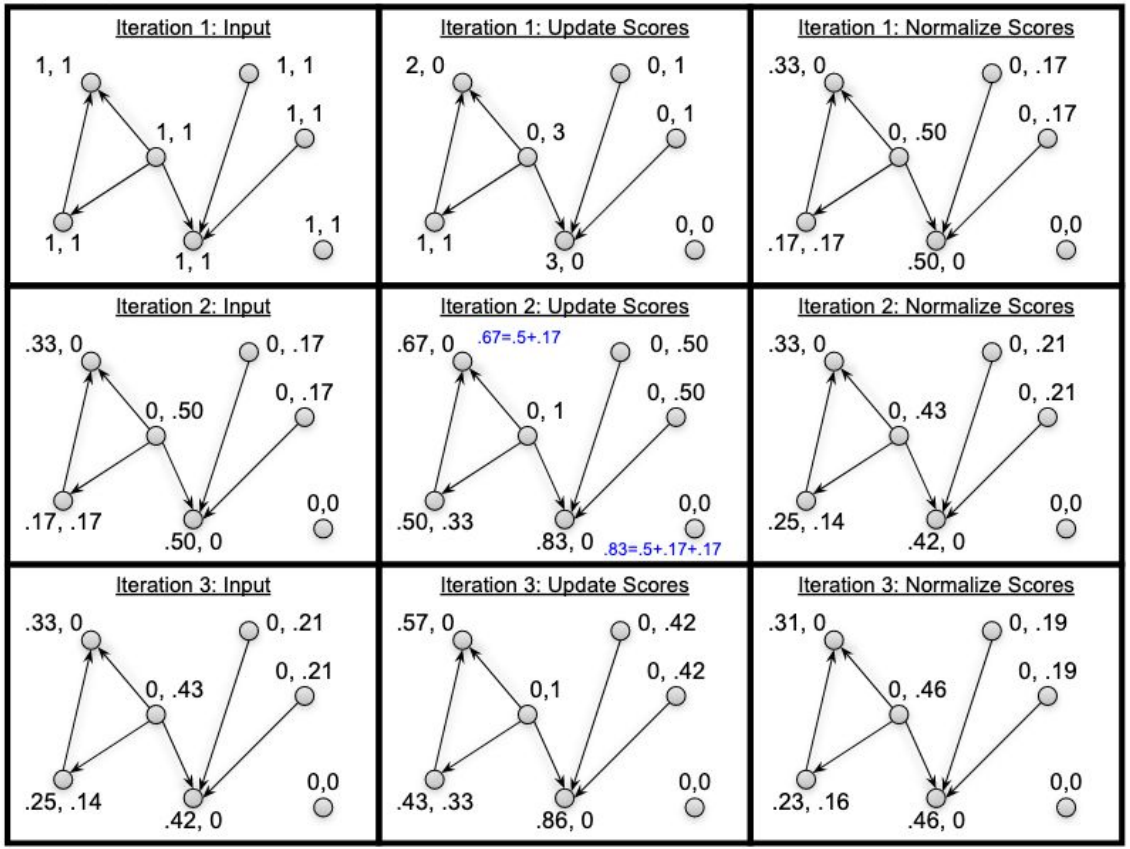
Init auth scores

Resulting hub scores

Fig 10.3 from *Search Engines: Information Retrieval in Practice*

HITS Example

(Authority, Hub)



Problems with HITS

- Has not been widely used
  - IBM holds patent
- Query dependence
  - Later implementations have made query independent
- Topic drift
  - Pages in expanded base set may not be on same topic as root pages
  - Solution is to examine link text when expanding

Levene (2010), An Introduction to Search Engines and Web Page Navigation

Objectives

- Describe the main steps needed for web search.
- Describe what a web crawler does.
- Explain the difference between precision and recall.
- Explain the difference between false positives and false negatives.
- Explain TF-IDF and how it is used to determine relevance in web search.
- Explain the importance of inlinks and outlinks in the Page Rank algorithm.
- Given a query term, compute TF-IDF for the term in a set of documents.