Web Crawling and Basic Text Analysis

Hongning Wang
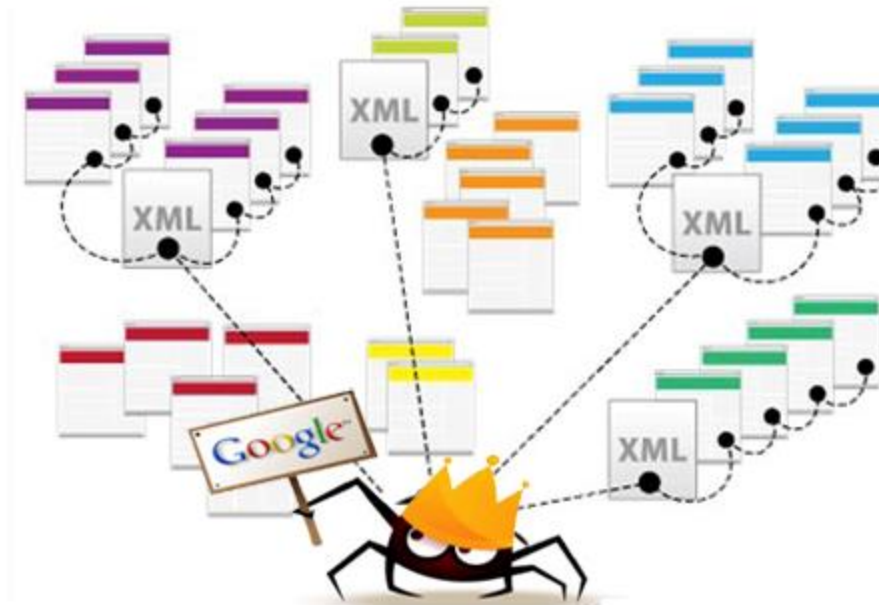## CS@UVa

Abstraction of search engine architecture

Indexed corpus

Crawler

Ranking procedure

Doc Analyzer

Doc Representation

Indexer

Index

Feedback ← **Evaluation**

(Query)

Query Rep

User

Ranker

results
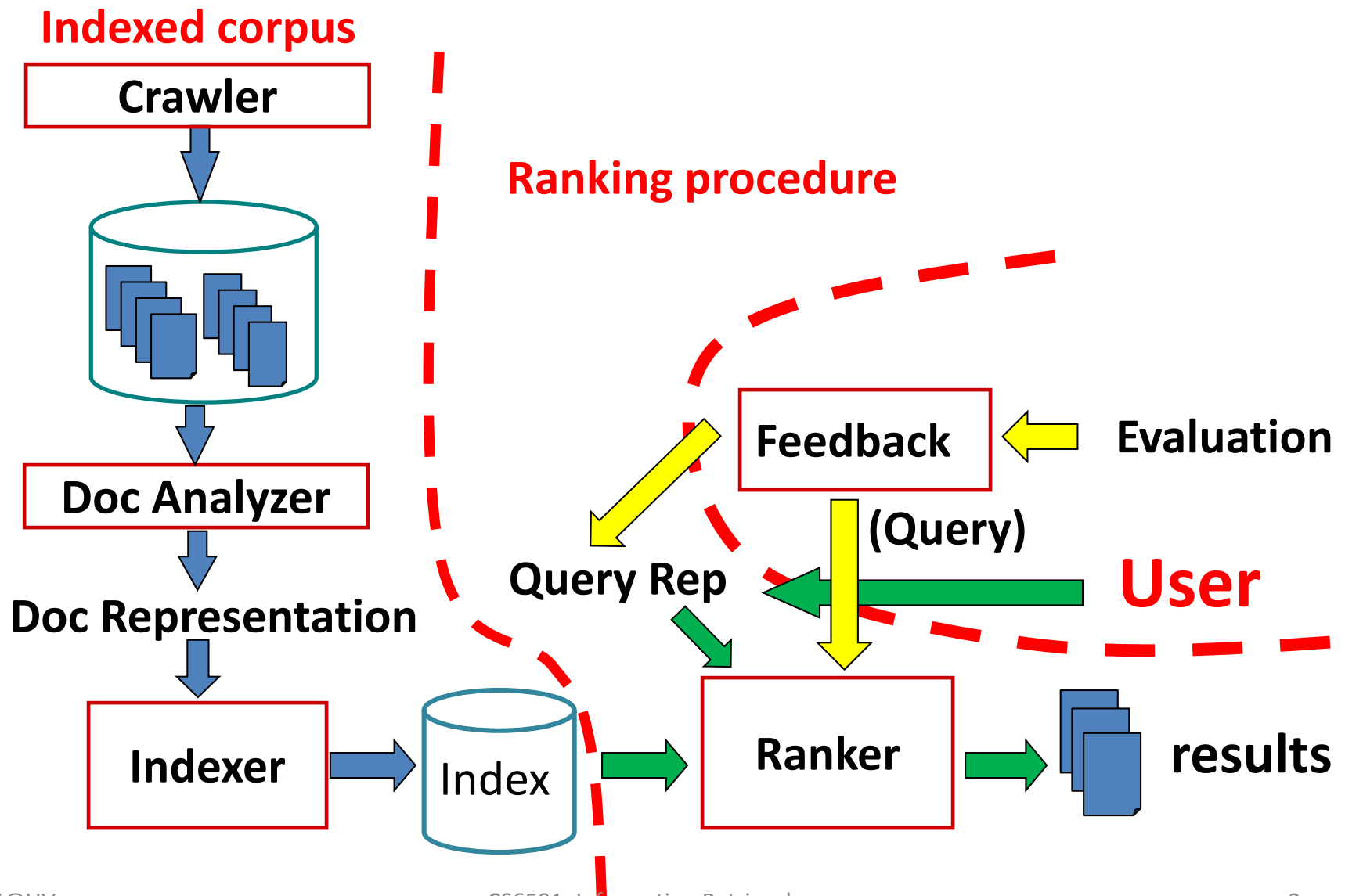
Web crawler

- A automatic program that systematically browses the web for the purpose of Web content indexing and updating
  - Synonyms: spider, robot, bot

How does it work

- In pseudo code

```
Def Crawler(entry_point) {
    URL_list = [entry_point]
    while (len(URL_list)>0) {              Which page to visit next?
        URL = URL_list.pop();
        if (isVisited(URL) or !isLegal(URL) or !checkRobotsTxt(URL))
            continue;
        HTML = URL.open();
        for (anchor in HTML.listOfAnchors()) {
            URL_list .append(anchor);
        }
        setVisited(URL);
        insertToIndex(HTML);
    }
}
```

Is it visited already?
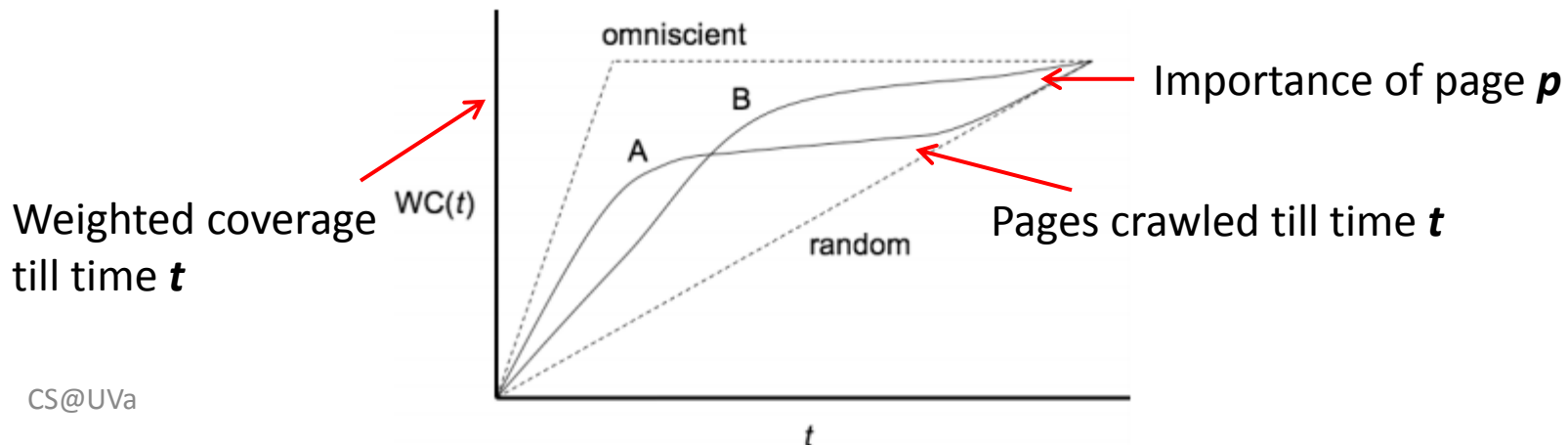*Or shall we visit it again?*

Is the access granted?

Visiting strategy

- Breadth first
  - Uniformly explore from the entry page
  - Memorize all nodes on the previous level
  - As shown in pseudo code

- Depth first
  - Explore the web by branch
  - Biased crawling given the web is not a tree structure

- Focused crawling
 - Prioritize the new links by predefined strategies

Focused crawling

- Prioritize the visiting sequence of the web
  - The size of Web is too large for a crawler (even Google) to completely cover
  - Not all documents are equally important
    - In 1999, no search engine indexes more than 16% of the Web
    - In 2005, large-scale search engines index no more than 40-70% of the indexable Web
  - Emphasize more on the high-quality documents
    - Maximize weighted coverage

Weighted coverage till time **t**

WC(*t*)

omniscient

B

A

random

t
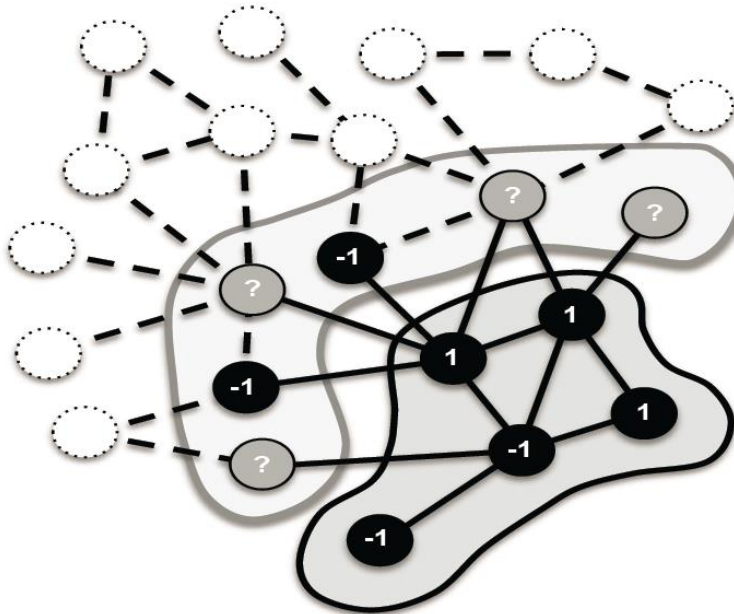
Importance of page **p**

Pages crawled till time **t**

Focused crawling

- Prioritize by in-degree [Cho et al. WW
  - The page with the highest numbe
    hyperlinks from previously downl
    downloaded next
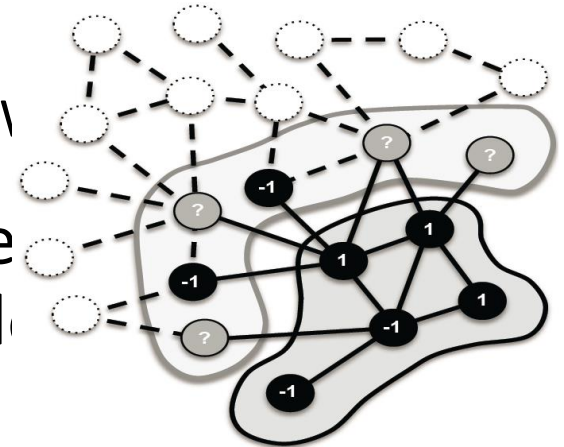
- Prioritize by   l. WWW'07, Cho and
  Uri VLDB'07]
  - Breadth-fi
    compute/   eriodically
  - More con   nce [Fetterly et al.
    SIGIR'09]

Focused crawling

- Prioritize by topical relevance
  - In vertical search, only crawl relevant pages [De et al. WWW'94]

    - E.g., restaurant search engine should only crawl restaurant pages

  - Estimate the similarity to current page by anchortext or text near anchor [Hersovici et al. WWW'98]

  - User given taxonomy or topical classifier [Chakrabarti et al. WWW'98]

Avoid duplicate visit

- Given web is a graph rather than a tree, avoid loop in crawling is important
- What to check
  - URL: must be normalized, not necessarily can avoid all duplication
    - http://dl.acm.org/event.cfm?id=RE160&CFID=516168213&CFTOKEN=99036335
    - http://dl.acm.org/event.cfm?id=RE160
  - Page: minor change might cause misfire
    - Timestamp, data center ID change in HTML
- How to check
  - trie or hash table

Politeness policy

- Crawlers can retrieve data much quicker and in greater depth than human searchers

- Costs of using Web crawlers
  - Network resources
  - Server overload

- Robots exclusion protocol
  - Examples: CNN, UVa

Robot exclusion protocol examples

- Exclude specific directories:

```
User-agent: *
Disallow: /tmp/
Disallow: /cgi-bin/
Disallow: /users/paranoid/
```

- Exclude a specific robot:

```
User-agent: GoogleBot
Disallow: /
```

- Allow a specific robot:

```
User-agent: GoogleBot
Disallow:

User-agent: *
Disallow: /
```

Re-visit policy

- The Web is very dynamic; by the time a Web crawler has finished its crawling, many events could have happened, including creations, updates and deletions
  - Keep re-visiting the crawled pages
  - Maximize freshness and minimize age of documents in the collection
- Strategy
  - Uniform re-visiting
  - Proportional re-visiting
    - Visiting frequency is proportional to the page's update frequency

Analyze crawled web pages

- What you care from the crawled web pages

Analyze crawled web pages

- What machine knows from the crawled web pages

```
<!DOCTYPE HTML>
<html lang="en-US">
<head>
<title>Technology News - Computers, Internet, Invention and Innovation Tech from CNN.com</title>
<meta http-equiv="content-type" content="text/html;charset=utf-8"/>
<meta http-equiv="last-modified" content="2014-07-23T15:25:56Z"/>
<meta name="robots" content="index,follow"/>
<meta name="googlebot" content="noarchive"/>
<meta name="viewport" content="width=1024"/>
<meta name="title" content="Technology News - Computers, Internet, Invention and Innovation Tech from CNN.com"/>
<meta name="description" content="Find information about the latest advances in technology at CNN. CNN Technology news and video covers the internet,
business and personal tech, video games, and more."/>
<meta name="keywords" content="CNN, CNN news, CNN.com, CNN TV, news, news online, breaking news, U.S. news, world news, weather, business, CNN Money,
sports, politics, law, technology, entertainment, education, travel, health, special reports, autos, developing story, news video, CNN Intl"/>
<link rel="canonical" href="http://www.cnn.com/TECH/"/>
<link type="image/png" rel="apple-touch-icon" href="http://i.cdn.turner.com/cnn/.e/img/3.0/global/misc/apple-touch-icon.png"/>
<link type="application/rss+xml" rel="alternate" href="http://rss.cnn.com/rss/cnn_tech.rss" title="CNN - Tech [RSS]"/>
<link type="application/rss+xml" rel="alternate" href="http://rss.cnn.com/rss/cnn_topstories.rss" title="CNN - Top Stories [RSS]"/>
<link type="application/rss+xml" rel="alternate" href="http://rss.cnn.com/rss/cnn_latest.rss" title="CNN - Recent Stories [RSS]"/>
<link type="application/opensearchdescription+xml" rel="search" href="/tools/search/cnncom.xml" title="CNN.com"/>
<link type="application/opensearchdescription+xml" rel="search" href="/tools/search/cnncomvideo.xml" title="CNN.com Video"/>
<link href="https://plus.google.com/u/0/b/117515799321987910349/117515799321987910349/posts" rel="publisher"/>
<link type="text/css" rel="stylesheet" href="http://z.cdn.turner.com/cnn/tmpl_asset/static/www_section/2695/css/techlib-min.css"/>
<script>
var cnnCVPAdSection='cnn.com_technology_section_homepage',
cnnIsSectionPage=true,
cnnSectionName='Tech',
cnnSectionFront='Tech',
sectionName='tech';
</script>
<script src="http://z.cdn.turner.com/cnn/tmpl_asset/static/www_section/2695/js/techlib-min.js"></script>
<script>
var cnnPageType="Section";
if(typeof(cnn_metadata)=='undefined'){var cnn_metadata={};}
var cnn_edtnswtchver='www';
cnn_metadata.section=['tech','tch : frontpage'];
cnn_metadata.friendly_name='Tech Home Page';
cnn_metadata.template_type='section front';
var CNN_gallery_0_ad_0="/cnn_adspaces/3.0/technology/main/bot1.120x90.ad";
var CNN_gallery_0_ad_1="/cnn_adspaces/3.0/technology/main/bot2.120x90.ad";
var CNN_gallery_0_ad_2="/cnn_adspaces/3.0/technology/main/bot3.120x90.ad";
</script>
```

Basic text analysis techniques

- Needs to analyze and index the crawled web pages
  - Extract informative content from HTML
  - Build machine accessible data <u>representation</u>

HTML parsing

- Generally difficult due to the free style of HTML

- Solutions
  - Shallow parsing
    - Remove all HTML tags
    - Only keep text between <title></title> and <p></p>
  - Automatic wrapper generation [Crescenzi et al. VLDB'01]
    - Wrapper: regular expression for HTML tags' combination
    - Inductive reasoning from examples
  - Visual parsing [Yang and Zhang DAR'01]
    - Frequent pattern mining of visually similar HTML blocks

HTML parsing

- ## [jsoup](#)
  - Java-based HTML parser
    - scrape and parse HTML from a URL, file, or string to DOM tree

CSS

e()

- Pyt
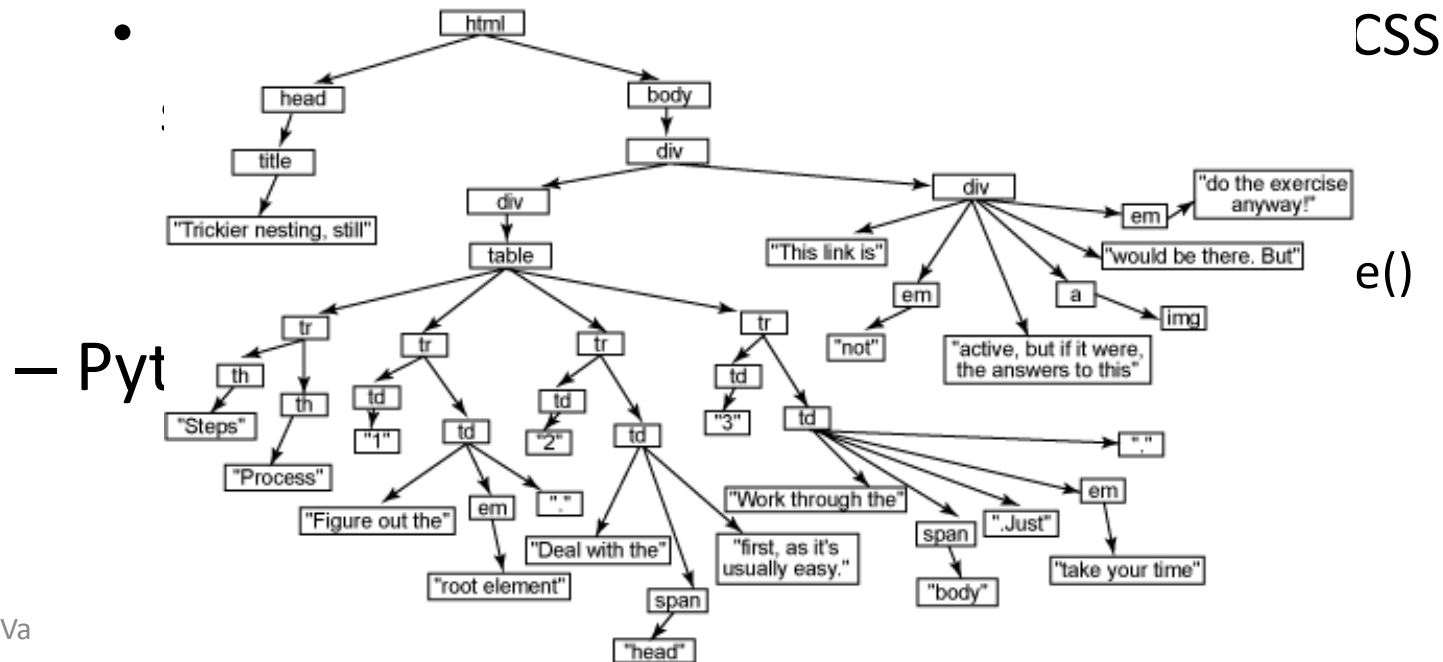
How to represent a document

- Represent by a string?

- <HEAD>Crowds in Liverpool to Mark 10th Anniversary of John Lennon's Death</HEAD>
<DATELINE>LIVERPOOL, England (AP)</DATELINE>
<TEXT>
   Dozens of fans of rock legend and former Beatle John Lennon gathered in the snow on a windy Saturday for a ceremony marking the 10th anniversary of his death. Liverpool's mayor, Dorothy Gavin, led Lennon devotees who laid wreaths at the foot of a bronze statue of The Beatles in the city's Cavern Walks shopping center.   The center was built on the original site of the Cavern Club, made famous when The Beatles played there in the 1960s, and has become a place of pilgrimage. ``Give peace a chance,'' the title of one of singer-songwriter Lennon's greatest hits, was the theme for the day.
   …
   Lennon and his wife, Yoko Ono, were returning to their apartment in New York's Dakota apartment building after a recording session on Dec. 8, 1980, when Lennon was shot to death by Mark David Chapman, a deranged fan to whom Lennon had

- given his autograph only hours before. Lennon was 40.   A spokesman for the Lennon family said Ms. Ono and the couple's son, Sean, were in Europe and would spend the anniversary privately.
   …
   Peebles said late in 1980 that Lennon had just recovered from a period when he had ``gone off the rails'' and his relationship with Ms. Ono had suffered. ``But (when I saw him) they'd had the baby, Sean had been born, and everything was great.''
</TEXT>

  – Bag-of-Words representation!

Tokenization

- Break a stream of text into meaningful units
  - Tokens: words, phrases, symbols
    - **Input:** It's not straight-forward to perform so-called "tokenization."
    - **Output(1):** 'It's', 'not', 'straight-forward', 'to', 'perform', 'so-called', '"tokenization."'
    - **Output(2):** 'It', '''', 's', 'not', 'straight', '-', 'forward, 'to', 'perform', 'so', '-', 'called', '"', 'tokenization', '.', '"'
  - Definition depends on language, corpus, or even context

Tokenization

- Solutions
  - Regular expression
    - [\w]+: so-called -> 'so', 'called'
    - [\S]+: It's -> 'It's' instead of 'It', ''s'
  - Statistical methods
    - Explore rich features to decide where is the boundary of a word
      - Apache OpenNLP (http://opennlp.apache.org/)
      - Stanford NLP Parser (http://nlp.stanford.edu/software/lex-parser.shtml)
    - Online Demo
      - Stanford (http://nlp.stanford.edu:8080/parser/index.jsp)
      - UIUC (http://cogcomp.cs.illinois.edu/curator/demo/index.html)

Full text indexing

- Bag-of-Words representation
  - Doc1: Information retrieval is helpful for everyone.
  - Doc2: Helpful information is retrieved for you.

	information	retrieval	retrieved	is	helpful	for	you	everyone
Doc1	1	1	0	1	1	1	0	1
Doc2	1	0	1	1	1	1	1	0

Word-document adjacent matrix

Full text indexing

- Bag-of-Words representation
  - Assumption: word is independent from each other
  - Pros: simple
  - Cons: grammar and order are missing
  - *The most frequently used document representation*
    - *Image, speech, gene sequence*

Full text indexing

- Improved Bag-of-Words representation
  - N-grams: a contiguous sequence of n items from a given sequence of text
    - E.g., Information retrieval is helpful for everyone
    - Bigrams: 'information_retrieval', 'retrieval_is', 'is_helpful', 'helpful_for', 'for_everyone'
  - Pros: capture local dependency and order
  - Cons: purely statistical view, increase vocabulary size $O(V^N)$

Full text indexing

- Index document with all the occurring word
  - Pros
    - Preserve all information in the text (hopefully)
    - Fully automatic
  - Cons
    - Vocabulary gap: cars v.s., car
    - Large storage: e.g., in N-grams $O(V^N)$
  - Solution
    - Construct controlled vocabulary

Statistical property of language

- Zipf's law

  discrete version of power law

  – Frequen[cy of any word is] portional to its rank

  – Formal[ly]

    • $f(k$...)

    where ... bulary size; s

    is langu[age]

A plot of word frequ[ency]
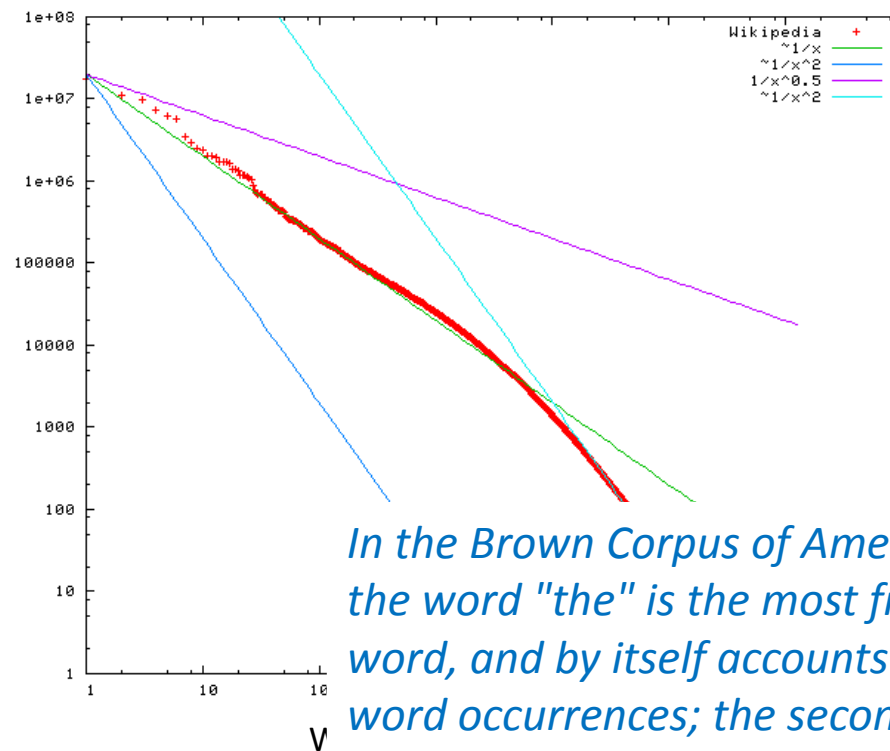
*In the Brown Corpus of American English text, the word "the" is the most frequently occurring word, and by itself accounts for nearly **7%** of all word occurrences; the second-place word "of" accounts for slightly over **3.5%** of words.*

Zipf's law tells us

- Head words may take large portion of occurrence, but they are semantically meaningless
  - E.g., the, a, an, we, do, to
- Tail words take major portion of vocabulary, but they rarely occur in documents
  - E.g., dextrosinistral
- The rest is most representative
  - To be included in the controlled vocabulary

Automatic text indexing
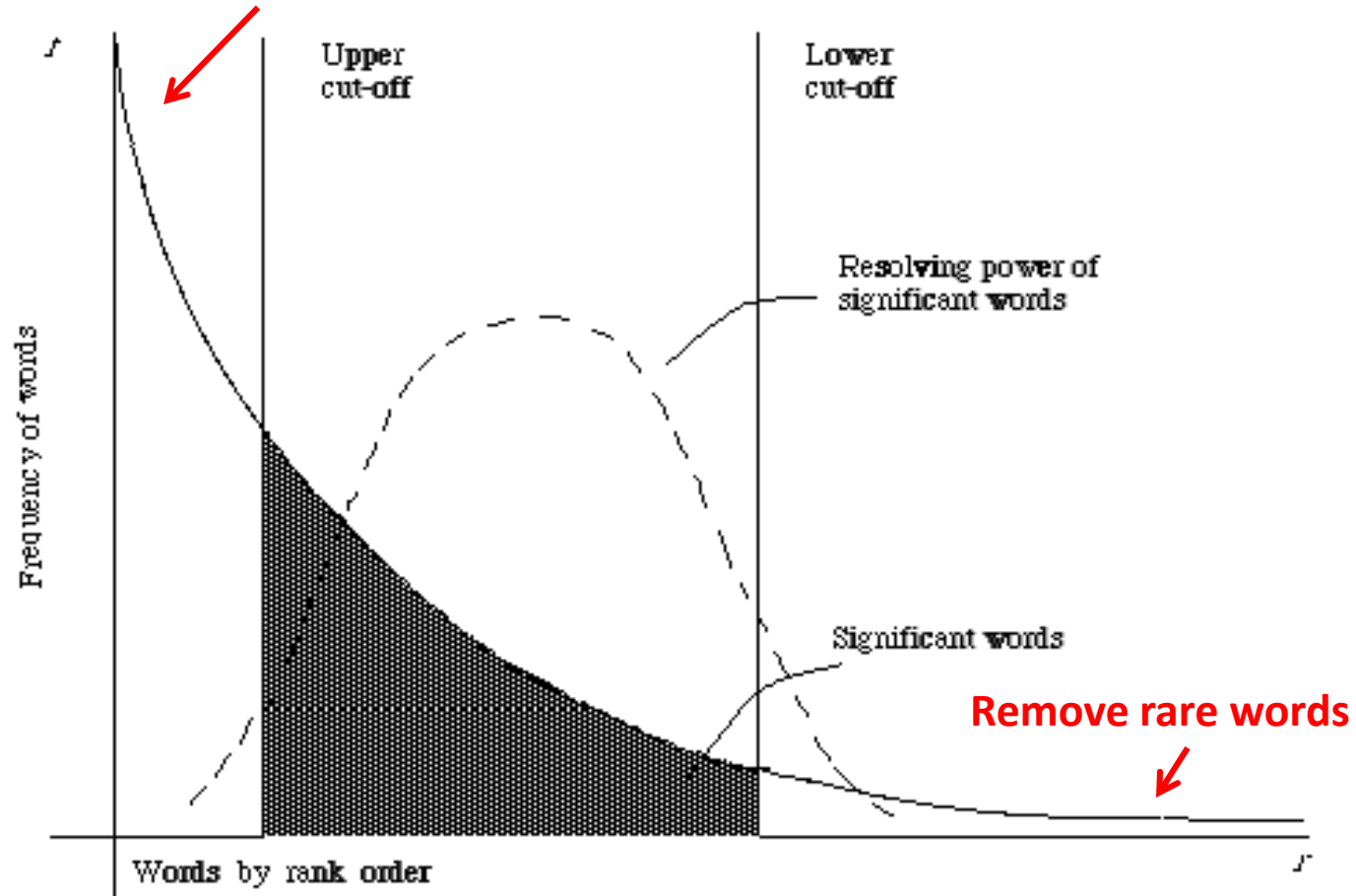
Figure 2.1. A plot of the hyperbolic curve relating f, the frequency of occurrence and r, the rank order (Adapted from Schultz[44] page 120)

Normalization

- Convert different forms of a word to normalized form in the vocabulary
  - U.S.A -> USA, St. Louis -> Saint Louis
- Solution
  - Rule-based
    - Delete periods and hyphens
    - All in lower case
  - Dictionary-based
    - Construct equivalent class
      - Car -> "automobile, vehicle"
      - Mobile phone -> "cellphone"

Stemming

- Reduce inflected or derived words to their root form
  - Plurals, adverbs, inflected word forms
    - E.g., ladies -> lady, referring -> refer, forgotten -> forget
  - Bridge the vocabulary gap
  - Risk: lose precise meaning of the word
    - E.g., lay -> lie (a false statement? or be in a horizontal position?)
  - Solutions (for English)
    - Porter stemmer: pattern of vowel-consonant sequence
    - Krovetz Stemmer: morphological rules

Stopwords

- U ... is

  − ... ?

  − ... of

Nouns	Verbs	Adjectives	Prepositions	Others
1. time	1. be	1. good	1. to	1. the
2. person	2. have	2. new	2. of	2. and
3. year	3. do	3. first	3. in	3. a
4. way	4. say	4. last	4. for	4. that
5. day	5. get	5. long	5. on	5. I
6. thing	6. make	6. great	6. with	6. it
7. man	7. go	7. little	7. at	7. not
8. world	8. know	8. own	8. by	8. he
9. life	9. take	9. other	9. from	9. as
10. hand	10. see	10. old	10. up	10. you
11. part	11. come	11. right	11. about	11. this
12. child	12. think	12. big	12. into	12. but
13. eye	13. look	13. high	13. over	13. his
14. woman	14. want	14. different	14. after	14. they
15. place	15. give	15. small	15. beneath	15. her
16. work	16. use	16. large	16. under	16. she
17. week	17. find	17. next	17. above	17. or
18. case	18. tell	18. early		18. an
19. point	19. ask	19. young		19. will
20. government	20. work	20. important		20. my
21. company	21. seem	21. few		21. one
22. number	22. feel	22. public		22. all
23. group	23. try	23. bad		23. would
24. problem	24. leave	24. same		24. there
25. fact	25. call	25. able		25. their

The OEC: Facts about the language

Abstraction of search engine architecture

Indexed corpus

Crawler

1. Visiting strategy
2. Avoid duplicated visit
3. Re-visit policy

Doc Analyzer

1. HTML parsing
2. Tokenization
3. Stemming/normalization
4. Stopword/controlled vocabulary filter

Doc Representation

Terms	Documents													
	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12	M13	M14
abnormalities	0	0	0	0	0	0	0	1	0	1	0	0	0	0
age	1	0	0	0	0	0	0	0	0	0	0	1	0	0
behavior	0	0	0	0	1	1	0	0	0	0	0	0	0	0
blood	0	0	0	0	0	0	0	1	0	0	1	0	0	0
close	0	0	0	0	0	0	1	0	0	0	1	0	0	0
culture	1	1	0	0	0	0	0	1	1	0	0	0	0	0
depressed	1	0	1	1	1	0	0	0	0	0	0	0	0	0
discharge	1	1	0	0	0	1	0	0	0	0	0	0	0	0
disease	0	0	0	0	0	0	0	0	1	0	1	0	0	0
fast	0	0	0	0	0	0	0	0	0	1	0	1	1	1
generation	0	0	0	0	0	0	0	0	1	0	0	0	1	0
oestrogen	0	0	1	1	0	0	0	0	0	0	0	0	0	0
patients	1	1	0	1	0	0	0	1	0	0	0	0	0	0
pressure	0	0	0	0	0	0	0	0	0	0	1	0	0	1
rats	0	0	0	0	0	0	0	0	0	0	0	0	1	1
respect	0	0	0	0	0	0	0	1	0	0	0	1	0	0
rise	0	0	0	1	0	0	0	0	0	0	0	0	0	1
study	1	0	1	0	0	0	0	0	1	0	0	0	0	0

BagOfWord representation!

Automatic text indexing

Query: "to be or not to be"

- In modern search engine
  - **No** stemming or stopword removal, since computation and storage are no longer the major concern
  - More advanced NLP techniques are applied
    - Named entity recognition
      - E.g., people, location and organization
    - Dependency parsing
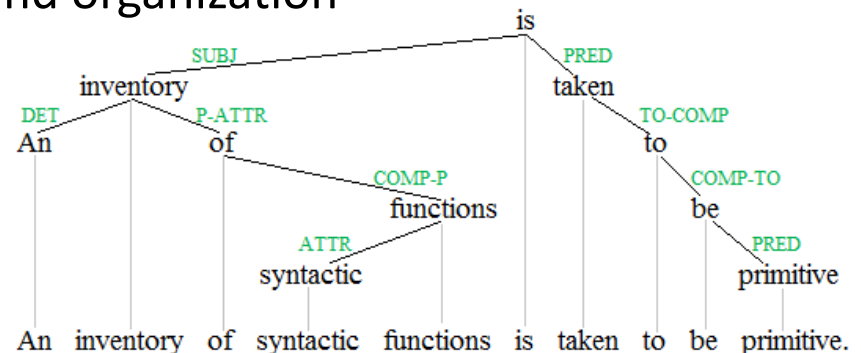
What you should know

- Basic techniques for crawling
- Zipf's law
- Procedures for automatic text indexing
- Bag-of-Words document representation

Reference I

- Cho, Junghoo, Hector Garcia-Molina, and Lawrence Page. "Efficient crawling through URL ordering." *Computer Networks and ISDN Systems* 30.1 (1998): 161-172.

- Abiteboul, Serge, Mihai Preda, and Gregory Cobena. "Adaptive on-line page importance computation." *Proceedings of the 12th international conference on World Wide Web*. ACM, 2003.

- Cho, Junghoo, and Uri Schonfeld. "RankMass crawler: a crawler with high personalized pagerank coverage guarantee." *Proceedings of the 33rd international conference on Very large data bases*. VLDB Endowment, 2007.

- Fetterly, Dennis, Nick Craswell, and Vishwa Vinay. "The impact of crawl policy on web search effectiveness." *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*. ACM, 2009.

- De Bra, Paul ME, and R. D. J. Post. "Information retrieval in the World-Wide Web: making client-based searching feasible." *Computer Networks and ISDN Systems* 27.2 (1994): 183-192.

- Hersovici, Michael, et al. "The shark-search algorithm. An application: tailored Web site mapping." *Computer Networks and ISDN Systems* 30.1 (1998): 317-326.

Reference II

- Chakrabarti, Soumen, Byron Dom, Prabhakar Raghavan, Sridhar Rajagopalan, David Gibson, and Jon Kleinberg. "Automatic resource compilation by analyzing hyperlink structure and associated text." *Computer Networks and ISDN Systems* 30, no. 1 (1998): 65-74.

- Crescenzi, Valter, Giansalvatore Mecca, and Paolo Merialdo. "Roadrunner: Towards automatic data extraction from large web sites." *VLDB*. Vol. 1. 2001.

- Yang, Yudong, and HongJiang Zhang. "HTML page analysis based on visual cues." *Document Analysis and Recognition, 2001. Proceedings. Sixth International Conference on*. IEEE, 2001.