Large-Scale Programming and Testing

Fall 2017 – CSCI 4963/6963 – Week 04

David Goldschmidt – goldschmidt@gmail.com

Office: Amos Eaton 115

Office hours: Mon/Thu 1:00-1:50PM; Wed 1:00-2:50PM

What are we searching for?

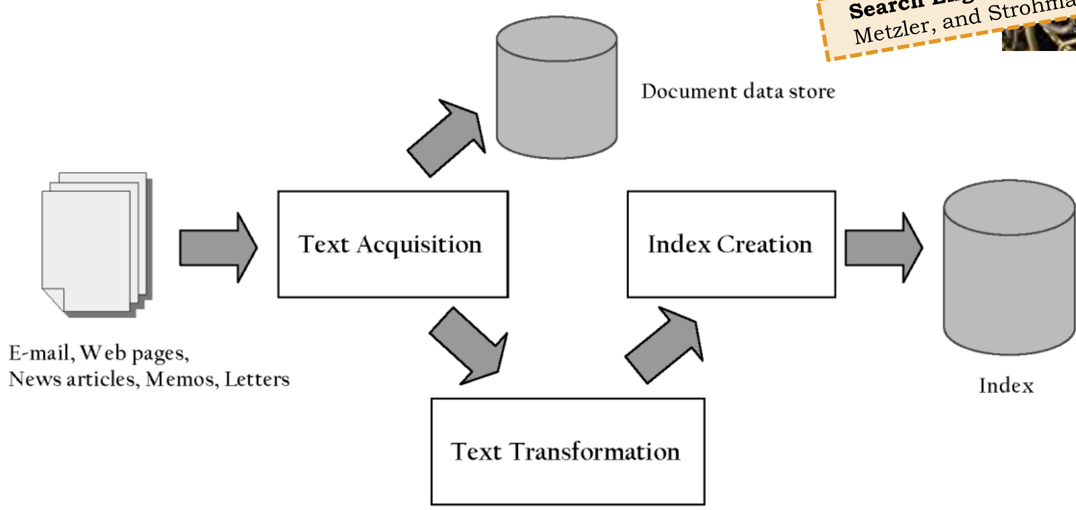
- What is search?
- Where do we use search?
- What are we searching for?
- How many searches are processed per day?
- What is the average number of words in text-based searches?
- What applications and varieties of search do we make use of?
- How do search engines and search functionality scale up?

Finding things

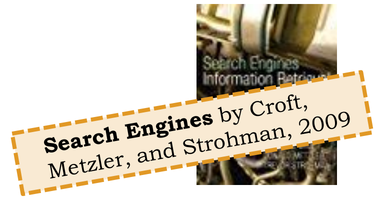
- Applications and varieties of search include:
  - Web search
  - Site search
  - Vertical search
  - Enterprise search
  - Desktop/mobile/local search
  - Proximity search
  - App search
  - People search (social media)
  - Location search (maps)
  - Text-based search
  - Image/video search
  - Siri; Alexa; Google Assistant
  - As-you-type search
  - Find-in-page search

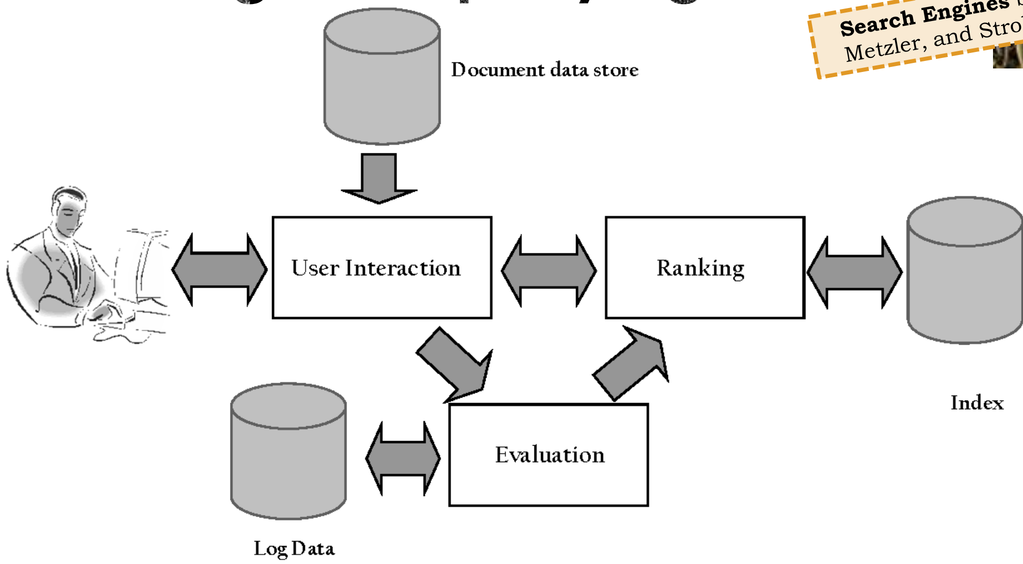
Acquisition and indexing

Search Engines by Croft, Metzler, and Strohman, 2009



Ranking and querying





How do we measure success?

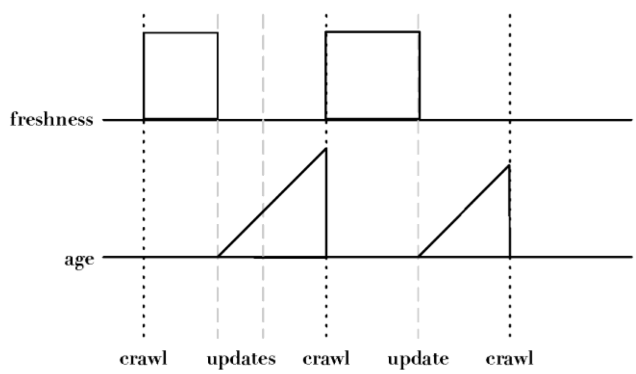
- Relevance
  - Do the presented search results contain information that the user was actually looking for?
  - Problems of context and vocabulary mismatch often occur here (e.g., homonyms)
- User relevance
  - Search results relevant to one user may be completely irrelevant to another user

How do we measure success?

- Timeliness
  - Do the presented search results contain information that is current?
  - Are more recent results ranked higher?
- Performance
  - Users expect sub-second response times
- Spam-resistance

How do we measure success?

- Freshness
  - Do the presented search results contain information that is very recent (e.g., breaking news stories)
- Age
  - To what degree are the presented search results out of date?
- How often should we crawl (and re-index) everything?



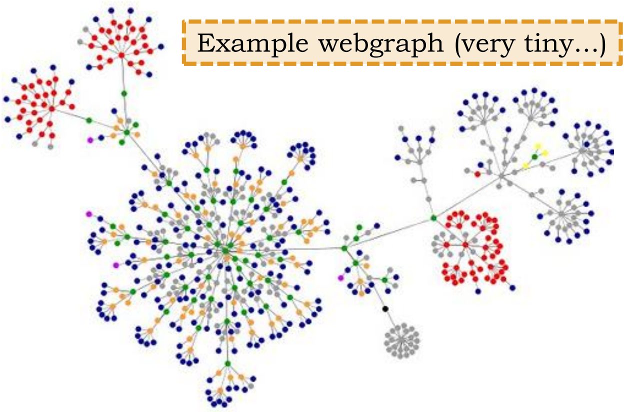
How do we measure success?

- Precision
  - How precise are the presented search results?
  - Precision measures the proportion of retrieved documents that were relevant to the given query versus those that were not relevant
  - Focuses only on retrieved documents (i.e., not the entire corpus)
- Recall
  - Did we retrieve all of the relevant documents?
  - Recall measures the proportion of relevant documents actually retrieved versus all possible (indexed) documents
  - Includes all documents in the given corpus

How do we measure success?

- Scalability
  - How well does the search engine scale?
  - Can we easily increase the number of documents (or users or queries) by an order of magnitude (or more)?
  - What hardware and parallelization techniques can we use here?
  - What data indexing optimizations can we use here?
  - The goal here is to achieve a design that performs equally well as the system grows and expands *by orders of magnitude*

A day in the life of a crawler....



Example webgraph (very tiny...)

```
procedure CRAWLERTHREAD(frontier)
    while not frontier.done() do
        website ← frontier.nextSite()
        url ← website.nextURL()
        if website.permitsCrawl(url) then
            text ← retrieveURL(url)
            storeDocument(url, text)
            for each url in parse(text) do
                frontier.addURL(url)
            end for
        end if
        frontier.releaseSite(website)
    end while
end procedure
```

Information retrieval (IR)

Information retrieval is "a field concerned with the structure, analysis, organization, storage, searching, and retrieval of information."
– Gerard Salton (1968)

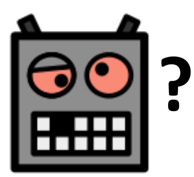
- Note that this is 1968, before the Internet, the Web, Unix, etc.

Structured information

- Structured information:
  - Often stored in a relational database or a set of related files
  - Organized via predefined tables, columns, keys, relationships, triggers, etc.
  - Queries are also structured, adhering to some implementation-specific flavor of SQL
  - Databases are often private and therefore not widely accessible to general search applications

Unstructured information

- Unstructured information:
  - Often the real or raw data that we wish to store, index, query, etc.
  - Consists of documents, images, video, audio (much of which uses textual tags and annotations)
  - Sometimes stored within a database (e.g., as a "blob" data type)
  - Contains information that humans can easily extract, but machines face major difficulties doing so (e.g., identifying headings, words, phrases, semantics)



Processing text

- Search and information retrieval has primarily focused on text processing and documents
- Search typically uses various statistical properties of text, including:
  - Word counts
  - Word frequencies
  - Phrases (i.e., n-grams, including bigrams, trigrams, etc.)
  - Linguistic and parts-of-speech features (e.g., nouns, verbs, etc.)
- The entire collection of documents is often called a corpus

What can text statistics tell us?

- English documents are rather predictable:
  - The top two most frequently occurring words are *the* and *of*, accounting for 10% of all word occurrences
  - The top six most frequently occurring words account for approximately 20% of all word occurrences
  - The top 50 most frequently occurring words account for approximately 50% of all word occurrences
  - Given all unique words in a (relatively large) document/corpus, approximately 50% occur only once
- Very similar predictions and observations can easily be made for other languages

Zipf's law

George Kingsley Zipf (1902-1950)

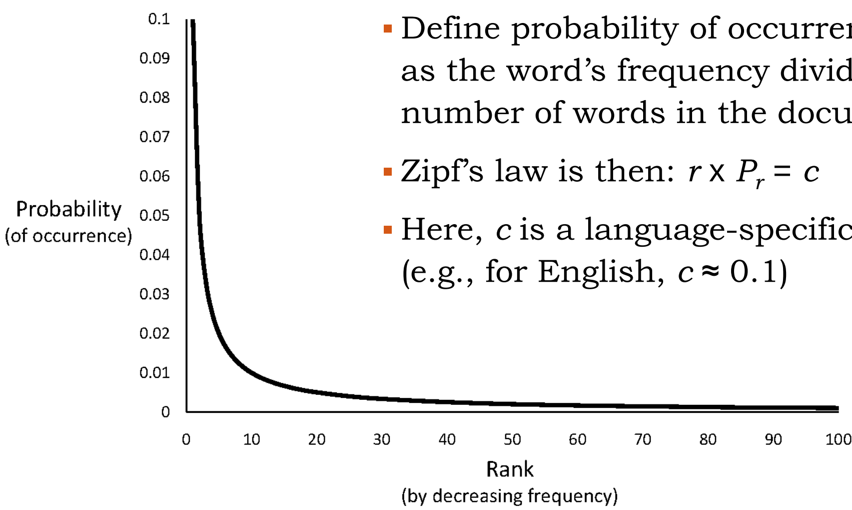
American linguist and philologist

- Let's count word occurrences, then rank words in order of decreasing frequency
- The rank *r* of a word multiplied by its frequency *f* is approximately equal to constant *k*:

$$k = r \times f$$

  - In other words, the frequency of the *r*th most common word is inversely proportional to *r*

Zipf's law



- Define probability of occurrence P_r of a word as the word's frequency divided by the total number of words in the document
- Zipf's law is then: $r \times P_r = c$
- Here, c is a language-specific constant (e.g., for English, $c \approx 0.1$)

Example word statistics

- AP89 is an extensive dataset containing all Associated Press (AP) news stories from 1989 (http://trec.nist.gov)
  - Total documents: 84,678
  - Total word occurrences: 39,749,179
  - Vocabulary size (i.e., unique words): 198,763
  - Words occurring more than 1000 times: 4169
  - Words occurring only once: 70,064

Top 50 words

- Here are the top 50 words from the AP89 dataset
- Statistics include:
  - Word frequencies
  - Rank *r*
  - Probability of occurrence P_r
  - What does $r \times P_r$ tell us?

Word	*Freq.*	*r*	$P_r(\%)$	$r.P_r$	*Word*	*Freq*	*r*	$P_r(\%)$	$r.P_r$
the	2,420,778	1	6.49	0.065	has	136,007	26	0.37	0.095
of	1,045,733	2	2.80	0.056	are	130,322	27	0.35	0.094
to	968,882	3	2.60	0.078	not	127,493	28	0.34	0.096
a	892,429	4	2.39	0.096	who	116,364	29	0.31	0.090
and	865,644	5	2.32	0.120	they	111,024	30	0.30	0.089
in	847,825	6	2.27	0.140	its	111,021	31	0.30	0.092
said	504,593	7	1.35	0.095	had	103,943	32	0.28	0.089
for	363,865	8	0.98	0.078	will	102,949	33	0.28	0.091
that	347,072	9	0.93	0.084	would	99,503	34	0.27	0.091
was	293,027	10	0.79	0.079	about	92,983	35	0.25	0.087
on	291,947	11	0.78	0.086	i	92,005	36	0.25	0.089
he	250,919	12	0.67	0.081	been	88,786	37	0.24	0.088
is	245,843	13	0.65	0.086	this	87,286	38	0.23	0.089
with	223,846	14	0.60	0.084	their	84,638	39	0.23	0.089
at	210,064	15	0.56	0.085	new	83,449	40	0.22	0.090
by	209,586	16	0.56	0.090	or	81,796	41	0.22	0.090
it	195,621	17	0.52	0.089	which	80,385	42	0.22	0.091
from	189,451	18	0.51	0.091	we	80,245	43	0.22	0.093
as	181,714	19	0.49	0.093	more	76,388	44	0.21	0.090
be	157,300	20	0.42	0.084	after	75,165	45	0.20	0.091
were	153,913	21	0.41	0.087	us	72,045	46	0.19	0.089
an	152,576	22	0.41	0.090	percent	71,956	47	0.19	0.091
have	149,749	23	0.40	0.092	up	71,082	48	0.19	0.092
his	142,285	24	0.38	0.092	one	70,266	49	0.19	0.092
but	140,880	25	0.38	0.094	people	68,988	50	0.19	0.093

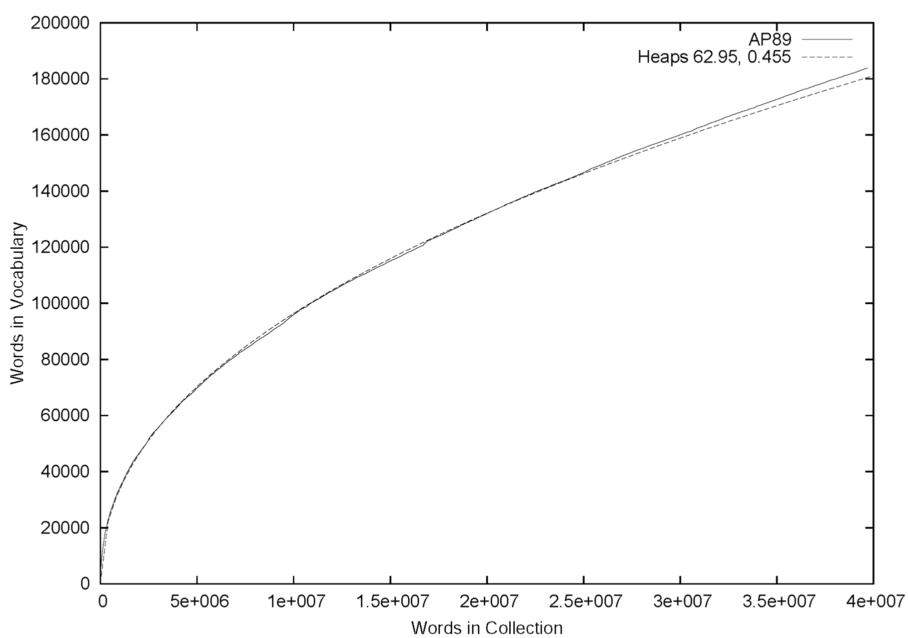
Vocabulary growth over time

- As the given corpus grows, so does vocabulary size
  - This growth slows down when the corpus becomes large
- The relationship between corpus size n and vocabulary size v was defined empirically by Herdan (1960), then by Heaps (1978)
- Heaps' law or Herdan's law:

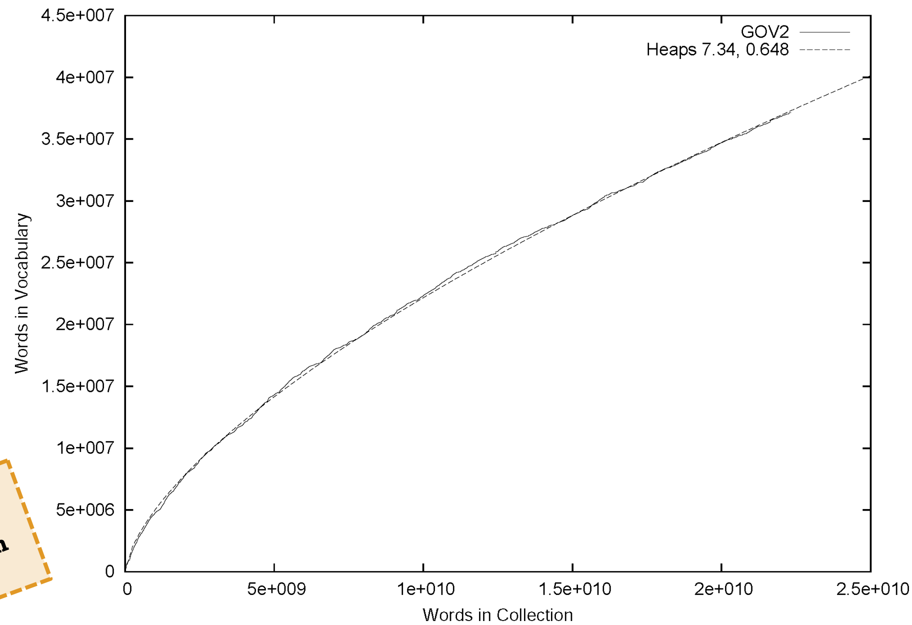
$$v = k \times n^{\beta}$$

  - Here, constants k and β vary
  - Typically $10 \leq k \leq 100$ and $\beta \approx 0.5$

AP89



GOV2



The GOV2 dataset contains Web pages crawled from .gov in early 2004