CAIM: Cerca i Anàlisi d'Informació Massiva

FIB, Grau en Enginyeria Informàtica

Slides by Marta Arias, José Luis Balcázar,
Ramon Ferrer-i-Cancho, Ricard Gavaldá
Department of Computer Science, UPC

Fall 2018

`http://www.cs.upc.edu/~caim`

1. Introduction: Concept. The Information Retrieval (IR) process

A Quote to Ponder About, I

> “Consider a future device for individual use, which is a sort of mechanized private file and library. It needs a name, and, to coin one at random, ’memex’ will do. A memex is a device in which an individual stores all his books, records, and communications, and which is mechanized so that it may be consulted with exceeding speed and flexibility. It is an enlarged intimate supplement to his memory. [...]
>
> Books of all sorts, pictures, current periodicals, newspapers, are thus obtained and dropped into place. Business correspondence takes the same path. [...]
>
> There is, of course, provision for consultation of the record by the usual scheme of indexing. [...]

A Quote to Ponder About, II

It affords an immediate step, however, to associative indexing, the basic idea of which is a provision whereby any item may be caused at will to select immediately and automatically another. This is the essential feature of the memex. The process of tying two items together is the important thing. [...]

Thereafter, at any time, when one of these items is in view, the other can be instantly recalled merely by tapping a button below the corresponding code space. Moreover, when numerous items have been thus joined together to form a trail, they can be reviewed in turn [...]

Wholly new forms of encyclopedias will appear, ready made with a mesh of associative trails running through them, ready to be dropped into the memex and there amplified."

A Quote to Ponder About, III

In case you did not recognize the text, guess in which year it was published!

Vannevar Bush: As We May Think

The Atlantic Magazine, july 1945

`http://www.theatlantic.com/magazine/archive/1945/07/as-we-may-think/303881/4/?single_page=true`

At the time, Dr. Vannevar Bush was the Director of the Office of Scientific Research and Development, USA.

Information Retrieval, the origins

The technology of information retrieval started on very limited digitalization and had quite restricted usage (librarians, government agencies…)

But now, we all depend on it through an amazing degree of digitalization!

And most of the information will never move outside the digital realm.

The web

The web changed everything:

everybody (. . .) can set up a site and publish information.

The web 2.0 changed everything again:

- (almost) everybody can participate,
- everybody (. . .) can affect everybody else’s info.

Web search as a comprehensive of Computing

Algorithms, data structures, computer architecture, networking, logic, discrete mathematics, interface design, user modelling, databases, software engineering, programming languages, multimedia technology, image and sound processing, data mining, artificial intelligence, . . .

Think about it: Search billions of pages and return satisfying results in tenths of a second

Information Retrieval versus Database Queries, I

Why is there a difference?

To retrieve my phone number, it is necessary to have it.

But this is not sufficient.

You need to know where you have it.

Database queries rely both on the data tuples and on the database schema.

Information Retrieval versus Database Queries, II

Why is there a difference?

In Information Retrieval,

- We may not know where we have the information we want to retrieve,
- We may not know whether we have the information we want to retrieve,
- We may not even know what information we actually want to retrieve.
- For instance, note the large qualitative difference between:
  - “Find me somebody’s phone number” versus
  - “Tell me about the influences of late XVI century European composers on Beethoven”.

User Expectations, I

The focus of this course

Thus, often, we

- do not know really much about what we want to ask exactly,
- and we know that the retrieval system will simply try to help us on the basis of just a large document collection.

Then, we are in Information Retrieval:

Assessing relevance is far from nontrivial!

- Heuristic approaches become required.
- There is nothing that looks remotely like keys (although we will call “keywords” the search terms).
- “Too literal” answers may well be inadequate.

User Expectations, II

The focus of this course

Yet, we may want to ask for something that clearly is not available to the retrieving system,

- like a prediction of a future customer trend,
- obviously unavailable: at most, we can “try”.

Then, we are in Data Mining, Statistics, and Machine Learning.

Big Data:

Having more observations within the same space leads to more precise estimates and better predictions.

- But they become harder to manage, due to sheer size.
- Most often, the new observations are not within the same space!
- Google this expression (with the quotes): “Big data is worth nothing without big science”.

Hierarchical/Taxonomic vs. Faceted Search

Biology:

Animalia → Chordata → Mammalia → Artiodactyla → Giraffidae → Giraffa

Universal Decimal Classification (e.g. Libraries):

0 Science and knowledge →
00 Prolegomena. Fundamentals of knowledge and culture. Propaedeutics →
004 Computer science and technology. Computing →
004.6 Data →
004.63 Files

Taxonomic vs. Faceted Search

Faceted search:

By combination of features (facets) that we have indexed

“It is yellow and black & lives near the equator”

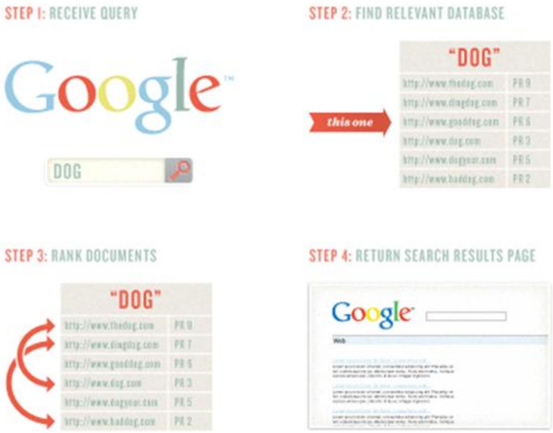
Elementary Set-Up: Textual Information

Focus in this course: Document retrieval from the web.

- Web documents contain terms and links.
- Users issue queries to look for documents.
- Queries typically formed by terms as well.

We do not consider explicitly retrieving audio, video, images, binaries, . . . or other forms of queries.

Search engines



STEP 1: RECEIVE QUERY

STEP 2: FIND RELEVANT DATABASE

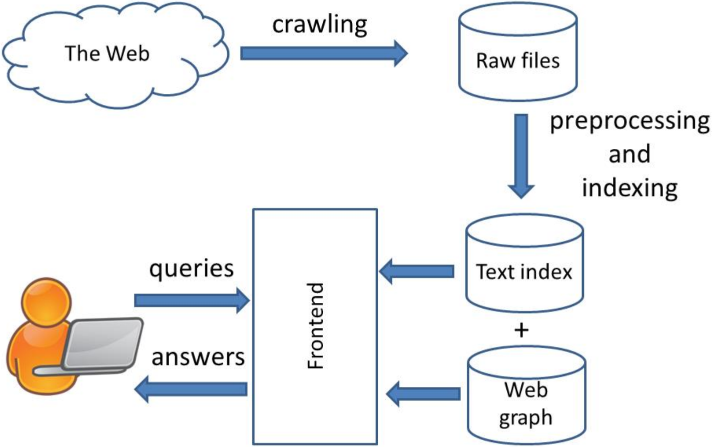
"DOG"	
http://www.thedog.com	PR 9
http://www.dingdog.com	PR 7
http://www.gooddog.com	PR 6
http://www.dog.com	PR 3
http://www.dogyear.com	PR 5
http://www.baddog.com	PR 2

STEP 3: RANK DOCUMENTS

"DOG"	
http://www.thedog.com	PR 9
http://www.dingdog.com	PR 7
http://www.gooddog.com	PR 6
http://www.dog.com	PR 3
http://www.dogyear.com	PR 5
http://www.baddog.com	PR 2

STEP 4: RETURN SEARCH RESULTS PAGE

The Information Retrieval process, I



The Information Retrieval process, I

Offline process:

- Crawling (today)
- Preprocessing (today)
- Indexing (later)

Goal:

Prepare data structures to make online process fast.

- Can afford long computations. For example, scan each document several times.
- Must produce reasonably compact output (data structure).

The Information Retrieval process, II

Online process:

- Get query
- Retrieve relevant documents
- Rank documents
- Format answer, return to user

Goal:

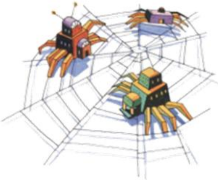
Instantaneous reaction, useful visualization.

- May use additional info: user location, ads, ...

Crawling

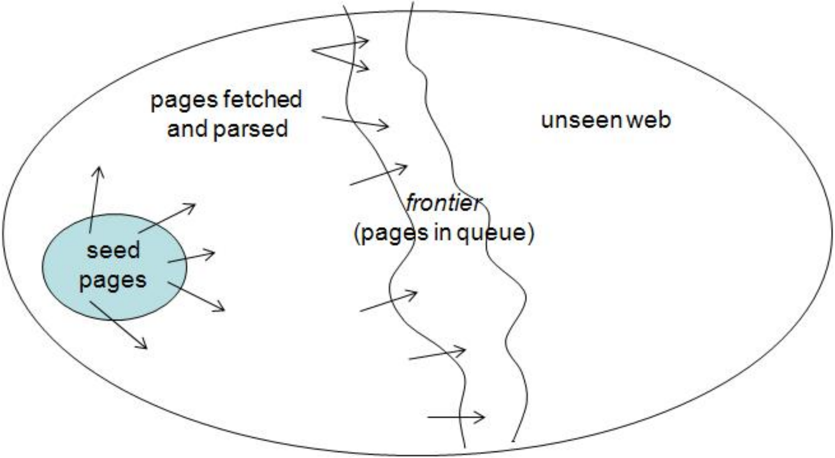
Crawler, robot, spider, wanderer . . .

Systematically explores the web & collect documents.



```
add ``seed'' URLs to queue
loop
   choose a URL from queue
   fetch page, parse it
   discard it or add it to DB
   add (new) URL's it contains to queue
end loop
```

Crawling as graph exploration



Crawling process

Exploration may be:

- breadth-first, depth-first (?)
- focused (or not): uses expressed focus or interests
  - by keywords
  - implicitly in choice of seed pages
- pages in the queue closer to focus get explored first

- Pages must be refreshed periodically.
- Pages with higher interest fetched first, refreshed more often.

The crawling process

Crawlers must be

- efficient
- robust
- polite

Crawling efficiency

- Distributed: use several machines
- Scalable: can add more machines for more throughput

- Connections have high latency
- Keep many open connections (100's?) per machine
- Try to keep all threads busy
- DNS server tends to be the bottleneck

Crawling efficiency

Some pages may be discarded:

- Duplicates
  - Fast duplicate detection a problem in itself
  - Fingerprints or k-shingles (similar to n-grams)
- Irrelevant for crawler's goals (e.g., focused crawlers)
- Unreliable or spam

Crawling robustness

- Dead URL’s: *Very* common. Timeout mechanisms
- Syntactically incorrect pages
- Spider traps. Often dynamically generated
- Webspam
- Mirror sites

Crawling politeness

- Don't hit the same server too often, esp. downloads
- Insert wait times
- Respect robot exclusion standard
  - `/robots.txt` file: administrator preferences
  - "If you are agent X, please don't explore directory Y"

```
User-agent: *
Disallow: /cgi-bin/
Disallow: /images/
Disallow: /tmp/
Disallow: /private/
```

Information Retrieval Process, III

Preprocessing: Term extraction

Potential actions:

- Parsing: Extracting structure (if present, e.g. HTML).
- Tokenization: decomposing character sequences into individual units to be handled.
- Enriching: annotating units with additional information.
- Either Lemmatization or Stemming: reduce words to roots.

Tokenization, I

Group characters

Join consecutive characters into “words”: use spaces and punctuation to mark their borders.
Similar to lexical analysis in compilers.
Many difficulties:

- “David A. Mix Barrington”, “Fahrenheit 451”, “September 11, 1714”,
- IP and phone numbers, email addresses, URL’s,
- “R+D”, “H&M”, “C#”, “I.B.M.”, “753 B.C.”,
- Hyphens are complicated:
  - change “afro-american culture” to “afroamerican culture”?
  - but “state-of-the-art” is not “stateoftheart”,
  - and what about “cheap San Francisco-Los Angeles flights”.

Tokenization, II

Reduce the list of tokens by identification

Case folding:

Move everything into lower case, so searches are case-independent…

But better be careful,

- “USA” might not be “usa”,
- “Windows” might not be “windows”,
- “bush” versus various famous members of a US family…

A very active current research area is Named Entity Recognition.

Tokenization, III

Reduce the list of tokens by simple deletion

Consider removing stopwords: words that will appear in essentially any document, or that will not help to find out what a document is about for some other reason,

- prepositions, articles, some adverbs,
- “emotional flow” words like “essentially”, “hence”. . .
- very common verbs like “be”, “may”, “will”. . .
- But note that “may”, “will”, “can” as nouns are not stopwords!

May reduce index size by up to 40 %.

Current computing power might afford to keep everything in the index anyway, and let the only filter be relevance of each document after found.

Tokenization, IV

Summary

- Language dependent...
- Application dependent...
  - search on a library?
  - search on an intranet?
  - search on the Web?
- Crucial for efficient retrieval!
- Requires to laboriously hardwire into retrieval systems many many different rules and exceptions.

Enriching

Enriching means that each term is associated to additional information that can be helpful to retrieve the "right" documents. For instance,

- Synonims: gun → weapon;
- Related words, definitions: laptop → portable computer;
- Categories: fencing → sports;
- POS tags (part of speech labels):
  - Part-of-speech (POS) tagging.
  - "Un hombre bajo me acompaña cuando bajo a esconderme bajo la escalera a tocar el bajo."
  - "a ship has sails" vs. "John often sails on weekends".
  - "fencing" as sport or "fencing" as setting up fences?

Again, a very active current research area is Word Sense Disambiguation.

Lemmatizing and Stemming, I

Two alternative options

Lemmatizing: reducing the words to their linguistic roots. Sometimes we just analyze suffixes:

swim, swimming, swimmer, swimmed → swim

But in other cases the situation is not that easy:

be, am, are, is → be

gave → give

feet → foot, teeth → tooth,

mice → mouse, dice → die

Stemming is a substitute process that simply removes suffixes and prefixes, with occassional letter changes. It is a replacement for lemmatizing, with “good enough” results, and much simpler and faster.

Lemmatizing and Stemming, II

The stemming choice

Stemmers

are based on rules that indicate when and which prefix or suffix can be removed or rewritten.

- The approach works well for English and Romance languages.
- The most famous algorithm is the Porter stemmer.
  - Available as a Lucene class, a KNIME node, implementations in many programming languages...
  - `http://tartarus.org/martin/PorterStemmer`
- A similar, more evolved, slightly better algorithm based on the same principles is the Snowball stemmer.

Lemmatizing and Stemming, III

An example of the Porter stemmer in action

The Porter Stemmer

is the most commonly employed stemming algorithm. In this example, we apply some preprocessing (case folding and punctuation erasure) and then Porter stemming:

> Stemming is a process that simply removes suffixes and prefixes, with occasional letter changes; it is a replacement for lemmatizing, with good enough results, and much simpler and faster.
>
> stem i a process that simpli remov suffix and prefix with occassion letter chang it i a replac for lemmat with good enough result and much simpler and faster

Lemmatizing and Stemming, IV

Inside stemmers

Inside the Porter Stemmer:

It consists of about 60 rules, distributed in 5 phases.

- Some rules applicable only under conditions (long enough word, other rules have / have not been applied. . .);
- for each phase, choose rule that applies to longest suffix. Some example rules:
  - In phase 1, “sses” replaced by “ss” (“caresses” → “caress”)
  - In phase 2, “(>1)ement” removed (“replacement” → “replac” but does not apply to “cement”).

Lemmatizing and Stemming, V

Pros and Cons of stemming

Pros:

Improves somewhat the effectivity of retrieval;

is almost as effective as lemmatization;

involves less language knowledge than lemmatization;

is faster, easier to describe, and easier to implement.

Cons:

Human-unreadable output, often counter-intuitive;

May reduce to the same stem vastly different words.

Probability Review

Fix distribution over probability space. Technicalities omitted.

$Pr(X)$: probability of event X

$Pr(Y|X) = Pr(X \cap Y)/Pr(X)$ = prob. of Y conditioned to X.

Bayes' Rule (prove it!):

$$Pr(X|Y) = \frac{Pr(Y|X) * Pr(X)}{Pr(Y)}$$

Independence

X and Y are independent if

$$Pr(X \cap Y) = Pr(XY) = Pr(X) * Pr(Y)$$

equivalently (prove it!) if

$$Pr(Y|X) = Pr(Y)$$

Expectation

$$E[X] = \sum_x (x * Pr[X = x])$$

(In continuous spaces an integral is needed instead of the sum.)

Major property: Linearity

- $E[X + Y] = E[X] + E[Y]$,
- $E[\alpha * X] = \alpha * E[X]$,
- and, more generally, $E[\sum_i \alpha_i * X_i] = \sum_i (\alpha_i * E[X_i])$.
- Additionally, if X and Y are independent events, then $E[X * Y] = E[X] * E[Y]$.

Harmonic Series

And its relatives

The harmonic series is $\sum_i \frac{1}{i}$:

- It diverges:

  $\lim_{N\to\infty} \sum_{i=1}^{N} \frac{1}{i} = \infty$.
- Specifically, $\sum_{i=1}^{N} \frac{1}{i} \approx \gamma + \ln(N)$,

  where $\gamma \approx 0.5772\ldots$ is known as Euler's constant.

However, for $\alpha > 1$, $\sum_i \frac{1}{i^\alpha}$ converges.

For example $\sum_i \frac{1}{i^2} = \frac{\pi^2}{6} \approx 1.6449\ldots$

How are texts constituted?

Obviously, some terms are very frequent and some are very infrequent.
Basic questions:

- How many different words do we use frequently?
- How much more frequent are frequent words?
- Can we formalize what we mean by all this?

It turns out there are quite precise empirical laws in most human languages.

Text Statistics, I

Heavy tails

In many phenomena (mostly human-related but also non-human), the governing probability distribution “decreases slowly” compared to Gaussians or exponentials.

This means: very unfrequent objects have substantial weight in total. Some such cases:

- texts, where they were observed by Zipf;
- distribution of people’s names;
- website popularity;
- wealth of individuals, companies, and countries;
- number of links to most popular web pages;
- earthquake intensity.

Text Statistics, II

The frequency of words in a text follows a powerlaw.
For (corpus-dependent) constants a, b, c

$$\text{Frequency of } i\text{-th most common word} \approx \frac{c}{(i+b)^a}$$

(Zipf-Mandelbrot equation).

Postulated by Zipf with $a = 1$ in the 30's.

$$\text{Frequency of } i\text{-th most common word} \approx \frac{c}{i^a}.$$

Further studies: a varies above and below 1.

Text Statistics, III

Power laws

How to detect power laws?

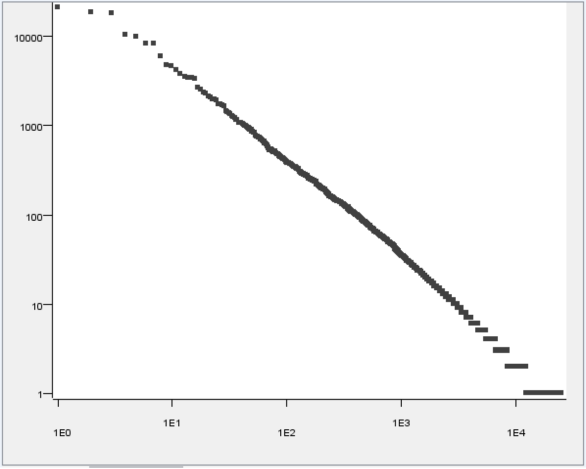
Try to estimate the exponent of an harmonic sequence.

- Sort the items by decreasing frequency.
- Plot them against their position in the sorted sequence (rank).
- Probably you do not see much until adjusting to get a log-log plot:

  That is, running both axes at log scale.
- Then you should see something close to a straight line.
- Beware the rounding to integer absolute frequencies.
- Use this plot to identify the exponent.

Text Statistics, III

Zipf's law in action



Word frequencies in Don Quijote (log-log scales).

Text Statistics, VI

Amount of terms in use

Naturally, longer texts tend to use wider lexicon.

However,

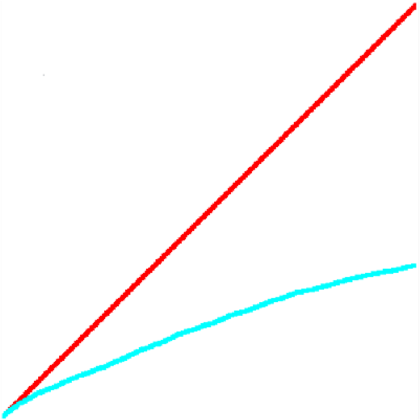
the longer the text already seen, the lesser the chances of finding novel terms.

- The first 2500 words of the Vannevar Bush article we used in the lab include only about 900 different words.
- The first 2500 words of Don Quijote include slightly over 1100 different words.
- The total text of Don Quijote reaches about 383000 words, but only less than 40000 different ones.

Text Statistics, VII

The first 2500 words in the Vannevar Bush article

(The blue line indicates number of different words.)



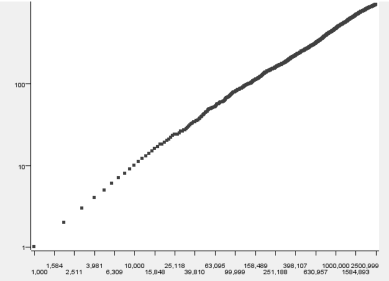
Text Statistics, VIII

Herder’s law, also known as Heaps’ law

The number of different words

is described by a polynomial of degree less than 1.

Again this can be seen by resorting to log-log plots. The blue curve in the previous slide then becomes “more straight”:



Text Statistics, IX

Deriving the formula for Heaps' law

For a text of length N:

Say that we tend to find d words; how to relate d to N?

As a straight line in the log-log plot, we get:

$$\log d = k_1 + \beta * \log N, \text{ that is, } d = k * N^\beta$$

- The value of β varies with language and type of text.
- For the article by Vannevar Bush, we find $\beta \approx 0.836$;
- for Don Quijote, we find $\beta \approx 0.806$.
- In English, lower values of β, down to 0.5, are common.
- Finite vocabulary implies no further growth for very large N (but note: misspellings, proper names, foreign words. . .).

Text Statistics, X

Advanced reading

- Zipf's law and Heaps' law in large corpora: the appearence of two power-law regimes (with different exponents each): Ferrer-i-Cancho & Solé (2001), Gerlach & Altmann (2013).
- The most accurate model for Heaps' law (vocabulary growth as a function of text length): Font-Clos & Corral (2015)
- The hidden invariance in the distribution of word frequencies: Font-Clos et al (2013)

References

- Ferrer-i-Cancho, R. & Solé, R. V. (2001). Two regimes in the frequency of words and the origin of complex lexicons: Zipf's law revisited. Journal of Quantitative Linguistics 8, 165-173. doi: 10.1076/jqul.8.3.165.4101.
- Font-Clos, F., Boleda, G. & Corral (2013). A scaling law beyond Zipf's law and its relation to Heaps'law. New Journal of Physics 15, 093033. doi: 10.1088/1367-2630/15/9/093033
- Font-Clos, F., Corral, A. (2015). Log-log convexity of type-token growth in Zipf's systems. Physical Review Letters 114, 237801. doi: 10.1103/PhysRevLett.114.238701
- Gerlach, M. & Altmann, E. G. (2013). Stochastic Model for the Vocabulary Growth in Natural Languages. Phys. Rev. X 3, 021006. doi: 10.1103/PhysRevX.3.021006

Text Statistics, XI

Advanced reading

The origins of Zipf’s law

- Zipf’s principle of least effort (conflict between hearer and speaker needs)
- Random typing?
- Information theoretic principles (compression).

References

- Ferrer-i-Cancho, R. & Elvevåg, B. (2010). Random texts do not exhibit the real Zipf’s law-like rank distribution. PLoS ONE 5 (3), e9411. doi: 10.1371/journal.pone.0009411
- Ferrer-i-Cancho, R. (2005). Zipf’s law from a communicative phase transition. European Physical Journal B 47, 449-457. doi: 10.1140/epjb/e2005-00340-y
- Ferrer-i-Cancho, R. (2016). Compression and the origins of Zipf’s law for word frequencies. Complexity, in press. doi: 10.1002/cplx.21820