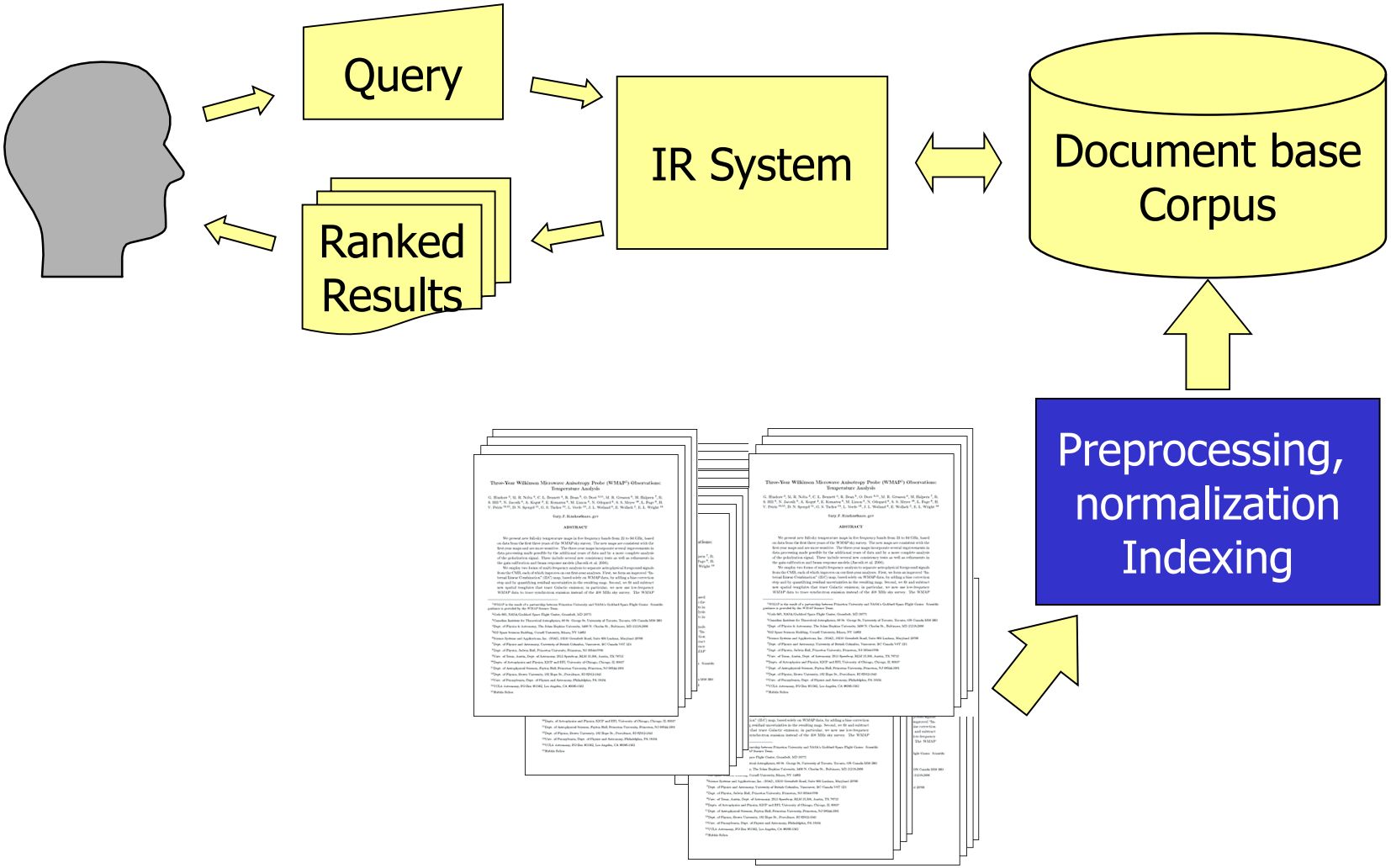
Information Retrieval

Text Preprocessing

Ulf Leser

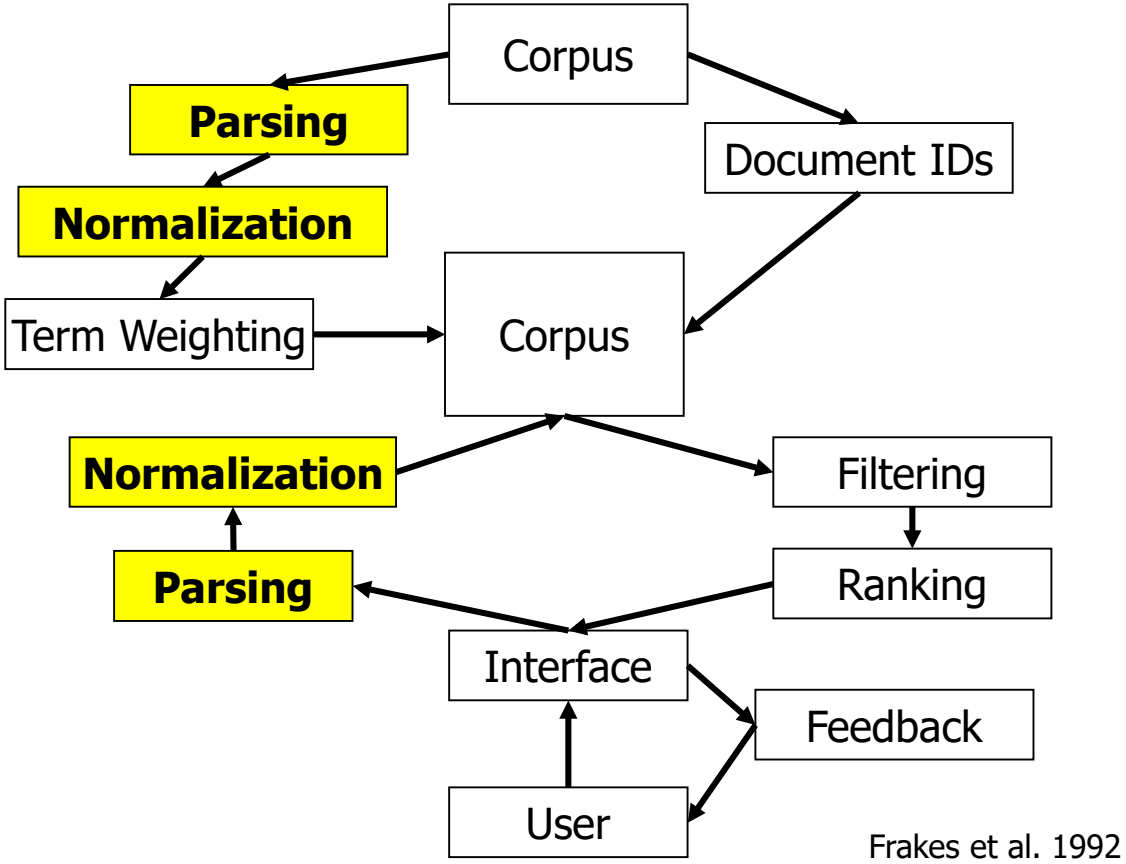
Structure of an IR System



Logical View

- Definition
  - *The logical view of a document denotes its representation inside the IR system*
- Determines what we can query
  - Only metadata, only title, only abstract, full text, …
- Creating the logical view of a doc involves transformations
  - Stemming, stop word removal
  - Transformation of special characters
    - Umlaute, greek letters, XML/HTML encodings, …
  - Removal of formatting information (HTML), tags (XML), …
  - …

Processing Pipeline

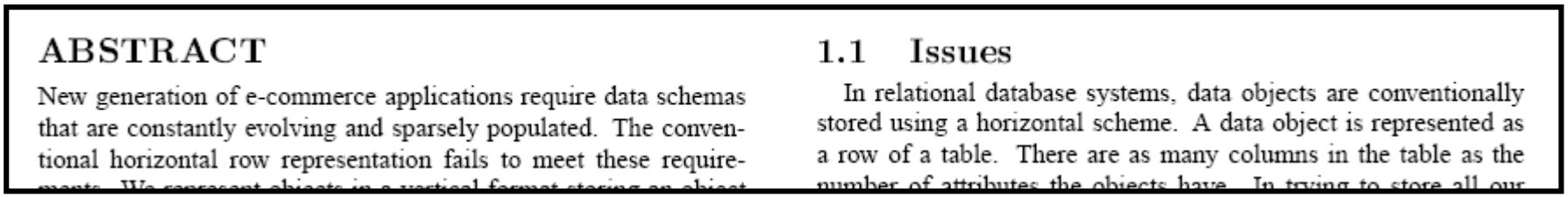


Frakes et al. 1992

Content of this Lecture

- Text Preprocessing
  - Special characters and case
  - Tokenization
  - Stemming and lemmatization
  - Stop words
  - Zipf's law
  - Proper names
  - Document Summarization
  - Annotation and Vocabularies

Format Conversion



ABSTRACT

New generation of e-commerce applications require data schemas that are constantly evolving and sparsely populated. The conventional horizontal row representation fails to meet these require-

1.1 Issues

In relational database systems, data objects are conventionally stored using a horizontal scheme. A data object is represented as a row of a table. There are as many columns in the table as the number of attributes the objects have. In trying to store all our

New generation of e-commerce applications require data schemas In relational database systems, data objects are conventionally that are constantly evolving and sparsely populated. The conven- stored using a horizontal scheme. A data object is represented as tional horizontal row representation fails to meet these require- …

- Transform PDF, XML, DOC, … into ASCII / UniCode
- Problems: Formatting instruction, special characters, formulas, figures and captions, tables, section headers, footnotes, page numbers, margin text, …
- Diplomacy: To what extend can one reconstruct the original document from the normalized representation?

Special Characters

- Umlaute, Greek letters, math symbols, …
- Even if part of ASCII, IR systems don't like them
  - Small alphabets make indexing, searching etc. much easier
  - Indexing Unicode doubles space requirements
- Options
  - Remove special characters
  - Normalize: ü->ue, α-> alpha, $\forall$->for all, Σ->sum? sigma? …
    - XML/HTML: ä <
  - Work with large alphabets (Unicode)
- Filtering characters implies that they cannot be queried
  - How to query for α, Σ, € etc.?

Case – A Difficult Case

- Should all texts be converted to lower case letters?
- Advantages
  - Makes queries simpler
  - Decreases index size
  - Implicitly leads to some "fuzziness" in search
- Disadvantages
  - Fuzziness not always beneficial
  - No abbreviations
  - Loss of important hints for sentence splitting
  - Loss of important hints for tokenization, NER, …
- Different impact in different languages (German / English)
- Often: Convert to lower case only after all other steps

Recognizing Structure Within Documents

- Many documents have structure
  - Chapter, sections, subsections
  - Abstract, introduction, results, discussion, material&methods, …
- Recognizing structure may be very helpful
  - Entities in material&methods are not in the focus of the paper
  - Using only introduction + conclusions often improves ranking
- Approaches
  - Search tags (XML embedding, <h1><h2> in HTML, CSS-Tags, …)
  - Search hints (empty lines, **format changes**, 1.2.3 numbering)
  - Search single-line keywords (“Introduction”, “Results”, …)
- Usage
  - Pre-filtering when creating the logical view
  - As scope for search (“where DMD is contained in the abstract”)
  - As boost for weighting matches

Definitions

- Definition
  - *A document as a sequence of sentences*
  - *A sentence is a sequence of tokens*
  - *A token is the smallest unit of text (words, numbers, …)*
  - *A concept is the mental representation of a "thing"*
  - *A term is a token or a set of tokens representing a concept*
    - *"San" is a token, but not a term*
    - *"San Francisco" are two tokens but one term*
    - *Printed dictionaries usually contain terms, not tokens*
  - *A homonym is a term representing multiple concepts*
  - *A synonym is a term representing a concept which may also be represented by other terms*
- A word can denote either a token or a term

Tokenization

- The basic elements of IR are the token
- Simple approach: Search for „ „ (blanks)
  - "A state-of-the-art Z-9 Firebird was purchased on 3/12/1995."
  - „SQL commands comprise SELECT … FROM … WHERE clauses; the latter may contain functions such as leftstr(STRING, INT)."
  - "This LCD-TV-Screen cost 3,100.99 USD."
  - "[Bis[1,2-cyclohexanedionedioximato(1-)-O]-[1,2-cyclohexanedione dioximato(2-)-O]methyl-borato(2-)-N,N0,N00,N000,N0000,N00000)-chlorotechnetium) belongs to a family of …"
- Typical approach (but many (domain-specific) variations)
  - Treat hyphens / parentheses as blanks
  - Remove "." (after sentence splitting)

Stems versus Lemmas

- Morphology: How words change to reflect tense, case, number, gender, …
- Common idea: Normalize words to a basal "normal" form
  - car,cars -> car; gives, gave, give -> give
- Stemming heuristically reduces words to their stems
  - Stems often not a proper word of the language
  - Quick and dirty, linguistically non-sense
- Lemmatization reduces words to their lemma
  - Finds the linguistic root of a word
  - The lemma must itself be a proper word of the language
  - No algorithmic solution, linguistically meaningful

Example: Porter Stemmer

- Simple, rule-based stemmer for English
  - Porter (1980). "An Algorithm for Suffix Stripping." *Program* 14(3)
  - Based on successive application of a small set of rewrite rules (V: vowels and y; C: consonants)
    - **sses → ss, ies → i, ss → s, s → ∅**
    - **If ((C)*((V)+(C)+)+(V)*eed) then eed → ee**
    - **If (*V*ed or *V*ing) then**
      - **ed → ∅**
      - **ing → ∅**
    - …
- Fast, often-used, available, reasonable results
- Many errors: Arm – army, police – policy, organ – organization, …

Lemmatization

- If possible, lemmatization is the way to go
  - Less false homonyms than with stemming
  - Advantage not so big for English as for German
    - Detached particles: "Kaufst du bitte ein Brot ein?"
    - More forms: "Kaufen, kauf, kaufe, kauft, gekauft, kaufst, …)
- Typical approach: "Vollformenlexikon"
  - Contains an entry for every possible form of a word plus its lemma
  - None available for free for German (to my knowledge)

Content of this Lecture

- Text Preprocessing
  - Special characters and case
  - Tokenization
  - Stemming and lemmatization
  - Stop words
  - Zipf's law
  - Proper names
  - Document Summarization
  - Annotation and Vocabularies

Stop Words

- Stop words: Words that are so frequent that their removal (hopefully) does not change the meaning of a document
  - English: Top-2: 10% of all tokens; Top6: 20%; Top-50: 50%
  - English (top-10; LOB corpus): the, of, and, to, a, in, that, is, was, it
  - German(top-100): aber, als, am, an, auch, auf, aus, bei, bin, …
- Removing stop words reduces a positional index by ~40%
- Hope: Increase in precision due to less spurious hits
  - But be careful with phrase queries
- Variations
  - Remove top 10, 100, 1000, … words
  - Language-specific, domain-specific, or corpus-specific stop word list

Example

The children of obese and overweight parents have an increased risk of obesity. Subjects with two obese parents are fatter in childhood and also show a stronger pattern of tracking from childhood to adulthood. As the prevalence of parental obesity increases in the general population the extent of child to adult tracking of BMI is likely to strengthen.

100 stop words

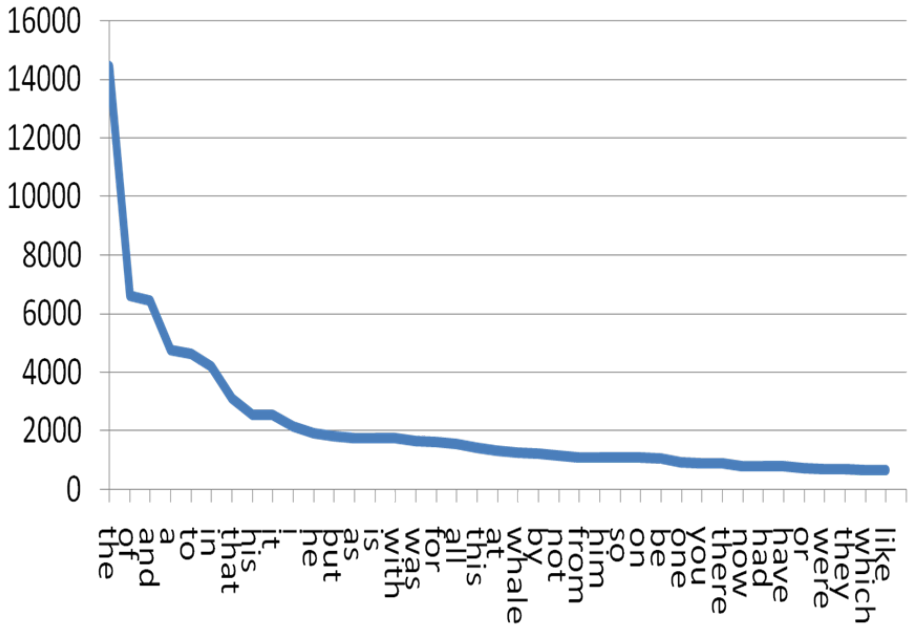
children obese overweight parents increased risk obesity. Subjects obese parents fatter childhood show stronger pattern tracking childhood adulthood. prevalence parental obesity increases general population extent child adult tracking BMI likely strengthen.

10 000 stop words

obese overweight obesity obese fatter adulthood prevalence parental obesity BMI

Zipf's Law (George Kingsley Zipf, 1902-1950)

- Let f be the frequency of a word and r its rank in the list of all words sorted by frequency
- Zipf's law: f ~ k/r for some constant k
- Example
  - Word ranks in Moby Dick
  - Good fit to Zipf's law
  - Some domain-dependency (whale)
- Fairly good approximation for most corpora



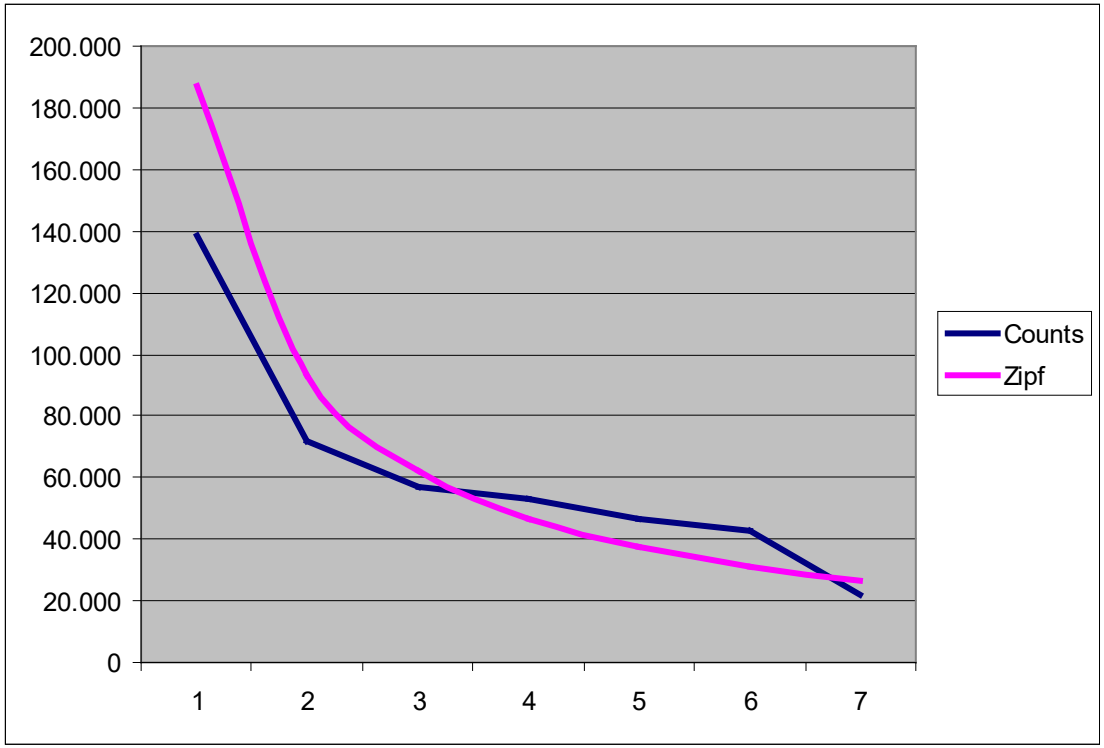
Source: http://searchengineland.com/the-long-tail-of-search-12198

Experiment

Rang $r(w)$	Anzahl $h(w)$	$\frac{r(w) \cdot h(w)}{100.000}$	Wort bzw. Term
1	138.323	1,38323	the
2	72.159	1,4432	of
3	56.750	1,7025	and
4	52.941	2,1176	to
5	46.523	2,3262	a
6	42.603	2,5562	in
7	22.177	1,5524	that
...	...	...	...
2804	73	2,0476	destroy
2805	73	2,0476	determination
...	...	...	...
12032	11	1,3235	would-be
12033	11	1,3236	yachting
12034	11	1,3237	yell

Tabelle 4.1 — Ergebnisse für den *Brown und Lob-Textkorpus*

Quelle: [Hen07]



Proper Names – Named Entity Recognition

- Proper names are different from ordinary terms
  - Often comprise more than one token
  - Often not contained in dictionaries
  - Appear and disappear all the time
  - May contain special characters
  - Very important for information retrieval and text mining
    - Entity search
    - Search for "Bill Clinton"
      - Match "B. Clinton", but not "Clinton ordered the bill"
- Recognizing proper names: Named entity recognition
  - Multi-token, ambiguous, conflict with tokenization, abbreviations, …
  - See lectures in information extraction / text mining

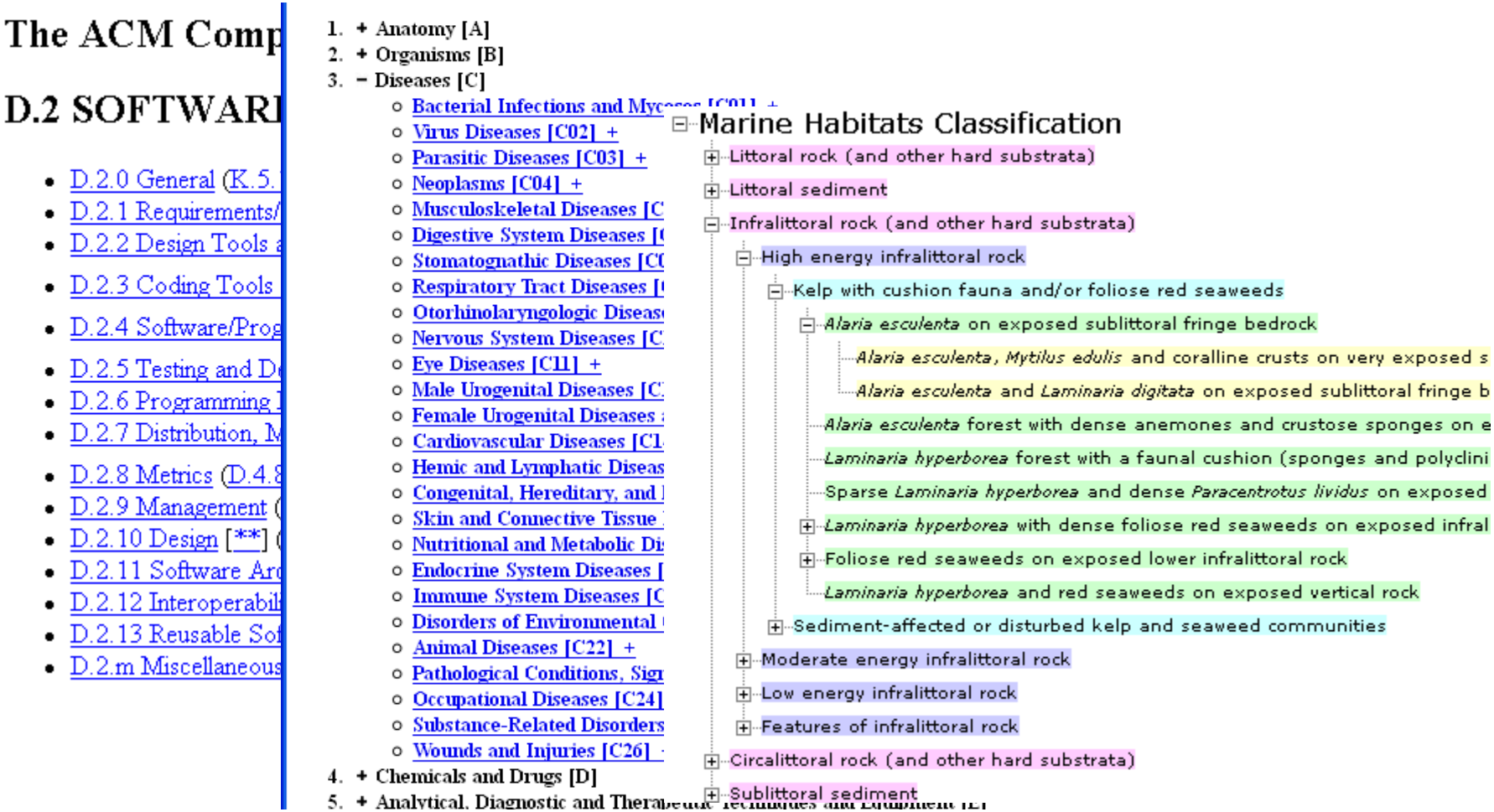
Document Summarization

- Why search the entire document?
- Statistical summarization: Remove token that are not "representative" for the text
  - Only keep words which are more frequent than expected by chance
    - "Chance": Over all documents in the collection (corpus-view), in the language (naïve-view)
  - Related to stop word removal
- Semantic summarization: "Understand" text and create summary of content
- Annotation
  - Categorize / annotate a text with concepts from a controlled dictionary (taxonomy / thesaurus) or with free texts (folksonomy)
  - That's what libraries are crazy about and Yahoo is famous for

Thesaurus

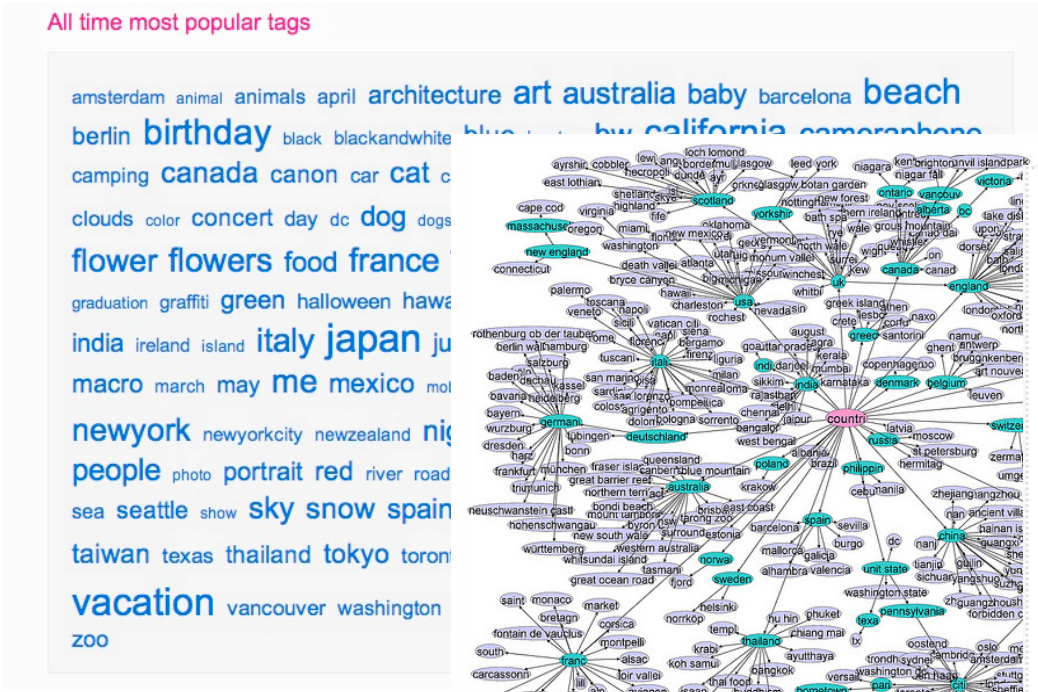
- ISO 2788:1986: Guidelines for the establishment and development of monolingual thesauri
  - „The vocabulary of a controlled indexing language, formally organized so that the a priori relationships between concepts … are made explicit."
- A thesaurus is a set of fixed terms and relationships between them
  - Term = concept = multi token word
  - Relationships: ISA, SYNONYM_OF, PART_OF, …
    - Transitivity: "A goose's leg is part of the goose; a goose is part of a flock of geese; but a goose's leg is not part of the flock of geese"
- Every library has a thesaurus
- Examples: Gene Ontology; MeSH; ACM keywords; …

Thesauri - Examples



Quellen: www.acm.org/; www.ncbi.nih.gov/mesh/; http://www.searchmesh.net/

Folksonomy - Examples



Quellen: www.flickr.com/; www.blogspot.com/

Improvements (as commonly assumed)

Action	Typical Effect
Stemming Latent semantic indexing	Increase recall
Synonym expansion	Increase recall & decrease precision
Domain-specific term weights	Increase precision
Stop-word removal	Not clear

Self Assessment

- List 5 important steps in document preprocessing and their expected impact on precision and recall
- Advantages / disadvantages of lemmatization versus stemming?
- Difference between a folksonomy and a taxonomy?
- Which preprocessing steps are affected if you work with a multi language corpus?