Information Retrieval

Evaluating IR Systems

Text Preprocessing

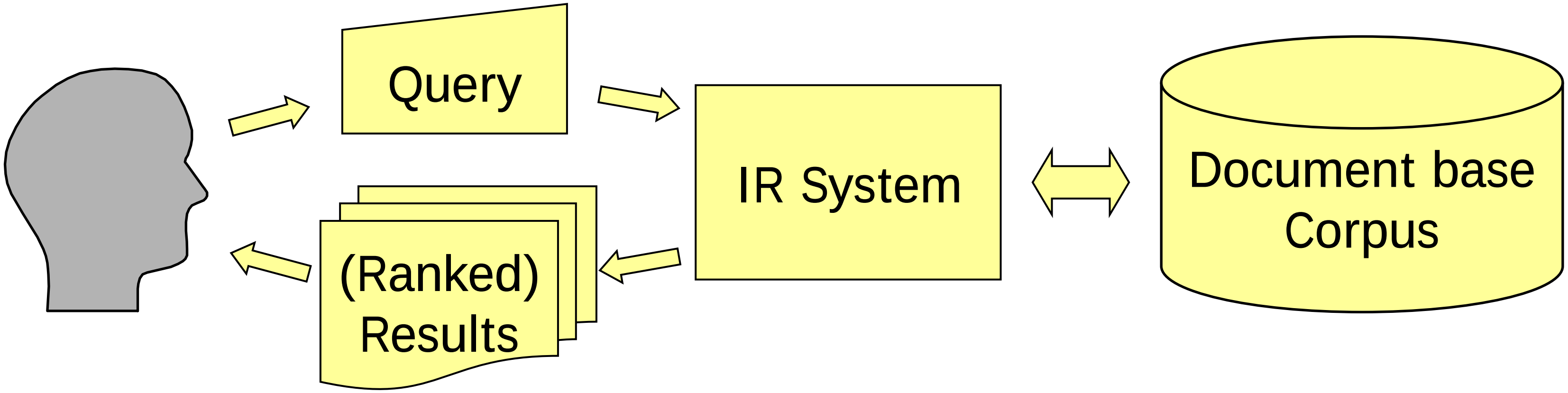
Ulf Leser

Content of this Lecture

- Evaluating IR Systems
- Text Preprocessing

The Informal Problem

- Help user in quickly finding the requested information within a given set of documents
  - Set of documents: Corpus, library, collection, ...
  - Quickly: Few queries, fast responses, simple interfaces, ...
  - Requested: The “best-fitting” documents; the “most relevant” content



Evaluation: Binary Model

- We assume that for a query q and any d∈D, somebody (the truth) determines whether d is relevant for q or not
  - An expert? An average user?
  - Binary decisions: No ranking (for now)
- The IR system (IRS) returns all docs it considers relevant
- Let T be the set of all truly relevant docs, X the set of all IRS-computed docs: |T|=TP+FN, |X|=TP+FP

	Truth: relevant	**Truth: not relevant**
IRS: relevant	True positives	False positives
IRS: not relevant	False negatives	True negatives

Precision and Recall

- Precision = TP/(TP+FP)
  - What is the fraction of relevant answers in X?
- Recall = TP/(TP+FN)
  - What is the fraction of found answers in T?
- The perfect world

	Truth: Relevant	Truth: Not relevant
IRS: Relevant	A	0
IRS: Not relevant	0	B

Example

- Let |D| = 10.000, |X|=15, |T|=20, |X∩T|=9

	Truth: Positive	Truth: Negative
IRS: Positive	TP = 9	FP = 6
IRS: Negative	FN = 11	TN= 9.974

  - Precision = TP/(TP+FP) = 9/15 = 60%
  - Recall = TP/(TP+FN) = 9/20 = 45%

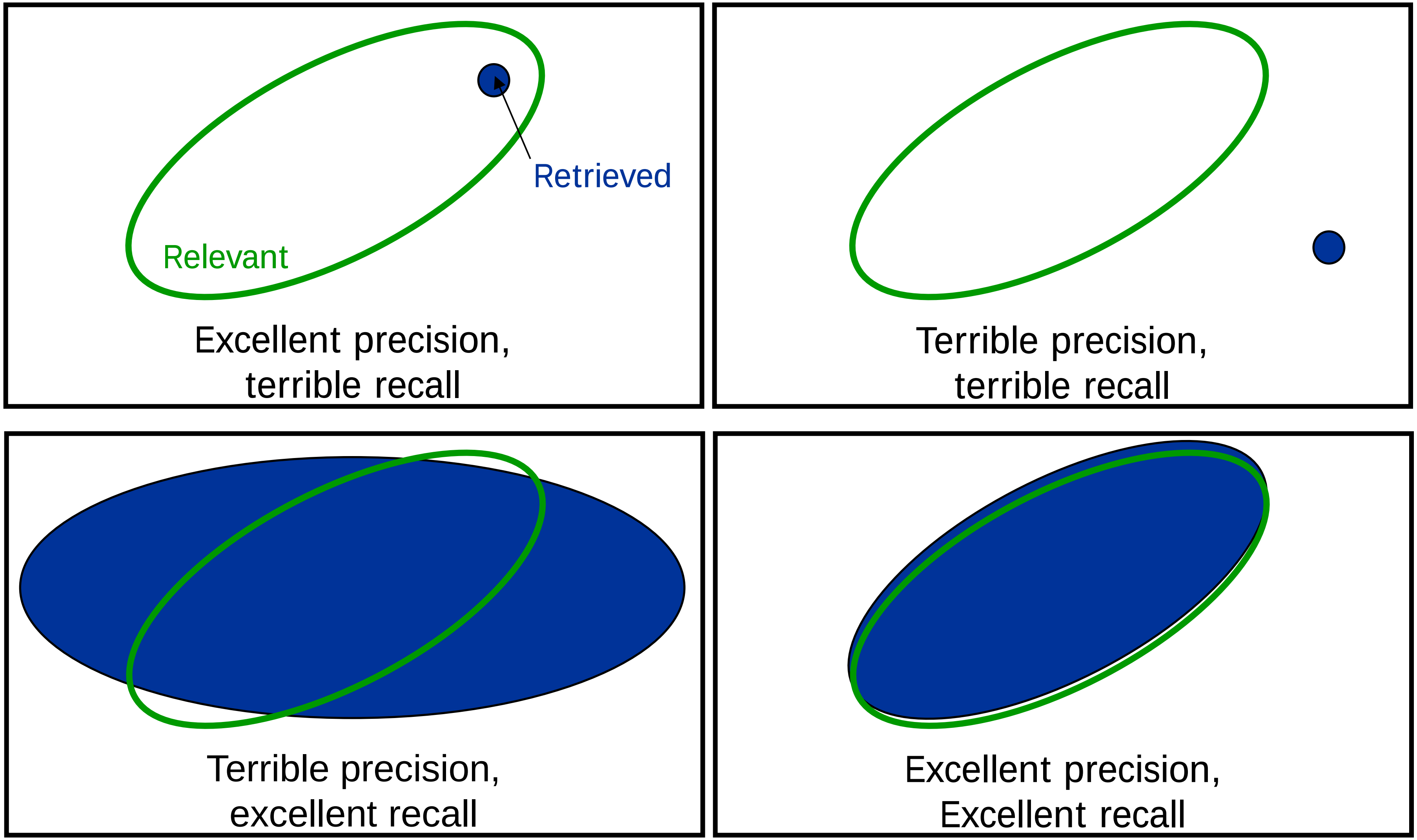
- Assume another result: |X|=10, |X∩T|=7

	Truth: Positive	Truth: Negative
IRS: Positive	TP = 7	FP = 3
IRS: Negative	FN = 13	

  - Precision: 70%, recall = 35%

A Different View

Quelle: A. Nürnberger, VL IR



Excellent precision, terrible recall

Terrible precision, terrible recall

Terrible precision, excellent recall

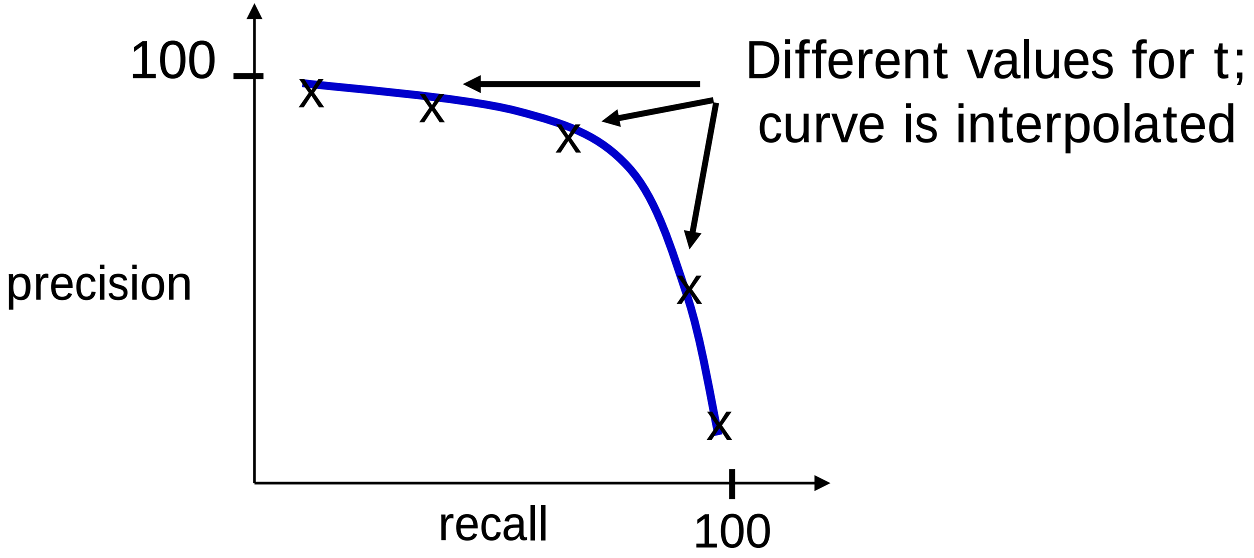
Excellent precision, Excellent recall

Trade-off

- Inherent Trade-off between precision and recall
- Example
  - Most methods compute a similarity score between docs (e.g. VSM)
    - Assume a reasonable score: High sim-score implies high probability of being relevant and vice-versa
  - They use a threshold t to enforce a binary decision
  - Increase t: Less results, most of them very likely relevant
    Precision increases, recall drops
    Set t=1: P~100%, R= ?
  - Decrease t: More results, some might be wrong
    Precision drops, recall increases
    Set t=0: P=?, R=100%

Precision / Recall Curve

- Sliding the threshold t gives a curve



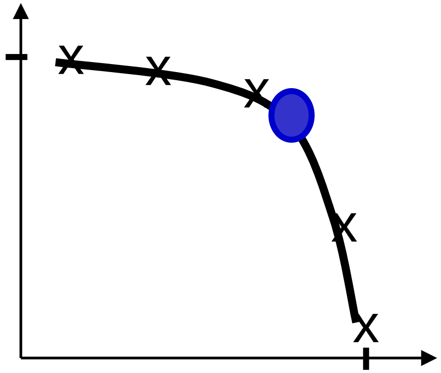
Different values for t; curve is interpolated

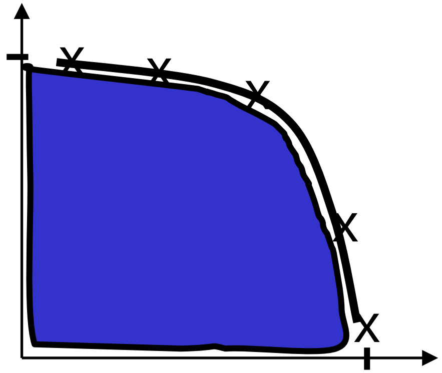
- Typical IR evaluation: Find best point within curve
  - E.g. balanced precision / recall

F-Measure

	Truth: relevant	Truth: not relevant
IR: relevant	TP	FP
IR: not relevant	FN	TN

- Defining one measure instead of two
  - E.g. to rank different IR-systems
- Classical: F1-Measure = 2*P*R / (P+R)
  - F-Measure is harmonic mean between precision and recall
  - Favors balanced P/R values
  - Defines the “best value” for t
- Alternative: Area-under-the-curve, (AUC)





Accuracy

	Truth: relevant	Truth: not relevant
IR: relevant	TP	FP
IR: not relevant	FN	TN

- Accuracy= (TP+TN) / (TP+FP+FN+TN)
  - Which percentage of the systems decision were correct?
  - Makes only sense with small corpora and large result set
  - Typically in IR, TN >>> TP+FP+FN
  - Thus, accuracy is always good (~0,9999995)
- Used in problems with balanced sets of TN / TP

From user/query to users/queries

- What if we have many queries?
  - For evaluation, one should always use a range of different queries
  - Compute average P/R values over all queries
  - Of course, stddev is also important
- What if we have different users?
  - Different users may have different thoughts about what is relevant
  - This leads to different gold standards
  - Compute inter-annotator agreement as upper bound
- Who can judge millions of docs?
  - Evaluate on small gold standard corpus
    - But: Extrapolation difficult: Are the properties of application/corpus really equal to properties of GS?
  - Use implicit feedback, e.g. click-through rates in top-K results

Micro- versus Macro Averages

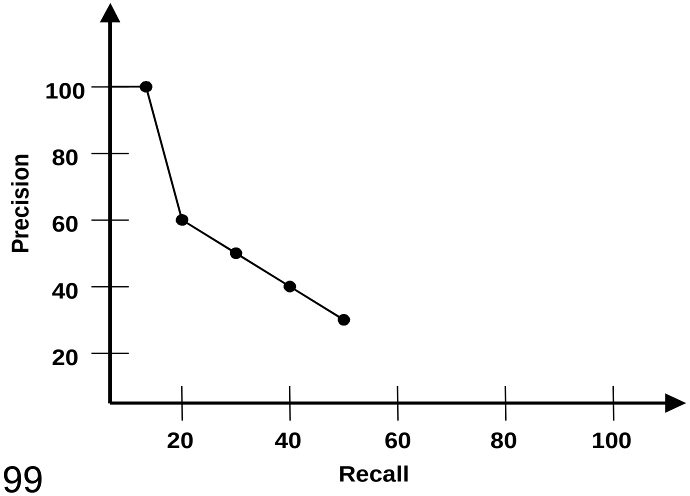
- Two ways of computing an average over m queries
  - Macro-Average: Average P and R over P_1, R_1, ... values of queries
  - Micro-Average: Compute P and R over all TP_1, FP_1, ... values

$$\frac{\sum_{i=1..m} P_i}{m} \neq \frac{\sum_{i=1..m} TP_i}{\sum_{i=1..m} TP_i + \sum_{i=1..m} FP_i}$$

- Comparison
  - Micro-Average implicitly weights queries with result size
  - Macro-Average is less affected by outliers (with large result sizes)

Evaluating Rankings

- Modern IR systems compute ranked answers (sim-score)
- Assume we still have a binary gold standard
- Typical approach: “P/R/F at k”
  - Move a pointer down the sorted list
  - Consider docs above the pointer as “IRS: relevant”
  - Gives one P/R value per list position



Quelle: BYRN99

- Assume there are 10 truly relevant docs and result = {**5**,9,**7**,67,9,**4**,17,3,90,**21**,...}
- At 1st position, IR scores P=100 and R=10 (1 out of 10)
- At 2nd position, P=50, R=10
- Pos 3: 66/20
- Pos 6: 50/30
- ...

Advanced: Evaluate Rankings with Rankings

- Assume uses also have several grades for „relevance“
  - Lickert-scale: Very relevant, quite relevant, neutral …
- Compare user ranking with IRS ranking
  - Solution: Distance function for rankings
    - E.g. Kendall-Tau: Percentage of pairs-wise disagreements
- Users with different rankings: What is the GS-ranking?
  - Solution: Median ranking; ranking with the least total distance to all user rankings
- Things get difficult when rankings may have ties, different rankings rank different sets of objects, or rank-distance should be included
  - Median-ranking becomes NP-hard
  - See: Brancotte et al. (2015). "Rank aggregation with ties„, VLDB

Critics

- Precision and recall are not independent from each other
- Both assume a static process – no user feedback, no second chance
  - Does not evaluate the process-view of IR
- Both ignore important aspects
  - Documents might be relevant yet boring (e.g. duplicates)
  - Different users find different results interesting (personalization)
- Both rely on gold standard
  - Which often don’t exist / are very expensive to create
  - Which might have been defined with a different conception than that of an average user

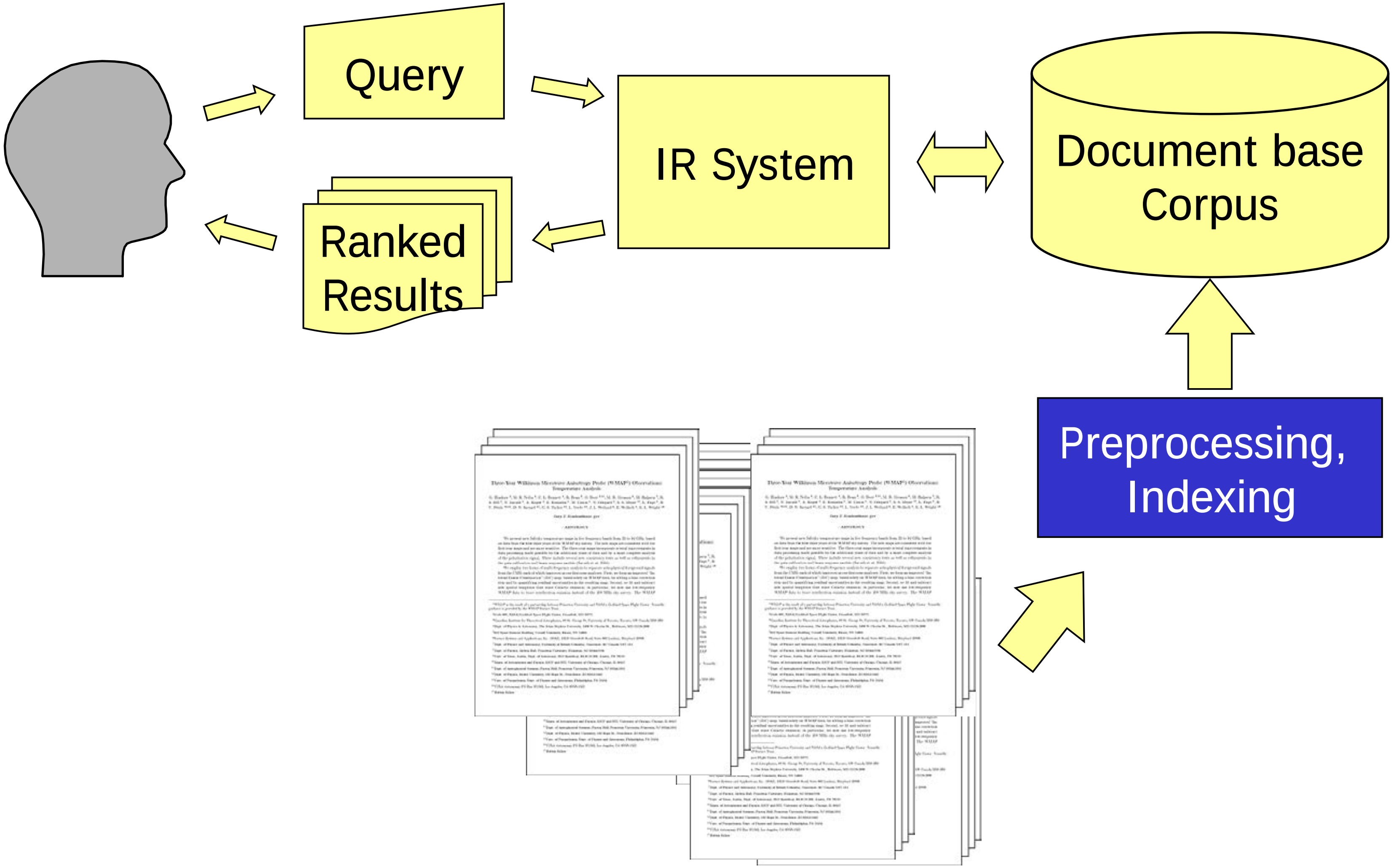
Why „F“-measure [Dave Lewis]

- http://metaoptimize.com/qa/questions/1088/f1-score-name-origin
  - Why is the F1 score called F1?
  - Yes, it was a bizarre lucky break! I was on the MUC program committee, and there was pressure for a single measure of how effective a system was. I knew of the E-measure from Van Rijsbergen's textbook on Information Retrieval, so thought of that.
  - However, *lower* values of E are better, and that just wouldn't do for a government-funded evaluation. I took a quick look in the book, and mistakenly interpreted another equation as being a definition of F as 1-E. I said great, we'll call 1-E the "F-measure". Later I discovered my mistake, but it was too late. Still later, I was reading Van Rijsbergen's dissertation, and saw that he had used E and F in the same relationship, but that hadn't made it into his textbook. Whew.
  - It's a somewhat unfortunate name, since there's an F-test and F-distribution in statistics that has nothing to do with the F-measure. But I guess that's inevitable with only 26 letters. :-)

Content of this Lecture

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

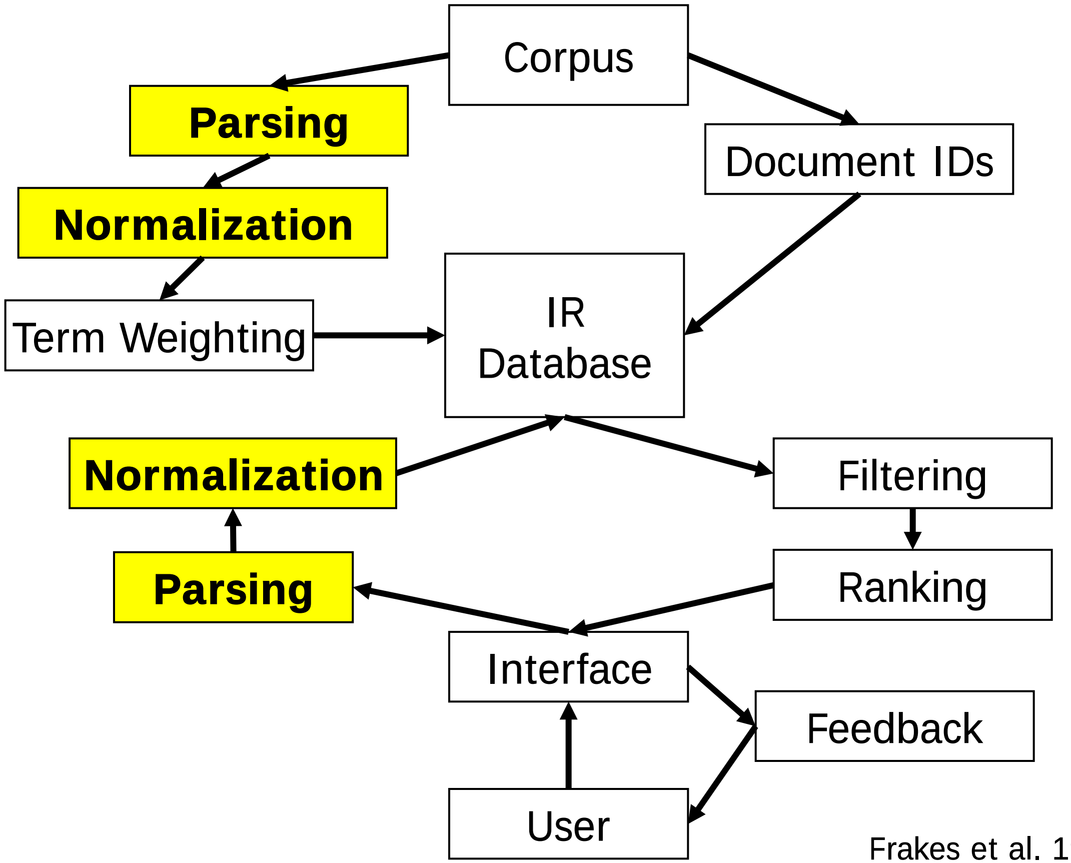
Structure of an IR System



Logical View

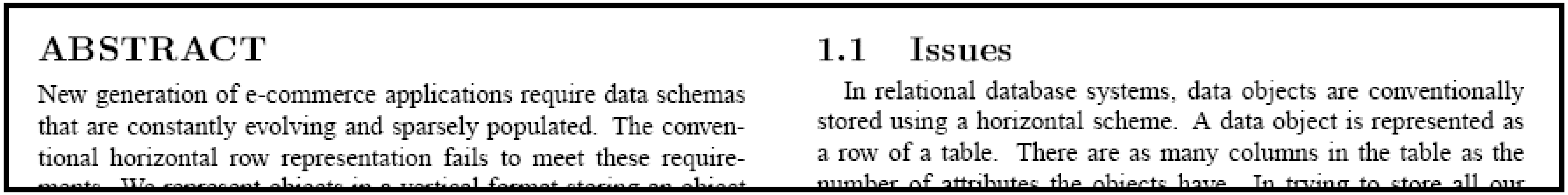
- Definition
  - *The logical view of a document denotes its representation inside the IR system*
- Determines what we can query
  - Only metadata, only title, only abstract, full text, ...
- Creating the logical view of a doc involves transformations
  - Stemming, stop word removal
  - Transformation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

Processing Pipeline



Frakes et al. 1992

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 requirements. We represent objects in a vertical format storing an object

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, ...
- Often part of ASCII/Unicode, but IR systems don’t like them
  - Small alphabets make indexing, searching etc. much easier
- Different way of representing special characters
  - XML/HTML: &nbsp; &auml; &lt;
- Treatment determines of what can be queried
  - How to query for α, ∑, € etc.?
- Options
  - Remove special characters
  - Normalize: ü->ue, α-> alpha, ∀->for all, Σ->sum? sigma? ...
  - Work with large alphabets (Unicode)

Case – A Difficult Case

- Should all text be converted to lower case letters?
- Advantages
  - Makes queries simpler
  - Decreases index size
  - Allows for some “fuzziness” in search
- Disadvantages
  - 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 improves classification
- Approaches
  - Search tags (XML embedding, <h1><h2> in HTML, CSS-Tags, ...)
  - Search hints (empty lines, **format changes**, 1.2.3 numbering)
- 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*
- Word can denote either a token or a term

Tokenization

- Fundamental 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 reduce words to their stems
  - Stems often not a proper word of the language
  - Quick and dirty, linguistically incorrect
- Lemmatization reduce words to their lemma
  - Finds the linguistic root of a word
  - The lemma must itself be a proper word of the language
  - Rather difficult, 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
  - Semantically more helpful, better IR results
  - Does not produce artificial words
  - Advantage not so big for English as for German
    - Detached particles: “Kaufst du bitte ein Brot ein?”
    - Composed nouns: “Donaudampfschifffahrtsgesellschaftskapitän”
- Typical approach: A “Vollformenlexikon”
  - Contains an entry for every possible form of a word plus its lemma
  - Very difficult to build automatically
  - Requires semantic analysis in case of ambiguity
  - None available for free for German

Content of this Lecture

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

Stop Words

- Words that are so frequent that their removal (hopefully) does not change the meaning of a doc
  - 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

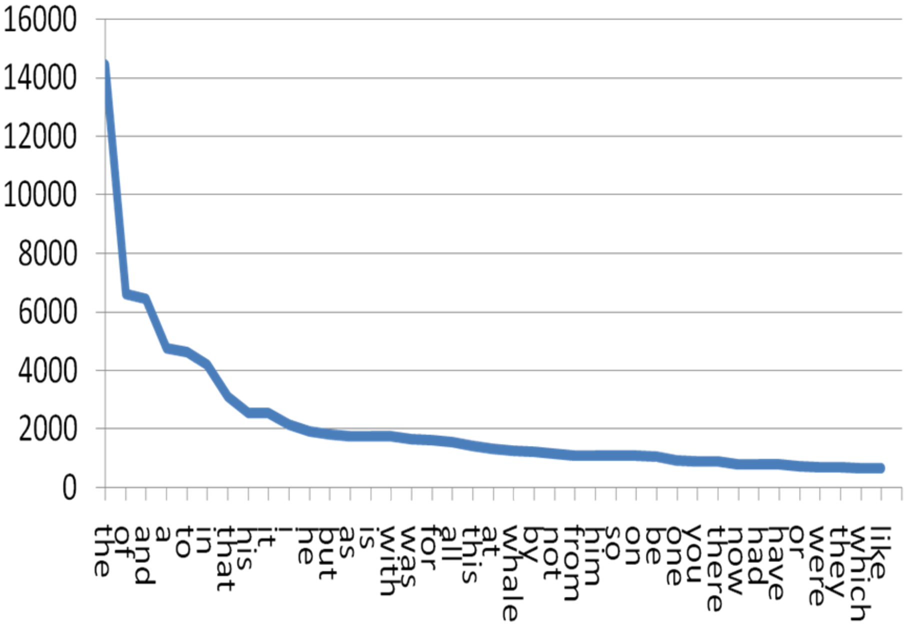
children obese overweight parents increased risk obesity. Subjects obese parents fatter childhood show stronger pattern tracking childhood adulthood. prevalence parental obesity increases general population extent child adult tracking BMI likely strengthen.

10 000 stop words

obese overweight obesity obese fatter adulthood prevalence parental obesity BMI

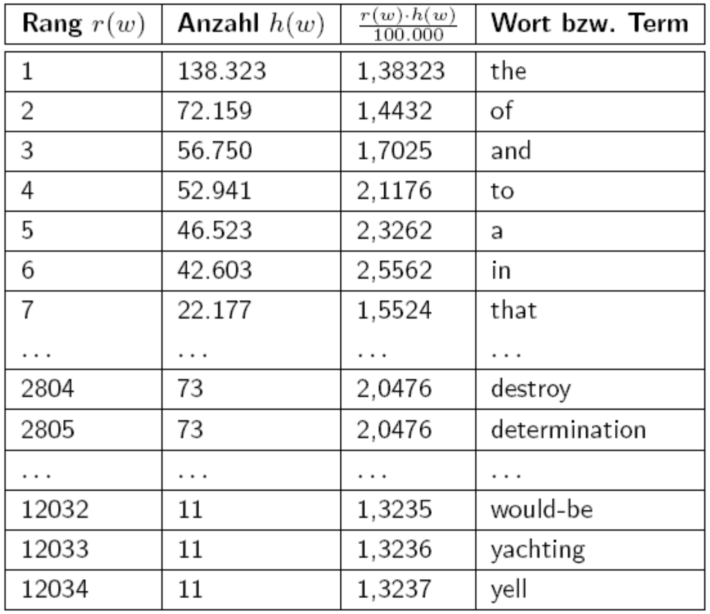
Zipf‘s Law

- 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

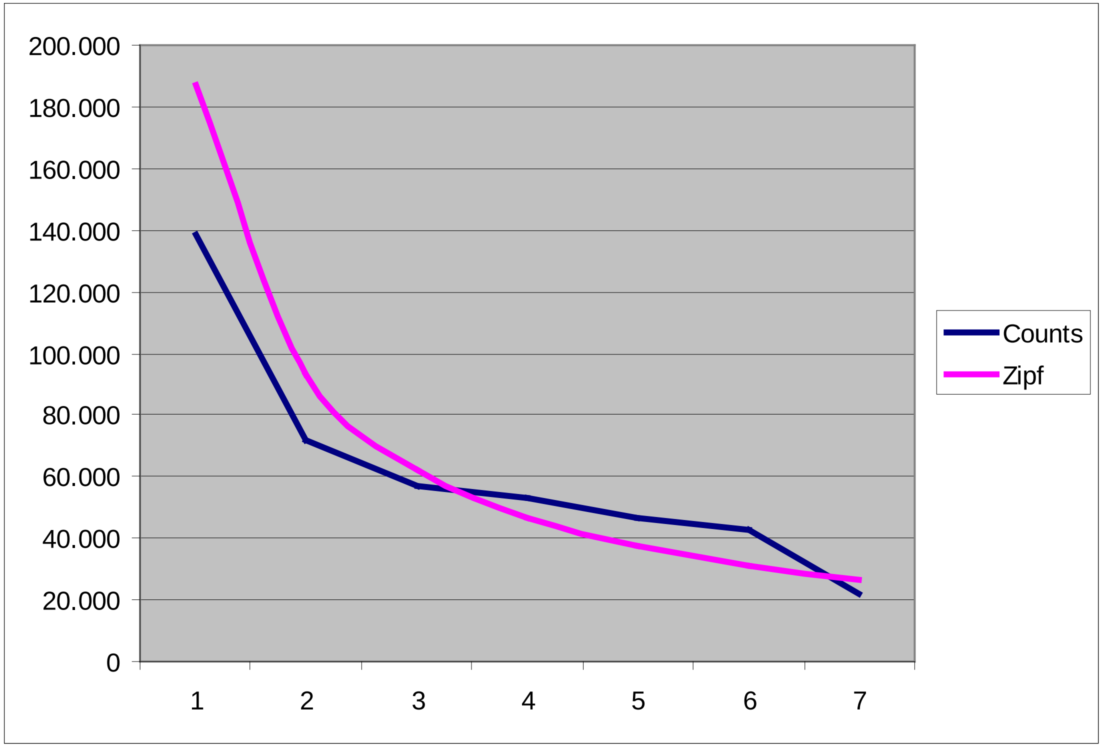
Experiment



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 words
  - Often comprise more than one token
  - Often not contained in language dictionaries
  - Appear and disappear all the time
  - May contain special characters
  - Very important for information retrieval and text mining
    - Search for “Bill Clinton” (not “Clinton ordered the bill”)
- Recognizing proper names: Named entity recognition
  - Multi-token, ambiguous, conflict with tokenization, abbreviations, ...
  - “dark” is a proper name (of a gene)
  - “broken wing” is not, “broken-wing gene” possibly is one
  - “dark” is a gene of the Fruitfly and a common English word

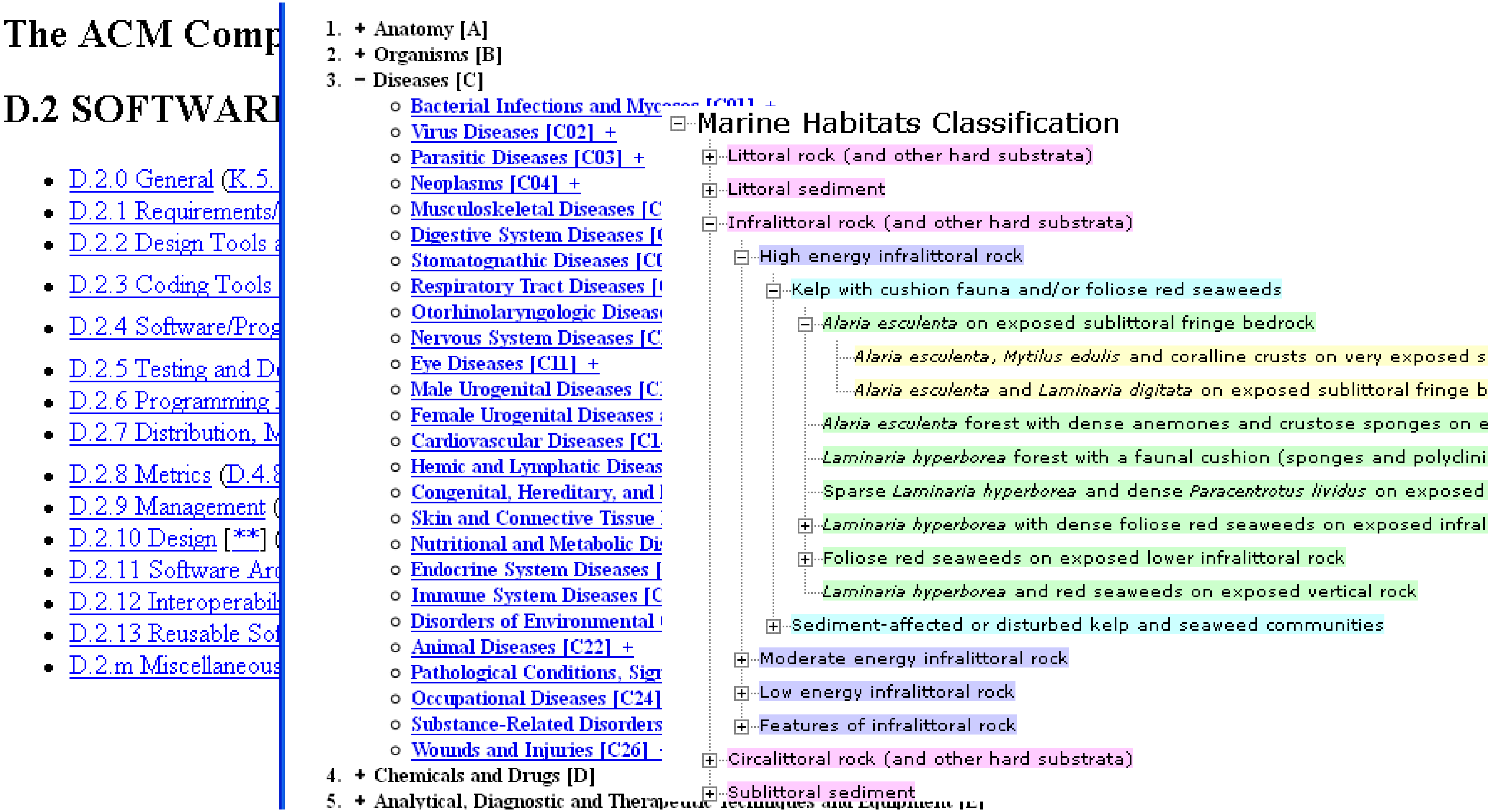
Document Summarization

- 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)
  - Closely related to stop word removal
- Semantic summarization: "Understand" text and create summary of content
- Annotation / classification
  - (Manually) categorize / annotate a text with concepts from a controlled dictionary (taxonomy) 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, ...
    - Aware: “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“
  - The graph made up of one relationship usually must be cycle-free
- Every library has a thesaurus
- Examples: Gene Ontology; MeSH; ACM keywords; ...

Thesauri - Examples



The ACM Comp

D.2 SOFTWARI

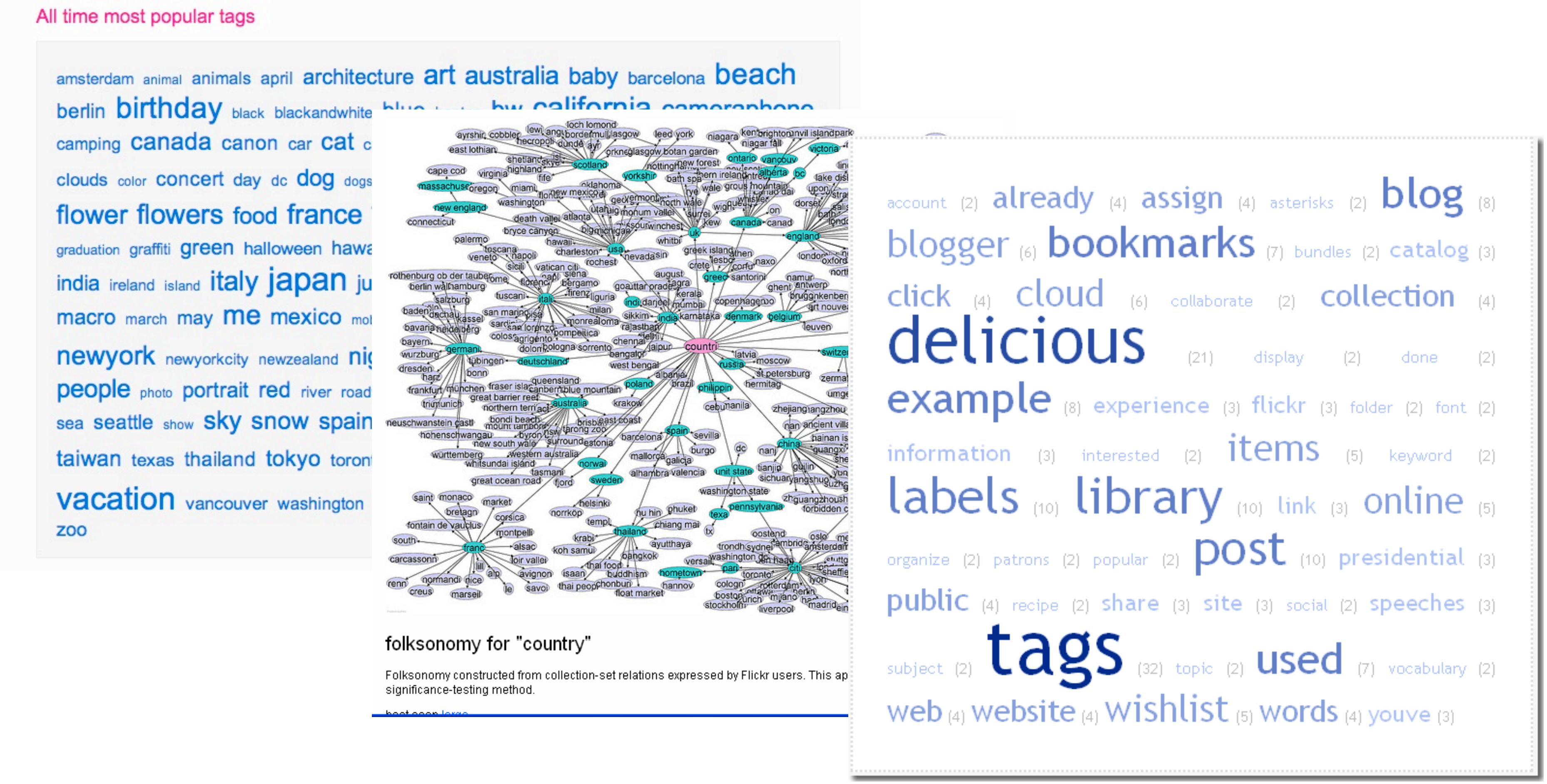
- D.2.0 General (K.5.
- D.2.1 Requirements/
- D.2.2 Design Tools a
- D.2.3 Coding Tools
- D.2.4 Software/Prog
- D.2.5 Testing and De
- D.2.6 Programming I
- D.2.7 Distribution, M
- D.2.8 Metrics (D.4.8
- D.2.9 Management (
- D.2.10 Design [**] (
- D.2.11 Software Arc
- D.2.12 Interoperabil
- D.2.13 Reusable Sof
- D.2.m Miscellaneous

1. + Anatomy [A]
2. + Organisms [B]
3. − Diseases [C]
   - Bacterial Infections and Myc
   - Virus Diseases [C02] +
   - Parasitic Diseases [C03] +
   - Neoplasms [C04] +
   - Musculoskeletal Diseases [C
   - Digestive System Diseases [
   - Stomatognathic Diseases [C
   - Respiratory Tract Diseases [
   - Otorhinolaryngologic Diseas
   - Nervous System Diseases [C
   - Eye Diseases [C11] +
   - Male Urogenital Diseases [C
   - Female Urogenital Diseases
   - Cardiovascular Diseases [C1
   - Hemic and Lymphatic Diseas
   - Congenital, Hereditary, and
   - Skin and Connective Tissue
   - Nutritional and Metabolic Di
   - Endocrine System Diseases [
   - Immune System Diseases [C
   - Disorders of Environmental
   - Animal Diseases [C22] +
   - Pathological Conditions, Sign
   - Occupational Diseases [C24]
   - Substance-Related Disorders
   - Wounds and Injuries [C26]
4. + Chemicals and Drugs [D]
5. + Analytical, Diagnostic and Therap

⊟ Marine Habitats Classification
- ⊞ Littoral rock (and other hard substrata)
- ⊞ Littoral sediment
- ⊟ Infralittoral rock (and other hard substrata)
  - ⊟ High energy infralittoral rock
    - ⊟ Kelp with cushion fauna and/or foliose red seaweeds
      - ⊟ *Alaria esculenta* on exposed sublittoral fringe bedrock
        - *Alaria esculenta*, *Mytilus edulis* and coralline crusts on very exposed s
        - *Alaria esculenta* and *Laminaria digitata* on exposed sublittoral fringe b
      - *Alaria esculenta* forest with dense anemones and crustose sponges on e
      - *Laminaria hyperborea* forest with a faunal cushion (sponges and polyclini
      - Sparse *Laminaria hyperborea* and dense *Paracentrotus lividus* on exposed
      - ⊞ *Laminaria hyperborea* with dense foliose red seaweeds on exposed infral
      - ⊞ Foliose red seaweeds on exposed lower infralittoral rock
      - *Laminaria hyperborea* and red seaweeds on exposed vertical rock
    - ⊞ Sediment-affected or disturbed kelp and seaweed communities
  - ⊞ Moderate energy infralittoral rock
  - ⊞ Low energy infralittoral rock
  - ⊞ Features of infralittoral rock
- ⊞ Circalittoral rock (and other hard substrata)
- ⊞ Sublittoral sediment

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

Folksonomy - Examples



All time most popular tags

amsterdam animal animals april architecture art australia baby barcelona beach
berlin birthday black blackandwhite blue bw california cameraphone
camping canada canon car cat c
clouds color concert day dc dog dogs
flower flowers food france
graduation graffiti green halloween hawa
india ireland island italy japan ju
macro march may me mexico mo
newyork newyorkcity newzealand ni
people photo portrait red river road
sea seattle show sky snow spain
taiwan texas thailand tokyo toron
vacation vancouver washington
zoo

folksonomy for "country"

Folksonomy constructed from collection-set relations expressed by Flickr users. This ap
significance-testing method.

account (2) already (4) assign (4) asterisks (2) blog (8)
blogger (6) bookmarks (7) bundles (2) catalog (3)
click (4) cloud (6) collaborate (2) collection (4)
delicious (21) display (2) done (2)
example (8) experience (3) flickr (3) folder (2) font (2)
information (3) interested (2) items (5) keyword (2)
labels (10) library (10) link (3) online (5)
organize (2) patrons (2) popular (2) post (10) presidential (3)
public (4) recipe (2) share (3) site (3) social (2) speeches (3)
subject (2) tags (32) topic (2) used (7) vocabulary (2)
web (4) website (4) wishlist (5) words (4) youve (3)

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 in VSM Relevance feedback	Increase precision
Stop-word removal	Not clear

Self Assessment

- List 5 important steps in document preprocessing and their expected impact on precision and recall
- Give a definition of recall, precision, and accuracy
- Which relevance models produce a Boolean answer, i.e., no ranking?
- What is “recall at k”? How could we turn this into a single value?
- What is the difference between micro and macro average
- Advantages / disadvantages of lemmatization versus stemming?
- Which preprocessing steps are affected if you work with a multi language corpus?