Text Analytics

Evaluating IR Systems

Text Preprocessing

Ulf Leser

Information Retrieval (IR)

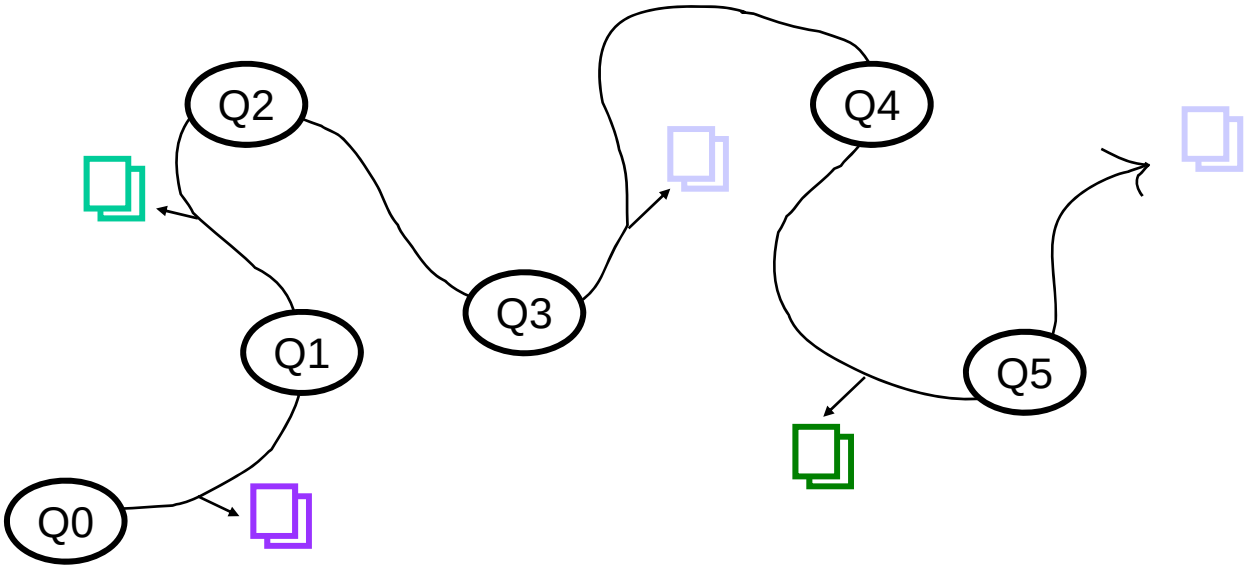
- IR is about helping a user
- IR is about finding information, not about finding data [BYRN99]
- IR builds systems for end users, not for programmers
  - No SQL
  - IR (web) is used by almost everybody, databases are not
- IR usually searches unstructured data (e.g. text)
  - But: Keyword search in relational databases

- 90% of all information is believed to be presented in unstructured form

Why is it hard?

- Homonyms (context)
  - Fenster (Glas, Computer, Brief, ...), Berlin (BRD, USA, ...), Boden (Dach, Fussboden, Ende von etwas, ...), ...
- Synonyms
  - Computer, PC, Rechner, Server, ...
- Specific queries
  - What was the score of Bayern München versus Stuttgart im DFB Pokal in 1998? Who scored the first goal for Stuttgart?
  - How many hours of sunshine on average has a day in Crete in May?
- Typical web queries have 1,6 terms
- "Information broker" is a profession

IR: An Iterative, Multi-Stage, Complex Process



- IR process: “Moving through many actions towards a general goal of satisfactory completion of research related to an information need.”
  - “Berry-picking” [Bates 89]

Documents

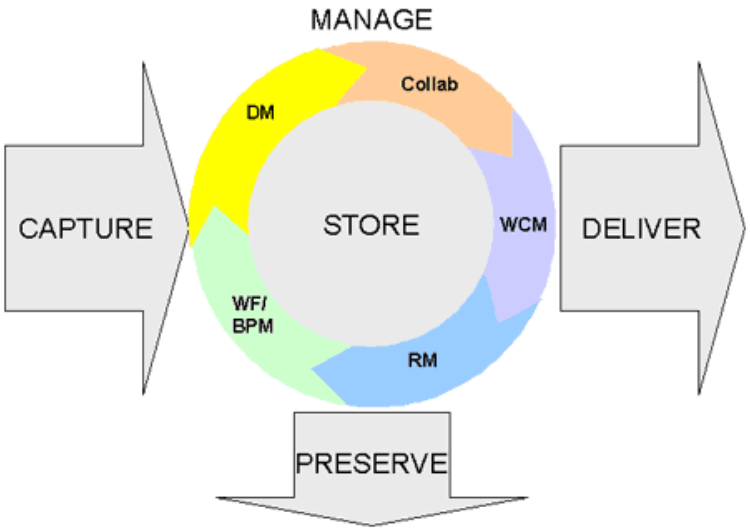
- This lecture: Natural language text
- Might be grammatically correct (books, newspapers) or not (BLOGs, newsgroups, spoken language)
- May have structure (abstract, summary, chapters, ...) or not (plain text)
- May have (explicit or in-text) metadata (title, author, year, ...) or not
- May be in many different languages or even mixed
  - Foreign characters
- May refer to other documents (hyperlinks)
- May have various formats (ASCII, PDF, DOC, XML, ...)

IR Queries

- Users formulate queries
  - Keywords
  - Phrases
  - Logical operations (AND, OR, NOT, ...)
    - Web search: “-ulf +leser”
  - Real questions (e.g. MS-Word help)
  - Structured queries (author=... and title~ ...)
- Querying with document
  - Find documents similar to this one
- Query refinement
  - Refine my query based on first results
  - Find documents within the result set of the previous search

Enterprise Content Management

- „The technologies used to capture, manage, store, deliver, and preserve information to support business processes"



Quelle: AIIM International

- Authorization and authentication
- Business process management and document flow
- Compliance: legal requirements
  - Record management
  - Pharma, Finance, ...
- Collaboration and sharing
  - Inter and intra organizations
  - Transactions, locks, ...
- Publishing: What, when, where
  - Web, catalogues, mail push, ...
- ...

Logical View

- Definition
  - *The logical view on a document denotes its representation inside the corpus*
- 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)

A First Approach: Boolean Queries (rough idea)

- Search all docs which match a logical expression of words
  - bono AND u2, bono AND (Florida OR Texas), bono AND NOT u2
- A simple evaluation scheme
  - Compute BoW of all documents: bow(d)
  - Compute all atoms of the query (w)
  - An atom w evaluates to true on b iff $w \in bow(d)$
  - Compute values of all atoms for each d
  - Compute value of logical expression for each d
- Advantages: Fast, simple
  - Efficient implementations do not scan all documents
- Disadvantages
  - No ranking: Either relevant or not
  - Logic is hard to understand for ordinary users
  - Typical queries are too simple – too many results (and no ranking)

Vector Space Model (rough idea)

- Let d by a doc, D be a collection of docs with $d \in D$
- Let v_d by a vector with
  - $v_d[i]=0$ if the i'th word in bow(D) is not contained in bow(d)
  - $v_d[i]=1$ if the i'th word in bow(D) is contained in bow(d)
- Let v_q by the corresponding vector for query q
- Compute relevance score for q wrt. each d as

$$rel(d,q) = \frac{v_d \bullet v_q}{|v_d| * |v_q|} = \frac{\sum_{i=1}^{|w(D)|} v_d[i] * v_q[i]}{\sqrt{\sum v_d[i]^2} * \sqrt{\sum v_q[i]^2}}$$

Content of this Lecture

- Evaluating IR Systems
- Text Preprocessing

Evaluation: Binary Model

- We assume that for any $d \in D$, the user could decide whether it is relevant or not, if she would read it
- We assume that the IR systems returns all docs it thinks that are relevant
  - No ranking for now
- Let T be the set of all relevant docs, X the set of all returned docs: $|T| = TP + FN$, $|X| = TP + FP$

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

Precision and Recall

		Reality: +	Reality: -
Prediction	+	TruePositive (TP)	FalsePositive (FP)
	-	FalseNegative (FN)	TrueNegative (TN)

- 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

	Real: Positive	Real: Negative
IR: Positive	X	0
IR: Negative	0	Y

Example

- Let |DB| = 1000, |X|=15, |T|=20, |X∩T|=9

	Real: Positive	Real: Negative
IR: Positive	TP = 9	FP = 6
IR: Negative	FN = 11	TN= 9.974

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

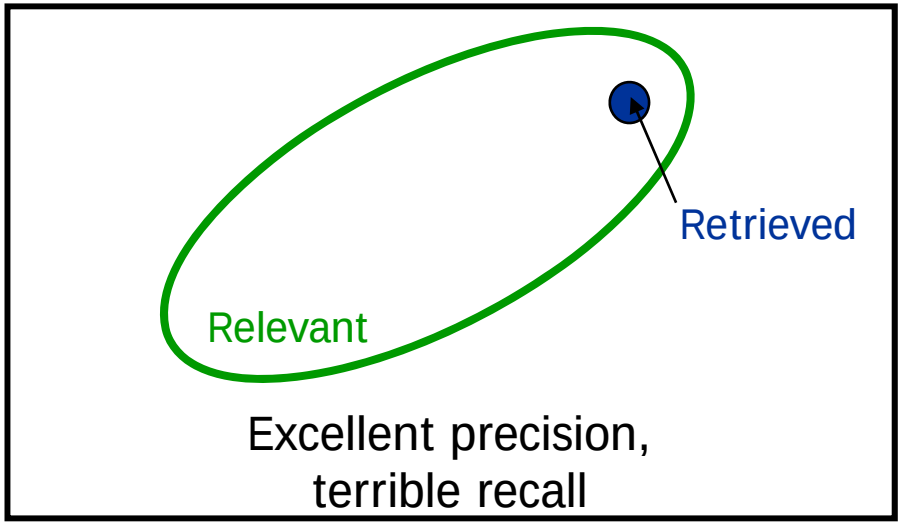
- Assume another result: |X|=10, |X∩T|=7

	Real: Positive	Real: Negative
IR: Positive	TP = 7	FP = 3
IR: Negative	FN = 13	

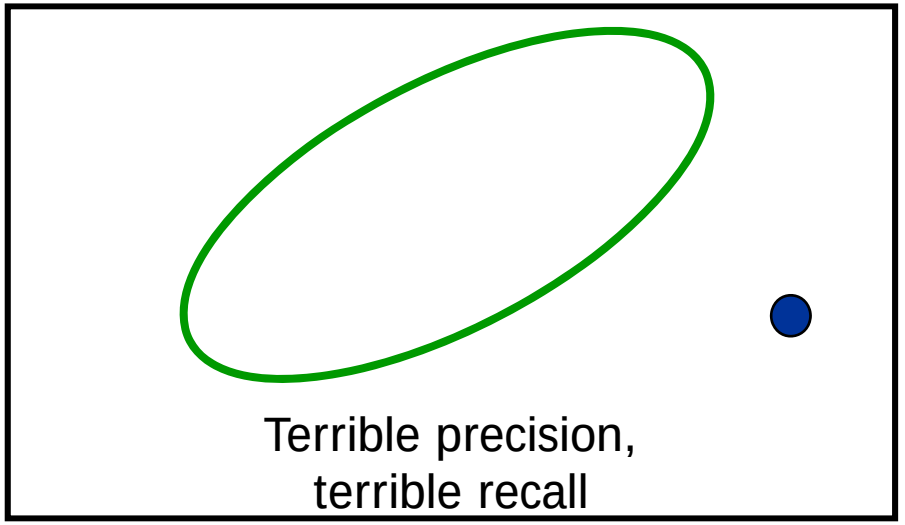
  - Precision: 70%, recall = 35%

A Different View

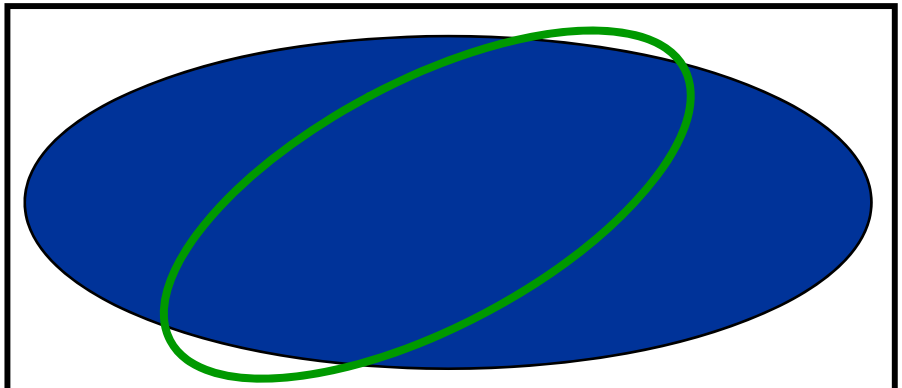
Quelle: A. Nürnberger, VL IR



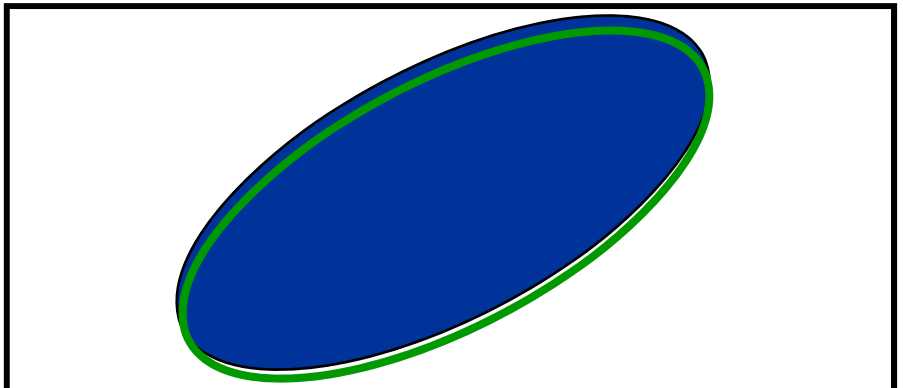
Excellent precision,
terrible recall



Terrible precision,
terrible recall



Terrible precision,
excellent recall



Excellent precision,
Excellent recall

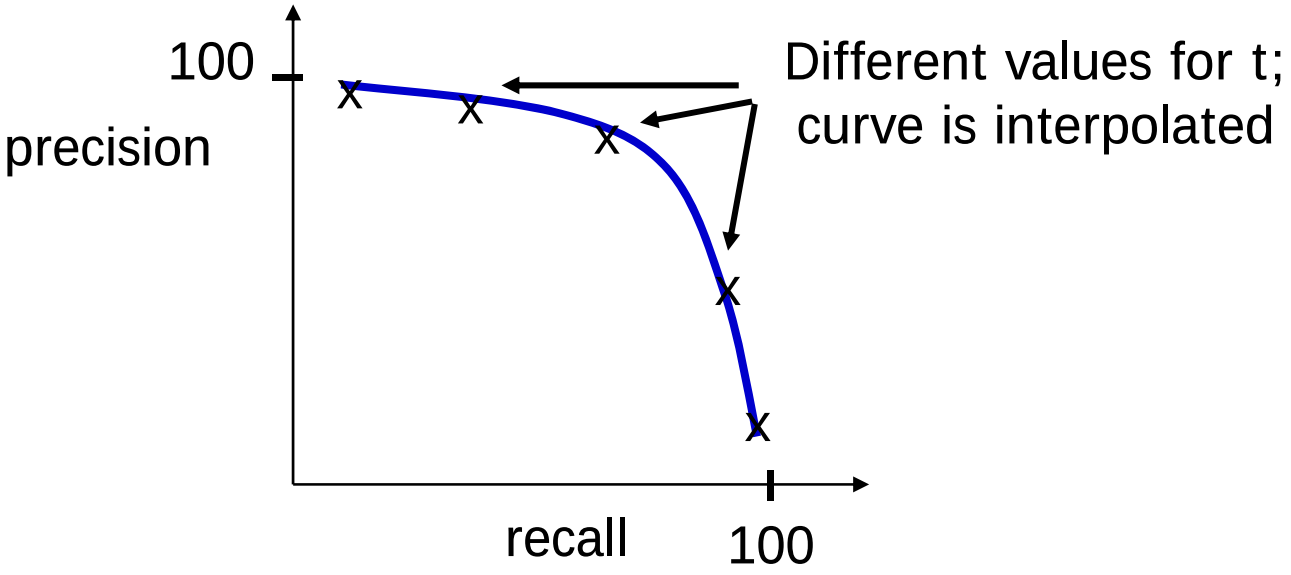
Trade-off

- Trade-off between precision and recall

- Example:
  - Think VSM with a threshold t to enforce a binary decision
  - Assume that docs with a high relevance score are most likely really relevant
  - 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
  - Similar to Receiver-Operator-Curves (ROC)



Accuracy, F-Measure

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

- Accuracy
  - A = (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, and thus accuracy is always good
- F-Measure
  - Sometimes, one wants one measure instead of two
    - E.g. to rank different IR-systems
  - Usual measure: Harmonic mean between precision and recall
    - F-Measure = 2*P*R / (P+R)
    - Favors balanced P/R values
    - Also useful for finding the “best value” for t
- Alternative: Area-under-the-curve, AUC
  - Compute integral of P/R curve

From user/query to users/queries

- What if we have many queries?
  - You should always use a range of different queries
  - Compute average P/R values over all queries
  - Be careful if results sets are of largely differing sizes
  - Of course, mean and stddev are also useful
- 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 a upper bound
- Who can judge millions of docs?
  - Nobody
  - Evaluate on small gold standard corpus
  - Extrapolate to your application
    - Difficult – are properties of application equal to properties of GS

Micro- versus Macro Averages

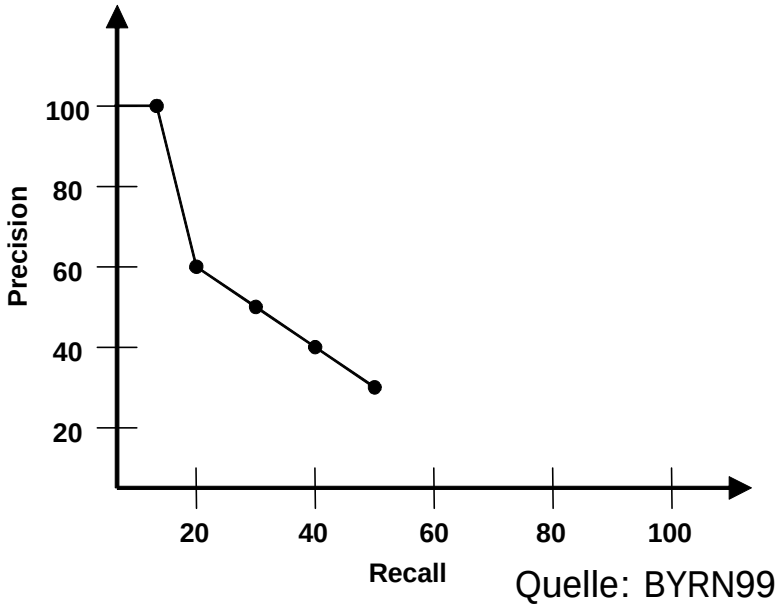
- Two ways of computing an average over many queries
  - Macro-Average: Average P, R, ... over P , R, ... values of queries
  - Micro-Average: Compute P, R, ... over all results

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

- Comparison
  - Micro-Average can cope with queries without results
  - Micro-Average implicitly weights queries with n# of positive results
  - Macro-Average is less affected by outliers (with large result sizes)

Evaluating Rankings

- IR systems do not compute binary but ranked answers
- Typical approach: Standard recall levels / “P/R/F at k”
  - Move a pointer down the sorted list
  - Consider docs above the pointer as “IR: relevant”
  - Gives on 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
- ...

Critics

- Evaluating IR is non-trivial
- Precision and recall are not independent measures
- Both assume a static process – no user feedback, no second chance
- Both ignore the user
  - Documents might be “relevant”, but very boring (e.g. duplicates)
  - Different users may find different results interesting
- Both worship the gold standard, which might have been defined with a different conception than that of an average user
  - Consider inter-annotator agreement

Improvements (as commonly assumed)

Trick	Effect
Stemming Synonym expansion Latent semantic indexing	Increase recall
Give domain-specific weight to terms in VSM Relevance feedback	Increase precision
Stop-word removal	Not clear

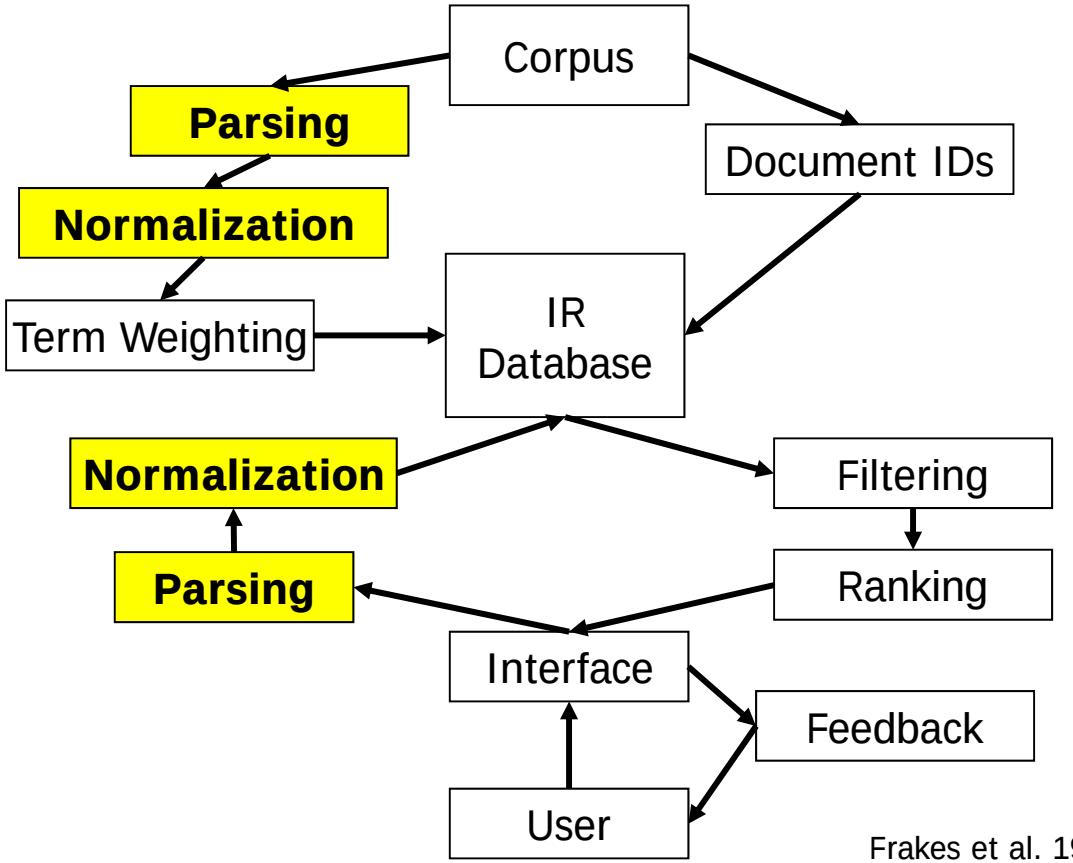
Why „F“-measure

- 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. :-)

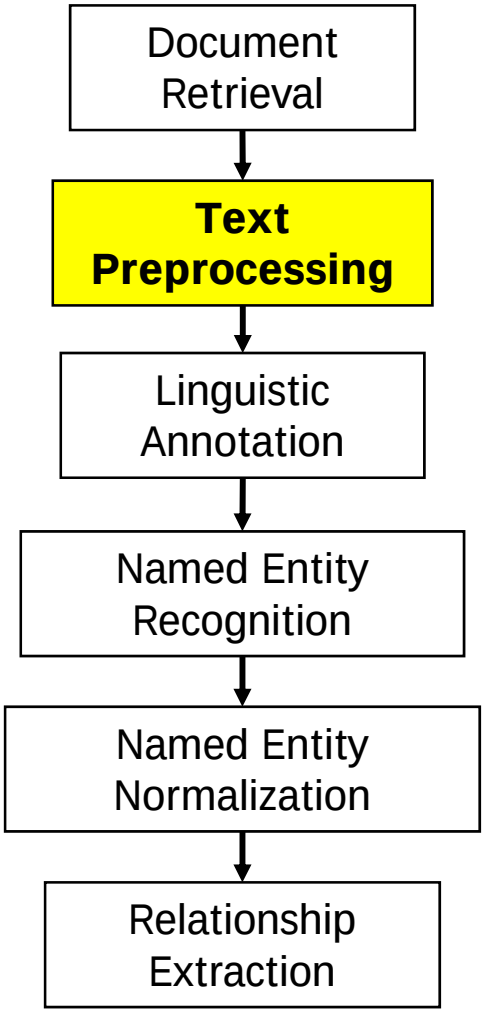
Content of this Lecture

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

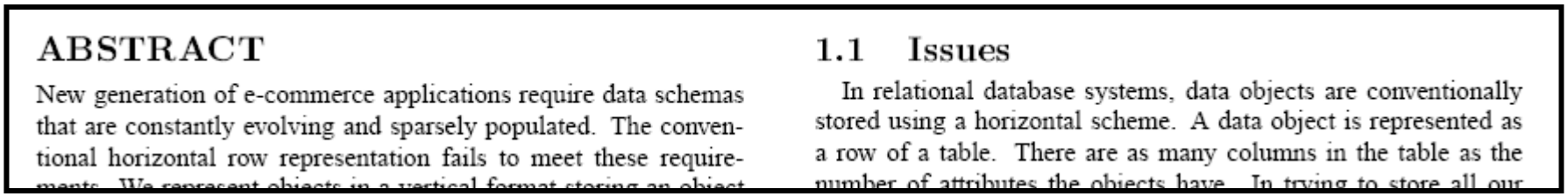
Processing Pipeline



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 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 text into ASCII / UniCode
  - From PDF, XML, DOC, images, ...
- 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, ...
- Many are actually contained in ASCII/UniCode, but IR programs don't like them
  - Standard assumption: ~25 letter alphabet
  - Makes indexing, searching, GUIs etc. much easier
- Some doc formats have their own way of representing special characters
  - XML/HTML: , ä, <,
- Options
  - Remove special characters
  - Transcribe: ü->ue, α-> alpha, $\forall$->for all, Σ->sum? sigma? ...

Case – A Difficult Case

- Should all text be converted to lower case letters?
- Advantages
  - Makes queries simpler
  - Decreases index size
  - Allows for some “fuzzyness” in search
- Disadvantages
  - No abbreviations
  - Loss of important hints for sentence splitting
    - Sentences start with upper case letters
  - Loss of important hints for tokenization, NER, ...
- Difficult impact in different languages (German / English)
- Often: Convert to lower case after all other steps

Recognizing Structure Within Documents

- Many documents have structure
  - Chapter, sections, subsections
  - abstract, introduction, results, discussion, material&methods, ...
- Recognizing them may be very helpful
  - Entities in material&methods are not in the focus of the paper
  - Using only introduction + conclusions improves doc classification
- Approaches
  - Search tags (XML embedding, <h1><h2> in HTML, ...)
  - Search hints (empty lines, **format changes**, 1.2.3 numbering)
- Usage
  - Filtering when creating the logical view
  - As scope for search (“where DMD is contained in the abstract”)

Definitions

- Definition
  - *A document 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*
    - *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

??? Bis hierhin

Sentence Splitting

- Most linguistic analysis works on sentence level
- Sentences group together objects and statements
  - But mind references: “What is this white thing? It is a house.”
- Naive approach: Reg-Exp search for “[.?!;] “
  - (note the blank)
  - Abbreviations
    - “C. Elegans is a worm which ...”; “This does not hold for the U.S.”
  - Errors (due to previous normalization steps)
    - “is not clear.Conclusions.We reported on ...”
  - Proper names
    - “DOT.NET is a technique for ...”
  - Direct speech
    - “By saying “It is the economy, stupid!”, B. Clinton meant that ...”
  - ...

Algorithms

- Place putative sentence boundaries after all occurrences of . ? ! (and maybe ; : —)
- Move the boundary after following quotation marks, if any.
- Disqualify a period boundary in the following circumstances:
  - If it is preceded by a known abbreviation of a sort that does not normally occur sentence finally, but is commonly followed by a capitalized proper name, such as *Prof.* or *vs.*
  - If it is preceded by a known abbreviation and not followed by an uppercase word. This will deal correctly with most usages of abbreviations like *etc.* or *Jr.* which can occur sentence medially or finally.
- Disqualify a boundary with a ? or ! if:
  - It is followed by a lowercase letter (or a known name).
- Regard other putative sentence boundaries as sentence boundaries.

Figure 4.1 Heuristic sentence boundary detection algorithm.

Quelle: [MS99]

- Advanced approaches (Schmid (2000), Mikheev (1998))
  - Machine learning (classification of each "."): Reaches 99.5% accuracy on brown corpus, but slow

Tokenization

- 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
  - Treat hyphens / parentheses as blanks
  - Remove “.” (after sentence splitting)

Stems versus Lemmas

- Common idea: Normalize words to a basal, “normal” form
  - car,cars -> car; gives, gave, give -> give
- Stemming reduce words to a base form
  - Base form often is not a word in 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
- Morphology: How words change to reflect tense, case, number, gender, ...

Stemming

- Separate word stem from suffixes, infixes, prefixes
  - Health, healthy, heal, health-system
- Different rules for different languages / words
  - soy, eres, es, somos, sois, son; compro, compras, compra, compramos, comprais, compran
  - am, are, is, are, are, are; buy, buy, buys, buy, buy, buy
  - Bin, bist, ist, sind, seid, sind; kaufe, kaufst, kauft, kaufen, kauft, kaufen
  - Compre, compraramos; bought, will buy; kaufte, wir werden kaufen
- Particularly hard in German
  - Detached particles: “Kaufst du bitte ein Brot ein?”
  - Composed substantives: “Donaudampfschiffahrtsgesellschaftskapitän”

Example: Porter Stemmer

- Porter, M. F. (1980). "An Algorithm for Suffix Stripping." *Program* **14**(3): 130-137.
  - Simple, rule-based stemmer for English
  - Based on successive application of a small set of rewrite rules (V: vowels and y; C: consonants)
    - **sses → ss, ies → i, ss → ss, 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 meaningful that stemming
  - Does not produce artificial words
  - Advantage not so big for English (as for German)
- Typical approach: A dictionary
  - “Vollformenlexikon”
  - Contains an entry for every possible form of a word plus its lemma
  - Very difficult to build (who knows the lemma?)
  - 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
  - Sentence splitting
  - 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, ...
- Fact: Removing stop words reduces a positional token index by ~40%
- Hope: Increase in precision due to less spurious hits
  - Note stop words in the query
- Variations
  - Remove top 10, 100, 1000, ... words
  - Language-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

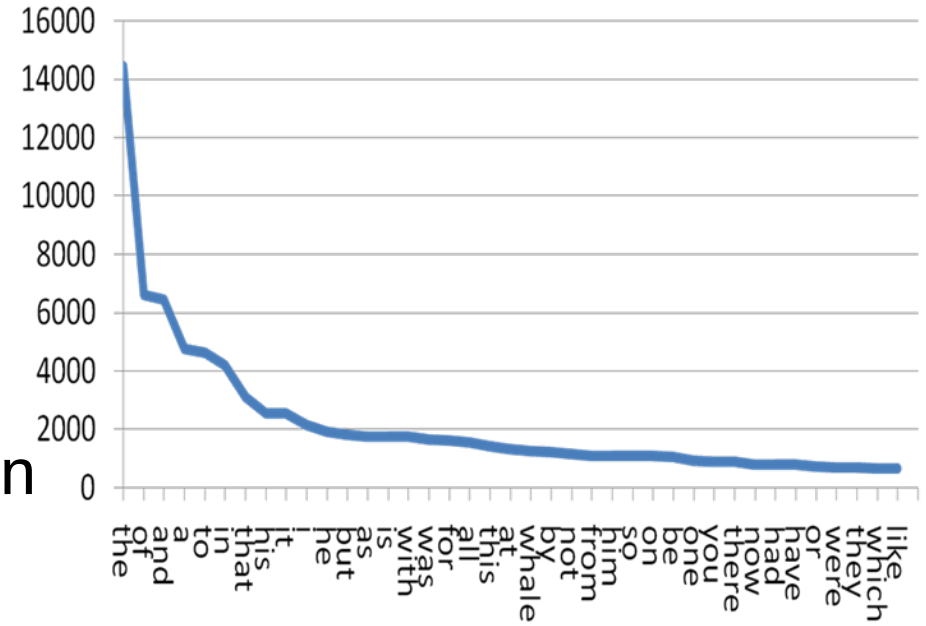
children obese overweight parents increased risk obesity. Subjects obese parents fatter childhood show stronger pattern tracking childhood adulthood. prevalence parental obesity increases general population extent child adult tracking BMI likely strengthen.

10 000 stop words

obese overweight obesity obese fatter adulthood prevalence parental obesity BMI

Zipf‘s Law – Important!

- 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!)
- Zipf’s law is a fairly good approximation for the frequency distribution in most corpora

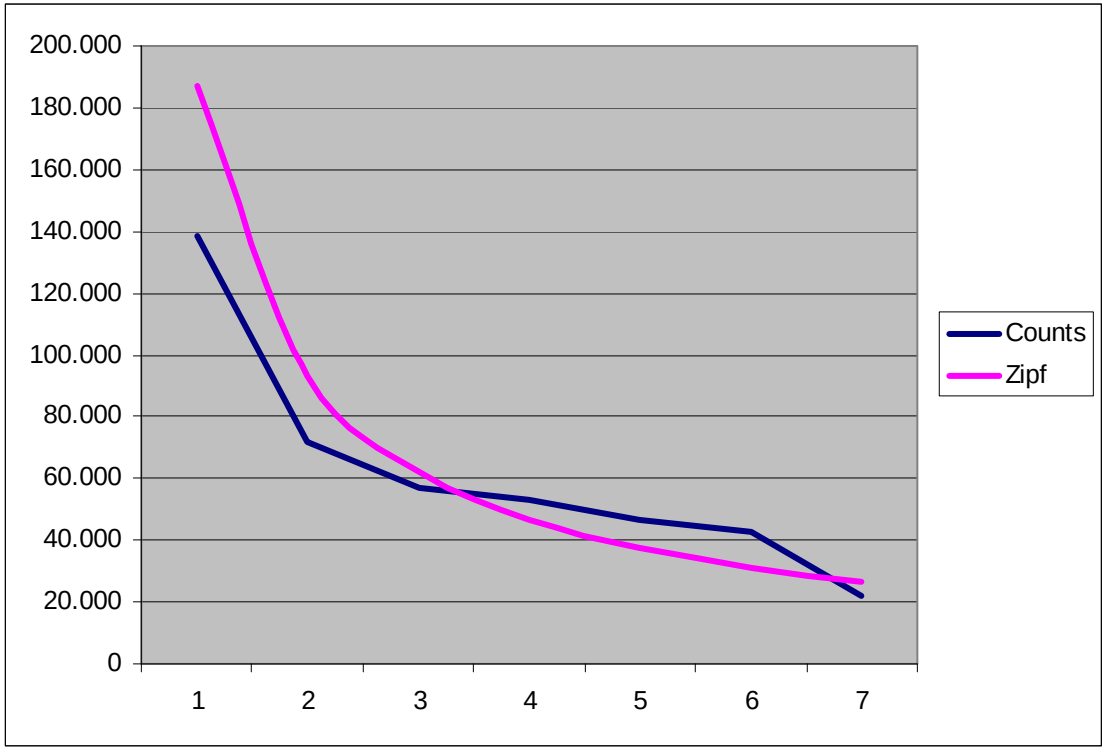


Experiment

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
  - Very often comprise more than one token
  - Often not contained in lexica
  - Appear and disappear all the time
  - Often contain special characters
  - Are very important for information retrieval and text mining
    - Search for “Bill Clinton” (not “Clinton ordered the bill”)
- Recognizing proper names is difficult (more later)
  - Named entity recognition
    - “dark” is a proper name (of a gene)
    - “broken wing” is no proper name, “broken-wing gene” possibly is one
  - Disambiguation
    - “dark” is a gene of the Fruitfly and a common English word

Document Summarization

- Preprocessing may go far beyond what we discussed
- Statistical summarization
  - Remove all 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)
  - Makes amount of text much smaller
- Semantic summarization
  - "Understand" text and create summary of content
- Annotation / classification
  - Have a human understand the text and categorize it / describe it with words from a controlled dictionary
  - 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 (for example as "broader" and "narrower") 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; ...