Text Processing

Günther Specht
Eva Zangerle

Summer Term 2019

Motivations

- Not all words are equally significant for representing the semantics of a document
  - Usually, noun words (or groups of noun words) are the most representative of a document content. (Gender aspect!)

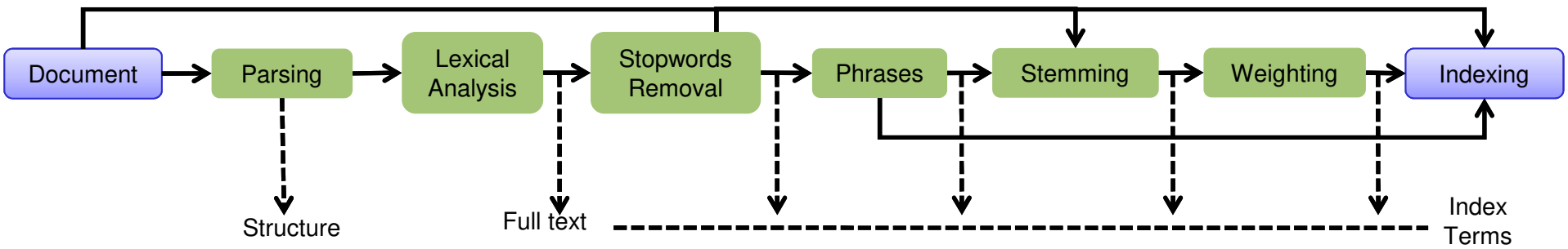
- It is worthwhile to preprocess the text to determine the terms to be used as **index terms**
  - Subset of words selected to represent a document's content

Index Terms and Performance

- Goal: trade off of
  - Exhaustiveness: to assign a big number of terms to a document
  - Specificity: exclude generic terms, concentrate on specific terms
    - Generic terms: low discriminative power, their frequency is high in all the documents (e.g., “and”, “or”, “of”, etc.)
    - Specific terms: higher discriminative power, variable document frequency → their frequency denotes their document’s representativeness

Analysis Process

- **Document Parsing**
- **Lexical analysis**: manage digits, hyphens, punctuation marks, letter cases
- Elimination of **stopwords**
- Matching with a **thesaurus**
- Determination of **phrases** (noun groups)
- **Stemming** (reduction of a word to its grammatical root)
- **Selection** and **weighting** of index terms (noun, adjectives, etc…)



Document Parsing

- What format is it in?
  - pdf/word/excel/html/zip?
  - What language is it in?
  - What character set is in use? (UTF-8, CP1252, …)

- Problems:
  - Documents being indexed can include docs from many different **languages**
  - Sometimes a document or its components can contain **multiple** languages/formats (German email with a English pdf attachment.)

Document Parsing

- After query processing we return “documents” as answer sets,
  but there are often interesting questions of grain size:

- What is a unit document?
  - A file?
  - An email? (Perhaps one of many in a single mbox file)
  - What about an email with 5 attachments?
  - A group of files (e.g., PPT or LaTeX split over HTML pages)

Lexical Analysis

- Process that transforms an input character stream (the original document's text) into a flow of words (**tokens**)
- GOAL: identification of words in the text
- Example
  - Input: "***Friends, Romans and Countrymen***"
  - Output: Tokens
    - ***Friends***
    - ***Romans***
    - ***Countrymen***
  - Each such token is now a candidate for an index entry
  - The general case is not so clear….

Tokenization

- Input: "***Friends, Romans and Countrymen***"
- Output: Tokens
  - ***Friends***
  - ***Romans***
  - ***Countrymen***
- A token is an instance of a sequence of characters
- Each such token is now a candidate for an index entry, after further processing
  - Described below
- But what are valid tokens to emit?

Tokenization

- Issues in tokenization:
  - ***Finland's capital*** $\rightarrow$ ***Finland*** AND ***s***? ***Finlands***? ***Finland's***?
  - ***Hewlett-Packard*** $\rightarrow$ ***Hewlett*** and ***Packard*** as two tokens?
    - ***state-of-the-art***: break up hyphenated sequence.
    - ***co-education***
    - ***lowercase***, ***lower-case***, ***lower case*** ?
    - It can be effective to get the user to put in possible hyphens
  - ***San Francisco***: one token or two?
    - How do you decide it is one token?

Tokenization: Numbers

- ***3/20/91*** ***Mar. 12, 1991*** ***20/3/91***
- ***55 B.C.***
- ***B-52***
- ***My PGP key is 324a3df234cb23e***
- ***(800) 234-2333***

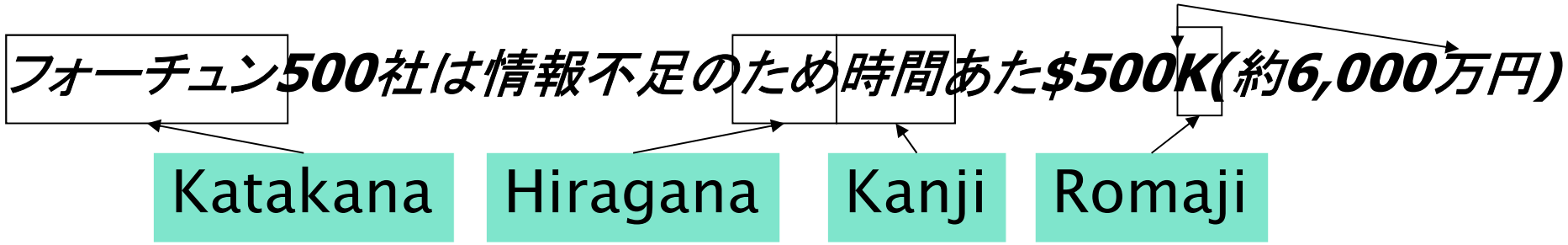
- ***Numbers***
  - Often have embedded spaces
  - Older IR systems may not index numbers
    - But often very useful: think about things like looking up error codes/stacktraces on the web
    - (One answer is using n-grams: IIR ch. 3)
  - Will often index "meta-data" separately
    - Creation date, format, etc.

Tokenization: Language Issues

- French
  - ***L'ensemble*** → one token or two?
    - ***L*** ? ***L'*** ? ***Le*** ?
    - Want ***l'ensemble*** to match with ***un ensemble***
      - Until at least 2003, it didn't on Google
        - Internationalization!
- German noun compounds are not segmented
  - ***Lebensversicherungsgesellschaftsangestellter***
  - 'life insurance company employee'
  - German retrieval systems benefit greatly from a **compound splitter** module
    - Can give a 15% performance boost for German

Tokenization: Language Issues

- Chinese and Japanese have no spaces between words:
  - 莎拉波娃现在居住在美国东南部的佛罗里达。
  - Not always guaranteed a unique tokenization
- Further complicated in Japanese, with multiple alphabets intermingled
  - Dates/amounts in multiple formats

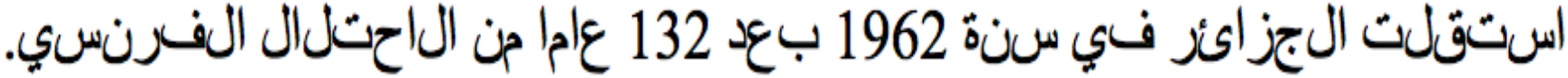


End-user can express query entirely in hiragana!

Tokenization: Language Issues

- Arabic (or Hebrew) is basically written right to left, but with certain items like numbers written left to right

- Words are separated, but letter forms within a word form



← → ← → ← start

(‘Algeria achieved its independence in 1962 after 132 years of French occupation.’)

- With Unicode, the surface presentation is complex, but the stored form is straightforward

Stop Words

- With a stop list, you exclude from the dictionary entirely the commonest words. Intuition:
  - They have little semantic content: *the, a, and, to, be*
  - There are a lot of them: ~30% of postings for top 30 words

- But the trend is away from doing this:
  - Good compression techniques means the space for including stop words in a system is very small
  - Good query optimization techniques mean you pay little at query time for including stop words.
  - You need them for:
    - Phrase queries: “King of Denmark”
    - Various song titles, etc.: “Let it be”, “To be or not to be”
    - “Relational” queries: “flights to London”

Normalization to Terms

- We may need to "normalize" words in indexed text as well as query words into the same form
  - We want to match ***U.S.A.*** and ***USA***
- Result are terms: a term is a (normalized) word type, which is an entry in our IR system dictionary
- We most commonly implicitly define equivalence classes of terms by, e.g.,
  - deleting periods to form a term
    - ***U.S.A., USA***
    - deleting hyphens to form a term
    - ***anti-discriminatory, antidiscriminatory***

Normalization: Other Languages

- Accents: e.g., French ***résumé*** vs. ***resume*.**
- Umlauts: e.g., German: ***Tübingen*** vs. ***Tuebingen***
  Should be equivalent
- Most important criterion:
  - How are your users like to write their queries for these words?

- Even in languages that standardly have accents, users often may not type them
  - Often best to normalize to a de-accented term
    - ***Tuebingen, Tübingen, Tubingen*** → ***Tubingen***

Normalization: Other Languages

- Normalization of things like date forms
  - ***7月30日 vs. 7/30***
  - ***Japanese use of kana vs. Chinese characters***

- Tokenization and normalization may depend on the language and so is intertwined with language detection

- Crucial: Need to "normalize" indexed text as well as query terms identically

Case Folding

- Reduce all letters to lower case
  - Exception: upper case in mid-sentence?
    - e.g., General Motors
    - Fed vs. fed
    - SAIL vs. sail
  - Often best to lower case everything, since users will use lowercase regardless of 'correct' capitalization…

- Longstanding Google example: [fixed in 2011…]
  - Query C.A.T.
  - #1 result is for "cats" (well, Lolcats) not Caterpillar Inc.

Normalization to Terms

- An alternative to equivalence classing is to do asymmetric expansion of search queries

- An example of where this may be useful
  - Enter: ***window*** Search: ***window, windows***
  - Enter: ***windows*** Search: ***Windows, windows, window***
  - Enter: ***Windows*** Search: ***Windows***

- Potentially more powerful, but less efficient

Thesauri and Soundexes

- Do we handle synonyms and homonyms?
  - E.g., by hand-constructed equivalence classes
    - ***car*** = ***automobile*** ***color*** = ***colour***
  - We can rewrite to form equivalence-class terms
    - When the document contains ***automobile***, index it under ***car-automobile*** (and vice-versa)
  - Or we can expand a query
    - When the query contains ***automobile***, look under ***car*** as well
- What about spelling mistakes?
  - One approach is Soundex, which forms equivalence classes of words based on phonetic heuristics

Thesaurus

- A thesaurus is as a **classification** scheme composed of words and phrases whose organization aims at facilitating the expression of ideas in written text
  - Example entry from Roget's thesaurus: cowardly **adjective**
    - Ignobly lacking in courage.
    - Syns: chicken (slang) chicken-hearted, craven, dastardly, faint-hearted, gutless, lily-livered
- A thesaurus can be
  - Thematic: specific to the IR system's domain of application (most frequent case)
    - E.g.: Thesuarus of Engineering and Scientific Terms
  - Generic
- A thesaurus can be used to
  - **Help** user formulate queries
  - **Modification** of queries by the system
  - **Select** index terms

Thesauri

- Many kinds of thesauri have been developed for IR systems
  - Hierarchical: *synonyms* (RT → related terms, UF → use for), *generalization* (BT → broader term), *specialization* (NT → narrower term)
    - Manually built and updated by domain experts
  - Clustered: cluster (or synset) of words having strong semantic relationship
  - Associative: graph of words, where nodes represents words and edges represents *semantic similarity* among words
    - Edges can be oriented or not, according to the symmetry of the similarity relationship
    - Edged can be weighted (fuzzy pseudo-thesauri)
    - Can be automatic generated from a collection of documents using a co-occurrence relationships

WordNet

WordNet Search - 3.1

- WordNet home page - Glossary - Help

Word to search for: dog Search WordNet

Display Options: (Select option to change) Change

Key: "S:" = Show Synset (semantic) relations, "W:" = Show Word (lexical) relations

Display options for sense: (frequency) {offset} <lexical filename > [lexical file number] (gloss) "an example sentence"

Display options for word: word#sense number (sense key)

Noun

- (42){02086723} <noun.animal>[05] S: (n) **dog#1 (dog%1:05:00::)**, domestic dog#1 (domestic_dog%1:05:00::), Canis familiaris#1 (canis_familiaris%1:05:00::) (a member of the genus Canis (probably descended from the common wolf) that has been domesticated by man since prehistoric times; occurs in many breeds) *"the dog barked all night"*
  - ***direct hyponym*** / *full hyponym*
    - {01325095} <noun.animal>[05] S: (n) puppy#1 (puppy%1:05:00::) (a young dog)
    - {02087384} <noun.animal>[05] S: (n) pooch#1 (pooch%1:05:00::), doggie#1 (doggie%1:05:00::), doggy#1 (doggy%1:05:00::), barker#2 (barker%1:05:00::), bow-wow#2 (bow-wow%1:05:00::) (informal terms for dogs)
    - {02087513} <noun.animal>[05] S: (n) cur#1 (cur%1:05:00::), mongrel#2 (mongrel%1:05:00::), mutt#1 (mutt%1:05:00::) (an inferior dog or one of mixed breed)
    - {02087924} <noun.animal>[05] S: (n) lapdog#1 (lapdog%1:05:00::) (a dog small and tame enough to be held in the lap)
    - {02088026} <noun.animal>[05] S: (n) toy dog#1 (toy_dog%1:05:00::), toy#5 (toy%1:05:00::) (any of several breeds of very small dogs kept purely as pets)

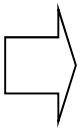
Lemmatization

- Reduce inflectional/variant forms to base form
- E.g.,
  - *am, are, is* $\rightarrow$ *be*
  - *car, cars, car's, cars'* $\rightarrow$ *car*
- *the boy's cars are different colors* $\rightarrow$ *the boy car be different color*
- Lemmatization implies doing “proper” reduction to dictionary headword form

Stemming

- Reduce terms to their “roots” before indexing
- “Stemming” suggests crude affix chopping
  - language dependent
  - e.g., ***automate(s), automatic, automation*** all reduced to ***automat***.

for example compressed and compression are both accepted as equivalent to compress.



for exampl compress and compress ar both accept as equival to compress

Stemming Example

Sample text: Such an analysis can reveal features that are not easily visible from the variations in the individual genes and can lead to a picture of expression that is more biologically transparent and accessible to interpretation

Lovins stemmer: such an analys can reve featur that ar not eas vis from th vari in th individu gen and can lead to a pictur of expres that is mor biolog transpar and acces to interpres

Porter stemmer: such an analysi can reveal featur that ar not easili visibl from the variat in the individu gene and can lead to a pictur of express that is more biolog transpar and access to interpret

Paice stemmer: such an analys can rev feat that are not easy vis from the vary in the individ gen and can lead to a pict of express that is mor biolog transp and access to interpret

Stemming

- *Many different algorithms*:
  - Porter's algorithm
    - Commonest algorithm for stemming English
      - Porter, Martin F. 1980. An algorithm for suffix stripping. *Program 14:130–137.*
      - http://www.tartarus.org/˜martin/PorterStemmer/
  - One-pass Lovins stemmer
    - Lovins, Julie Beth. 1968. Development of a stemming algorithm. *Translation and*
  - Lancaster
    - http://www.comp.lancs.ac.uk/computing/research/stemming/
    - Paice, Chris D. 1990. Another stemmer. *SIGIR Forum 24:56–61*
    - http://snowball.tartarus.org/demo.php
  - Snowball Stemmer
- Full morphological analysis (lemmatization)
  - At most modest benefits for retrieval
- Stemming increases recall while harming precision

Porter's algorithm

- Most common algorithm for stemming English
  - Results suggest it's at least as good as other stemming options
- Conventions + 5 phases of reductions
  - Phases applied sequentially
  - Each phase consists of a set of commands
  - Sample convention: *Of the rules in a compound command, select the one that applies to the longest suffix.*

Typical rules in Porter

- *sses* $\rightarrow$ *ss*
- *ies* $\rightarrow$ *i*
- *ational* $\rightarrow$ *ate*
- *tional* $\rightarrow$ *tion*

- Weight of word sensitive rules
- *(m>1) EMENT* →
  - *replacement* → *replac*
  - *cement* → *cement*

Language-Specificity

- The above methods embody transformations that are
  - Language-specific, and often
  - Application-specific
- These are “plug-in” addenda to the indexing process
- Both open source and commercial plug-ins are available for handling these

Does Stemming Help?

- English: very mixed results. Helps recall for some queries but harms precision on others
  - E.g., operative (dentistry) ⇒ oper
- Definitely useful for Spanish, German, Finnish, …
  - 30% performance gains for Finnish!

Credits

- Slides partly adapted from
  - Eva Zangerle , DBIS Innsbruck (2014/15)
  - Stefano Ceri, Alessandro Bozzon , Marco Brambilla,Emanuele Della Valle, Piero Fraternali, Silvia Quarteroni: Web Information Retrieval
  - Günther Specht, DBIS Innsbruck (former lectures)