Lecture 5:
Text Operations

Johan Bollen
Old Dominion University
Department of Computer Science

jbollen@cs.odu.edu
http://www.cs.odu.edu/~jbollen
Text operations

The traditional substrate of IR is text\textsuperscript{a}

1. Logical view

2. Requirement to translate user information need and document semantics to common representation

3. Concept of an index term: user information need (query) and document →

4. Selection of index terms: highest importance

Term weights

1. Some words are better than others

(a) Objective is to represent semantics of documents: what is it about?

(b) Some terms: very non-specific, others: very specifically tied to document

(c) This reasoning is represented by TDIDF weighing schemes

(d) Reduction of noise: irrelevant terms, unspecific terms

2. Given a collection of documents:

(a) Extraction or stipulation of index terms

(b) Text pre-processing

(c) Term pruning

\textsuperscript{a}This does not mean IR is inherently tied to text systems!
1. Description of content and semantics of document
2. Structure
3. Derivation of index terms:
   (a) removal of stopwords
   (b) stemming
   (c) identification of nouns
4. Pull vs. push
Index Terms

- **Assigned terms**
  1. Manual indexing
     (a) Professional indexer
     (b) Authors
  2. Classification efforts (Library of Congress Classification, ACM)
     (a) Coarseness: large collections?
     (b) Categories are mutually exclusive
  3. Subjectivity, inter-indexer consistency (:very low!)
  4. Ontologies and classifications:

- **Extracted terms**
  1. Extraction from title, abstract or full-text
  2. Methodology to extract must distinguish significant from non-significant terms
  3. Controlled (limit to pre-defined dictionary vs. non-controlled vocabulary (all terms as they are))
  4. Controlled → less noise, better performance?

limited appeal for retrieval
Document pre-processing

**Five main operations**

1. Lexical analysis: separation of text into words, treating digits, hyphens, punctuation marks and letter case
2. Elimination of stopwords: words with low discrimination values
3. Stemming
4. Selection of index terms
5. Construction of dictionary, thesaurus

All operations are optional except separation of text into words or terms. Most intend to automatically construct adequate list of index terms for collection of documents.
Lexical Analysis

1. Word separation:
   (a) adequate delimiters:
      i. spaces: deduplication
      ii. dashes, underscores?
      iii. commas
      iv. semi-colons
      v. All may be part of actual index terms
   (b) problems:
      i. index terms can contain more than 1 word
      ii. recognize frequent pairs and triples: “computer science”,

   "world wide web"
   iii. problem is more prominent in English
   iv. delimiters in other non-Western alphabets

(c) Lower-case, spurious punctuation

2. Demonstration: Kafka’s Metamorphosis
   (a) Note effects of punctuations, high number of irrelevant terms
   (b) mk_pairs.pl?
   (c) German version?
#!/usr/bin/perl

while (<>){
    $line = $_
    chomp ($line);

    if ($line =~ /[a-z]/){
        # split on spaces
        @words = split /\s+/, $line;

        # separately print all words in line
        foreach $term (@words){
            # lower case
            $term =~ tr /[A-Z]/[a-z]/;
            # remove punctuation
            $term =~ s/\./|\.|;|\:|"\'|\?|\!/g;

            print $term . 
        }
    }
}
Search pairs

#!/usr/bin/perl

while (<>){
    $w1 = $_;
    chomp ($w1);

    if ( $w1 && $w2 ){
        print $w2 . "−" . $w1 . "\n";
    }

    $w2=$w1;
}

Elimination of stopwords

1. Stopwords:
   (a) Too frequent, \( P(k_i) > 0.8 \)
   (b) Irrelevant in linguistic terms
      i. Do not correspond to nouns, concepts etc
      ii. Could however be vital in natural language processing

2. Removal from collection on two grounds:
   (a) Retrieval efficiency
      i. High frequency: no distinction between documents
      ii. Useless for retrieval
   (b) Storage efficiency
      i. Zipf’s law
      ii. Heap’s law
On the frequency of terms and powerlaws

- First consider Zipf’s law
  1. relation between word frequency and rank
  2. frequency of i’th most frequent word is $\frac{1}{i^\theta}$
  3. or frequency, $f$, and rank, $r$: $f = r^{-b}$
  4. Applies to many natural phenomena

- Zipf: implications for stopwords
  1. removal of stopwords will greatly affect size of dictionary: -40%
  2. Similar effect in libraries: 80/20 rule

Demo: Kafka metamorphosis, look at distribution of words
Heap’s Law

1. Relation between text size and dictionary size:
2. size of vocabulary, $V = O(n^\beta)$

3. $\beta < 1, \beta \in \mathbb{R}^+$
4. $\beta$ is usual 0.4-0.6
5. Shown to apply to WWW
Back to stopword removal

1. Two approaches:
   (a) Frequency: cut off at frequency threshold
   (b) Stopword lists
2. Simply apply filter: demo
3. Note stark reduction
Search pairs

#!/usr/bin/perl

# read list of stopwords in stoplist.txt
open (IN, "< stoplist.txt");
@words = <IN>;
close (IN);

foreach $word (@words){
  chomp ($word);
  $h{$word} = 1;
}

while (<>){
  $word = $_;
  chomp ($word);

  if (!$h{$word}){
    print $word . "\n";
  }
}
Example stoplist

a, about, above, according, across, actually, adj, after, afterwards, again, against, all, almost, alone, along, already, also, although, always, among,

amongst, an, and, another, any, anyhow, anyone, anything, anywhere, are, aren’t, around, as, at, b, be, became, because, become, becomes,
becoming, been, before, beforehand, begin, beginning, behind, being, below, beside, besides, between, beyond, billion, both, but, by, c, can, can’t,
cannot, caption, co, co., could, couldn’t, d, did, didn’t, do, does, doesn’t, don’t, down, during, e, each, eg, eight, eighty, either, else, elsewhere, end,
ending, enough, etc, even, ever, every, everyone, everything, everywhere, except, f, few, fifty, first, five, for, former, formerly, forty, found, , four,
from, further, g, h, had, has, hasn’t, have, haven’t, he, he’d, he’ll, he’s, ···, hence, her, here, here’s, hereafter, hereby, herein, hereupon, hers, herself,
him, himself, his, how, however, hundred, i, i’d, i’ll, i’m, i’ve, ie, if, in, inc., indeed, instead, into, is, isn’t, it, it’s, its, itself, j, k, l, last, later, latter,
latterly, least, less, let, let’s, like, likely, ltd, m, made, make, makes, many, maybe, me, meantime, meanwhile, might, million, miss, more, moreover,
most, mostly, mr, mrs, much, must, my, myself, n, namely, neither, never, nevertheless, next, nine, ninety, no, nobody, none, nonetheless, noone, nor,
not, nothing, now, nowhere, o, of, off, often, on, once, one, one’s, only, onto, or, other, others, otherwise, our, ours, ourselves, out, over, overall,
own, p, per, perhaps, q, r, rather, recent, recently, s, same, seem, seemed, seeming, seems, seven, seventy, several, she, she’d, she’ll, she’s, should,
shouldn’t, since, six, sixty, so, some, somehow, someone, something, sometime, sometimes, somewhere, still, stop, such, t, takin ···, up, upon, us,
used, using, v, very, via, w, was, wasn’t, we, we’d, we’ll, we’re, we’ve, well, were, weren’t, what, what’ll, what’s, what’ve, whatever, when, whence,
whenever, where, where’s, whereafter, whereas, whereby, who’ll, who’s, whoever, whole, whom, whomever, whose, why, will, with, within,
without, won’t, would, wouldn’t, x, y, yes, yet, you, you’d, you’ll, you’re, you’ve, your, yours, yourself, yourselves, z
Stemming

Another technique to bring us closer to list of “pure” index terms

   Issue:

1. Grammatical variations of essentially same word
2. inflexions, conjugations, plurals, etc
3. For example: \{house, housing, housed\}, or \{international, internationalization, internationalized, internationals\}
4. Search: root word, root concept
5. Intention to reduce of false-negative rates

 technique: stemming

1. Affix removal
2. Table lookup
3. successor variety
4. n-grams

Most often used: affix removal

1. Porter’s stemming algorithm is standard
2. Long set of rules to remove word-endings and replace with neutral ones
Porter’s stemming algorithm

**Suffix list**

1. set of rules to removes suffixes
2. Greedy matching
   (a) Tries to always match longest suffix
   (b) Avoid non-suffixes: distinguish between suffix and genuine part of word
3. Design to gradually strip increasing numbers of suffixes
4. Works relatively blindly: efficient

**Examples**

Plurals and past participles (longest suffix)
- sses → ss
- ies → i
- ss → ss
- s → Ø

More suffixes
- ational → ate
- tional → tion
- ousness → ous

etc...
Porter’s stemming algorithm

Demos on Kafka’s masterpiece
Index terms selection

Options

1. full text representation
   (a) All words in document are taken into account
   (b) Used by some search engines
   (c) Avoids problems with e.g. “to be or not to be”

2. Use of filtered set of index terms
   (a) Based on extraction parameters and characteristics
   (b) Ad hoc and partially manual
   (c) Accept that set of index terms will be noisy

3. Controlled dictionary
   (a) Nouns carry most semantics
   (b) Retain only nouns
   (c) Or better: noun groups, e.g. “computer science”
   (d) Use dictionary to validate set
Index terms selection: problems

1. Specific jargon
   (a) Words in specific domain may not be in dictionary
   (b) insensitive to key-phrases, such as “to be or not to be”
   (c) renders certain information impossible to find

2. Non-lexical information
   (a) Chemistry, math
   (b) Specific use of non-alphanumerics
   (c) certain nouns: meaningless, e.g. “result”, “theorem”, etc
   (d) notation problems across domains

3. Automated extraction is not failproof and can be quite cumbersome
Thesauri

**Characteristics**

1. Pre-compiled list of words in a domain
2. For each word, list of related words (synonyms)
3. Often accompanied by phrases, definitions, categorization

**Applications**

1. Normalization of vocabulary
2. Adoption of standardized vocabulary for specific domain
3. Assist users to find proper terms
4. Classification to broaden and narrow search fields
5. Provide controlled vocabulary for IR systems

**Limitations**

1. Absence of sufficient data for domain
2. Scale and dynamic nature of collection of documents
3. Applies to WWW
4. Automated approaches to thesaurus constructions
5. Semantic Web may provide solution

---

*aNote: Word-Association research*
Word association norms, Taxonomies and Ontologies

1. Efforts related to generation of thesauri
   (a) different approach
   (b) idea is to represent empirical or logical relations among terms
   (c) important in disambiguation and interpretation of user queries

2. Aspects of queries that can be addressed:
   (a) Terms related to other terms
      i. In population, not by oc-occurrence
      ii. Psychological applications
   (b) Taxonomical information
      i. Is a whale a fish?
      ii. Is car a vehicle?
   (c) Logical inference on relationships of concepts
      i. Is it true that all creatures with gills are fish?
      ii. Is it true that all students are lazy?
      iii. Is it true or false that professors work too much?

3. Difficult to establish: large-scale efforts
Word association norms

Douglas L. Nelson comments on the usefulness of these norms:

We assume that a dynamic associative structure is created in memory that involves representations of the words themselves as well as connections to other words, and we have reasons to believe that this lexical structure plays a critical role in any task involving familiar words. The role is complex because it differs for different goals and for different tasks but we presume its omnipresence is essential whenever and wherever meaning is sought.

1. Capture degree of word associations
2. Establish associations empirically
Word association norms

1. Measured in human subjects
   (a) Asked to respond to “cue” word
   (b) Produce list of 5 or 10 “target”-word
   (c) Frequency of target given cue can be calculated

2. Open and close
   (a) Open: not all targets are normed
   (b) Closed: attempt to norm all targets
      i. Major effort
      ii. With every new target: new cue
      iii. Attempt to gradually expand existing base
   (c) Nelson - USF - http://w3.usf.edu/FreeAssociation/
Ontologies

1. Ontologies and Knowledge representation
   (a) Vocabulary
   (b) Logical relations
   (c) formally represents our knowledge on given domain

2. WordNet prime example
   (a) Created to represent ontology of English language
   (b) Based on present psycho-linguistic theories

   (c) Available for download and on-line services

3. Problem:
   (a) Portable representations
   (b) Efficient generation
   (c) Validation

Demo:

http://www.cogsci.princeton.edu/~wn/
Example

Diagram:
- Company
  - Part-of
  - Deals-with...
  - employed-by
  - located-in
  - manages
  - deploys
- Person
  - name
- Product/Service
  - name
  - category, description
- Technology
  - used-in...
  - name
- Facility
  - name
  - function, city, region
  - manufactured-by...
Readings for next week

The University of South Florida Word Association, Rhyme and Word Fragment Norms
http://w3.usf.edu/FreeAssociation/Intro.html

James Hendler (2002)
Integrating applications on the World Wide web
http://www.w3.org/2002/07/swint