Text Features
Features

• Key to machine learning is having good features

• In industrial data mining, large effort devoted to constructing appropriate features
Cooper’s concordance of Wordsworth was published in 1911. The applications of full-text retrieval are legion: they include résumé scanning, litigation support and searching published journals on-line.

- Cooper’s vs. Cooper vs. Coopers.
- Full-text vs. full text vs. {full, text} vs. fulltext.
- résumé vs. resume.
Punctuation

• *Ne’er*: use language-specific, handcrafted “locale” to normalize.
• *State-of-the-art*: break up hyphenated sequence.
• *U.S.A. vs. USA* - use locale.
• *a.out*
Numbers

• 3/12/91
• Mar. 12, 1991
• 55 B.C.
• B-52
• 100.2.86.144
  – Generally, don’t index as text
  – Creation dates for docs
Case folding

• Reduce all letters to lower case
• Exception: upper case in mid-sentence
  – *e.g.*, *General Motors*
  – *Fed vs. fed*
  – *SAIL vs. sail*
Thesauri and Soundex

• Handle synonyms and homonyms
  – Hand-constructed equivalence classes
    • e.g., car = automobile
    • your ≠ you’re

• Index such equivalences?
• Or expand query?

slide from Raghavan, Schütze, Larson
Spell Correction

• Look for all words within (say) edit distance 3 (Insert/Delete/Replace) at query time
  – *e.g.*, *Alanis Morisette*

• Spell correction is expensive and slows the query (up to a factor of 100)
  – Invoke only when index returns zero matches?
  – What if docs contain mis-spellings?
Lemmatization

• Reduce inflectional/variant forms to base form

- *am, are, is* → *be*
- *car, cars, car's, cars'* → *car*

the boy's cars are different colors

→

the boy car be different color
Stemming

• Reduce terms to their “roots” before indexing
  – 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 compres and compres are both accept as equivel to compres.

slide from Raghavan, Schütze, Larson
Porter’s algorithm

• Common algorithm for stemming English
• 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.*
• Porter’s stemmer available:
  [http://www.sims.berkeley.edu/~hearst/irbook/porter.html](http://www.sims.berkeley.edu/~hearst/irbook/porter.html)

slide from Raghavan, Schütze, Larson
Typical rules in Porter

• *sses* → *ss*
• *ational* → *ate*
• *tional* → *tion*
Challenges

• Sandy
• Sanded
• Sander

→ Sand ??
Properties of Text

• Word frequencies - skewed distribution
• `The’ and `of’ account for 10% of all words
• Six most common words account for 40%

Zipf’s Law:
Rank * probability = c
Eg, c = 0.1

From [Croft, Metzler & Strohman 2010]
Associate Press Corpus `AP89’

Total documents: 84,678
Total word occurrences: 39,749,179
Vocabulary size: 198,763
Words occurring > 1000 times: 4,169
Words occurring once: 70,064

From [Croft, Metzler & Strohman 2010]
Middle Ground

• Very common words $\rightarrow$ bad features

• Language-based stop list:
  words that bear little meaning
  20-500 words
  http://www.dcs.gla.ac.uk/idom/ir_resources/linguistic_utils/stop_words

• Subject-dependent stop lists

• Very rare words *also* bad features
  Drop words appearing less than k times / corpus
Beyond Words

- Look at capitalization (may indicated a proper noun)

- Look for commonly occurring sequences
  - E.g. New York, New York City
  - Limit to 2-3 consecutive words
  - Keep all that meet minimum threshold (e.g. occur at least 5 or 10 times in corpus)