COM6115: Text Processing

Information Retrieval: Term Manipulation

Mark Hepple

Department of Computer Science
University of Sheffield

Overview

- Definition of the information retrieval problem
- Approaches to document indexing
  - ◇ manual approaches
  - ◇ automatic approaches
- Automated retrieval models
  - ◇ boolean model
  - ◇ ranked retrieval methods (e.g. vector space model)
- Term manipulation:
  - ◇ stemming, stopwords, term weighting
- Web Search Ranking
- Evaluation

What counts as a term?

Common to just use the **words**, but pre-process them for generalisation

- **Tokenisation**: split words from punctuation (get rid of punctuation)
  e.g. word-based. → word based   three issues: → three issues
- **Capitalisation**: normalise all words to lower (or upper) case
  e.g. Cat and cat should be seen as the same term, but should we conflate Turkey and turkey?
- **Lemmatisation**: conflate different inflected forms of a word to their basic form (singular, present tense, 1st person):
  e.g. cats, cat → cat   have, has, had → have   worried, worries → worry

What counts as a term? (ctd)

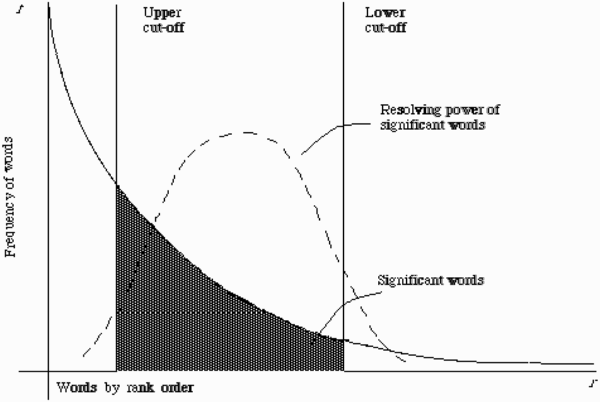
- **Stemming**: conflate morphological variants by chopping their affix:

CONNECT	WORRY	GALL
CONNECTED	WORRIED	GALLING
CONNECTING	WORRIES	GALLED
CONNECTION	WORRYING	GALLEY
CONNECTIONS	WORRYINGLY	GALLERY

- **Normalisation**: heuristics to conflate variants due to spelling, hyphenation, spaces, etc.

  e.g. USA and U.S.A. and U S A → USA
  e.g. chequebook and cheque book → cheque book
  e.g. word-sense and word sense → word-sense

Word Frequency and Term Usefulness



- The most and least frequent terms are not the most useful for retrieval
  - ◇ (Figure from van Rijsbergen (1979) *Information Retrieval* `http://www.dcs.gla.ac.uk/Keith/Preface.html`)

Stop words

- Use **Stop list** removal to exclude “non-content” words
- Usually most frequent (and least useful for retrieval)

a	always	both
about	am	being
above	among	co
across	amongst	could

  - ◇ greatly reduces the size of the inverted index
  - ◇ but what if we want to search for *phrases* that include these terms?
    - Kings of Leon
    - Let it be
    - To be or not to be
    - Flights to London

Single vs. Multi-word Terms

- To aid recognition of phrases, might allow *multi-word terms*

  e.g. Sheffield University

- Possible approach — allow *multi-word indexing*

  e.g. bigram indexing: store each bigram as a term in index

  For pease porridge in the pot get:

  pease porridge
  porridge in
  in the
  the pot

  ◇ Problem: number of bigrams is v.large c.f. number of words

    - leads to a huge increase in size of the index

- Alternative: identify multi-word phrases during retrieval

  ◇ Positional indexes, storing position terms in documents, can help

    - use to compute if occurrences of search terms in document are adjacent / close / far apart

Single vs. Multi-word Terms (ctd)

- Positional indexes:

Doc	*Text*
1	Pease porridge hot, pease porridge cold
2	Pease porridge in the pot
3	Nine days old
4	Some like it hot, some like it cold
5	Some like it in the pot
6	Nine days old

⟹

Num	*Token*	*Docs*
1	cold	1:(6), 4:(8)
2	days	3:(2), 6:(2)
3	hot	1:(3), 4:(4)
4	in	2:(3), 5:(4)
5	it	4:(3, 7), 5:(3)
6	like	4:(2, 6), 5:(2)
7	nine	3:(1), 6:(1)
8	old	3:(3), 6:(3)
9	pease	1:(1, 4), 2:(1)
10	porridge	1:(2, 5), 2:(2)
11	pot	2:(5), 5:(6)
12	some	4:(1, 5), 5:(1)
13	the	2:(4), 5:(5)