COM6115: Text Processing

Information Retrieval: Term Weighting

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

Term Weighting

What values do we assign for terms in document (and query) vectors?

- binary weights - 0/1: whether or not term is present in document
  - ◇ But documents with multiple occurrences of query keyword may be more relevant
- Frequency of term in document: like the examples we have seen
  - ◇ But what if the term is also frequent in collection?
  - ◇ Common terms: not very useful for discriminating relevant documents
- Frequency in document vs in collection: weight terms highly if
  - ◇ They are **frequent** in relevant documents . . . *but*
  - ◇ They are **infrequent** in collection as a whole

Term Weighting (ctd)

- Key concepts:

document collection	D	collection (set) of documents
size of collection	$\|D\|$	total number of documents in collection
term freq	$tf_{w,d}$	number of times w occurs in document d
collection freq	cf_w	number of times w occurs in collection
document freq	df_w	number of documents containing w

Term Weighting (ctd)

The informativeness of terms

- Idea that *less common* terms are *more useful* to finding relevant docs:

  i.e. these terms are more *informative*

- Is this idea best addressed using *document frequency* or *collection frequency*?

- Consider following counts (from New York Times data, $|D| = 10000$):

Word	cf_w	df_w
insurance	10440	3997
try	10422	8760

  ◇ term *insurance* semantically focussed, term *try* very general

  - document frequency reflects this difference
  - collection frequency fails to distinguish them (i.e. very similar counts)

Term Weighting (ctd)

- Informativeness is inversely related to (document) frequency

  i.e. *less common* terms are *more useful* to finding relevant documents
  more common terms are *less useful* to finding relevant documents

- Compute metric such as: $\frac{|D|}{df_w}$

  ◇ Value reduces as df_w gets larger, tending to 1 as df_w approaches $|D|$

  e.g. $\frac{10000}{3997} = 2.5$ (insurance) $\frac{10000}{8760} = 1.14$ (try)

  ◇ Value very large for small df_w — over-weights such cases

  e.g. $\frac{10000}{350} = 28.6$ (mischief)

- To moderate this, take *log*: **Inverse document frequency** (idf)

$$idf_{w,D} = log\frac{|D|}{df_w}$$

$log\frac{10000}{3997} = 0.398$ (insurance) $log\frac{10000}{8760} = 0.057$ (try) $log\frac{10000}{350} = 1.456$ (mischief)

Term Weighting (ctd)

- **BUT** Not all terms describe a document equally well
- Putting it all together: **tf.idf**
  - Terms which are frequent in a document are better:

$$tf_{w,d} = freq_{w,d}$$

  - Terms that are rare in the document collection are better:

$$idf_{w,D} = log \frac{|D|}{df_w}$$

  - Combine the two to give **tf.idf** term weighting:

$$tf.idf_{w,d,D} = tf_{w,d} \cdot idf_{w,D}$$

- Most commonly used method for term weighting.
  - Used in other fields too (e.g. summarisation)

Term Weighting (ctd)

tf.idf example:

Term	tf	df	$\|D\|$	idf	$tf.idf$
the	312	28,799	30,000	0.018	5.54
in	179	26,452	30,000	0.055	9.78
general	136	179	30,000	2.224	302.50
fact	131	231	30,000	2.114	276.87
explosives	63	98	30,000	2.486	156.61
nations	45	142	30,000	2.325	104.62
haven	37	227	30,000	2.121	78.48

For term the:

$$idf(the) = \log_{10}(\frac{30,000}{28,799}) = 0.018$$

$$tf.idf(the) = 312 \cdot 0.018 = 5.54$$

Putting things together

Example: Vector Space Model, tf.idf term weighting, cosine similarity

- tf.idf values for words in two documents D_1 and D_2, and in a query Q "hunter gatherer Scandinavia":

	Q	D_1	D_2
hunter	19.2	56.4	112.2
gatherer	34.5	122.4	0
Scandinavia	13.9	0	30.9
30,000	0	457.2	0
years	0	12.4	0
BC	0	200.2	0
prehistoric	0	45.3	0
deer	0	0	23.6
rifle	0	0	452.2
Mesolithic	0	344.2	0
$\sqrt{\sum_{i=1}^{n} x_i^2}$	41.9	622.9	467.5

(i.e. length of vector)

Putting things together (ctd)

- $$sim(\vec{q},\vec{d}) = cos(\vec{q},\vec{d}) = \frac{\sum_{i=1}^{n} q_i d_i}{\sqrt{\sum_{i=1}^{n} q_i^2}\sqrt{\sum_{i=1}^{n} d_i^2}}$$

$$\begin{aligned}
cos(Q, D_1) &= \frac{(19.2 * 56.4) + (34.5 * 122.4) + \cdots + (0 * 0) + (0 * 344.2)}{41.9 * 622.9} \\
&= \frac{5305.68}{26071.72} \\
&= 0.20
\end{aligned}$$

$$\begin{aligned}
cos(Q, D_2) &= \frac{(19.2 * 112.2) + (34.5 * 0) + \cdots + (0.0 * 452.2) + (0.0 * 0.0)}{41.9 * 467.5} \\
&= \frac{2583.8}{19570.0} \\
&= 0.13
\end{aligned}$$

- so document D_1 is more similar to Q than D_2