Introduction to Information Retrieval

Colm O’Riordan

Information Technology,
NUI Galway.

Outline

Vector Space Model - recap

Attempts to improve upon the Boolean model by removing the limitation of binary weights for index terms.
Terms can have a non-binary value both in queries and documents.
Hence we can represent documents and query as n-dimensional vectors.

$$\vec{d_j} = (w_{1,j}, w_{2,j}, \ldots w_{n,j})$$

$$\vec{q} = (w_{1,q}, w_{2,q} \ldots w_{n,q})$$

We can calculate the similarity between a document and a query by calculating the similarity between the vector representations.

We can measure this similarity by measuring the cosine of the angle between the two vectors.
Inner product:

$$a \bullet b = |a||b|cos(a, b)$$

$$\Rightarrow cos(a, b) = \frac{a \bullet b}{|a||b|}$$

We can therefore calculate similarity between document and query as:

$$sim(\vec{d_j}, \vec{q}) = \frac{d_j \bullet q}{|d_j||q|}$$

$$\Rightarrow \mathrm{sim}(\vec{d_j}, \vec{q}) = \frac{\sum_{i=1}^{n} w_{i,j} \times w_{i,q}}{\sqrt{\sum_{i=1}^{n} w_{i,j}^2} \times \sqrt{\sum_{i=1}^{n} w_{i,q}^2}}$$

Vector Space - *tf-idf* weighting

- We need means to calculate the term weights in the document and query vector representations.
- A term's frequency within a document quantifies how well a term describes a document. The more frequent a term occurs in a document, the better it is at describing that document and vice-versa.
- This frequency is known as the term frequency or *tf factor*.

idf

- Also, if a term occurs frequently across all the documents that term does little to distinguish one document from another.
- This factor is known as the inverse document frequency (idf-frequency).
- The most commonly used weighting schemes are known as *tf-idf* weighting schemes.

A tf-idf weighting scheme

For all terms in a document, the can be weight assigned is calculated by:

$$w_{i,j} = f_{i,j} \times log \frac{N}{n_i}$$

where

- $f_{i,j}$ is the normalised frequency of term t_i in document d_j
- N is the number of documents in the collection
- n_i is the number of documents that contain term t_i

A similar weighting scheme can be used for queries. The main difference is that the tf and idf are given less credence and all terms have an initial value of 0.5 which is increased or decreased based on tf-idf across the document collection. Weighting schemes due to Salton (1983).

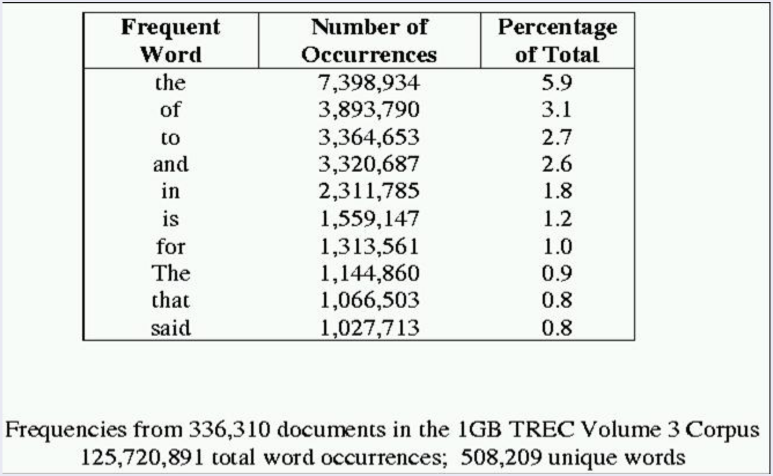
Text Properties

- Not all words are equally important for capturing meaning of a document
- Text documents comprise symbols from a finite alphabet

- What is the distribution of the frequency of different words?
- How fast does vocabulary size grow with the size of a document collection?
- Such factors affect the performance of information retrieval
- Can be used to select appropriate term weights and other aspects of an IR system.

Word frequencies

- A few words are very common. e.g. the two most frequent words (e.g. “the", “of") can account for about 10% of word occurrences.
- Most words are very rare. Half the words in a corpus appear only once i.e. a “heavy tailed"/Zipfian distribution



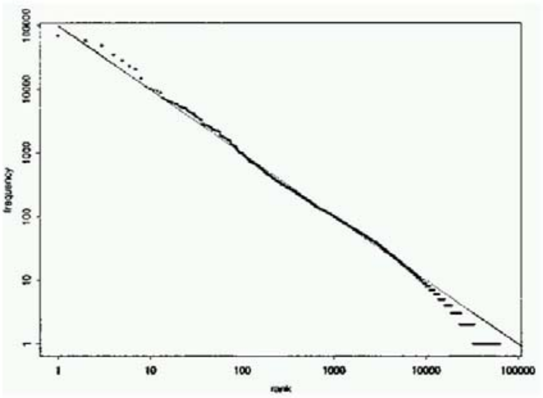
Frequent Word	Number of Occurrences	Percentage of Total
the	7,398,934	5.9
of	3,893,790	3.1
to	3,364,653	2.7
and	3,320,687	2.6
in	2,311,785	1.8
is	1,559,147	1.2
for	1,313,561	1.0
The	1,144,860	0.9
that	1,066,503	0.8
said	1,027,713	0.8

Frequencies from 336,310 documents in the 1GB TREC Volume 3 Corpus
125,720,891 total word occurrences; 508,209 unique words

Zipf

- Gives an approximate model for the distribution of different words in a document
- Rank(r): The numerical position of a word in a list sorted by decreasing frequency (f).
- Zipf's law: $f \times r =$ constant
- Represents a power law; straight line on a log-log plot

- Accurate model except for extremes
- Model on the Brown corpus



Luhn - resolving power

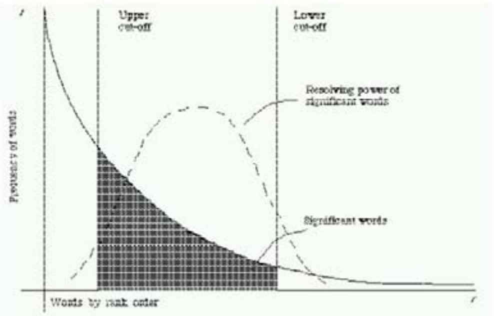


Figure 2.1. A plot of the hyperbolic curve relating f, the frequency of occurrence and r, the rank order (Adapted from Schultz[44] page 120)

Vocabularly Growth

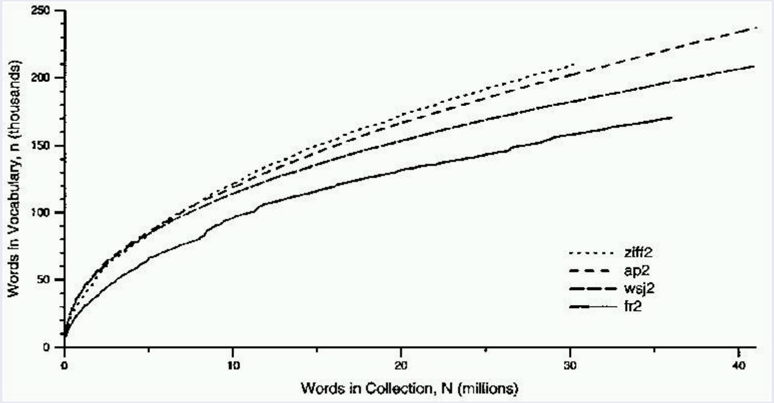
- How does the size of the vocabularly increase with the size of the document collections?
- Has impact on choice of indexing strategies and algorithms
- Note: the size of a vocabularly is not really bounded in real world cases due to the existence of mis-spellings, proper names etc. and document identifiers

Vocabularly Growth

- If V is the size of the vocabulary and n is the length of the document collection in word occurrences:

$$V = K.n^{\beta}, 0 < \beta < 1$$

- Standard values:
  - $K \approx 10 - 100$
  - $\beta \approx 0.4 - 0.6$


Words in Vocabulary, n (thousands)
0
50
100
150
200
250
Words in Collection, N (millions)
0
10
20
30
40
ziff2
ap2
wsj2
fr2

Weighting schemes

- Quality of performance of information retrieval system depends on the quality of the weighting scheme
- Want to assign high weights to those terms with a high resolving power.
- tf*idf is one such approach where weight is increased for frequently occurring terms but decreased again for those that are frequent across the collection

- ‘bag of words’ model usually adopted
- Term independence assumption usually adopted

This means that for ..

- Doc1 = *Mary is quicker than John*
- Doc2 = *John is quicker than Mary*
- Doc1 and Doc2 are viewed as equivalent

Some variants

- It is unlikely that 30 occurrences of a term in a document truly carries thirty times the signifiance of a single occurrence of that term.
- A common modification is to use the logarithm of the term frequency
- $w_{i,d} = 1 + log(tf_{i,d})$, if $tf_{i,d} > 0$
- $w_{i,d} = 0$, otherwise

Maximum term normalisation

- Usually of the form:

$$ntf = a + (1 - a)\frac{tf_{i,d}}{tfmax(d)}$$

- *a* is a smoothing factor which can be used to dampen the impact of the second term

Problems with maximum term normalisation

- Stopword removal may have effects on distribution of terms; this normalisation is unstable and may require tuning per collection
- Possibility of outliers with unusually high frequency
- Those documents with a more even distribution of term frequencies should be treated differently to those with a skewed distribution

- More sophisticated forms of normalisation exist - will look at some of them in later classes

Normalisation because ..

- We observe higher frequencies in longer documents merely because longer documents tend to repeat same words more frequently
- Consider a document $d^{`}$ created by concatenating a document d to itself.
- $d^{`}$ is no more relevant to any query than document $d$, but yet according to vector space type similarity, $\text{sim}(d^{`}, q) \geq \text{sim}(d, q)\ \forall q$

Structure of modern weighting schemes

- Many, if not all, of the developed or learned weighting schemes can be epresented in the following format:

$$sim(q,d) = \sum_{t \in q \cap d} (ntf(D) \times gw_t(C) \times qw_t(Q))$$

- where
  - $ntf(D)$ is the normalised term frequency in a document
  - $gw_t(C)$ is the global weight of a term across a collection
  - $qw_t(Q)$ is the query weight of a term in Q, a query

BM25/Okapi

$$BM25(Q,D) = \sum_{t \in Q \cap D} \left(\frac{tf_t^D \cdot log(\frac{N - df_t + 0.5}{df_t + 0.5}) \cdot tf_t^Q}{tf_t^D + k_1 \cdot ((1-b) + b \cdot \frac{dl}{dl_{avg}})} \right) \quad (1)$$

- Standard benchmark
- Relatively good performance
- Needs to be tuned for collection

Pivoted Normalisation

$$piv(Q,D) = \sum_{t \in Q \cap D} \left(\frac{1 + log(1 + log(tf_t^D))}{(1-s) + s.\frac{dl}{dl_{avg}}} \right) \times log\left(\frac{N+1}{df_t}\right) \times tf_t^Q \tag{2}$$

- Standard benchmark
- Needs to be tuned for collection
- Issues with normalisation

Axiomatic approaches

Constraint 1
Adding a query term to a document must always increase the score of a document

Constraint 2
Adding a non-query term to a document must always decrease the score of a document

Constraint 3
Adding successive occurrences of a term to a document must increase the score of a document less with successive occurrences. Essentially any term-frequency factor should be sub-linear.

Axiomatic approach

Constraint 4

- Using a vector length should be a better normalisation factor for retrieval. However, using the vector length will violate one of the existing constraints.
- Thus, ensuring that the document length factor is used in a sub-linear function will ensure that repeated appearances of non-query terms are weighted less.

- New weighting schemes which adhere to all these constraints outperform best known benchmarks

Summary

- Many text properties that can inform the development of weighting schemes
- Many ideas in the literature regarding normalisation, term weighting etc.
- Most modern weighting schemes hve similar structure. Many require tuning
- Recent axiomatic approach gives a means to guide the development of weighting schemes