Boolean Retrieval

- Views each document as a set of words
- Boolean Queries use AND, OR and NOT to join query terms
  - Simple SQL-like queries
  - Sometimes with weights attached to each component
- It is like exact match: document matches condition or not
  - Perhaps the simplest model to build an IR system
- Many current search systems are still using Boolean
  - Professional searchers who want to under control of the search process
  - e.g. doctors and lawyers write very long and complex queries with Boolean operators

Summary: Boolean Retrieval

- Advantages:
  - Users are under control of the search results
  - The system is nearly transparent to the user
- Disadvantages:
  - Only give inclusion or exclusion of docs, not rankings
  - Users would need to spend more effort in manually examining the returned sets; sometimes it is very labor intensive
  - No fuzziness allowed so the user must be very precise and good at writing their queries
    - However, in many cases users start a search because they don’t know the answer (document)

BM25

The *(Magical)* Okapi BM25 Model

- BM25 is one of the most successful retrieval models
- It is a special case of the Okapi models
  - Its full name is Okapi BM25
- It considers the length of documents and uses it to normalize the term frequency
- It is virtually a probabilistic ranking algorithm though it looks very ad-hoc
- It is intended to behave similarly to a two-Poisson model
- We will talk about Okapi in general

What is Behind Okapi?

- [Robertson and Walker 94]
- A two-Poisson document-likelihood Language model
  - Models within-document term frequencies by means of a mixture of two Poisson distributions
- Hypothesize that occurrences of a term in a document have a random or stochastic element
  - It reflects a real but hidden distinction between those documents which are “about” the concept represented by the term and those which are not.
- Documents which are “about” this concept are described as “elite” for the term.
- Relevance to a query is related to eliteness rather than directly to term frequency, which is assumed to depend only on eliteness.

Two-Poisson Model

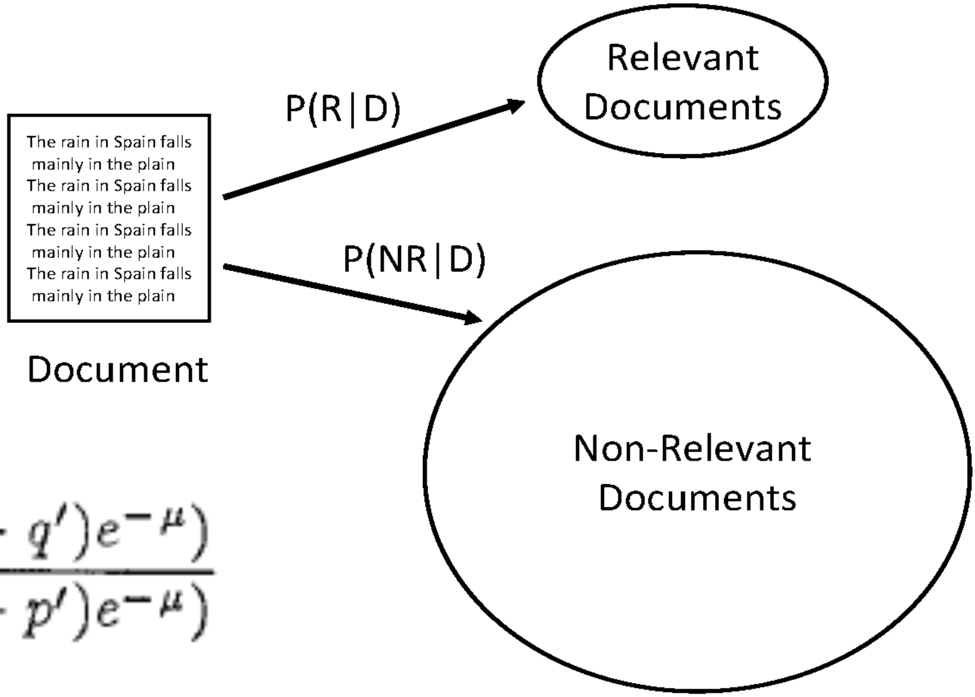

Figure adapted from “Search Engines: Information Retrieval in Practice” Chap 7

- Term weight for a term t:

$$w = \log \frac{\left(p'\lambda^{tf}e^{-\lambda} + (1-p')\mu^{tf}e^{-\mu}\right)\left(q'e^{-\lambda} + (1-q')e^{-\mu}\right)}{\left(q'\lambda^{tf}e^{-\lambda} + (1-q')\mu^{tf}e^{-\mu}\right)\left(p'e^{-\lambda} + (1-p')e^{-\mu}\right)}$$

where lambda and mu are the Poisson means for tf
In the elite and non-elite sets for t

p’ = P(document elite for t| R)
q’ = P(document elite for t| NR)

Characteristics of Two-Poisson Model

- It is zero for tf=0;
- It increases monotonically with tf;
- but to an asymptotic maximum;
- The maximum approximates to the Robertson/Sparck-Jones weight that would be given to a direct indicator of eliteness.

$$w = \log \frac{p(1-q)}{q(1-p)},$$

p = P(term present| R)
q = P(term present| NR)

Constructing a Function

- Constructing a function
  - Such that tf/(constant + tf) increases from 0 to an asymptotic maximum
- A rough estimation of 2-poisson

$$w = \boxed{\frac{tf}{(k_1 + tf)}}\, w^{(1)}$$

Approximated term weight

Robertson/Sparck-Jones weight;
Becomes the idf component of Okapi

constant

tf component of Okapi

Okapi Model

- The complete version of Okapi BMxx models

$$\sum_{t \in Q} \left(\log \frac{\mathrm{N} - \mathrm{df}_t + 0.5}{\mathrm{df}_t + 0.5} \right) \frac{(k_1 + 1)\, tf_t}{k_1 \left((1-b) + b \dfrac{\mathrm{doclen}}{\mathrm{avg\_doclen}} \right) + tf_t} \frac{(k_3 + 1)\, qtf_t}{k_3 + qtf_t}$$

idf (Robertson-Sparck Jones weight)

tf

user related weight

Original Okapi: k1 = 2, b=0.75, k3 = 0

BM25: k1 = 1.2, b=0.75, k3 = a number from 0 to 1000

Exercise: Okapi BM25

- Query with two terms, “president lincoln”, (*qtf = 1)*
- No relevance information (*r and R are* zero)
- N = 500,000 documents
- *“president”* occurs in 40,000 documents (df_1 = 40, 000)
- *“lincoln”* occurs in 300 documents (df_2 = 300)
- “president” occurs 15 times in the doc (tf_1 = 15)
- *“lincoln”* occurs 25 times in the doc (tf_2 = 25)
- document length is 90% of the average length ($dl/avdl$ = .9)
- $k_1 = 1.2$, $b = 0.75$, and $k_3 = 100$
- $K = 1.2 \cdot (0.25 + 0.75 \cdot 0.9) = 1.11$

Example from textbook “Search Engines: Information Retrieval in Practice” Chap 7

Answer: Okapi BM25

$$
\begin{aligned}
BM25(Q,D) \quad = \quad & \\
& \log \frac{(0+0.5)/(0-0+0.5)}{(40000-0+0.5)/(500000-40000-0+0+0.5)} \\
& \times \frac{(1.2+1)15}{1.11+15} \times \frac{(100+1)1}{100+1} \\
& + \log \frac{(0+0.5)/(0-0+0.5)}{(300-0+0.5)/(500000-300-0+0+0.5)} \\
& \times \frac{(1.2+1)25}{1.11+25} \times \frac{(100+1)1}{100+1} \\
\\
= \quad & \log 460000.5/40000.5 \cdot 33/16.11 \cdot 101/101 \\
& + \log 499700.5/300.5 \cdot 55/26.11 \cdot 101/101 \\
= \quad & 2.44 \cdot 2.05 \cdot 1 + 7.42 \cdot 2.11 \cdot 1 \\
= \quad & 5.00 + 15.66 = 20.66
\end{aligned}
$$

Example from textbook “Search Engines: Information Retrieval in Practice” Chap 7

Effect of term frequencies in BM25

Frequency of "president"	Frequency of "lincoln"	BM25 score
15	25	20.66
15	1	12.74
15	0	5.00
1	25	18.2
0	25	15.66