Dublin City University

School of Computing

CA4009: Search Technologies

Section 3b: Text Retrieval - BM25

See also the worked example on ranked retrieval.

Gareth Jones

October 2016

Probabilistic Model

Although the vector-space model employs frequency counts, its underlying mathematics as a retrieval model are fairly ad hoc.

- The matching score assigned to a document in a result set is not a probability of relevance, but rather a measure that attempts to estimate how much evidence there is in favour of a document being relevant

The probabilistic model of retrieval is an attempt to formalize the idea behind ranked retrieval in terms of probability theory.

It attempts *to compute the probability that a document is relevant to a query*, given that it possesses certain attributes or features, i.e. that it contains certain search terms.

Probabilistic Model

It is generally assumed that:

- each document is either relevant or irrelevant to the query.
- judging one document to be relevant or irrelevant tells us nothing about the relevance of another document.

The *Probability Ranking Principle* states that ranking documents by decreasing order of probability of relevance to a query will yield “optimal performance”, i.e. the best ordering based on the available data.

Probabilistic Model

The probabilistic model query-document matching score $ms(j)$ for a document j can be calculated in a similar manner to the other ranked retrieval schemes that we have already seen.

$$ms(j) = \sum_{i=0}^{I-1} w(i, j)$$

where $w(i, j)$ indicates a probabilistic term weighting scheme, and I is the set of all search terms.

This can be shown as part of the derivation of the probabilistic model to be a valid means of calculating the probability of the relevance of a document to a particular query.

Probabilistic Model

Derivation of a probabilistic model is beyond the scope of this module, but we can still examine and use a derived probabilistic model.

One effective probabilistic model is the *Okapi* BM25 *combined weighting* $cw(i,j)$ scheme developed at City University, London:

$$cw(i,j) = cfw(i) \times \frac{tf(i,j) \times (k_1 + 1)}{k_1 \times ((1-b) + (b \times ndl(j))) + tf(i,j)}$$

where $cw(i,j)$ indicates the combined weighting scheme, $ndl(j)$ is the normalised length of document j, and k_1 and b are experimentally determined constants that control the effect of $tf(i,j)$ and the degree of length normalisation respectively.

Okapi BM25

Let us look at the components of the Okapi BM25 term weighting function in a little detail.

First notice that it is composed of a $tf \times idf$ type function, since $cfw(i)$ is the same as $idf(i)$ and the other part of the function is a function $tf(i, j)$.

The k_1 factor determines the impact of term frequency in a document. A typical value for k_1 would be 1.5.

b determines the degree of document length normalisation. The value of b can vary in the range 0 to 1.

Okapi BM25

Let us consider the two exteme cases of $b = 0$ and $b = 1$.

$b = 0$

$$cw(i,j) = cfw(i) \times \frac{tf(i,j) \times (k_1 + 1)}{k_1 + tf(i,j)}$$

Here there is no adjustment of the term weight to take account of document length.

Londer documents will tend to have higher term weights, since on average they will have higher $tf(i,j)$ values.

Okapi BM25

$b = 1$

$$cw(i,j) = cfw(i) \times \frac{tf(i,j) \times (k_1 + 1)}{k_1 \times ndl(j) + tf(i,j)}$$

Here there is maximu application of adjustment of term weights to account for document length.

Shorter documents will have $ndl(j)$ values less than 1.0 - leading to an increase in the $cw(i,j)$ value compared to a document of average length.

Longer documents will have $ndl(j)$ values greater than 1.0 - leading to an decrease in the $cw(i,j)$ value compared to a document of average length.

Okapi BM25

The overall effect should be to provide a degree of compensation to document matching scores $ms(j)$ based on document length, i.e there should be less tendency to score longer documents higher, and so less tendency to place them nearer the top of a ranked retrieval list.

Note the b value is typically set between 0.0 and 1.0 to give optimal retrieval effectiveness foc a document collection using a set of training queries and corresponding relevance assessments.

Okapi BM25

Returning to the cae of $b = 0.0$.

$$cw(i,j) = cfw(i) \times \frac{tf(i,j) \times (k_1 + 1)}{k_1 + tf(i,j)}$$

We can see that the function increases more slowly with respect to $tf(i,j)$ for large values of k_1 and more quickly for smaller values of smaller values of k_1.

When $tf(i,j) = 1$, $cw(i,j)$ is simply equal to $cfw(i)$, i.e. there is no impact of the $tf(i,j)$ function here when a term occurs in a document only once.