BM25- a variant of tf-idf

> Bm25 is an algorithm used to evaluate the correlation between search terms and documents. It is an algorithm based on the probability retrieval model.

Instruction: we have a query and a batch of documents Ds. Now we need to calculate the correlation score between the query and each document D. Our approach is to first split the query to get the word q, and the score of the word consists of three parts:

- Correlation between words and D
- Correlation between words and query
- Weight of each word

Finally, we calculate the sum each word scores, and get the score between the query and the document.

formula

For query Q and document d, we have BM25 d of Q: $BM25_{score}(Q,d) = \sum_{t \in Q} w(t,d)$

$w(t,d) = \frac{(k_2+1)qf_i}{k_2+qf_i} \times \frac{(k_1+1)\times f_i}{f_i+k_1(1-b+b\times l_d/avg\_l)} \times log_2 \frac{(r_i+0.5)/(R-r_i+0.5)}{(n_i-r_i+0.5)/(N-n_i-R+r_i+0.5)}$

- r_i is the # of relevant documents containing term i
- n_i is the # of docs containing term i
- N is the total # of docs in the collection
- R is the number of relevant documents for this query (set to 0 if no relevancy info is known)
- f_i is the frequency of term i in the doc under consideration
- qf_i is the frequency of term i in the query
- k_1 determines how the tf component of the term weight changes as f_i increases. (if 0, then tf component is ignored.)
- k_2 has a similar role for the query term weights. Typical values make the equation less sensitive to k2 than k1 because query term frequencies are much lower and less variable than doc term frequencies.
- K ($k_1(1-b+b\times l_d/avg\_l)$) is more complicated. Its role is basically to normalize the tf component by document length.
- b regulates the impact of length normalization. (0 means none; 1 is full normalization.)