Scoring Function Document

Okapi TF:

It is a vector space model; in this we represent the document and query terms by a vector of term weights. So, the document which having the higher occurrence of query terms, the rank for that document is higher than other documents for the same query. Here, the score is calculated using the cosine similarity between the query and document. The formulae for the cosine similarity is as given below:

$$\cos\,(\text{document}, \text{query}) = \frac{\vec{d} \cdot \vec{q}}{|\vec{d}||\vec{q}|}$$

The Numerator of the above formulae is calculated using the dot product and the denominator is basically the product of the vector magnitude.

Expectation:

The algorithm performs good for the short nature of queries because the algorithm assumes that terms are statistically independent. The Ranking of the document for the particular query is totally dependent on the appropriateness of the query i.e the query should be meaningful to the document in order to give the good rank to the relevant documents. So basically higher the number of terms in the query lowers the similarity score because if the frequency of the term is more in the document then it will going to increase the magnitude of the query vector(denominator part of the above formulae) which in turn decrease the score of the relevant document and increase the score of the irrelevant document.

TF-IDF:

Basically it stands for “Term Frequency, Inverse Document “. The IDF stands for Inverse Document Frequency, which can be calculated as log of total number of documents in the corpus divided by the total number of documents in which query word presents. The inverse document frequency will improve the efficiency of search because term is weighted relative to the entire corpus. Suppose the term of the query comes in many documents, then those documents are not useful. But if the term comes in less documents, then those documents are important and probably the one which need to be ranked high.

Expectation:

After analyzing the formulae of the TF-IDF we can expect the following things:

1) If the frequency of the terms in a document is higher then the score for that document will be the highest for that query.
2) If the frequency of the terms is good in one or more document, then it score will be less because that terms are less unique.

BM-25:

It is a probabilistic Scoring Model. Probabilistic scoring model means it estimates the probability document d relevant to the query q. BM-25 gives us the relevant feedback which gives us a good approximation of a document score. This model sum up all documents and query terms weights using the equation given in the problem description. Through this model the documents can be ranked based on the query term appearing in each document, independent of the inter-relationship between the query terms within a document.

Expectation:

In this model we are considering the query inverse frequency, it means longer sensible queries will going to generate better performance to this model and yield more relevant document. And the relationship between the query terms in the document is not taken into the account while assigning scores to relevant documents.

Laplace Model

It is a language model in which a topic in a document or query can be represented as probability of the terms i.e the terms that are occurring frequently when discussing a topic will have high probabilities in the corresponding language model. In this language model, the documents are ranked by the probability that the documents language model could have generated the user’s query term.

Expectation:

Basically the language model is calculating the probability on the basis that higher the frequency of the terms in the document**,** the higher will be the result. Therefore its difficult to accurately determine how relevant the query for the document.

Jelinek-Mercer Model

This is also a language model in which the documents are ranked according to the probability they are getting for the query, means this document language model might have generated the user's query terms. It basically combines the relative frequency of a query term in the Document D with the relative frequency of the term in the collection as a Whole.

Exceptions:

There will be the variation in the result because result dependent on lambda which vary with the each corpus collections and the set of the query. So it may happen that we may not get consistent result for queries or the same query with different length and different collections. We can expect documents which contain all query terms with reasonable frequency to be ranked higher.

Result Analysis

Okapi TF:

Now from the result we can analyze that document with the high frequency of query terms means importance of the term in document will be ranked greater than other documents. Let see the query 202(uss carl vinson) which doesn't do well because when each term ranked separately, there is no dependency between the term. This is as per the expectation return the irrelevant document may be served due to such query.
Query234 is longer that Query 243 but Query 234 score higher than Query243. This is due to the presence of good common words.

Query ID	**GAP Score**
202	0.00735294117647
214	0.540617410236
216	0.395370981098
221	0.461059091585
227	0.162253943855
230	0.280029078668
234	0.665989992928
243	0.259722572053
246	0.233916113403
250	0.252199921731
avg	**0.325851204673**

TF-IDF:

Now if we see the query202(uss carl vinson) again score low, this is because the word uss has high frequency in most of the documents, which means its not a common query, and having a very high presence in corpus.

Query ID	**GAP Score**
202	0.00689655172414
214	0.497441659633
216	0.447169070129
221	0.387552392289
227	0.135222218749
230	0.193492758315

234	0.593412092249
243	0.231501330264
246	0.214848223592
250	0.217415905867
avg	**0.292495220281**

BM-25:

Now according to the expectation the query which are meaning full should have the better score that we can see 234(Dark chocolates health benefits) having a good score because the query makes sense and 250(ford edge problems) is a kind of vague query which score low than 234 which is according to the expectation. But the query 202 (uss carl vinson) scores very low even though it is specific.

Query ID	**GAP Score**
202	0.00657894736842
214	0.528940122509
216	0.428670973708
221	0.419002623348
227	0.236254798007
230	0.284960922239
234	0.689762043577
243	0.439864619141
246	0.189048804601
250	0.384181848741
avg	**0.360726570324**

Laplace:

Now the queries which are according to the document language model produces better scores in the laplace case. Thus query 202(uss carl vinson) scored very less score because the words is difficult to get in corpus

Query ID	**GAP Score**
202	0.0046511627907

214	0.492480425041
216	0.436716273662
221	0.412867025902
227	0.12593092272
230	0.154559280027
234	0.5804594258
243	0.182201491398
246	0.118569655184
250	0.289273678897
avg	**0.279770934142**

Jelinek – Mercer:

According to the Jelinek Mercer the difference between the scores of the consecutively ranked documents is not drastic. So the query with high frequency keyword are generally ranked higher thus leading to a very high score for 202(uss carl vinson). Thus the curve made through it is smoother. After comparing with all the function this is the function with the good average GAP score.

Query ID	**GAP Score**
202	1.0
214	0.505702594049
216	0.359451599824
221	0.333155111994
227	0.121704061109
230	0.236587229931
234	0.598288287077
243	0.452927361117
246	0.207346678433
250	0.246078715721
avg	**0.406124163926**