CAB431 Tutorial (Week 6): BM25-based Simple Search Engine

****

BM25 is a bag-of-words retrieval function that ranks a set of documents based on the query terms appearing in each document, regardless of the inter-relationship between the query terms within a document (e.g., their relative proximity). It is based on binary independence model. The equation is defined:

$$\sum_{i \in Q} \log \frac{(r_i + 0.5)/(R - r_i + 0.5)}{(n_i - r_i + 0.5)/(N - n_i - R + r_i + 0.5)} \cdot \frac{(k_1+1) f_i}{K + f_i} \cdot \frac{(k_2+1) q f_i}{k_2 + q f_i}$$

For each term i in a query Q:

- N is total number of documents in given documents collection;
- n_i is the document frequency of term i i.e. the number of document the term i occurs in;
- R is total number of relevant documents in given document collection, usually not given so we just assume **it is zero**;
- r is the number of relevant documents which contain term i, usually not given, so we just assume **it is zero**;
- f_i is term i frequency in given document D;
- qf_i is term i frequency in given query Q;
- $K = k_1((1 - b) + b * \frac{dl}{avgdl})$
- k_1, k_2 and b are parameters, typically ***k₁* = 1.2, *b* = 0.75, and *k₂* = 100**
- dl is document length of D i.e. words count of D, $avgdl$ is the average document length in given documents collection

TASK 1: Calculate and store the average document length in given documents set.

- In the BowDocument class, add a field (class variable) of docLength, write accessor (get) and mutator (set) methods for it.
- Modify the processor class, while parsing every input documents, for BowDocument object creation, call mutator method of docLength to save the document length to the BowDocument object. In the same time, sum up every BowDocument's docLength as totalDocLength, then in the end the parsing all

input document, calculate the average document length. You may keep the $avgdl$ as processor class variable for later use. Or you can create a method taking a BowDocument collection as a parameter, then calculate the average document length for it.

- Print out a list of docID : docLength, and average document length for given document collection.

TASK 2: Create a method to calculate a given document's BM25 score for a given query. Please note you should parse query using same stemming method as parsing document. Print out the BM25 scores for every document in collection for query "stock market"

TASK 3: Rank every document in a given document collection using BM25 score with a given query.

- You may create a ranking method taking a BowDocument collection as the parameter. Or simply add ranking process in the parsing method.
- Sample query input:
    - British fashion
    - fashion awards
    - Stock market
    - British Fashion Awards
    - Car failure
- Print out the ranking result of top 3 relevant documents with given queries. Sample Ranking result output:
    - For your query "stock market", top 3 relevant documents is: docId3:BM25score3, docId9:BM25score9, docId2:BM25score2
- OPTIONAL: ask user to input a query (like Google) then give the rank result.

Example of output

Average document length 272 for query: British fashion

Document: 741299, Length: 199 and BM25 Score: 0.000000
Document: 741309, Length: 104 and BM25 Score: 2.937976
Document: 780718, Length: 107 and BM25 Score: 0.000000
Document: 780723, Length: 124 and BM25 Score: 0.000000
Document: 783802, Length: 120 and BM25 Score: 8.550640
Document: 783803, Length: 490 and BM25 Score: 0.000000
Document: 807600, Length: 538 and BM25 Score: 0.000000
Document: 807606, Length: 187 and BM25 Score: 0.000000
Document: 809481, Length: 151 and BM25 Score: 0.000000
Document: 809495, Length: 703 and BM25 Score: 0.000000

For query "British fashion", three recommended relevant documents and their BM25 score:

783802 : 8.550640
741309 : 2.937976
807606 : 0.000000

Appendix

TASK 2 Specification

- Create a method in your processor class:

```
/**
   * @param aDoc - a BowDocument
   * @param aQuery – a String of query
   * @param avgDocLen - a int of the average document length
   * @param docNo – a int of total number of document in a collection
   * @return a double(float) value of given document's BM25 score.
   */
  private double calculateBM25(BowDocument aDoc, String aQuery, int avgDocLen, int
docNo) {
   //your code here, please refer the introduction of BM25 equation above for
parameters setup and calculation.  If some of them are not given, it means you can get
them from the result of previous tutorial tasks or you could work out somehow.
}
```

- Please note you should parse query using same stemming method as parsing document.
- Print out the BM25 scores for every document in collection for query "stock market"