IMPLEMENTATION REPORT FOR ASSIGNMENT 3

This assignment has two programs.

1. indexer.py
2. bm25.py

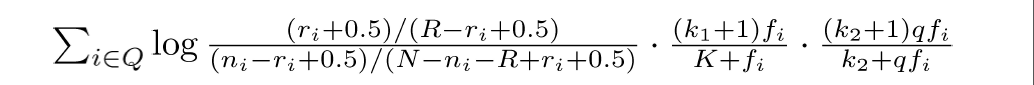
indexer.py:

This module is used to create an inverted index file from the document collection. The file creates index with Unigram terms. The inverted index has the following format,

Term -> (docID, FrequencyinDocument) (docID, FrequencyinDocument)

docID is an incremental number generated on the fly. The mapping between DocID and filename is contained in the docID mapping file. The inverted index is stored in JSON format. Before index terms are generated I have tokenized the words in the Wikipedia article removing all unnecessary symbols except hyphen. Also certain sections of Wikipedia like references are ignored.

bm25.py

The bm25.py files takes the index file, query file and returns the top 100 documents according to the BM25 score. Average document length and total number of documents are calculated from the index. The following formula is used to compute BM25,

The data structure for inverted index is a dictionary in which index term is the key and the value is a list of tuples of the type (document\_id, count). We use list of tuples so that we can efficiently iterate over the list while ranking the documents for queries

Apart from the inverted index, we also have a dictionary of token count in which document-id is the key and frequency of occurrence is the value.

The query file is parsed to obtain a list of words for each query. This word is then compared against the inverted index and BM25 score is calculated iteratively over each frequency term of query.

The documents are displayed in decreasing order of BM25 scores. i.e The document with highest BM25 score is displayed first and document with lesser scores are displayed later