CS 453, Fundamentals of Information Retrieval, Spring 2016

Project Assignment 1

Processing, Retrieving, and Ranking Documents in a Wikipedia collection

due Monday, May 9

1 Project Description

The purpose of this project is to (i) apply various *text operations* (i.e., stopword removal and stemming) on a text document collection C to create the corresponding indexed structure, and (ii) use the *indexed structure* to retrieve and rank documents in C that are relevant to a user's query based on $TF \times IDF$.

1.1 Text Processing

Using a small Wikipedia document collection (posted under http://students.cs.byu.edu/∼cs453ta/projs.html), denoted $Wiki$, you are required to

a. Implement and run a *word tokenizer* on $Wiki$. The tokenizer you are to implement should remove capitalization, punctuation symbols, and hyphens. Note that if you have decided to simply remove hyphens and concatenate the words (before and after each hyphen), you should check the spelling of each concatenated word using the provided dictionary for the project assignment. If a concatenated word is a misspelled word, you should not concatenate the corresponding words in the first place.

b. *Remove stopwords* from (the documents in) $Wiki$. To facilitate this task we provide a Java implementation of a stopword-removal tool, along with the stopword list, which can be downloaded from http://students.cs.byu.edu/∼cs453ta/projs.html.

c. Reduce the non-stopwords in (the documents in) $Wiki$ to their grammatical stems using the *Porter Stemmer* algorithm. We provide a Java implementation of the *Porter Stemmer*, which can be downloaded from http://students.cs.byu.edu/∼cs453ta/projs.html. You should make sure that before calling the *Porter Stemmer* algorithm, any spaces before or after the words to be stemmed MUST be removed.

d. Create an indexed structure for the $Wiki$ collection, which should include for each stem s in $Wiki$ (i) the *documents* (identified by their IDs) in which s appears and (ii) the *frequency of occurrence* of s in each document.

1.2 Evaluation of Keyword Queries

For each of the keyword queries listed in Section 2, you are required to *retrieve* and *rank the top-10 most relevant documents* from the $Wiki$ collection based on their ranking scores. The *ranking score* of each document d (in the $Wiki$ collection) with respect to a query q is computed as

$$\begin{aligned}
Score(q,d) &= \sum_{w \in q} TF(w,d) \times IDF(w) \\
TF(w,d) &= \frac{freq(w,d)}{max_l(freq(l,d))} \\
IDF(w) &= log_2 \frac{N}{n_w}
\end{aligned}$$

where w is a non-stop, stemmed word in q, $freq(w,\ d)$ is the number of times w appears in d, N is the number of documents in the $Wiki$ collection, and n_w is the number of documents in which w appears.

2 Keyword Queries to be Evaluated

For each of the ten test queries q given below, you are required to retrieve and rank the top-10 documents with the highest ranking score (among the others) with respect to q. For each retrieved document you are required to include its ID, first sentence, and computed ranking score, as shown in Table 1. Note that you must perform *stopword removal* and *stemming* on the ten test queries prior to processing them.

1. killing incident
2. suspect charged with murder
3. court
4. jury sentenced murderer to prison
5. movie
6. entertainment films
7. court appeal
8. action film producer
9. drunk driving accusations
10. actor appeared in movie premiere

Query:			
Ranked Documents	Document ID	First Sentence	Ranking Score
1			
2			
3			
4			
5			
6			
7			
8			
9			
10			

Table 1: Expected information for the top-10 documents retrieved for each of the ten test queries to be processed

This assignment is worth 100 points.