University of Texas at Dallas
CS 6322.001 : Information Retrieval
Spring 2016
Homework # 3

Instructor: Dr. Sanda Harabagiu
Grader: Ramon Maldonado

Issued: March 21st 2016
Due April 11th 2016 before midnight

Problem (100 points)
Ranked Retrieval

In this assignment you will implement a simple statistical relevance model in your retrieval system based on the vector relevance model, using the inverted list index that you built in the last assignment. The system should retrieve and rank the documents that satisfy the queries from the file:

/people/cs/s/sanda/cs6322/hw3.queries

The relevance model of your retrieval system must (1) read a query, (2) parse it by determining the tokens, (3) discard stop-words, (4) generate the lemmas for the content words and then (5) compute the weights of the query vector.

In addition, the relevance model needs to use the index to compute the weights that constitute the vector representations of the documents in your collection. This allows you to have vector representations for queries as well as for documents.

The vector representations of queries and documents are used to determine scores that inform the ranking of documents against the queries. The scores are obtained by computing the cosine similarity for every query-document vector pair.

In this homework you are asked to implement and compare two term weighting functions for the query and document vector representations. The weighting functions are provided by W1 and W2:

W1 = (0.4 + 0.6 * log (tf + 0.5) / log (maxtf + 1.0))
* (log (collectionsize / df)/ log (collectionsize))

W2 = (0.4 + 0.6 * (tf / (tf + 0.5 + 1.5 *
(doclen / avgdoclen))) * log (collectionsize / df)/
log (collectionsize))

where:

tf: the frequency of the term in the document,

maxtf: the frequency of the most frequent indexed term in the document,

df: the number of documents containing the term,

doclen: the length of the document, in words, discounting stop-words, - you may use the same stopword list as in the previous homework;

avgdoclen: the average document length in the collection, considering the doclen of each document, and

collectionsize: the number of documents in the collection.

W1 is a variation of older, but well-known, 'max tf' term weighting. W2 is a variation on Okapi term weighting. Both TW1 and TW2 use a fairly standard idf, namely:

idf = log (collectionsize/df)

When evaluating the relevance of the documents in the collection against each of the queries, documents should be presented in ranked order of the total scores.
FOR each query:

1. Turn in the vector representation of the query (10 points per weighting scheme), and the **top 5 documents** for the query under both weighting schemes (*50 points, with 25 points per weighting scheme). You are also required to present the vector representations for each of the first 5 ranked documents.*
2. Indicate the rank, score, external document identifier, and headline, for each of the top 5 documents for each query. (*5 points*)
3. Identify which documents you think are relevant and non-relevant for each query. (*10 points*)
4. Describe why the top-ranked non-relevant document for each query did not get a lower score. (*5 points*)
5. Briefly discuss the different effects you notice with the two weighting schemes, either on a query-by-query basis or overall, whichever is most illuminating. For example, you can point out that the weighting scheme seems to be working for this query as well as a list of other queries, but not for some other queries you have noticed. Try to explain why it works and why it does not work.(*5 points)*
6. Describe the design decisions you made in building your ranking system. (*5 points*)