RYERSON UNIVERSITY

CIND-110

DATA ORGANIZATION
FOR DATA ANALYSTS

Assignment III
On Information Retrieval Approaches

Instructor:
Tamer ABDOU, PhD

This assignment counts for 10% of the final grade

Instructions

- The attached script reads 100 transcripts exported from https://www.ted.com/talks, and creates a Term Document Matrix to store the frequencies of words/terms and the references to the documents that contain them. Then, it ranks 99 transcripts according to their similarity to the first transcript 'Query' Using Cosine Similarity function, and displays the top 10 similar transcripts on screen.
- Use RStudio for this assignment. Edit the file 'A3_F20_Q.Rmd' and insert your R code, then click the Knit button to generate a document that includes both the content and the output of the embedded R code chunks.
- When you are done with your answers and before submitting, save the file with the following naming convention: your Lastname.Firstname
- **Submit** the source (in RMD format) and the output (either in PDF, WORD, or HTML format) files. Failing to submit both will be subject to mark deduction.

Question 1

Apply three different text pre-processing techniques to cleanse the data before creating the Term Document Matrix. For example, you might trim the suffix and prefix of the original words by applying a Stemming algorithm. In addition, you might remove the most commonly used words in the language which seldom contribute to the meaning of the sentence, by applying a Stopword Removal algorithm.

Question 2

For each applied pre-processing technique, explain why this technique is necessary for better information retrieval.

Question 3

The given RMD script - on the course shell - is using the following formulae to compute the weight of each term/word (TF-IDF) across all documents/transcripts.

$TF_{ij} = f_{ij} / \sum_{i=1}^{|V|} f_{ij}$

$IDF_i = log(\frac{N}{n_i})$

In this formula, the meaning of the symbols is:

- TF_{ij} is the normalized term frequency of term i in document D_j.
- f_{ij} is the number of occurrences of term i in document D_j.
- IDF_i is the inverse document frequency weight for term i.
- N is the number of documents in the collection.
- n_i is the number of documents in which term i occurs.

Apply the following three TF-IDF variants (one at a time), then compare the outputs against the original TF-IDF output.

1. First Variant:
   $TF_{ij} = f_{ij} / \sum_{i=1}^{|V|} f_{ij}$, $IDF_i = 1$
2. Second Variant:
   $TF_{ij} = 1 + log(f_{ij})$, $IDF_i = log(1 + \frac{N}{n_i})$
3. Third Variant:
   $TF_{ij} = f_{ij}$, $IDF_i = log(\frac{N}{n_i})$

Question 4

If we redefine the 'Term' in Question 3 to be 'two adjacent words' rather than 'one word' and the TF-IDF weights are computed accordingly, do you think this approach would be worth pursuing? Explain and show your work.