UCC

University College Cork, Ireland
Coláiste na hOllscoile Corcaigh

Information Storage and Retrieval - CS6409
Lecturer: Dr. Alejandro Arbelaez

Submission:

This assignment is due on March/1/2021 at 9:00AM. You should submit a file with the source code of your solution electronically via Canvas. You can complete the assignment in Python, Java, C++, C, or R. (Recommending Python).

Please note that this assignment will account for **10 Marks** of your module grade

Declaration:

By submitting this assignment. I agree to the following:

"I have read and understand the UCC academic policy on plagiarism and I agree to the requirements set out thereby in relation to plagiarism and referencing. I confirm that I have referenced and acknowledged properly all sources used in preparation of this assignment.

I declare that this assignment is entirely my own work based on my personal study. I further declare that I have not engaged the services of another to either assist me in, or complete this assignment"

This project is intended to give you hands-on experience in using practical ranking retrieval systems. You will implement TF-IDF (term frequency X inverse document frequency) and the cosine similarity for ranking.

In this work we will use an online dataset of paper reviews sent to an international conference mostly in Spanish (some are in English). In particular we will use a set of more than 100 documents available in Canvas. This dataset is useful for a variety of information retrieval and text classification projects.

Exploration of the dataset (2 Marks)

Your first task is to get a feel of the data that you will dealing with in the rest of the assignment.

- Use tokenization and compute the top 200 most popular words (total occurrences)
- Repeat the previous step, but now filter tokens by length (min. size = 4 and max. size = 20). Please indicate the total occurrences of the words.

Retrieval System (5 Marks)

In this part, you will build a retrieval system for exact top k retrieval (use k=10). Using TF-IDF and Cosine Similarities.

I recommend the use of dictionaries to store the frequency of the each word. (Note that keys of the dictionary should correspond to all words in the vocabulary and the values should specify how often they occur). For example, if the word "brilliant" occurs 5 times in the document your dictionary should be <"brilliant" : 5 >. You need to record the frequency and TF-IDF values for all words in your documents.

Your program should take as input a document (from the dataset) and output a list with the 10 most relevant documents (from the dataset), using TF-IDF and Cosine similarities. These documents need to be ordered by relevance, with the first document being the most relevant.

Code requirements: (3 Marks)

Program Correctness: Your program should work correctly on all inputs. Also if there is any specifications about how the program should be written, or how the output should appear, those specifications should be followed.

Readability: Variables functions should have meaningful names. Code should be organized into functions/methods where appropriate.
There should be an appropriate amount of white space so that the code is readable, and indentation should be consistent.

Documentation: your code and functions/methods should be appropriately commented. However, not every line should be commented because that makes your code overly busy. Think carefully about where comments are added.

Code Elegance: There are many ways to write the same functionality into your code, and some of them are needlessly slow or complicated. For example, if you are repeating the same code, it should be inside creating a new method/function or for loop

Code efficiency: The implementation is logically well designed without inappropriate design choices (e.g., unnecessary loops).