25. 10. 2021

Homework Assignment 3

Data Structures and Algorithms I, WT 2021

Due: 5. 11. 2021

1. (14 Points) Let $X = \{r_1, r_2, \dots, r_N\}$ be a set containing N product reviews r_i from an online retailer. Each review r_i is represented as an array $r_i = [w_1, w_2, \dots, w_{M_i}]$ where w_1 is the first word of the review and w_{M_i} is the last word of the review. All words are lower case and r_i does not contain special characters (they were removed from the review). For example, the array r for the review "Stunning even for non-gamer: This sound track was beautiful." is given by $r =$ ['stunning', 'even', 'for', 'non', 'gamer', 'this', 'sound', 'track', 'was', 'beautiful']. The reviews should be automatically classified, based on r_i, into positive and negative reviews. Classifying the reviews can be accomplished by looking at the relative frequency of words that occur in the reviews. For example, words like "beautiful" and "amazing" will appear more often in positive reviews than in negative reviews. However, the accuracy of the classifier can be improved by considering a sequence of k consecutive words (e.g., it is better to consider "not good" as a whole than "not" and "good" individually).

   You do not have to understand all of the details of the classification algorithm (we will provide the algorithm). Your task is to prepare the data in a way such that the classification can be done efficiently. More specifically, your task is to design an algorithm that computes an array $y = [y_1, y_2, \dots, y_L]$ and an array $c = [c_1, c_2, \dots, c_L]$. The array y should contain each text fragment, that is, k consecutive words, of X exactly once, and the ith entry in c should correspond to the number of times y_i occurs in X. Your algorithm should work for arbitrary values of k, and y should be sorted in descending order (i.e., y_1 is the most frequent word). For example:

   Let $X = \{$['the', 'worst', 'a', 'complete', 'waste', 'of', 'time'], ['the', 'best', 'cookie', 'mix', 'of', 'all', 'time']$\}$.

   If we use $k = 1$ we want:

   $y =$ ['the', 'of', 'time', 'worst', 'a', 'complete' 'waste', 'best', 'cookie', 'mix', 'all'] and
   $c = [2, 2, 2, 1, 1, 1, 1, 1, 1, 1, 1]$.

   Setting $k = 2$ should give:

   $y =$ ['the worst', 'worst a', 'a complete', 'complete waste', 'waste of', 'of time' 'the best', 'best cookie', 'cookie mix', 'mix of', 'of all', 'all time'] and $c = [1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1]$.

   You should design, analyze, and implement your algorithm. The algorithm should be as efficient as possible.

   1a. Describe your algorithm in words and in pseudo code.

   2a. Analyze the running time of your algorithm.

   3a. Argue that your solution is correct.

   1b. Implement your algorithm in Python.

   Assume that a function `CompareString`($s1, s2$) exists. This function returns `True` if $s1 = s2$; otherwise, `False`. Algorithms that were discussed in the lecture (e.g., sorting algorithms) do not have to be explicitly written down in the pseudo code (however, you have to consider their runtime for your analysis).

   You will have to hand in two files, one written report (containing 1a. through 3a.) and a Python file containing the implementation of your algorithm. Please note the naming convention of the Python file for the submission: `group-number_sheet-number_name_surname.py`, example: `01_03_John_Smith.py`. Note the leading zeros. For the implementation use Python version 3.x and the template (`ex3_template.py`) provided on the course website. The sections to be completed by you are marked in the template with `TODO begin` and `TODO end`. Do not make any further changes in the template. You have to implement the (sorting) algorithm by yourself (e.g., you are not allowed to use `sort`, `sorted`, etc nor any Numpy function like `unique`).

Note: Although Python 3 is installed on some systems, the interpreter of version 2 is called with the `python` command (can be checked using `python --version`). In this case you have to work with the command `python3`. Numpy, Scipy, Scikit-learn and NLTK are used in the Python program. You may need to install these packages. On Linux this is relatively easy with pip. Use `pip install numpy`, `pip install scipy`, `pip install scikit-learn`, `pip install nltk`.