Ivan Kostine
17324427
21.11.22

Algorithmics and Data Management
Project 1 – Report

Description :
For this project I worked completely alone.

Code explanation:
The program can be executed by calling main.py. When done so, the function get_highest_similarities() gets called, which calls in chain, all the other functions.

List of functions and brief explanation:

- Read_reference_text(): This function reads the provided reference text and stores all the sentences in a list of strings. It is also responsible for getting rid of all punctuation and transforming all characters to lowercases.
- Make_word_vector(): This function creates a dictionary of key/values containing the occurences of a **word w1** in the same sentence as a **reference word W**. The key is w1, and the value is its number of occurences. To do so, it verifies that W is not a stop word and that w1 is not equal to W. It also verifies that w1 is longer that 3 characters. The function returns the dictionary with key/value pairs where key contains the string and value contains the number of occurences.
- Sim_word_vect(): This function computes the cosine similarity between two strings. To do that it calls the function product() which computes the scalar product between two vectors (produced by make_word_vector()).
- Compute_all_similarities(): It allows to compute the similarities between all words that are part of a list of words. To do that, it is composed of two for loops that iterate through the list of words. The first for loop takes the first word to compute and stores it in the variable "**w1**", the second for loop takes the second word to compute and stores it in the variable "**w2**". Because the similarity **w1->w2** is equivalent to the similarity of **w2->w1**, it is redundant to calculate both. To calculate the cosine similarities, the function **sim_word_vect(w1, w2)** is called. When the for loops arrive to values of w1/w2 or w2/w1 that have already been computed, it simply skips them and continues the loop until reaching the end of the list. All the similarities are stored in a list "**cosine_sim_list**" and appears in the form of [w1, w2, similarity]. In addition to that, all the results are stored in a text file named **all_results.txt**.
- Get_highest_similarities(): This function takes the list computed by **compute_all_similarities()** and a list of words as parameters. Its goal is to sort the list of similarities and to keep only the results where the similarities are the highest between a pair of words that are part of the list of words (passed as parameters). It produces a file named **highest_results_only.txt** that contains only the pairs of words with the highest cosine similarity.