CS 562 Proposal – What to Test

Haoxiang Wang, Student ID:932359049

For this testing project, a third party python library named "***fuzzywuzzy***" will be tested. The library is implemented to accomplish the fuzzy string matching. It is significantly useful and well known. It has been downloaded from Python Software Foundation for 70,000 times just in last month. It is a really powerful, light weighted, and easy to use string matching tool implemented as a python library. It can be found here on the Github: https://github.com/seatgeek/fuzzywuzzy.

The fuzzywuzzy is a fuzzy string matching tool which is able to handle several complicated situations. It uses *Levenshtein Distance* to help calculate differences between sequences. It contains several function calls that can handle difference situations. The most basic call is "*fuzz.ratio*". It takes two strings as input and calculates the percentage of matching in between these two strings. It is also capable of partial ratio, which detects whether one string is part of another one, by using the function call "*fuzz.partial_ratio*". It can ignore the order of the words in one sentence by calling "*fuzz.token_sort_ratio*". This function call basically sorts the tokens in two strings and then compares them. It also can ignore the repeated words in two strings by using the function "*fuzz.token_set_ratio*", each different word in the string will only be counted once and applying to the matching. The fuzzywuzzy also applies a kind of searching tool with the function "*process*". It can do both one word search and multi word search with two functions. The function "*process.extractOne*" will return all of the strings contain one word using as the input, and the function "*process.extract*" could find the strings with several words out of one selected string. All of the outputs of these functions contain the ratio of matching in percentage form.

To test this library, all of the functions listed above will be tested for sure. For all of the functions, the first test I am planning to run on them is to use TSTL to examine the length of the string that the function can take. Since all of the functions return the ration of matching, the value of the ratio will also be tested. The ratio result will be compared to the manually calculation result to check the correctness. Then for some more complicated function, for example "*fuzz.partial_ratio*" or "*fuzz.token_set_ration*", I am also planning to run the complication examination on them. These functions ignore some part of the original string and compare it to another one, so the length of "common part" and its complication should be tested using TSTL. As for the functions belong to the "*process*", the tests mentioned above will all be applied on them. Since these functions provide search-like operations, the robust of the search should be tested for sure. The word number limit for "*process.extract*" will be tested by TSTL. The complication and the length of the search strings will also be tested. The correctness for this library should be considered based on two parts: the ratio that functions return and the ability handling long and complicated strings. These are the main things I would like to test in this project.