Testing FuzzyWuzzy using TSTL

Deepthi S Kumar (onid: kumarde)

CS 562 - Winter 2016

Software under test

FuzzyWuzzy (https://github.com/seatgeek/fuzzywuzzy)

Description

It is a pure python library offering the following four string matching techniques.

1 Ratio - Encapsulates existing method 'ratio()' from python's difflib or using Levenshtein distance, if the package is imported.

2 Partial Ratio - Computes the best partial match between 2 strings of noticeably different lengths.If the shorter string is length m, and the longer string is length n, it returns the score of the best matching length-m substring.

3 Token Sort Ratio - Similarity between 2 strings which are out of order in their constructions. As the name suggests, this technique involves tokenizing the string, sorting the tokens in alphabetical order, form a string of the sorted token and then compute simple ratio of the sorted string.

4 Token Set Ratio - Similar to token sort ratio, but provides better result when the difference in string lengths is quite big.

Apart from the above mentioned functions, the library also offers two other methods called extract() and extractOne().

extract() - Extracts best matches of string from a list of strings based on the limit specified. The default technique used here is weighted Ratio but can be changed by passing relevant arguments.

extractOne() - Same as extract() but returns exactly one string that has highest match score from the list of strings.

Usecase

The original use case for this library was to map different search strings to the the same result in a ticket search engine. Since different users could search for a particular ticket by giving different words, the search engine has to identify the tickets being searched.

Testing Scope

Correctness of the edit distance calculated by the library for different techniques can be checked against the original libraries used. A value of 100 is returned when the strings completely match (both the strings need not be same) and a value of 0 when the strings do not match at all.

Since it is based on fuzzy-concept, the results of the operations are neither completely correct nor completely wrong unless the values are 0 or 100. The values have a range within which the results are grouped as matched or unmatched.

There is no specification as to what the range is and hence testing the functions with different combination of strings will probably yield an acceptable range. Four different matching techniques will be tested on functionality and performance and compared against each other. All the functions will be tested with unicode characters.

References

http://chairnerd.seatgeek.com/fuzzywuzzy-fuzzy-string-matching-in-python/

http://en.wikipedia.org/wiki/Fuzzy_concept