Assignment 4

Corvaglia Salvatore, Savino Franco, Villani Marco

28/11/17

0.1 Spelling Corrector

The Peter Norving algorithm results very functional and reliable, moreover it does not take too much time to compute, thing that can be improved by implementing the algorithm in another programming language which is compiled instead of Python which is interpreted.

Substantially the algorithm estimate a words probability by counting the occurrences of itself into a big size file, storing the results in a data structure like a dictionary.

Then the word is processed, generating every possible word with edit distance $d <= 2$, assuming that it cannot be possible to make more than two mistakes.

In the end, the obtained words are searched in the dictionary, checking the probability and returning the most likely, if no match is found, the word itself is returned.

In his assay, Norving point out particular aspect which could improve his algorithm, however not implemented because out of scope.

One possible improvement to this approach in terms of performance, could be gained through the use of a table, which associate a weight to every possible edit operation, going to lighten the computation phase and avoiding useless iterations.

An additional algorithm bringing further improvements in term of performance is the Symmetric Delete Spelling Correction, which results independent from the language, because it does not require the use of an alphabet to generate the words to be analysed.

In this algorithm, there is a pre-calculation stage where we generate all the words with edit distance $d <= 2$, by deletion only, for each item in the dictionary and stored into it maintaining a link between them.

Afterwards, in the execution stage, from the input, we generate all the possible words with edit distance $d <= 2$, which then are searched in the dictionary.

The cost of this approach is the pre-calculation time and storage space of x deletes for every original dictionary entry, which is acceptable in most cases.