------------------SPELL\_CHECKER\_EN----------------

SPELL CHECKER EN is a setting up a simple spell checking algorithm. It uses a Levenshtein Distance algorithm to find permutations within an edit distance of 2 from the original word. It then compares all permutations (insertions, deletions, replacements, and transpositions) to known words in a word frequency list. Dictionary was generated using the WordFrequency project on GitHub.

The Levenshtein distance is a string metric for measuring the difference between two sequences. Informally, the Levenshtein distance between two words is the minimum number of single-character edits (insertions, deletions or substitutions) required to change one word into the other.

Installation

The easiest method to install is using pip:

pip install pyspellchecker

Additional Methods:

On-line documentation is available; below contains the cliff-notes version of some of the available functions:

correction(word): Returns the most probable result for the misspelled word

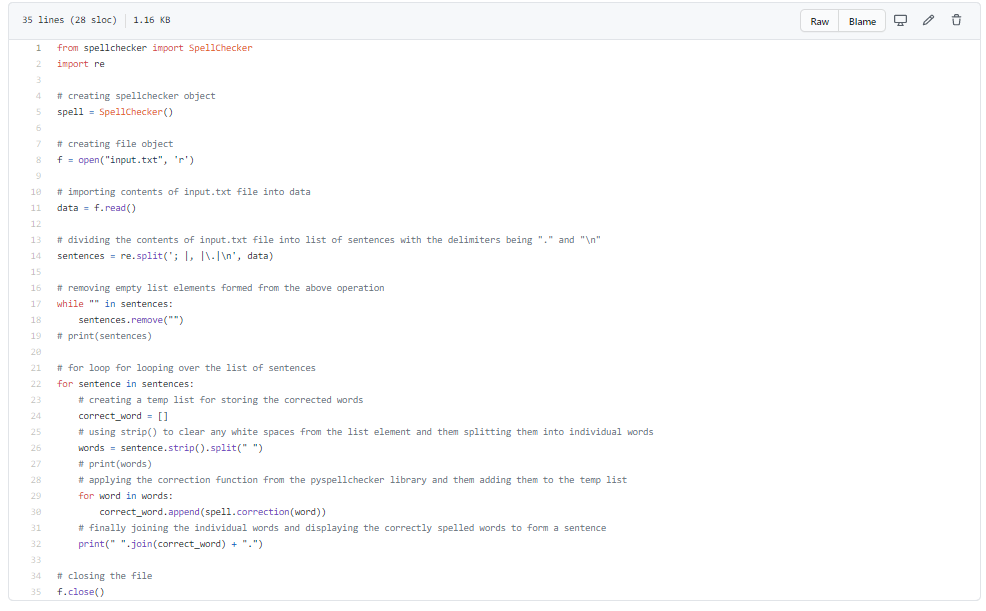
candidates(word): Returns a set of possible candidates for the misspelled word

known([words]): Returns those words that are in the word frequency list

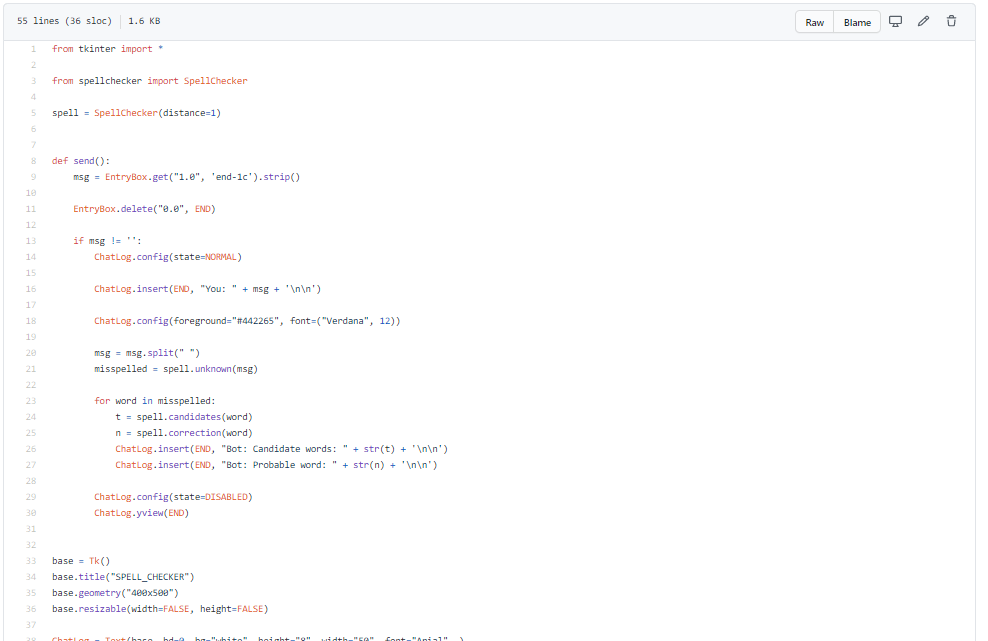
unknown([words]): Returns those words that are not in the frequency list

word\_probability(word): The frequency of the given word out of all words in the frequency list

It has three parts :

1. A script.py file that takes input as a paragraph and gives a sequence of lines in their most appropriate words.It uses the module spellchecker to find out the correct words using function correction() to do so.
2. A GUI interface that takes multiple words ,sorts out multiple misspelled words and displays their most appropriate words . It uses the module tkinter to implement the GUI and also implements

the spellchecker module to find out the corrected spellings.



3 A CLI that takes a word and if it is misspelled then return the various appropriate words. It also makes use of spell checker module.

