

Fall Semester 2016
Iowa State University

ENGL 520 - Computational Analysis of English

Problem Set 2
Basic Text Processing Practice
(ungraded)

1. Compare the works of two of your favorite writers in terms of the most frequently used words in their novels, using a computer program. Use the free ebooks available on project gutenber website for getting the corpus you want.
2. Write a program that converts a user entered sentence into Morse code. Google for morse code to know the mapping between English characters and Morse code sequences.
3. Build a small English to German dictionary file (collect about 100-150 words from the web and save in a .txt file), and write a program that takes a English sentence from user as input, tokenizes it, and looks word by word in the dictionary, and translates the sentence to German. If the word is not found in the dictionary, leave it as it is. If it is found, replace with the translation.
4. Write a program (using regular expressions or any HTML parser tool you know of) that takes a Wikipedia URL as input and returns the number of languages in which the page exists as output. The languages in which the same wiki page exists can be seen in the left frame, at the very end, in all wikipedia pages.
5. Implement a Caesar cipher encoder. Caesar cipher is a simple encryption technique named after Julius Caesar, who used it to communicate with his team in cryptic messages. Usually, Ceasar cipher refers to any encryption where each letter is replaced by another letter a fixed number of places down in the alphabet. For example, if the fixed number is 3, A becomes D, B becomes E and so on. Now, your program should implement a Ceasar cipher of the fixed size 4. Leave the punctuation markers unchanged. Your input will be a normal English sentence, and the output should be a Caesar cipher.
6. Implement a Caesar decoder for the above cipher. Your input is a Ceasar cipher encoded sentence, and your program should decode it and return the decrypted English sentence.
7. Implement the Hangman program as described in this url: <http://openbookproject.net/pybiblio/practice/wilson/hangman.php>

8. Write a program that generates a jumbled sentence and asks the user to enter the correctly ordered sentence. If the user enters correctly, inform them that they are right. Otherwise, ask them to enter the correct text. Repeat for a maximum of 5 times. For the generated sentence: Have 10 sentences stored somewhere, and randomly choose one of them, and jumble the order of words before presenting to user.
9. Do problem 46 in http://www.ling.gu.se/~lager/python_exercises.html. Although the link calls it python exercises, you can solve it in any language you want.
10. Write a program that can read a microsoft word document, and print out the text in it. Refer to the website <https://automatetheboringstuff.com> for some relevant Python libraries. Perl users - search for Text::Extract::Word or some such relevant perl module.