Jeff Lee
1/16/17
Programming Project Part 1

So far, this code is working very well. Each function operates as it is intended to, however there are complications with stemming due to the complexity of the English language. The code is clean and well-segmented, making it easy for me to go back and make minor edits to certain functions. The output looks very similar to the given examples, and even the longer Alice text cleans, stems, and sorts properly.

The design of my program begins with import statements and global variables at the top, providing easy access for quick changes. Then, each of the functions are defined. The order of the definitions is similar to how they are executed, beginning with the cleaning functions and then moving on to the stemming functions. Next, there are two functions which group cleaning and stemming into singular functions, so that they may be called in a group more easily. After that, I went on to write the functions for organizing the list, beginning with counting and then sorting. Finally, I wrote the print functions as they are the final methods executed. After the definitions, I have a main execution section that I can easily tweak to include new definitions.

I have found that it is convenient to clean the list before stemming, then again calling the removeBoringWords function to ensure that stemming did not produce any more boring words that previously had suffixes. removeBoringWords is a fairly simple operation, and it is quicker for it to be called twice than to run stemming functions on words already deemed boring.

List of functions:

def removeBoringWords(list):

The removeBoringWords function removes all words from a list that appear in a list of words noted as "boring". The input is a list of strings.

def removePunctuation(list):

The removePunctuation function removes all punctuation from a string. The input is a list of strings.

def removeNonWords(list):

The removeNonWords function removes all strings that do not contain a vowel. The input is a list of strings.

def removeEd(list):

The removeEd function removes the past tense suffix -ed from a string. The input is a list of strings.

def removeEr(list):

The removeEr function removes the suffix -er from a string. The input is a list of strings.

def removeIng(list):

The removeIng function removes the suffix -ing from a string. The input is a list of strings.

def removePlural(list):

The removePlural function removes the plural suffix -s from a string. The input is a list of strings.

def cleanList(list):

The cleanList function cleans a list of strings by executing three functions to remove punctuation, non-words, and boring words. The input is a list of strings.

def stemList(list):

The stemList function executes four stemming methods to simplify words to their base form. The input is a list of strings.

def wordCounter(list):

The wordCounter function creates a list of strings and a list of integers in parallel, and counts how many times each word appears. The input is a list of strings.

def wordSorter(listTuple):

The wordSorter function takes a parellel list tuple and sorts the first list of strings in accordance to the parallel second list of integers. It also shortens the lists to the global limit MAX_WORDS so as to not crowd the console. The input is a two-tuple of lists, the first being strings and the second being integers.

def printDictionary(listTuple):

The printDictionary function prints a parallel list tuple in the format provided in the assignment. The input is a two-tuple of lists.

def printTextCloud(sortedListTuple):

The printTextCloud function prints a parallel list tuple in the format provided in the assignment. The input is a two-tuple of lists.