Lab11 Shakespeare (11.11.19) CS103 Fall 2019

author john k johnstone jkj at uab dot edu
course CS103 Fall 2019
license MIT
version Fall 2019

materials

- *lab11_19fa103.pdf* (this document)
- *lab11_19fa103.py*
- remove_punctuation.py (do not read me)
- text data:
  - a_midsummer_nights_dream_folger.txt
  - romeo_and_juliet_folger.txt
  - sonnet_18_shakespeare.txt

purpose

- explore the Python dictionary

in-class exercises

- 15 minutes in: algorithm for `vocabDict` and a hint for testing

coding exercises

- consider the `vocabDict` function that builds a Python dictionary (a structure of type 'dict') that encodes the vocabulary of a piece of text;
  the dictionary counts the occurrences of words in the text; similar words should be collapsed before counting, so that, for example, 'cat', 'Cat', and 'cat!' all contribute to the count for 'cat';
  the key of a key/value pair is a string w in the text;
  the value of this key/value pair is the number of times w occurs in the text
- before implementing this function, decide how to attack it: write an algorithm for `vocabDict`
- `removePunctuation`: you may find the helper function 'removePunctuation' useful: it is a function with one string parameter that returns this string with all punctuation removed (e.g., input 'Cat!?&#' is transformed to output 'Cat'); it is provided in the script 'remove_punctuation.py' and leverages string.punctuation in the string module; there is no need to read, let alone understand, this function: just use it (e.g., removePunctuation ('Cat!?&#')); you can use it since it has already been imported (see top of lab11_19fa103.py)
- implement the `vocabDict` function

- implement the `vocabSize` function that finds the size of the vocabulary (number of unique words) in some text; collapsing is again used in counting (not just punctuation, but lowercase/uppercase)
- questions you should answer after implementing vocabSize
  (please place your answers as a clean comment at the top of lab11_19fa103.py)
  - how many unique words in Shakespeare's 18th sonnet?
  - how many unique words in "A Midsummer's Night Dream"?
  - how many unique words in "Romeo and Juliet"?

- implement the `mostPopular` function that finds the most popular word in some text;
  it returns the most popular word and the number of times it occurred, combined into a tuple;
  does this computation feel familiar? use the same approach you used earlier for this problem
- questions you should answer after implementing mostPopular
  - what is the most popular word in Shakespeare's 18th sonnet?
  - what is the most popular word in "A Midsummer's Night Dream"?
  - what is the most popular word in "Romeo and Juliet"?

- implement the `nMostPopular` function that finds the n most popular words in some text;
  useful idea from dictionary (del to delete an entry in a dictionary)
- questions you should answer after implementing nMostPopular:
  - which character is mentioned most in Romeo and Juliet?
  - what is the most common adjective in Romeo and Juliet?
  - which character is mentioned most in A Midsummer Night's Dream?
  - what is the most common adjective in A Midsummer Night's Dream?

challenge

- ignore stop words in your counts; stop words are words like 'a', 'the', 'and', 'or' that may not be considered important enough to count (and which often will be the most popular words, adding noise to the count of popular words)

deliverables

A+: A exercises and a challenge
A: B exercises, answers to the counting questions, vocabDict code
B: C exercises and in-class exercises
C: attendance