INLS 560 - Assignment 6:
Spell Checker

Date Assigned: Fri. Oct 31, 2014

Completion Date: Fri. Nov 7, 2014 (midnight)

Software issues:

If you feel there are mistakes in this assignment, check the web page and Sakai for corrections, and report them to us if they have not been made.

Description

You will write a program that checks a file (input_text.txt) for misspelled words, prints the number of misspelled words and the frequency that the misspelled words occur within the file (input_text.txt), and underlines the misspelled words.

Requirements

This program requires you to use:

- The file open function to read in text from a file.
- The file open function to append text to a file.
- The print function to output the results.
- At least one loop
- Accessing individual characters within a string

Assignment Specification

Write a program that:

a) reads the contents of file (input.txt) into a list
b) reads the contents of file (dictionary.txt) into a list
c) determines if each word in input.txt is in dictionary.txt.
d) prints each misspelled word and the number of times the word is misspelled.

e) prints the text to a file named spelling_correction.html with misspelled words underlined and bold. To see view spelling_correction.html, open the file in a browser (Chrome, Firefox, Internet Explorer)

The location of input.txt is:

http://ils.unc.edu/courses/2014_fall/inls560_001/assignments/input.txt

The location of dictionary.txt is
http://ils.unc.edu/courses/2014_fall/inls560_001/assignments/dictionary.txt

To print the html file copy and paste the following functions into PyCharm. (make sure you indent the code correctly).

```
def print_header():
        output_file = open("spelling_correction.html", "a")
        output_file.writelines("<html><head><title>Correction</title></head><body><br />")

def print_footer():
          output_file = open("spelling_correction.html", "a")
          output_file.writelines("</body></html>")
```

Before you print the text to the html file call print_header(). Then print the text to the html file with misspellings bolded and underlined. Finally, call print_footer().

The following code is an example of how to underline and bold text.

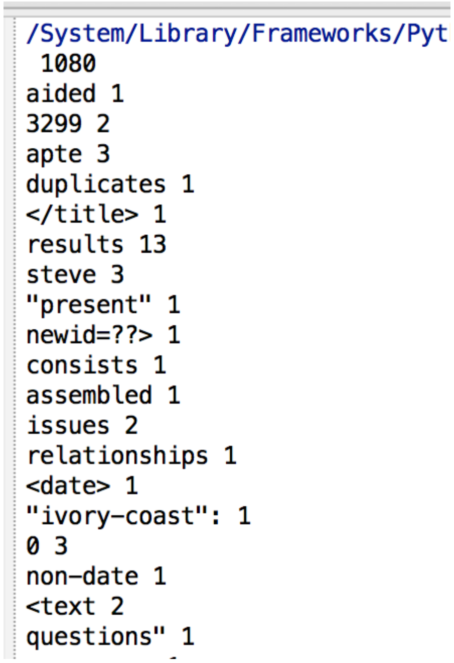
```
output_file.writelines("<u><b>" + word + "</b></u> ")
```

output_file.writelines, will write text to a file. The function writelines takes a string as a parameter. The text, "<u><b>", means start underline and bold, the plus symbol means concatenate the next string. In our case, the next string is word, which is a word from the input.txt file. "</b></u> means end underline and bold.

Be sure to comment your code.

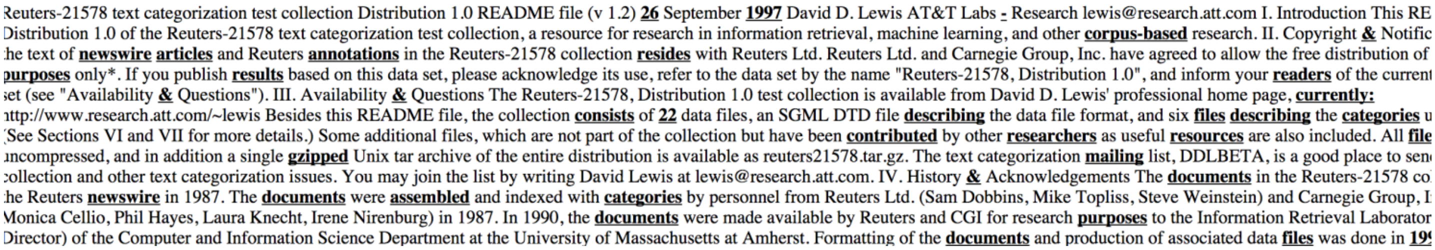
Sample Interactions

Example



```
/System/Library/Frameworks/Pyt
 1080
aided 1
3299 2
apte 3
duplicates 1
</title> 1
results 13
steve 3
"present" 1
newid=??> 1
consists 1
assembled 1
issues 2
relationships 1
<date> 1
"ivory-coast": 1
0 3
non-date 1
<text 2
questions" 1
```

Example of spelling_correction.html:



Reuters-21578 text categorization test collection Distribution 1.0 README file (v 1.2) **26** September **1997** David D. Lewis AT&T Labs **-** Research lewis@research.att.com I. Introduction This RE
Distribution 1.0 of the Reuters-21578 text categorization test collection, a resource for research in information retrieval, machine learning, and other **corpus-based** research. II. Copyright **&** Notific
he text of **newswire** **articles** and Reuters **annotations** in the Reuters-21578 collection **resides** with Reuters Ltd. Reuters Ltd. and Carnegie Group, Inc. have agreed to allow the free distribution of
purposes only*. If you publish **results** based on this data set, please acknowledge its use, refer to the data set by the name "Reuters-21578, Distribution 1.0", and inform your **readers** of the current
set (see "Availability **&** Questions"). III. Availability **&** Questions The Reuters-21578, Distribution 1.0 test collection is available from David D. Lewis' professional home page, **currently:**
http://www.research.att.com/~lewis Besides this README file, the collection **consists** of **22** data files, an SGML DTD file **describing** the data file format, and six **files** **describing** the **categories** u
(See Sections VI and VII for more details.) Some additional files, which are not part of the collection but have been **contributed** by other **researchers** as useful **resources** are also included. All **file**
uncompressed, and in addition a single **gzipped** Unix tar archive of the entire distribution is available as reuters21578.tar.gz. The text categorization **mailing** list, DDLBETA, is a good place to sen
collection and other text categorization issues. You may join the list by writing David Lewis at lewis@research.att.com. IV. History **&** Acknowledgements The **documents** in the Reuters-21578 co
he Reuters **newswire** in 1987. The **documents** were **assembled** and indexed with **categories** by personnel from Reuters Ltd. (Sam Dobbins, Mike Topliss, Steve Weinstein) and Carnegie Group, I
Monica Cellio, Phil Hayes, Laura Knecht, Irene Nirenburg) in 1987. In 1990, the **documents** were made available by Reuters and CGI for research **purposes** to the Information Retrieval Laborator
Director) of the Computer and Information Science Department at the University of Massachusetts at Amherst. Formatting of the **documents** and production of associated data **files** was done in **19**

Use the Pycharm Community Edition IDE to develop and execute the code.

Grading

Programs will be graded based on whether they display the correct output, the correct logic, and style.

In this assignment style means, make variable names meaningful. Do not create one letter variable names or variable names that do not have anything to with the assignment. The program must not only print the correct values, but the code must actually perform the correct operations.

Getting Help

If you have trouble, please post a question on Piazza before contacting me. Before posing a question, please check if this question has been asked before. This will reduce post clutter and reduce our burden. Repeat questions will be ignored by the instructors.

Piazza allows anyone to respond. So if you see a question that you think you can respond to, please do so, as that will reduce our burden and help you "teach" your fellow students.

Good luck!