21704985-cleart01_2270

Overall: 20/20

Testing: 12/ 12
Style: 4/4
Efficiency: 4/4

Excellent work

PROJECT CODE

```
# CITS1401: Computational Thinking with Pyth
# Project 2

# Name: Thomas Cleary
# Student Number: 21704985


import math


# Takes a filename and trys to open it.
# Returns None if there is an error opening the file.
# Else returns String of the contents of the file.
def read_file(file_name):

    try:
        text_file = open(file_name, "r")

    except FileNotFoundError:
        error("File Not Found")
        return None

    except:
        error("Unknown File Error Occurred")
        return None

    text = text_file.read().strip("\n") + " "
    text_file.close()
    return text


# Takes String of a text file and Boolean normalize
# Returns None if there is not atleast 1 complete sentence in the text.
# Else returns the Profile of the text as a dictionary.
def create_profile(text, normalize):

    counted = ["also", "although", "and", "as", "because", "before", "but",
               "for", "if", "nor","of", "or", "since", "that","though",
               "until", "when", "whenever", "whereas","which", "while",
               "yet", ",", ";", "'", "-", "sents_per_para", "words_per_sent"]

    profile = {}
    for item in counted:
        profile[item] = 0

    text = list(text)
    num_sentences = 0
    index = 0

    for char in text:
        remove = False

        # Removes extra new line characters so paragraph spaces are
        # only ever '\n\n'
        if char == "\n":
            if text[index + 2] == "\n":
                remove = True


        elif char in [".", "?", "!"]:
            if text[index+1] in [" ", "\t", "\n", '"', "'"]:
                num_sentences += 1
            remove = True

        elif char in [",", ";"]:
            profile[char] += 1
            remove = True

```

```
        elif char in ["’", "-"]:
            if (text[index-1] + text[index+1]).isalnum():
                profile[char] += 1
            else:
                remove = True

        elif char in ['"', "#", "$", "%", "&", "(", ")", "*", "+", "/", ":",
                      "<", "=", ">", "?", "@", "[", "\\", "]", "^", "_", "‘",
                      "{", "|", "}", "~"]:

            remove = True

        if remove:
            text[index] = " "

        index += 1

    if num_sentences == 0:
        error("File Does Not Contain A Complete Sentence.")
        return None

    clean_text = "".join(text)
    num_paragraphs = clean_text.count("\n\n") + 1

    words = clean_text.split()

    num_words = len(words)

    profile["sents_per_para"] = num_sentences / num_paragraphs
    profile["words_per_sent"] = num_words     / num_sentences

    for word in words:
        word = word.lower()
        if word in counted:
            profile[word] += 1

    if  normalize:
        normalize_profile(profile, num_sentences)

    return profile


# Takes a text file's profile and the number of sentences in the file.
# Returns a normalized version of the profile.
def normalize_profile(profile, num_sentences):
    for key in profile:
        if key in ["sents_per_para", "words_per_sent"]:
            continue

        profile[key] = profile[key] / num_sentences


# Takes 2 profiles of text files
# Returns the overall distance between the values in each profile.
def profile_distance(profile1, profile2):
    total = 0

    for key in profile1.keys():
        total += (profile1[key] - profile2[key]) ** 2

    distance = math.sqrt(total)

    return distance


# Takes a profile and the corresponding file name
# Displays a listing of the values within that profile.
def display_listing(profile, filename):
    keys = profile.keys()

```

21704985-cleart01_2270

```
139      title = "Profile of " + filename
140
141      print("{0}".format(title))
142      print("-" * len(title))
143
144      for key in keys:
145          print("{0}\t\t{1:>9.4f}".format(key, profile[key]))
146
147
148  # Returns a customer error message.
149  def error(message):
150      if message == None:
151          print("Program ended.")
152      else:
153          print("Error: " + message + "\nEnding program...")
154
155
156  def main(textfile1, arg2, normalize=False):
157
158      if type(normalize) != bool:
159          print("Non boolean value passed to argument - normalize.")
160          print("Data will not be normalized.\n")
161          normalize = False
162
163      profile1_text = read_file(textfile1)
164
165      if profile1_text == None:
166          error(profile1_text)
167          return
168
169      profile1 = create_profile(profile1_text, normalize)
170
171      if profile1 == None:
172          error(profile1)
173          return
174
175      if arg2.lower() != "listing":
176          profile2_text = read_file(arg2)
177
178          if profile2_text == None:
179              error(profile2_text)
180              return
181
182          profile2 = create_profile(profile2_text, normalize)
183
184          if profile2 == None:
185              error(profile2)
186              return
187
188          distance = profile_distance(profile1, profile2)
189          print("The distance between the 2 texts is: {0:.4f}".format(distance))
190
191      else:
192          display_listing(profile1, textfile1)
193

-------------------------------------

BEGINNING OF END TESTING
Correct inputs
Test 0: Corpus 1 sample1, listing, not normalised Corpus 2 or 'listing' : sample1.txt nor
malise: listing                                                          1/1

Profile of _corpus1
-------------------
also               1.0000
although                   0.0000
and               13.0000
as                 8.0000
because            0.0000
```

21704985-cleart01_2270

```
before              0.0000
but                 2.0000
for                 0.0000
if                  0.0000
nor                 0.0000
of                  8.0000
or                  0.0000
since               1.0000
that                6.0000
though              0.0000
until               1.0000
when                0.0000
whenever                    0.0000
whereas             0.0000
which               0.0000
while               0.0000
yet                 0.0000
,                  27.0000
;                   5.0000
'                   1.0000
-                  10.0000
sents_per_para              6.0000
words_per_sent             26.7500


Execution Times - User: 0.00 Sys: 0.00


---------------------------
Test 1: Corpus 1 sample2, listing, normalised Corpus 2 or 'listing' : sample2.txt normali
se: listing                                1/1

Profile of _corpus1
-------------------
also                0.0000
although                    0.0000
and                 1.2857
as                  0.0952
because             0.0000
before              0.0952
but                 0.1905
for                 0.0952
if                  0.0952
nor                 0.0000
of                  0.6190
or                  0.0952
since               0.0000
that                0.4762
though              0.0952
until               0.0000
when                0.1429
whenever                    0.0000
whereas             0.0000
which               0.0000
while               0.0000
yet                 0.0000
,                   1.9524
;                   0.1429
'                   0.8095
-                   0.0476
sents_per_para              1.7500
words_per_sent             25.4286


Execution Times - User: 0.00 Sys: 0.00


---------------------------
Test 2: Corpus 1 sample3, listing, third arg missing (un-normalised) Corpus 2 or 'listing
' : sample3.txt normalise: listing
```

```
Profile of _corpus1
-------------------
also                0.0000
although                    0.0000
and                 4.0000
as                  3.0000
because             1.0000
before              0.0000
but                 0.0000
for                 2.0000
if                  0.0000
nor                 0.0000
of                  4.0000
or                  0.0000
since               0.0000
that                1.0000
though              0.0000
until               0.0000
when                0.0000
whenever                    0.0000
whereas             0.0000
which               1.0000
while               0.0000
yet                 0.0000
,                  11.0000
;                   0.0000
’                   3.0000
-                   5.0000
sents_per_para              3.0000
words_per_sent             23.3333


Execution Times - User: 0.00 Sys: 0.00


----------------------------
Test 3: Corpus 1 sample4, hyphenated words, listing, un-normalised Corpus 2 or ’listing’
: sample4.txt normalise: listing
```



```
Profile of _corpus1
-------------------
also                0.0000
although                    0.0000
and                 1.0000
as                  1.0000
because             0.0000
before              0.0000
but                 0.0000
for                 1.0000
if                  1.0000
nor                 0.0000
of                  0.0000
or                  0.0000
since               0.0000
that                0.0000
though              0.0000
until               0.0000
when                0.0000
whenever                    0.0000
whereas             0.0000
which               0.0000
while               0.0000
yet                 0.0000
,                   9.0000
;                   0.0000
’                   4.0000
-                   5.0000
sents_per_para              2.0000
words_per_sent              6.7857
```

21704985-cleart01_2270

Execution Times - User: 0.00 Sys: 0.00

Test 4: Corpus 1 sample7, multiple lines between pars, listing, un-normalised Corpus 2 or 'listing' : sample7.txt normalise: listing

Profile of _corpus1 1/1

also	0.0000
although	0.0000
and	2.0000
as	1.0000
because	0.0000
before	0.0000
but	1.0000
for	0.0000
if	1.0000
nor	0.0000
of	1.0000
or	1.0000
since	0.0000
that	0.0000
though	0.0000
until	0.0000
when	0.0000
whenever	0.0000
whereas	0.0000
which	1.0000
while	0.0000
yet	0.0000
,	3.0000
;	1.0000
'	1.0000
-	0.0000
sents_per_para	0.5000
words_per_sent	25.3333

Execution Times - User: 0.00 Sys: 0.00

Test 5: Corpus 1 King James Bible, long full text, listing, un-normalised Corpus 2 or 'listing' : KingJamesBIble.txt normalise: listing

Profile of _corpus1 1/1

also	1769.0000
although	16.0000
and	51696.0000
as	3520.0000
because	1209.0000
before	1796.0000
but	3992.0000
for	8971.0000
if	1595.0000
nor	755.0000
of	34670.0000
or	1122.0000
since	70.0000
that	12912.0000
though	233.0000
until	366.0000
when	2834.0000
whenever	0.0000
whereas	33.0000
which	4413.0000

21704985-cleart01_2270

```
while            214.0000
yet              683.0000
,              70573.0000
;              10139.0000
'              1790.0000
-                21.0000
sents_per_para             1.2097
words_per_sent            28.6301


Execution Times - User: 4.55 Sys: 0.04


---------------------------
Test 6: Corpus 1 sample1 cf sample1, normalised Corpus 2 or 'listing' : sample1.txt norma
lise: sample1.txt
                                                    1/1
The distance between the 2 texts is: 0.0000


Execution Times - User: 0.00 Sys: 0.00


---------------------------
Test 7: Corpus 1 sample2 cf sample6, un normalised Corpus 2 or 'listing' : sample2.txt no
rmalise: sample6.txt
                                                    1/1
The distance between the 2 texts is: 43.2146


Execution Times - User: 0.00 Sys: 0.00


---------------------------
Test 8: Corpus 1 King James Bible cf Bleak House normalised Corpus 2 or 'listing' : KingJ
amesBIble.txt normalise: Bleak_House.txt
                                                    1/1
The distance between the 2 texts is: 14.3236


Execution Times - User: 6.64 Sys: 0.06


---------------------------

Error State Handling
Test 9: Corpus 1 Bogus second corpus Corpus 2 or 'listing' : sample1.txt normalise: Missi
ng
                                        1/1
Error: File Not Found
Ending program...
Program ended.


Execution Times - User: 0.00 Sys: 0.00


---------------------------
Test 10: Corpus 1 Empty file for second corpus Corpus 2 or 'listing' : sample2.txt normal
ise: empty_file                                 1/1

Error: File Does Not Contain A Complete Sentence.
Ending program...
Program ended.


Execution Times - User: 0.00 Sys: 0.00


---------------------------
```

21704985-cleart01_2270

```
Test 11: Corpus 1 File with only white space Corpus 2 or 'listing' : blank_lines.txt norm
alise: listing

Error: File Does Not Contain A Complete Sentence.
Ending program...
Program ended.


Execution Times - User: 0.00 Sys: 0.00


----------------------------
END OF END TESTING
```