Documentation

Marcin Michnik, Adam Dochtorowicz

June 2019

Contents

1 Introduction

The programme is a letter encoder. It intakes a text file and outputs a list of letter sequences according to specific rules.

2 The menu of the programme

At the beginning, the programme prints the following message: "Prosze podać nazwe pliku (np. 'powiedzenia.txt' albo 'expressions.txt'): " Then, there are four options to choose from:

1. Pierwsze litery słów (First letters of the words)
2. Ostatnie litery słów (Last letters of the words)
3. Pierwsze litery posortowanych słów (First letters of the sorted words)
4. Ostatnie litery posortowanych słów (Last letters of the sorted words)

```
import matplotlib.pyplot as plt
#The menu
#Input the name of the text file
print("Prosze podać nazwe pliku (np. 'powiedzenia.txt' albo 'Expressions.txt'): ")
nazwa=input()
#The programme opens the file and converts it to a list
tekst1=open(nazwa,'r',encoding="utf8")
tekst=list(tekst1)
print(tekst)
#Choose one of the options
print('''Prosze podać opcje programu(od 1 do 4): \n
1 - Pierwsze litery słów \n
2 - Ostatnie litery słów \n
3 - Pierwsze litery posortowanych słów \n
4 - Ostatnie litery posortowanych słów
''')
opcja = int(input())

```

The options to choose from are written in the following way (Only the first option is shown because the other are similar):

```
197 if opcja==1:
198     print("Prawdopodobieństwo wystąpienia sekwencji z pierwszych liter nieposortowanych słów:")
199     for segment1 in probability_of_occurence1(ul1):
200         print(*segment1, end='\n')
201     list_ready = []
202     for i in range(0, 10):
203         x = unique1(list1)
204         list_ready.append(x[i])
205 
206     height = []
207     for i in range(0, 10):
208         y = probability_of_occurence1(ul1)
209         height.append(y[i][1])
210 
211     plt.bar(list_ready, height)
212     plt.savefig("figure1.png")
213     plt.show()
```

3 The input

The programme accepts a file in the ".txt" format and 'utf-8' encoding. Two files were prepared beforehand: 'powiedzenia.txt', 'expressions.txt'. The user is supposed to enter the name of a file containing expressions in separate lines.

```
A blessing in disguise
A dime a dozen
Beat around the bush
Better late than never
Bite the bullet
Break a leg
Call it a day
Cut somebody some slack
Cutting corners
Easy does it
Get out of hand
Get something out of your system
Get your act together
Give someone the benefit of the doubt
Go back to the drawing board
Hang in there
Hit the sack
It's not rocket science
Let someone off the hook
Make a long story short
Miss the boat
No pain no gain
On the ball
Pull someone's leg
Pull yourself together
So far so good
Speak of the devil
That's the last straw
The best of both worlds
Time flies when you're having fun
To get bent out of shape
To make matters worse
Under the weather
We'll cross that bridge when we come to it
```

4 The functions

There are four functions, which all return a list in the form of the results of the according options. The lists consist of individual characters and backslash-n as separators.

```
21  def First_Letters(tekst):
22      lista_wyników1 = []
23      for element1 in tekst:
24          tekst_individual1 = element1.split()
25          for i in range(len(tekst_individual1)):
26              lista_wyników1.append(tekst_individual1[i][0])
27          lista_wyników1.append('\n')
28      return lista_wyników1
```

Then, the characters are concatenated to full strings.

```
34  def conc1(flt):
35      result1 = ''
36      for element1 in flt:
37          result1 += str(element1)
38      return result1
```

The individual strings are put in list in another function.

```
44  def unique1(list1):
45      unique_list1 = []
46      for x1 in list1:
47          if x1 not in unique_list1:
48              unique_list1.append(x1)
49      del unique_list1[-1]
50      return unique_list1
```

The probability of appearance is counted for every individual string.

```
55  def probability_of_occurence1(ul1):
56      probability_list1 = []
57      for unique_word1 in ul1:
58          occurence_count1 = list1.count(unique_word1)
59          probability1 = unique_word1, occurence_count1 / len(list1)
60          probability_list1.append(probability1)
61      return probability_list1
```

5 The output

The programme prints the probability of appearance of every unique string.

```
Program ×
Pigtsw 0.009900990099009901
Pjwdk 0.009900990099009901
Psnd 0.009900990099009901
Pkzp 0.009900990099009901
Pln 0.009900990099009901
Pwok 0.009900990099009901
Pkdw 0.009900990099009901
PknM 0.009900990099009901
```

Finally, the 'pyplot' module makes a bar chart for the ten first instances and saves the file in '.png' format.