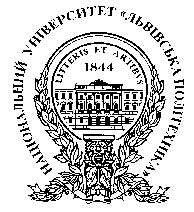
МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ

Національний Університет “Львівська політехніка”



Лабораторна робота № 5

***ВИВЧЕННЯ БІБЛІОТЕКИ ПРИКЛАДНИХ ПРОГРАМ NLTK, ДЛЯ ОПРАЦЮВАННЯ ТЕКСТІВ ПРИРОДНОЮ МОВОЮ.***

***ПОЧАТКОВА ОБРОБКА ТЕКСТІВ ПРИРОДНОЮ МОВОЮ.***

Виконав:

студент групи ПРЛс-11

Форманюк А.А.

Прийняв:

асистент

Дупак Б.П.

Львів 2015

**МЕТА РОБОТИ**

* Вивчення основ програмування на мові *Python*.
* Вивчення методів роботи з файлами на локальних дисках та з Інтернету.
* Використання Юнікоду при обробці текстів.
* Нормалізація текстів, стемінг, лематизація та сегментація.

**КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ**

* 1. ***Електронні книжки***

Частина електронних книжок з Project Gutenberg розповсюджується разом з NLTK у вигляді корпуса текстів. Для використання інших текстів з цього проекту можна переглянути каталог 25000 електронних книжок за адресою <http://www.gutenberg.org/catalog/> та встановити адресу (URL) потрібного текстового файлу в ASCII кодуванні. 90% текстів в Project Gutenberg є англійською мовою, але він включає також тексти більше ніж 50-ма іншими мовами (каталонська, китайська, датська, фінська, французька, німецька, італійська, португальська, іспанська…).

Текст за номером 2554 це переклад англійською *Crime and Punishment(Злочин і кара)*, і отримати доступ до тексту можна наступним чином:

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | |  | **>>> from urllib import urlopen**  **>>> url = "http://www.gutenberg.org/files/2554/2554.txt"**  **>>> raw = urlopen(url).read()**  **>>> type(raw)**  **<type 'str'>**  **>>> len(raw)**  **1176831**  **>>> raw[:75]**  **'The Project Gutenberg EBook of Crime and Punishment, by Fyodor Dostoevsky\r\n'** | |

Виконання read() займає певний час протягом якого відбувається завантаження цієї великої книжки. При використанні проксі сервера для доступу до Інтернету, при необхідності, його параметри потрібно вказати:

|  |  |
| --- | --- |
|  | **>>> proxies = {'http': 'http://www.someproxy.com:3128'}**  **>>> raw = urlopen(url, proxies=proxies).read()** |

Текст книжки збережений як значення змінної raw . Змінна raw містить стрічку довжиною 1,176,831 символів. (Перевірити тип змінної можна скориставшись type(raw).) Стрічка яка відповідає вмісту книжки містить багато не цікавої для аналізу інформації: пробіли, пусті стрічки, межі стрічки. Символи \r and \n , які є в тексті, це символи переводу каретки та початку нового рядка. Для подальшої роботи з текстом потрібно розділити текст на окремі слова та виділити розділові знаки. Такий процес називають токенізацією. При використанні програми токенізації з NLTK отримуємо список слів та розділових знаків.

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | |  | **>>> tokens = nltk.word\_tokenize(raw)**  **>>> type(tokens)**  **<type 'list'>**  **>>> len(tokens)**  **255809**  **>>> tokens[:10]**  **['The', 'Project', 'Gutenberg', 'EBook', 'of', 'Crime', 'and', 'Punishment', ',', 'by']** | |

Бібліотека NLTK використовувалась тільки на етапі токенізації і не використовувалась при доступі за адресою в Інтернеті та при читанні стрічки. Для подальшої роботи список перетворюється в NLTK текст і над ним можна здійснювати різноманітні операції:

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | |  | **>>> text = nltk.Text(tokens)**  **>>> type(text)**  **<type 'nltk.text.Text'>**  **>>> text[1020:1060]**  **['CHAPTER', 'I', 'On', 'an', 'exceptionally', 'hot', 'evening', 'early', 'in',**  **'July', 'a', 'young', 'man', 'came', 'out', 'of', 'the', 'garret', 'in',**  **'which', 'he', 'lodged', 'in', 'S', '.', 'Place', 'and', 'walked', 'slowly',**  **',', 'as', 'though', 'in', 'hesitation', ',', 'towards', 'K', '.', 'bridge', '.']**  **>>> text.collocations()**  **Katerina Ivanovna; Pulcheria Alexandrovna; Avdotya Romanovna; Pyotr**  **Petrovitch; Project Gutenberg; Marfa Petrovna; Rodion Romanovitch;**  **Sofya Semyonovna; Nikodim Fomitch; did not; Hay Market; Andrey**  **Semyonovitch; old woman; Literary Archive; Dmitri Prokofitch; great**  **deal; United States; Praskovya Pavlovna; Porfiry Petrovitch; ear rings** | |

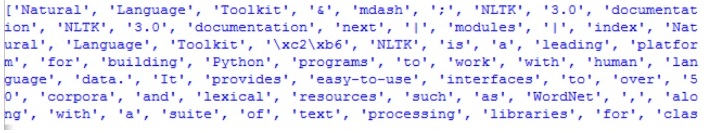
В побудованих колокаціях зустрічається Project Gutenberg, словосполучення, яке не міститься в тексті книжки. Завантажений текст з сайту містить метатекстову розмітку (інформацію про автора, про текст, про людей які готували електронний варіант та ін.). Ця інформація може бути, як на початку тексту так і в його кінці . Для роботи власне з текстом книжки потрібно в ручному режимі знайти межі цих додаткових даних і за допомогою зрізів доступитися до тексту.

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | |  | **>>> raw.find("PART I")**  **5303**  **>>> raw.rfind("End of Project Gutenberg's Crime")**  **1157681**  **>>> raw = raw[5303:1157681]**  **>>> raw.find("PART I")**  **0** | |

Методи find() та rfind() ("пошук з кінця") допомагають знайти потрібні індекси для їх подальшого використання в зрізах. Значення зрізу переприсвоюється змінній raw.

1. Напишіть функцію, яка приймає адресу URL, як аргумент, і повертає те що міститься за цією адресою з видаленням HTML розмітки. Використовувати urllib.urlopen для доступу до контенту наступним чином raw\_contents = urllib.urlopen('http://www.nltk.org/').read().

import urllib  
from urllib import urlopen  
url = '[http://www.gutenberg.org/files/2554/2554.txt](https://vk.com/away.php?utf=1&to=http%3A%2F%2Fwww.gutenberg.org%2Ffiles%2F2554%2F2554.txt)'  
raw\_contents = urllib.urlopen(url).read()  
raw\_contents[:40]  
import nltk  
raw = nltk.clean\_html(raw\_contents)  
tokens = nltk.word\_tokenize(raw)  
print tokens



1. Збережіть деякий текст у файлі corpus.txt. Визначити функцію load(f) для читання файлу, назва якого є її аргументом і повертає стрічку, яка містить текст з файлу.

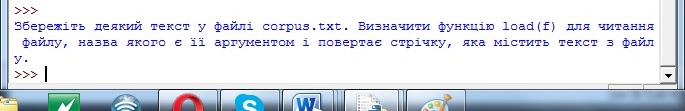
def load(f):

f = open(f)

raw = f.read()

return raw

print load("C://Python27/corpus.txt")



1. Перепишіть наступний цикл як list comprehension:

**>>> sent = ['The', 'dog', 'gave', 'John', 'the', 'newspaper']**

**>>> result = []**

**>>> for word in sent:**

**... word\_len = (word, len(word))**

**... result.append(word\_len)**

**>>> result**

**[('The', 3), ('dog', 3), ('gave', 4), ('John', 4), ('the', 3), ('newspaper', 9)]**

sent = ['The', 'dog', 'gave', 'John', 'the', 'newspaper']

result = []

for word in sent:

word\_len = (word, len(word))

result = [(word, len(word)) for word in sent]

print result

[('The', 3), ('dog', 3), ('gave', 4), ('John', 4), ('the', 3), ('newspaper', 9)]

1. Перевірити різницю між стрічками і цілим виконавши наступні дії: "3" \* 7 та 3 \* 7. Спробуйте здійснити конвертування між стрічками і цілими використавши int("3") та str(3).

>>> a = "3" \* 7

>>> print a

3333333

>>> b = 3 \* 7

>>> print b

21

>>> c = int("3")

>>> d = c \* 7

>>> print d

21

>>> e = str(3)

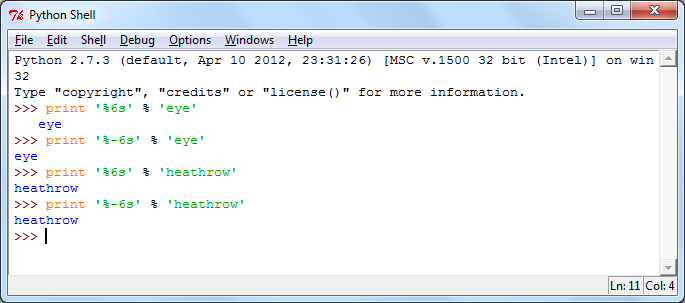
>>> f = e \* 7

>>> print f

3333333

Використовуючи int можна стрічку переконвертувати у число. Зворотня операція виконується за допомогою str.

1. Що станеться, коли стрічки форматування %6s та %-6s використовується для відображення стрічки довшої ніж 6 символів?



1. Створіть файл, який буде містити слова та їх частоту записані в окремих рядках через пробіл ( fuzzy 53). Прочитайте цей файл використовуючи open(filename).readlines(). Розділіть кожну стрічку на дві частини використовуючи split(), і перетворіть число в ціле значення використовуючи int(). Результат повинен бути у вигляді списку: [['fuzzy', 53], ...].

from \_\_future\_\_ import division

import nltk, re, pprint

q = open("C://Python27/corpus.txt").readlines()

print q

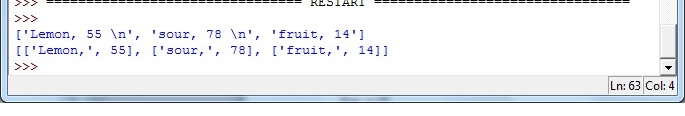
text = q

list = []

for i in text:

list.append([i.split()[0], int(i.split()[1])])

print list



1. Прочитайте деякий текст з корпуса, здійсніть його токенізацію і збережіть у список всі wh-слова, які в ньому зустрічаються.

import nltk

from nltk.corpus import gutenberg

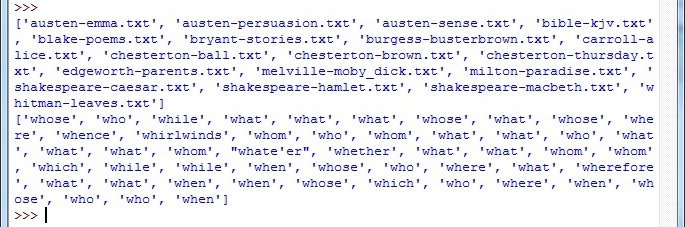
print gutenberg.fileids()

q = gutenberg.raw('milton-paradise.txt')

tokens = nltk.word\_tokenize(q)

wh\_words = [w.lower() for w in tokens if w.startswith('wh')]

print wh\_words[:50]



1. Міра оцінки читабельності використовується для оцінки складності тексту для читання. Нехай, μw - середня кількість літер у слові, та μs – середнє значення кількості слів у реченні в певному тексті. Automated Readability Index (ARI) тексту визначається згідно виразу: 4.71 μw + 0.5 μs - 21.43. Визначити значення ARI для різних частин корпуса Brown Corpus, включаючи частину f (popular lore) та j (learned). Використовуйте nltk.corpus.brown.words() для знаходження послідовності слів та nltk.corpus.brown.sents() для знаходження послідовності речень.

import nltk

from nltk.corpus import brown

print brown.categories()

num\_chars = len(brown.raw(categories='lore'))

num\_words = len(brown.words(categories='lore'))

num\_sents = len(brown.sents(categories='lore'))

Rw = int(num\_chars/num\_words)

print Rw

Rs = int(num\_words/num\_sents)

print Rs

ARI = 4.7\*Rw+0.5\*Rs-21.43

print ARI

num\_chars = len(brown.raw(categories='learned'))

num\_words = len(brown.words(categories='learned'))

num\_sents = len(brown.sents(categories='learned'))

Rw = int(num\_chars/num\_words)

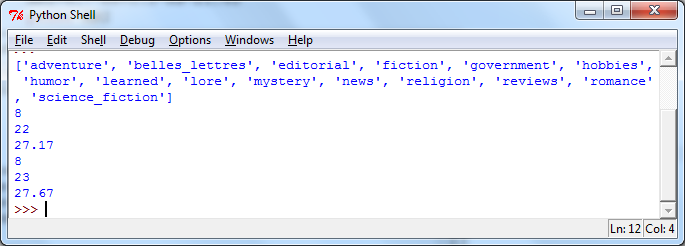
print Rw

Rs = int(num\_words/num\_sents)

print Rs

ARI = 4.7\*Rw+0.5\*Rs-21.43

print ARI



1. Перепишіть наступний цикл, як list comprehension:

**>>> words = ['attribution', 'confabulation', 'elocution',**

**... 'sequoia', 'tenacious', 'unidirectional']**

**>>> vsequences = set()**

**>>> for word in words:**

**... vowels = []**

**... for char in word:**

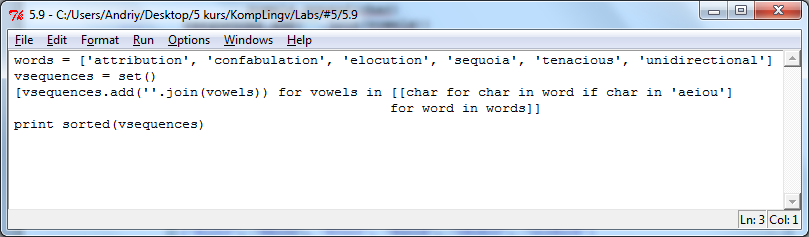
**... if char in 'aeiou':**

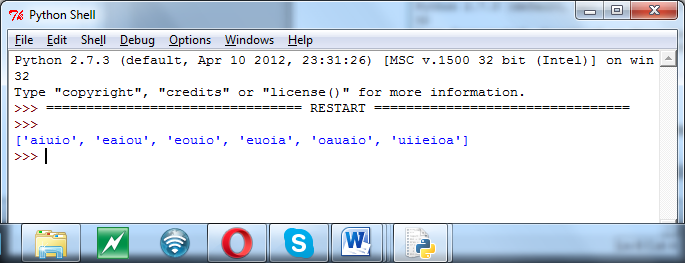
**... vowels.append(char)**

**... vsequences.add(''.join(vowels))**

**>>> sorted(vsequences)**

**['aiuio', 'eaiou', 'eouio', 'euoia', 'oauaio', 'uiieioa']**





**Висновок:** Під час виконання даної лабораторної роботи були вивчені основи програмування мови Python, а саме методи роботи з файлами на локальних дисках та з Інтернету. Також я ознайомився з використанням Юнікоду при обробці тексту.