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
Іваськів М.Є.

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

1.1. 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:

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:

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.

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.

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

1.2. Робота з HTML файлами.

Текст, який вивели на екран містить HTML розмітку (мета теги, JavaScript, форми , таблиці). Вилучення тексту з HTML файлу доволі розповсюджена задача, яка в NLTK вирішується за допомогою функції `nltk.clean_html()`. Ця функція обробляє HTML стрічку і повертає текст у вигляді зручному для подальшої обробки (токенізації).

Видалення іншої небажаної інформації проводиться в ручному режимі, аналогічно до попередніх прикладів з електронними книжками.

1.3. Обробка RSS стрічок

Блогосфера важливе джерело текстів, як формальних так і не формальних. З допомогою бібліотеки Python *Universal Feed Parser*, `http://feedparser.org/`, можна отримати доступ до вмісту блогів, як показано у наступному прикладі:

1.4. Читання локальних файлів.

Для читання локальних файлів необхідно використовувати вбудовану функцію Python `open()` та `read()` метод. Якщо існує файл `document.txt`, то змінній `raw` можна присвоїти його вміст. Для перевірки чи дійсно файл є в потрібній директорії у графічному інтерфейсі IDLE використовується команда *Open* з пункту меню *File.* Можна також перевірити вміст директорії наступним чином.

Інша можлива проблема при читанні текстових файлів – це різні способи маркування нового рядка у файлах різних операційних систем. При виклику функція `open()` може містити другий параметр для контролю відкривання файлу `open('document.txt', 'rU')` („r" файл для читання, "U" universal дозволяє ігнорувати різні способи, які використовуються для маркування нового рядка).

Для читання вмісту файлу можна використати багато різних методів. Метод *read()* , використаний до об'єкту файл (*f*), читає вміст файлу і представляє його стрічкою. Символ `'\n'` – це символ нового рядка.

Файл можна читати стрічка за стрічкою, використовуючи *for*-цикл і використовувати зріз *[:-1]* або метод strip() для видалення символів нового рядка.

За допомогою цих методів також можна доступитися і до файлів з корпусів, які розповсюджуються з NLTK. Потрібно використати `nltk.data.find()` для одержання шляху до будь-якого файлу корпуса, а далі відкривати та читати файл, як показано.

1.5. Ввід тексту з клавіатури.

Для вводу тексту з клавіатури (при взаємодії користувача з програмою) потрібно використати функцію `raw_input()`. Після збереження введеного тексту у змінній з ним можна працювати як зі звичайною стрічкою.

Тип об'єкту визначає, які операції можуть бути здійснені з цим об'єктом. Наприклад, можна додавати елементи до списку але не можна до стрічки. Стрічки та списки можна поєднувати з іншими стрічками та списками, але не можна поєднувати стрічки зі списками.

ВИКОНАННЯ ЗАВДАНЬ

Завдання 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'
raw_contents = urllib.urlopen(url).read()
raw_contents[:50]
import nltk
raw = nltk.clean_html(raw_contents)
tokens = nltk.word_tokenize(raw)
print tokens
```

```
['Natural', 'Language', 'Toolkit', '&', 'mdash', ';', 'NLTK', '3.0', 'documentat
ion', 'NLTK', '3.0', 'documentation', 'next', '|', 'modules', '|', 'index', 'Nat
ural', 'Language', 'Toolkit', '\xc2\xb6', 'NLTK', 'is', 'a', 'leading', 'platfor
m', 'for', 'building', 'Python', 'programs', 'to', 'work', 'with', 'human', 'lan
guage', 'data.', 'It', 'provides', 'easy-to-use', 'interfaces', 'to', 'over', '5
0', 'corpora', 'and', 'lexical', 'resources', 'such', 'as', 'WordNet', ',', 'alo
ng', 'with', 'a', 'suite', 'of', 'text', 'processing', 'libraries', 'for', 'clas
```

Завдання 2. 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
```

```
>>> load('corpus.txt')
'Ukraine is the best country in the world\nWould you like to go to Ukraine?\nCom
e visit us'
```

Завдання 3. 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)]
```

List comprehension:

```
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)]
```

Завдання 4. 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).

```
hey = "3" * 7
print hey
bye = 3 * 7
print bye
print "a few changes made: "
s = int("3")
a = s * 7
print a
q = str(3)
b = q * 7
print b
>>>
3333333
21
a few changes made:
21
3333333
```

Завдання 5. Що станеться, коли стрічки форматування %6s та %-6s використовується для відображення стрічки довшої ніж 6 символів?

```
>>> '%6s' % 'cat'
'   cat'
>>> '%-6s' % 'cat'
'cat   '
>>> '%6s' % 'marvelous'
'marvelous'
>>> '%-6s' % 'marvelous'
'marvelous'
```

Завдання 7. 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
p = open('C:\Python27\corp.txt').readlines()
print p
smth = p
list = []
for i in smth:
    list.append([i.split()[0], int(i.split()[1])])

print list
```

```
>>>
['fuzzy 53\n', 'school 5\n', 'unforgivable 30']
[['fuzzy', 53], ['school', 5], ['unforgivable', 30]]
```

Завдання 9. З тексту мовою, яка має гармонію голосних (Hungarian), вилучіть у словах послідовності голосних і створіть частотну таблицю біграмів голосних.

```
from __future__ import division
import nltk, re, pprint
import collections
f = 'Minden emberi leny szabadon szuletik es egyenlo joga van'
a = f.split()
s = set()
for word in a:
        vowels=[]
        for char in word:
                if char in 'ayouiei':
                        vowels.append(char)
                        s.add(''.join(vowels))

print sorted(s)
p = nltk.bigrams(s)
print p
freq = collections.Counter(s)
print freq
```

```
>>>
['a', 'aa', 'aao', 'e', 'ee', 'eei', 'ey', 'eye', 'eyeo', 'i', 'ie', 'o', 'oa',
'u', 'ue', 'uei']
[('a', 'aa'), ('aa', 'uei'), ('uei', 'e'), ('e', 'eye'), ('eye', 'i'), ('i', 'aa
o'), ('aao', 'o'), ('o', 'ee'), ('ee', 'eei'), ('eei', 'oa'), ('oa', 'ue'), ('ue
', 'ey'), ('ey', 'ie'), ('ie', 'u'), ('u', 'eyeo')]
Counter({'a': 1, 'aa': 1, 'uei': 1, 'eye': 1, 'i': 1, 'aao': 1, 'o': 1, 'ee': 1,
 'ey': 1, 'eei': 1, 'oa': 1, 'ue': 1, 'e': 1, 'ie': 1, 'u': 1, 'eyeo': 1})
```

Завдання 11. Здійсніть аналіз числового виразу в наступному реченні з корпуса MedLine: The corresponding free cortisol fractions in these sera were 4.53 +/- 0.15% and 8.16 +/- 0.23%, respectively. Чи можна сказати, що 4.53 +/- 0.15% це три окремих слова? Чи це одне складне слово? Чи це дев'ять слів "four point five three, plus or minus fifteen percent"? Чи це взагалі не можна вважати словом? При вирішенні яких задач потрібно вибирати ту чи іншу відповідь?

```
from __future__ import division
import nltk, re, pprint
f = 'The corresponding free cortisol fractions in these sera were 4.53 +/- 0.15%
print f
pattern = r'\w+|[^\w\s]+'
a = nltk.tokenize.regexp_tokenize(f, pattern)
print a
>>>
The corresponding free cortisol fractions in these sera were 4.53 +/- 0.15% and
8.16 +/- 0.23%, respectively
['The', 'corresponding', 'free', 'cortisol', 'fractions', 'in', 'these', 'sera',
 'were', '4', '.', '53', '+/-', '0', '.', '15', '%', 'and', '8', '.', '16', '+/-
', '0', '.', '23', '%,', 'respectively']
```

Програма розглядає кожне число та символ окремо. У цьому реченні такий аналіз не спрацьовує, тому що розривати «4,53» чи «0,15» не можна – це одне число. Так само не можна відділяти % від числа. Цей числовий вираз потрібно вважати єдиним цілим, неподільним. Проте, якщо аналізувати вираз як слова, то це окремі слова, які представляють одне значення.

Завдання 15. 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']
```

List comprehension:

```
words = ['attribution', 'confabulation', 'elocution','sequoia', 'tenacious', 'un
vsequences = set()
word='attribution'
[char for char in word if char in 'aeiou']
[vsequences.add(''.join(vowels)) for vowels in [char for char in word if char in
print vsequences
[vsequences.add(''.join(vowels)) for vowels in [[char for char in word if char i
print sorted(vsequences)
>>>
set(['a', 'i', 'u', 'o'])
['a', 'aiuio', 'eaiou', 'eouio', 'euoia', 'i', 'o', 'oauaio', 'u', 'uiieioa']
```