МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ "ЛЬВІВСЬКА ПОЛІТЕХНІКА"
ІНСТИТУТ КОМП'ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра "Системи автоматизованого проектування"

Звіт

до лабораторної роботи №5
на тему: ВИВЧЕННЯ БІБЛІОТЕКИ ПРИКЛАДНИХ ПРОГРАМ NLTK, ДЛЯ ОПРАЦЮВАННЯ ТЕКСТІВ ПРИРОДНОЮ МОВОЮ.ПОЧАТКОВА ОБРОБКА ТЕКСТІВ ПРИРОДНОЮ МОВОЮ.
з дисципліни "Комп'ютерна лінгвістика"

Виконала:
студентка групи ПРЛм-11
Гарбуз Л.В.
Прийняв:
викладач
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.

Тексти програм на мові *Python*.

Варіант – 3

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, nltk
from urllib import urlopen
url="http://www.nltk.org/"
raw_contents=urllib.urlopen(url).read()
raw=nltk.clean_html(raw_contents)
tokens=nltk.word_tokenize(raw)
print tokens [:50]
```

```
>>>
['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']
>>>
```

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):
    d=open(f)
    for line in d:
        print line.strip()
print load ('corpus.txt')
```

```
>>>
Many websites will try to tell you which careers offer the best prospects for th
e future. Their choices are usually based on percentage growth statistics for re
cent years, which are a clear indicator of where the numbers of jobs are increas
ing.

However, this does not reflect other concerns such as which careers pay best, wh
ich jobs are easiest to obtain, which need the longest periods of undergraduate
and postgraduate study, and so on. Despite this, some general trends hold true o
n a general level.
None
>>>
```

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)]
```

```
>>> sent = ['the', 'dog', 'gave', 'John','the','newspaper']
>>> [(word, len(word)) for word in sent]
[('the', 3), ('dog', 3), ('gave', 4), ('John', 4), ('the', 3), ('newspaper', 9)]
```

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).

```
>>> '3'* 7
'3333333'
>>> 3*7
21
>>> int('3') * 7
21
>>> str(3) * 7
'3333333'
>>>
```

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

```
>>> '%6s' % 'andandandandand'
'andandandandand'
>>> '%6s' % 'and'
'   and'
>>> '%-6s' % 'andandandandand'
'andandandandand'
>>> '%-6s' % 'and'
'and   '
```

7. Створіть файл, який буде містити слова та їх частоту записані в окремих рядках черезпробіл (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],...].

```
>>> f = open('E:\Lab5\words.txt').readlines()
>>> f
['night 30\n', 'day 21\n', 'room 5\n', 'bag 19\n', 'cinema 21\n', 'film 3\n', 'sweets 15']
>>> a=[]
>>> for i in f:
        b=i.split(' ')
        a+=[[b[0]]+[int(b[1])]]

>>> a
[['night', 30], ['day', 21], ['room', 5], ['bag', 19], ['cinema', 21], ['film', 3], ['sweets', 15]]
>>> for line in f:
        print(line.split())

['night', '30']
['day', '21']
['room', '5']
['bag', '19']
['cinema', '21']
['film', '3']
['sweets', '15']
```

12. 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.words(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
```

```
>>>
['adventure', 'belles_lettres', 'editorial', 'fiction', 'government', 'hobbies',
 'humor', 'learned', 'lore', 'mystery', 'news', 'religion', 'reviews', 'romance'
, 'science_fiction']
8
22
27.17
8
1
16.67
```

13. Використовуючи Porter стемер нормалізуйте будь-який токенізований текст. До того самого тексту застосуйте Lancaster стемер. Результати порівняйте та поясніть.

```
>>> from __future__ import division
>>> import nltk, re, pprint
>>> text = ['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', '.']
>>> porter = nltk.PorterStemmer()
>>> lancaster = nltk.LancasterStemmer()
>>> [porter.stem(t) for t in text]
['CHAPTER', 'I', 'On', 'an', 'except', 'hot', 'even', 'earli', 'in', 'Juli', 'a'
, 'young', 'man', 'came', 'out', 'of', 'the', 'garret', 'in', 'which', 'he', 'lo
dg', 'in', 'S', '.', 'Place', 'and', 'walk', 'slowli', ',', 'as', 'though', 'in'
, 'hesit', ',', 'toward', 'K', '.', 'bridg', '.']
>>> [lancaster.stem(t) for t in text]
['chapt', 'i', 'on', 'an', 'exceiv', 'hot', 'ev', 'ear', 'in', 'july', 'a', 'you
ng', 'man', 'cam', 'out', 'of', 'the', 'garret', 'in', 'which', 'he', 'lodg', 'i
n', 's', '.', 'plac', 'and', 'walk', 'slow', ',', 'as', 'though', 'in', 'hesit',
 ',', 'toward', 'k', '.', 'bridg', '.']
>>>
```

14. Доступіться до текстів ABC Rural News та ABC Science News з корпуса (nltk.corpus.abc). Знайдіть значення для оцінки читабельності текстів (аналогічно до задачі №12). Використовуйте Punkt для поділу тексту на окремі речення.

```
from __future__ import division
import nltk, re, pprint
from nltk.corpus import abc
abc.fileids()
for fileid in abc.fileids():
    ari=(4.71*(len(abc.raw (fileid))/len (abc.words(fileid)))+
         (0.5*(len(abc.words (fileid))/len (abc.sents (fileid)))) - 21.43)
    print ari, fileid


>>>
16.2873798544 rural.txt
16.7019225404 science.txt
>>>
```

Висновок: на цій лабораторній роботі я вивчила основи програмування на мові Python, методи роботи з файлами на локальних дисках та з Інтернету. Використала Юнікод при обробці текстів, нормалізацію текстів, стемінг, лематизацію та сегментацію.