МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”
ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

Кафедра САП

Звіт
Лабораторної роботи №4

ВИВЧЕННЯ БІБЛІОТЕКИ ПРИКЛАДНИХ ПРОГРАМ NLTK, ДЛЯ ОПРАЦЮВАННЯ ТЕКСТІВ ПРИРОДНОЮ МОВОЮ. ДОСТУП ТА РОБОТА З ЛЕКСИЧНИМИ РЕСУРСАМИ.

Виконала:
студентка групи ПРЛм-12
Іваськів М.Є.
Перевірив:
Дупак Б.П.

Львів 2015

1. Мета.

- Вивчення основ програмування на мові *Python*
- Вивчення методів доступу та роботи з лексичним ресурсами.
- Семантичний словник англійської мови WordNet.

2. Короткі теоретичні відомості.

Функція - це програмна конструкція, яку можна викликати з одним або більше вхідними параметрами, і отримувати результат на виході. Визначаємо функцію, використовуючи ключове слово *def* далі потрібно дати назву функції і визначити вхідні параметри, після двокрапки записується тіло функції. Ключове слово *return* використовується для відображення значення, яке ми хочемо отримати на виході функції.

Розробляючи ту чи іншу програму протягом довшого періоду додаючи до неї нові функції і змінюючи існуючі або розробляючи декілька версій однієї програми потрібно зберігати тексти програм в окремих файлах і організовувати доступ до відповідних функцій в цих програмах.

Множина змінних і функцій збережених у файлі називаються в Python – модулем. Множина пов'язаних між собою модулів називають – пакетом.

NLTK розповсюджується з деякими корпусами, які насправді є списками слів. Корпус words це файл з Unix, який використовується для перевірки правопису. В NLTK також включений корпус стоп-слів (незначущі слова). Ці слова часто зустрічаються в текстах, але переважно не мають окремого лексичного значення. Наступний корпус NLTK це корпус імен, об'ємом 8000 одиниць, які поділені на категорії за родами (чоловічим та жіночим).

Більш багатим лінгвістичним ресурсом може бути словник де кожному слову поставлена у відповідність певна інформація. NLTK включає CMU Pronouncing Dictionary американського варіанту англійської , який розроблений для використання в синтезаторах мови.

Словник в NLTK це порівняльний словник (Swadesh wordlists), який містить 200 спільних слів для 24 мов.

WordNet, це семантично орієнтований словник англійської мови, подібний до традиційних тезаурусів але з більш багатою структурою. У *WordNet* слова групуються у набори синонімів – синсети, кожен із своїм визначенням і зв'язками з іншими синсетами.

WordNet дозволяє легко переміщатися між поняттями. Наприклад для поняття *motorcar* ми можемо переглянути поняття, які є більш специфічними (гіпонім).

Аналогічно можна піднятися по ієрархії і переглянути більш широкі поняття ніж motorcar(гіпероніми). Деякі слова мають декілька шляхів вверх, ці слова можуть класифікуватися більш ніж одним способом.

Синсети зв'язані між собою складною мережею лексичних зв'язків. Для певного синсету можна переглянути зв'язки у WordNet і знайти синсети, які з ним зв'язані за змістом. Інформація про семантичні взаємозв'язки між словами цінна при класифікації текстів.

Кожен синсет має один або більше шляхів за яким він зводиться до ключового поняття entity.n.01.

Гіперніми та гіпоніми називають лексичними зв'язками тому що вони пов'язують один син сет з іншим. Ці два зв'язки вказують на рух вверх-вниз в ієрархії «is-a». Інший можливий шлях в ієрархії WordNet це від предмету до його складових (меронім), або до поняття яке містить предмет в собі (голоніми). Наприклад, частини дерева – стовбур, крона та ін. part_meronyms() . Речовина з якого дерево зроблено включає heartwood та sapwood; - substance_meronyms(). Багато дерев утворюють ліс - member_holonyms()

3. Виконання завдань.

Завдання 2. Використовуючи компаративний словник знайти близькі слова для німецької, італійської та англійської мов. Чи можуть отримані результати використовуватися для здійснення перекладу?

Результати можна використовувати для перекладу окремих слів аналізованих мов.

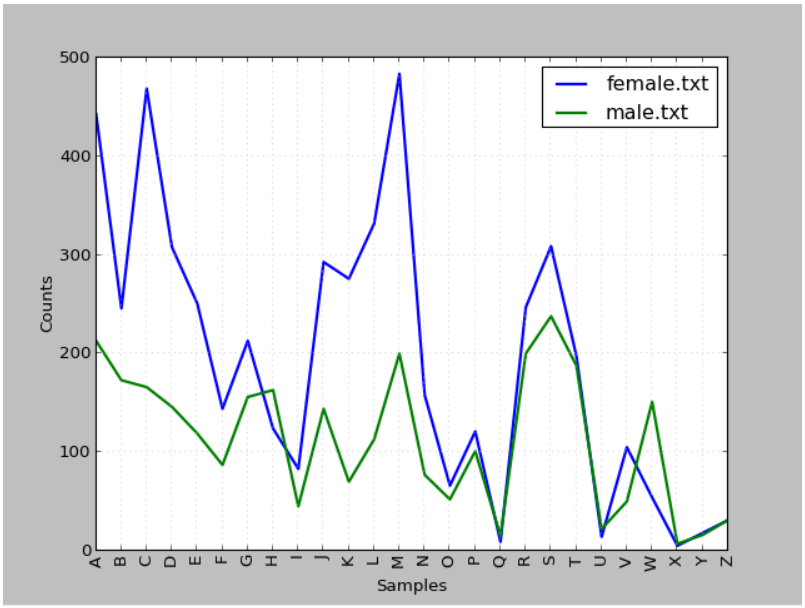
```
from nltk.corpus import swadesh
swadesh.fileids()
deiten = swadesh.entries(['de', 'it', 'en'])
print deiten[:50]
```

```
[('ich', 'io', 'I'), ('du, Sie', 'tu, Lei', 'you (singular), thou'), ('er', 'lui
, egli', 'he'), ('wir', 'noi', 'we'), ('ihr, Sie', 'voi', 'you (plural)'), ('sie
', 'loro, essi', 'they'), ('dieses', 'questo', 'this'), ('jenes', 'quello', 'tha
t'), ('hier', 'qui, qua', 'here'), ('dort', 'l\xc3\xa0', 'there'), ('wer', 'chi'
, 'who'), ('was', 'che', 'what'), ('wo', 'dove', 'where'), ('wann', 'quando', 'w
hen'), ('wie', 'come', 'how'), ('nicht', 'non', 'not'), ('alle', 'tutto', 'all')
, ('viele', 'molti', 'many'), ('einige', 'alcuni', 'some'), ('wenige', 'pochi',
'few'), ('andere', 'altro', 'other'), ('eins', 'uno', 'one'), ('zwei', 'due', 't
wo'), ('drei', 'tre', 'three'), ('vier', 'quattro', 'four'), ('f\xc3\xbcnf', 'ci
nque', 'five'), ('gro\xc3\x9f', 'grande', 'big'), ('lang', 'lungo', 'long'), ('b
reit, weit', 'largo', 'wide'), ('dick', 'spesso', 'thick'), ('schwer', 'pesante'
, 'heavy'), ('klein', 'piccolo', 'small'), ('kurz', 'corto', 'short'), ('eng', '
stretto', 'narrow'), ('d\xc3\xbcnn', 'sottile', 'thin'), ('Frau', 'donna', 'woma
n'), ('Mann', 'uomo', 'man (adult male)'), ('Mensch', 'uomo', 'man (human being)
'), ('Kind', 'bambino', 'child'), ('Frau, Ehefrau', 'moglie', 'wife'), ('Mann, E
hemann', 'marito', 'husband'), ('Mutter', 'madre', 'mother'), ('Vater', 'padre',
 'father'), ('Tier', 'animale', 'animal'), ('Fisch', 'pesce', 'fish'), ('Vogel',
 'uccello', 'bird'), ('Hund', 'cane', 'dog'), ('Laus', 'pidocchio', 'louse'), ('
Schlange', 'serpente', 'snake'), ('Wurm', 'verme', 'worm')]
```

Завдання 3. Побудувати умовний частотний розподіл для корпусу імен. Знайти які перші літери частіше використовуються в чоловічих та жіночих іменах.

```
import nltk
names = nltk.corpus.names
names.fileids()
male_names = names.words('male.txt')
female_names = names.words('female.txt')
print 'male names: ', male_names[:50]
print 'female names: ', female_names[:50]
cfd = nltk.ConditionalFreqDist(
          (fileid, name[0])
          for fileid in names.fileids()
        for name in names.words(fileid))
nam = cfd.plot()
print nam
```

```
male names:  ['Aamir', 'Aaron', 'Abbey', 'Abbie', 'Abbot', 'Abbott', 'Abby', 'Ab
del', 'Abdul', 'Abdulkarim', 'Abdullah', 'Abe', 'Abel', 'Abelard', 'Abner', 'Abr
aham', 'Abram', 'Ace', 'Adair', 'Adam', 'Adams', 'Addie', 'Adger', 'Aditya', 'Ad
lai', 'Adnan', 'Adolf', 'Adolfo', 'Adolph', 'Adolphe', 'Adolpho', 'Adolphus', 'A
drian', 'Adrick', 'Adrien', 'Agamemnon', 'Aguinaldo', 'Aguste', 'Agustin', 'Ahar
on', 'Ahmad', 'Ahmed', 'Ahmet', 'Ajai', 'Ajay', 'Al', 'Alaa', 'Alain', 'Alan', '
Alasdair']
female names:  ['Abagael', 'Abagail', 'Abbe', 'Abbey', 'Abbi', 'Abbie', 'Abby',
'Abigael', 'Abigail', 'Abigale', 'Abra', 'Acacia', 'Ada', 'Adah', 'Adaline', 'Ad
ara', 'Addie', 'Addis', 'Adel', 'Adela', 'Adelaide', 'Adele', 'Adelice', 'Adelin
a', 'Adelind', 'Adeline', 'Adella', 'Adelle', 'Adena', 'Adey', 'Adi', 'Adiana',
'Adina', 'Adora', 'Adore', 'Adoree', 'Adorne', 'Adrea', 'Adria', 'Adriaens', 'Ad
rian', 'Adriana', 'Adriane', 'Adrianna', 'Adrianne', 'Adrien', 'Adriena', 'Adrie
nne', 'Aeriel', 'Aeriela']
```



Завдання 6. Визначити функцію supergloss(s) , яка буде приймати синсет s як аргумент і повертати стрічку в якій будуть поєднані всі описи всіх значень синсету s та описи всіх гіпернімів та гіпонімів s.

```
import nltk
from nltk.corpus import wordnet as wn
def supergloss(s):
    print 'meaning:'
    print wn.synset(s).definition
    print 'Hypernyms'
    for hypernym in set(wn.synset(s).hypernyms()):
        print hypernym, hypernym.definition
    print "hyponyms"
    for hyponym in set (wn.synset(s).hyponyms()):
        print hyponym, hyponym.definition
```

```
>>> supergloss('love.n.01')
meaning:
a strong positive emotion of regard and affection
Hypernyms
Synset('emotion.n.01') any strong feeling
hyponyms
Synset('filial_love.n.01') the love of a child for a parent
Synset('ardor.n.02') intense feeling of love
Synset('heartstrings.n.01') your deepest feelings of love and compassion
Synset('worship.n.02') a feeling of profound love and admiration
Synset('puppy_love.n.01') temporary love of an adolescent
Synset('lovingness.n.01') a loving feeling
Synset('benevolence.n.01') disposition to do good
Synset('amorousness.n.01') a feeling of love or fondness
Synset('agape.n.01') (Christian theology) the love of God or Christ for mankind
Synset('agape.n.02') selfless love of one person for another without sexual impl
ications (especially love that is spiritual in nature)
Synset('devotion.n.01') feelings of ardent love
Synset('loyalty.n.02') feelings of allegiance
```

Завдання 9. Модифікувати програму генерації випадкового тексту для виконання наступного: тренування програми на текстах двох різних жанрів та генерації тексту об'єднаного жанру.

```
import nltk
from nltk.corpus import brown
brown.categories()
def generate_model(cfdist, word, num=15):
    for i in range(num):
        print word,
        word = cfdist[word].max()
text1 = nltk.corpus.brown.words(categories='religion')
bigrams1 = nltk.bigrams(text1)
cfd = nltk.ConditionalFreqDist(bigrams1)
generate_model(cfd, 'breaking')
text2 = nltk.corpus.brown.words(categories='romance')
bigrams2 = nltk.bigrams(text2)
cfd = nltk.ConditionalFreqDist(bigrams2)
generate_model(cfd, 'breaking')
bigrams=nltk.bigrams(text2+text1)
cfd=nltk.ConditionalFreqDist(bigrams)
generate_model(cfd, 'breaking')

>>>
breaking the world . The new members of the world . The new members of breaking
up and the same time , and the same time , and the same breaking the world , and
 the world , and the world , and the world
```

Завдання 13. Полісемія - це явище коли одне слово має декілька значень (іменник dog має 7 значень, кількість яких визначити можна як len(wn.synsets('dog', 'n'))). Знайдіть середнє значення полісемії для прислівників.

```
from __future__ import division
from nltk.corpus import wordnet as wn
d =[]
for i in wn.all_synsets('r'):
    for j in i.lemma_names:
        d.append(j)
num=len(set(d))
print num
q=0
for i in d:
    q += len(wn.synsets(i, 'r'))
print q
poly = q/num
print poly

>>>
4481
9478
2.11515286766
```

Завдання 17. Використовуючи один з методів визначення подібності слів побудуйте відсортований по спаданню список значень подібності для наступних пар слів: monk-oracle, cemetery-woodland, food-rooster, coast-hill, forest-graveyard, crane-implement, journey-car, coast-shore, asylum-madhouse, magician-wizard, midday-noon, furnace-stove, food-fruit, bird-cock.

```
from nltk.corpus import wordnet as wn
monk = wn.synset('monk.n.01')
oracle=wn.synset('oracle.n.01')
cemetery = wn.synset('cemetery.n.01')
woodland=wn.synset('woodland.n.01')
food =wn.synset('food.n.01')
rooster=wn.synset('cock.n.04')
coast=wn.synset('seashore.n.01')
hill=wn.synset('hill.n.01')
forest=wn.synset('forest.n.01')
graveyard=wn.synset('cemetery.n.01')
crane=wn.synset('crane.n.01')
implement=wn.synset('implement.n.01')
journey=wn.synset('journey.n.01')
car=wn.synset('car.n.01')
coast=wn.synset('seashore.n.01')
shore=wn.synset('shore.n.01')
asylum=wn.synset('refuge.n.03')
madhouse=wn.synset('bedlam.n.02')
magician=wn.synset('magician.n.01')
wizard=wn.synset('ace.n.03')
midday=wn.synset('noon.n.01')
noon=wn.synset('noon.n.01')
furnace=wn.synset('furnace.n.01')
stove=wn.synset('stove.n.01')
fruit=wn.synset('fruit.n.01')
bird=wn.synset('bird.n.01')
cock=wn.synset('monk.n.01')
simil=monk.path_similarity(oracle), cemetery.path_similarity(woodland), food.pat
print simil
list=[]
for n in range(14):
    list.append(simil[n])
rev=sorted(list)
rev.reverse()

print 'reversed - ', rev
>>>
(0.125, 0.1111111111111111, 0.0625, 0.2, 0.07142857142857142, 0.1, 0.05, 0.5, 0.
125, 0.16666666666666666, 1.0, 0.07692307692307693, 0.09090909090909091, 0.11111
11111111111)
reversed -  [1.0, 0.5, 0.2, 0.16666666666666666, 0.125, 0.125, 0.111111111111111
1, 0.1111111111111111, 0.1, 0.09090909090909091, 0.07692307692307693, 0.07142857
142857142, 0.0625, 0.05]
```