

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

Кафедра “Системи автоматизованого проектування”



Звіт

до лабораторної роботи №4

на тему: “ВИВЧЕННЯ БІБЛІОТЕКИ ПРИКЛАДНИХ ПРОГРАМ NLTK, ДЛЯ
ОПРАЦЮВАННЯ ТЕКСТІВ ПРИРОДНОЮ МОВОЮ.
ДОСТУП ТА РОБОТА З ЛЕКСИЧНИМИ РЕСУРСАМИ ”
з дисципліни “Комп’ютерна лінгвістика”

Виконала:

студентка групи ПРЛм-11

Липак О.В.

Прийняв:

викладач

Дупак Б.П.

Львів-2015

Мета роботи: вивчити основи програмування на мові *Python*, вивчити методи доступу та робот из лексичними ресурсами, семантичний словник англійської мови WordNet.

Теоретичні відомості.

Лексичні зв'язки в WordNet.

Таблиця 1 містить список найбільш важливих типів зв'язків, які реалізовані у WordNet. Таблиця 2 містить повний список зв'язків іменників.

Таблиця 1

Hypernym	more general	<i>animal</i> is a hypernym of <i>dog</i>
Hyponym	more specific	<i>dog</i> is a hypernym of <i>animal</i>
Meronym	part of	<i>door</i> is a meronym of <i>house</i>
Holonym	has part	<i>house</i> is a holonym of <i>door</i>
Synonym	similar meaning	<i>car</i> is a synonym of <i>automobile</i>
Antonym	opposite meaning	<i>like</i> is an antonym of <i>dislike</i>
Entailment	necessary action	<i>step</i> is an entailment of <i>walk</i>

Таблиця 2

Relation	Also Called	Definition	Example
Hypernym	Superordinate	From concepts to superordinates	<i>breakfast</i> ¹ → <i>meal</i> ¹
Hyponym	Subordinate	From concepts to subtypes	<i>meal</i> ¹ → <i>lunch</i> ¹
Instance Hypernym	Instance	From instances to their concepts	<i>Austen</i> ¹ → <i>author</i> ¹
Instance Hyponym	Has-Instance	From concepts to concept instances	<i>composer</i> ¹ → <i>Bach</i> ¹
Member Meronym	Has-Member	From groups to their members	<i>faculty</i> ² → <i>professor</i> ¹
Member Holonym	Member-Of	From members to their groups	<i>copilot</i> ¹ → <i>crew</i> ¹
Part Meronym	Has-Part	From wholes to parts	<i>table</i> ² → <i>leg</i> ³
Part Holonym	Part-Of	From parts to wholes	<i>course</i> ⁷ → <i>meal</i> ¹
Substance Meronym		From substances to their subparts	<i>water</i> ¹ → <i>oxygen</i> ¹
Substance Holonym		From parts of substances to wholes	<i>gin</i> ¹ → <i>martini</i> ¹
Antonym		Semantic opposition between lemmas	<i>leader</i> ¹ ⇔ <i>follower</i> ¹
Derivationally Related Form		Lemmas w/same morphological root	<i>destruction</i> ¹ ⇔ <i>destroy</i> ¹

Figure 19.2 Noun relations in WordNet.

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 -

1. Дослідити зв'язки голонім-меронім для іменників. Знайти іменники для демонстрації наступних зв'язків: `member_meronyms()`, `part_meronyms()`, `substance_meronyms()`, `member_holonyms()`, `part_holonyms()`, та `substance_holonyms()`.

```
import nltk
from nltk.corpus import wordnet as wn
for w in ['cat', 'house', 'water', 'door']:
    m = wn.synset(w+'.n.01')
    if (m is not None):
        print 'member_meronym ('+w+')=', m.member_meronyms()
        print 'part_meronym ('+w+')=', m.part_meronyms()
        print 'substance_meronym ('+w+')=', m.substance_meronyms()
        print 'member_holonym ('+w+')=', m.member_holonyms()
        print 'part_holonym ('+w+')=', m.part_holonyms()
        print 'substance_holonym ('+w+')=', m.substance_holonyms()

member_meronym (cat)= []
part_meronym (cat)= []
substance_meronym (cat)= []
member_holonym (cat)= []
part_holonym (cat)= []
substance_holonym (cat)= []
member_meronym (house)= []
part_meronym (house)= [Synset('porch.n.01'), Synset('study.n.05'), Synset('library.n.01'),
Synset('loft.n.02')]
substance_meronym (house)= []
member_holonym (house)= []
part_holonym (house)= []
substance_holonym (house)= []
member_meronym (water)= []
part_meronym (water)= []
substance_meronym (water)= [Synset('oxygen.n.01'), Synset('hydrogen.n.01')]
member_holonym (water)= []
part_holonym (water)= []
substance_holonym (water)= [Synset('tear.n.01'), Synset('perspiration.n.01'), Synset('snow
flake.n.01'), Synset('ice_crystal.n.01'), Synset('ice.n.01'), Synset('body_of_water.n.01')
]
member_meronym (door)= []
part_meronym (door)= [Synset('lock.n.01')]
substance_meronym (door)= []
member_holonym (door)= []
part_holonym (door)= [Synset('doorway.n.01')]
substance_holonym (door)= []
```

4. Здійснити аналіз словника вимов. Знайти скільки різних слів він містить.

Який відсоток слів з цього словника можуть мати різну вимову?

```
import nltk
from nltk.corpus import cmudict
entries=cmudict.entries()
len(entries)
words=[w for w, pron in entries]
len(words)
fdist=nltk.FreqDist(words)
s=[w for w in words if fdist[w]>1]
percent = (len(s) * 100)/len(words)
print len(entries)
print(percent)

>>>
133737
14
>>> |
```

5. Який відсоток синсетів іменників не мають гіпонімів? До всіх синсетів можна досягти за допомогою `wn.all_synsets('n')`.

```
>>> from nltk.corpus import wordnet as wn
>>> all = wn.all_synsets('n')
>>> l_all = 0
>>> awh = 0
>>> for syn in all:
>>>     if syn.hyponyms() == []:
>>>         awh += 1
>>>         l_all += 1

>>> l_all
82115
>>> awh
65422
>>> from __future__ import division
>>> awh/l_all*100
79.67119283931072
```

7. Модифікувати програму генерації випадкового тексту для виконання наступного: зберігати можливі наступні слова у списку та вибирати їх за допомогою `random.choice()` попередньо виконавши `import random`.

```
import nltk
from nltk.book import text3
import random
s=text3
print 'The text: ', s, '\n'
word=[]
for i in range(20):
    word.append(random.choice(s))
    print word
```

The text: <Text: The Book of Genesis> .

```
['in']
['in', 'is']
['in', 'is', 'go']
['in', 'is', 'go', 'saw']
['in', 'is', 'go', 'saw', 'And']
['in', 'is', 'go', 'saw', 'And', 'the']
['in', 'is', 'go', 'saw', 'And', 'the', 'cried']
['in', 'is', 'go', 'saw', 'And', 'the', 'cried', 'give']
['in', 'is', 'go', 'saw', 'And', 'the', 'cried', 'give', 'Unto']
['in', 'is', 'go', 'saw', 'And', 'the', 'cried', 'give', 'Unto', 'his']
```

12. 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 дієслів.

```
import nltk
from nltk.corpus import wordnet as wn
verb = []
for i in wn.all_synsets('v'):
    for j in i.lemma_names:
        verb.append(j)
len(set(verb))
print len(set(verb))
mean_amount = 0
for i in set(verb):
    mean_amount = mean_amount + len(wn.synsets(i, 'v'))
print mean_amount
poly_verb=float(mean_amount)/float(len(set(verb)))
print poly_verb
```

```
11531
25214
2.18662735235
```

16. 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, shore-woodland, monk-slave, coast-forest, lad-wizard, chord-smile, glass-magician, rooster-voyage, noon-string.


```

import nltk
from nltk.corpus import wordnet as wn

list_cort = [('monk','oracle'), ('cemetery','woodland'), ('food','rooster'), ('coast','hill'),
              ('monk','slave'), ('coast','forest'), ('lad','wizard'), ('chord','smile'), ('glass'
result_list=[]
for i,j in list_cort:
    first_word=wn.synsets(i)
    second_word=wn.synsets(j)
    for a in first_word:
        result_list.append('Znachenya podibnosti dlya:')
        result_list.append(a)
        for b in second_word:
            result_list.append('ta:')
            result_list.append(b)
            result_list.append(a.path_similarity(b))

print '\n Посортований список 1 \n'
print first_word
print '\n Посортований список 2 \n'
print second_word
print '\n Подібність \n'
print result_list

    Посортований список 1

[Synset('noon.n.01')]

    Посортований список 2

[Synset('string.n.01'), Synset('bowed_stringed_instrument.n.01'), Synset('string
.n.03'), Synset('string.n.04'), Synset('string.n.05'), Synset('drawstring.n.01')
, Synset('string.n.07'), Synset('string.n.08'), Synset('string.n.09'), Synset('c
hain.n.10'), Synset('string.v.01'), Synset('string.v.02'), Synset('string.v.03')
, Synset('string.v.04'), Synset('string.v.05'), Synset('string.v.06'), Synset('s
tring.v.07')]

    Подібність

['Znachenya podibnosti dlya:', Synset('monk.n.01'), 'ta:', Synset('prophet.n.01'
), 0.125, 'ta:', Synset('oracle.n.02'), 0.07142857142857142, 'ta:', Synset('orac
le.n.03'), 0.07692307692307693, 'Znachenya podibnosti dlya:', Synset('monk.n.02'
), 'ta:', Synset('prophet.n.01'), 0.1, 'ta:', Synset('oracle.n.02'), 0.0625, 'ta
:', Synset('oracle.n.03'), 0.06666666666666667, 'Znachenya podibnosti dlya:', Sy
nset('cemetery.n.01'), 'ta:', Synset('forest.n.02'), 0.1111111111111111, 'Znache
nya podibnosti dlya:', Synset('food.n.01'), 'ta:', Synset('cock.n.04'), 0.0625,

```

Висновок: у цій лабораторній роботі я вивчила основи програмування на мові *Python*, методи доступу та роботи із лексичними ресурсами, семантичний словник англійської мови WordNet.