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

Кафедра "Системи автоматизованого проектування"

Звіт

до лабораторної роботи №9

на тему: ВИВЧЕННЯ БІБЛІОТЕКИ ПРИКЛАДНИХ ПРОГРАМ NLTK, ДЛЯ ОПРАЦЮВАННЯ ТЕКСТІВ ПРИРОДНОЮ МОВОЮ. АВТОМАТИЧНИЙ МОРФОЛОГІЧНИЙ АНАЛІЗ (частина1).

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. Ознайомлення з автоматичним морфологічним аналізом в NLTK.

**Тексти програм на мові *Python*.**

Варіант – 3

15. Знайдіть заголовки газетних публікацій подібні до наступних «British Left Waffles on Falkland Islands», «Juvenile Court to Try Shooting Defendant». Промаркуйте ці заголовки і опишіть, яким чином знання про набори тегів корпуса Brown дозволяють розв'язати неднозначності.

```
import nltk
import numpy
a='Five Mistakes College Job Seekers Make'
b='Terrorist attacks night: how it was and who stands behind executions and explosions in Paris'
token1=nltk.word_tokenize(a)
token2=nltk.word_tokenize(b)
morph1=nltk.pos_tag(token1)
morph2=nltk.pos_tag(token2)
print(morph1,morph2)
```

```
>>>
([('Five', 'CD'), ('Mistakes', 'NNS'), ('College', 'NNP'), ('Job', 'NNP'), ('See
kers', 'NNP'), ('Make', 'NNP')], [('Terrorist', 'NNP'), ('attacks', 'NNS'), ('ni
ght', 'VBD'), (':', ':'), ('how', 'WRB'), ('it', 'PRP'), ('was', 'VBD'), ('and',
 'CC'), ('who', 'WP'), ('stands', 'NNS'), ('behind', 'VBP'), ('executions', 'NNS
'), ('and', 'CC'), ('explosions', 'NNS'), ('in', 'IN'), ('Paris', 'NNP')])
>>>
```

3. Опрацювати всі приклади з методичних вказівок по роботі зі словниками. Що станеться, якщо доступитися до неіснуючого запису звичайного словника та словника по замовчуванню?

```
import nltk
pos={}
print '1)',pos

pos['colorless']='ADJ'
print '2)', pos

pos['ideas']='N'
pos['sleep']='V'
pos['furiously']='ADV'
print '3)',pos

print '4)',pos['ideas']
print '5)',pos['colorless']
print '6)',list(pos)
print '7)',sorted(pos)

s=[w for w in pos if w.endswith('s')]
print '8)',s

for word in sorted(pos):
    print word+':',pos[word]

print '9)', pos.keys()
print '10)',pos.values()
print '11)',pos.items()

for key,val in sorted(pos.items()):
    print '12)',key+':',val

pos={'colorless':'ADJ','ideas':'N','sleep':'V','furiously':'ADJ'}
print '13)',pos

pos=dict(colorless='ADJ',ideas='N',sleep='V',furiously='ADV')
print '14)',pos

frequency=nltk.defaultdict(int)
frequency['colorless']=4
print '15)',frequency['ideas']

pos=nltk.defaultdict(list)
pos['sleep']=['N','V']
print '16)', pos['ideas']

pos=nltk.defaultdict(lambda:'N')
pos['colorless']='ADJ'
print '17)',pos['blog']

print '18)', pos.items()
```

```
>>>
1) {}
2) {'colorless': 'ADJ'}
3) {'furiously': 'ADV', 'sleep': 'V', 'ideas': 'N', 'colorless': 'ADJ'}
4) N
5) ADJ
6) ['furiously', 'sleep', 'ideas', 'colorless']
7) ['colorless', 'furiously', 'ideas', 'sleep']
8) ['ideas', 'colorless']
colorless: ADJ
furiously: ADV
ideas: N
sleep: V
9) ['furiously', 'sleep', 'ideas', 'colorless']
10) ['ADV', 'V', 'N', 'ADJ']
11) [('furiously', 'ADV'), ('sleep', 'V'), ('ideas', 'N'), ('colorless', '
12) colorless: ADJ
12) furiously: ADV
12) ideas: N
12) sleep: V
13) {'furiously': 'ADJ', 'sleep': 'V', 'ideas': 'N', 'colorless': 'ADJ'}
14) {'furiously': 'ADV', 'sleep': 'V', 'ideas': 'N', 'colorless': 'ADJ'}
15) 0
16) []
17) N
18) [('blog', 'N'), ('colorless', 'ADJ')]
>>>
```

7. Використовуючи sorted() та set() отримайте відсортований список всіх тегів корпуса Brown без їх дублювання.

```
import nltk
import pprint
from nltk.corpus import brown
def list_of_tags(tagged_words):
    tags=[]
    for(w,t) in tagged_words:
        tags.append(t)
    tags_list=set(tags)
    return pprint.pprint(sorted(tags_list))
print list_of_tags(nltk.corpus.brown.tagged_words())
```

```
>>>
["''",
 "``",
 '(',
 '(-HL',
 ')',
 ')-HL',
 '*',
 '*-HL',
 '*-NC',
 '*-TL',
 ',',
 ',-HL',
 ',-NC',
 ',-TL',
 '--',
 '---HL',
 '.',
```

10. Напишіть програму, яка обробить Brown Corpus і допоможе відповісти на наступне запитання: які теги найчастіше зустрічаються (створити список 20 тегів, які мають максимальну частоту);

```
import nltk
text=nltk.corpus.brown.tagged_words()
tags=[t for (w,t) in text]
fd = nltk.FreqDist(tags)
print fd.keys()[:20]
nltk.help.brown_tagset('AT')

>>>
['NN', 'IN', 'AT', 'JJ', '.', ',', 'NNS', 'CC', 'RB', 'NP', 'VB', 'VBN', 'VBD',
'CS', 'PPS', 'VBG', 'PP$', 'TO', 'PPSS', 'CD']
AT: article
    the an no a every th' ever' ye
>>>
```

14. Напишіть програму для збору статистичних даних по розмічених корпусах і відповіді на наступне запитання: який відсоток слів корпусу Brown мають неоднозначності.

```
import nltk
from nltk.probability import FreqDist
brown_mystery_tagged=nltk.corpus.brown.tagged_words(categories='mystery')
data=nltk.ConditionalFreqDist((word.lower(),tag)
                              for(word,tag)in brown_mystery_tagged)
suspicious=0
for word in sorted(data.conditions()):
    if len(data[word])>1:
        tags=[tag for tag in data[word]]
        suspicious=suspicious+1
print('Suspicious words:',suspicious)
print('All words:',len(brown_mystery_tagged))
print('Persentage of suspicious words is:', suspicious/len(brown_mystery_tagged)*100,'%')

>>>
('Suspicious words:', 751)
('All words:', 57169)
('Persentage of suspicious words is:', 0, '%')
>>>
```

18. Напишіть програми для знаходження слів та словосполучень згідно відповідних їм тегів для відповіді на наступне питання: яке співвідношення між займенниками (чоловічими і жіночими).

```
import nltk
from nltk import FreqDist,ConditionalFreqDist
from nltk.corpus import brown
fd=FreqDist()
cfd=ConditionalFreqDist()
for sent in brown.tagged_sents():
    for(token,tag) in sent:
        fd.inc(tag)
        cfd[token].inc(tag)
male=['man','he']
female=['woman','she']
n_male,n_female=0,0
for m in male:
    n_male+=cfd[m].N()
print n_male
for f in female:
    n_female+=cfd[m].N()
print n_female
print float(n_male)/n_female

>>>
7717
13132
0.587648492233
>>>
```

21. Написати програму побудови словника, записами якого будуть набори словників. Використовуючи створений словник, збережіть у ньому набори можливих тегів, які зустрічаються після заданого слова з певним тегом, наприклад $word_i \rightarrow tag_i \rightarrow tag_{i+1}$.

```
import nltk
from nltk.corpus import brown
pos=nltk.defaultdict(lambda:nltk.defaultdict(int))
brown_fiction_tagged=brown.tagged_words(categories='fiction',simplify_tags=True)
for ((w1,t1),(w2,t2)) in nltk.ibigrams(brown_fiction_tagged):
    pos[(w1,t1)][t2]+=1
print pos[('that','DET')]

>>>
defaultdict(<type 'int'>, {'ADV': 5, ':': 1, 'VD': 3, 'WH': 2, 'CNJ': 1, 'PRO':
3, 'DET': 4, ',': 9, 'VN': 3, 'N': 65, "''": 6, 'P': 9, 'NUM': 1, 'V': 21, 'NP':
 4, 'VBZ': 1, '.': 12, 'ADJ': 22, 'MOD': 8})
>>>
```

**Висновок:** на цій лабораторній роботі я ознайомилася з автоматичним морфологічним аналізом в NLTK.