МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
«ЛЬВІВСЬКА ПОЛІТЕХНІКА»

Кафедра
«Системи автоматизованого проектування»

Звіт
До лабораторної роботи №12
З курсу: «Комп'ютерна лінгвістика»

ВИВЧЕННЯ БІБЛІОТЕКИ ПРИКЛАДНИХ ПРОГРАМ NLTK, ДЛЯ ОПРАЦЮВАННЯ ТЕКСТІВ ПРИРОДНОЮ МОВОЮ. АВТОМАТИЧНИЙ СИНТАКСИЧНИЙ АНАЛІЗ (частина2).

Виконала:
ст. гр. ПРЛм-11
Зварич О.І.
Перевірив:
викладач
Дупак Б.П.

Львів-201

МЕТА РОБОТА

- Вивчення основ програмування на мові *Python*.
- Ознайомлення з автоматичним синтаксичним аналізом в NLTK.

1 Написати рекурсивну функцію для перегляду дерева, яка визначає його глибину. Дерево з одного вузла має глибину рівну нулю. (глибина піддерева це максимальна глибина його дітей плюс один)

```
import nltk
t = nltk.Tree('(S (NP Marry) (VP chased (NP the cat)))')
def traverse(t):
    try:
        t.node
    except AttributeError:
            print t,
    else:
            print '(', t.node,
            for child in t:
                traverse(child)
            print ')',
print t.height()
print traverse(t)
```

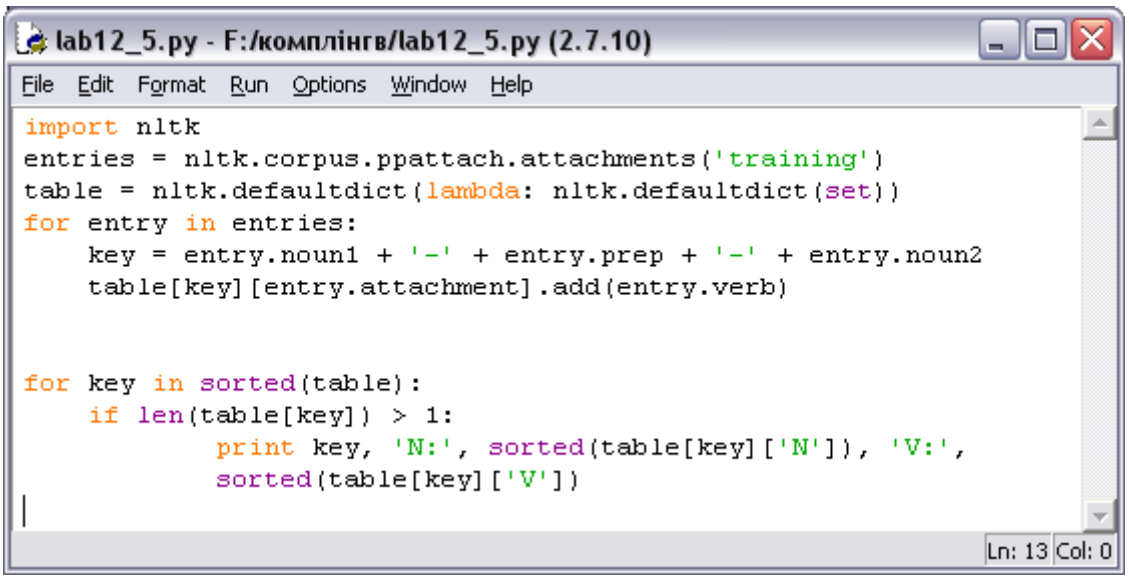
```
>>>
4
( S ( NP Marry ) ( VP chased ( NP the cat ) ) ) None
>>>
```

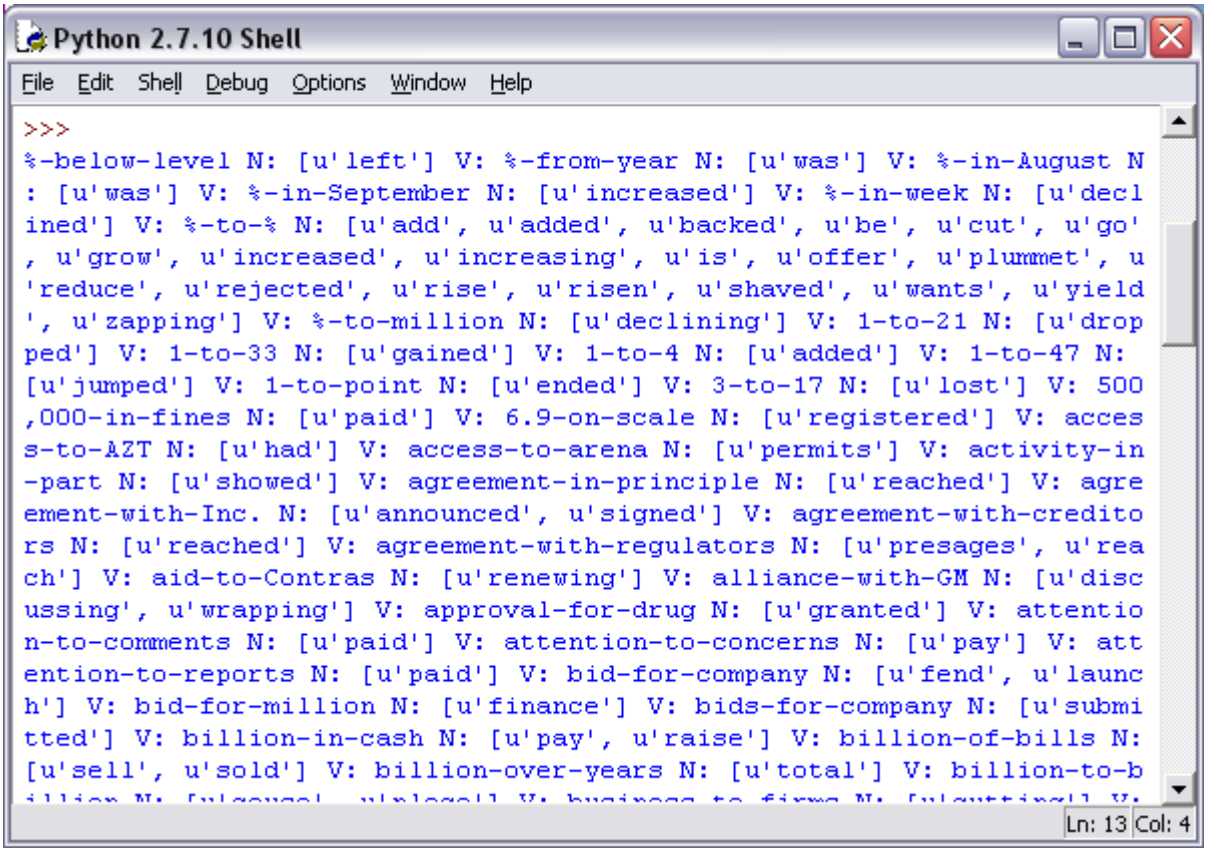
3.chart parser додає, але ніколи не видаляє дуги з chart. Чому?
Адже це динамічне програмування і воно запам’ятовує проміжні результати.

5. Вибрати декілька (2) загальних дієслова та напишіть програми для вирішення наступних задач:

Пошук дієслів в корпусі Prepositional Phrase Attachment Corpus `nltk.corpus.ppattach`. Пошук всіх випадків вживання дієслова з двома різними PP в яких перший іменник, або другий іменник або прийменник залишаються незмінними

Розробити правила CFG граматики для врахування цих випадків.





7. Використовуючи позиції в дереві побудувати список підметів перших 100 речень корпусу Penn treebank; для спрощення представлення результатів підмети представляти як піддерева з глибиною не більше 2.

```
import nltk
from nltk.corpus import treebank
treebank.penn=treebank.parsed_sents()[:100]
def filter(tree):
        child_nodes=[child.node for child in tree
                     if isinstance (child, nltk.Tree) and len(tree) <2]
        return tree.node=='NP-SBJ'
result=[subtree for tree in treebank.penn for subtree in tree.subtrees(filter)]
print result
```

```
[Tree('NP-SBJ', [Tree('NP', [Tree('NNP', ['Pierre']), Tree('NNP', ['Vinken'])]),
 Tree(',', [',']), Tree('ADJP', [Tree('NP', [Tree('CD', ['61']), Tree('NNS', ['y
ears'])]), Tree('JJ', ['old'])]), Tree(',', [','])]), Tree('NP-SBJ', [Tree('NNP'
, ['Mr.']), Tree('NNP', ['Vinken'])]), Tree('NP-SBJ', [Tree('-NONE-', ['*-1'])])
, Tree('NP-SBJ', [Tree('NP', [Tree('NP', [Tree('DT', ['A']), Tree('NN', ['form']
)]), Tree('PP', [Tree('IN', ['of']), Tree('NP', [Tree('NN', ['asbestos'])])])]),
 Tree('RRC', [Tree('ADVP-TMP', [Tree('RB', ['once'])]), Tree('VP', [Tree('VBN',
['used']), Tree('NP', [Tree('-NONE-', ['*'])]), Tree('S-CLR', [Tree('NP-SBJ', [T
ree('-NONE-', ['*'])]), Tree('VP', [Tree('TO', ['to']), Tree('VP', [Tree('VB', [
'make']), Tree('NP', [Tree('NNP', ['Kent']), Tree('NN', ['cigarette']), Tree('NN
S', ['filters'])])])])])])])]), Tree('NP-SBJ', [Tree('-NONE-', ['*'])]), Tree('N
P-SBJ', [Tree('NNS', ['researchers'])]), Tree('NP-SBJ', [Tree('NP', [Tree('DT',
['The']), Tree('NN', ['asbestos']), Tree('NN', ['fiber'])]), Tree(',', [',']), T
ree('NP', [Tree('NN', ['crocidolite'])]), Tree(',', [','])]), Tree('NP-SBJ', [Tr
ee('PRP', ['it'])]), Tree('NP-SBJ', [Tree('NP', [Tree('RB', ['even']), Tree('JJ'
, ['brief']), Tree('NNS', ['exposures'])]), Tree('PP', [Tree('TO', ['to']), Tree
('NP', [Tree('PRP', ['it'])])])]), Tree('NP-SBJ', [Tree('-NONE-', ['*T*-1'])]),
Tree('NP-SBJ', [Tree('NNS', ['researchers'])]), Tree('NP-SBJ', [Tree('NP', [Tree
('NNP', ['Lorillard']), Tree('NNP', ['Inc.'])]), Tree(',', [',']), Tree('NP', [T
ree('NP', [Tree('DT', ['the']), Tree('NN', ['unit'])]), Tree('PP', [Tree('IN', [
'of']), Tree('NP', [Tree('ADJP', [Tree('JJ', ['New']), Tree('JJ', ['York-based']
)]), Tree('NNP', ['Loews']), Tree('NNP', ['Corp.'])])]), Tree('SBAR', [Tree('WHN
```

12 Розробити програму обробки дерев корпуса Treebank `nltk.corpus.treebank` , яка вилучить всі правила з кожного з дерев за допомогою `Tree.productions()`. Правилами, які зустрічаються тільки один раз можна знехтувати. Правила з однаковими лівими частинами та подібними правими частинами об'єднати для отримання еквівалентного але більш компактного набору правил.

```
import nltk
from nltk.corpus import treebank
t = treebank.parsed_sents('wsj_0002.mrg')[0]
print t
p= t.productions()
print p
Fdist=nltk.FreqDist(p)
fd=Fdist.keys()
for i in Fdist.keys():
    if Fdist[i]>0:
        print i
```

```
>>>
(S
  (NP-SBJ-1
    (NP (NNP Rudolph) (NNP Agnew))
    (, ,)
    (UCP
      (ADJP (NP (CD 55) (NNS years)) (JJ old))
      (CC and)
      (NP
        (NP (JJ former) (NN chairman))
        (PP
          (IN of)
          (NP (NNP Consolidated) (NNP Gold) (NNP Fields) (NNP PLC)))))
    (, ,))
  (VP
    (VBD was)
    (VP
      (VBN named)
      (S
        (NP-SBJ (-NONE- *-1))
        (NP-PRD
          (NP (DT a) (JJ nonexecutive) (NN director))
          (PP
            (IN of)
            (NP
              (DT this)
              (JJ British)
              (JJ industrial)
              (NN conglomerate)))))))
  (. .))
[S -> NP-SBJ-1 VP ., NP-SBJ-1 -> NP , UCP ,, NP -> NNP NNP, NNP -> 'Rudolph', NN
P -> 'Agnew', , -> ',', UCP -> ADJP CC NP, ADJP -> NP JJ, NP -> CD NNS, CD -> '5
5', NNS -> 'years', JJ -> 'old', CC -> 'and', NP -> NP PP, NP -> JJ NN, JJ -> 'f
ormer', NN -> 'chairman', PP -> IN NP, IN -> 'of', NP -> NNP NNP NNP NNP, NNP ->
 'Consolidated', NNP -> 'Gold', NNP -> 'Fields', NNP -> 'PLC', , -> ',', VP -> V
BD VP, VBD -> 'was', VP -> VBN S, VBN -> 'named', S -> NP-SBJ NP-PRD, NP-SBJ ->
-NONE-, -NONE- -> '*-1', NP-PRD -> NP PP, NP -> DT JJ NN, DT -> 'a', JJ -> 'none
xecutive', NN -> 'director', PP -> IN NP, IN -> 'of', NP -> DT JJ JJ NN, DT -> '
```

```
IN -> 'of'
PP -> IN NP
-NONE- -> '*-1'
. -> '.'
ADJP -> NP JJ
CC -> 'and'
CD -> '55'
DT -> 'a'
DT -> 'this'
JJ -> 'British'
JJ -> 'former'
JJ -> 'industrial'
JJ -> 'nonexecutive'
JJ -> 'old'
NN -> 'chairman'
NN -> 'conglomerate'
NN -> 'director'
NNP -> 'Agnew'
NNP -> 'Consolidated'
NNP -> 'Fields'
NNP -> 'Gold'
NNP -> 'PLC'
NNP -> 'Rudolph'
NNS -> 'years'
NP -> CD NNS
NP -> DT JJ JJ NN
NP -> DT JJ NN
NP -> JJ NN
NP -> NNP NNP
NP -> NNP NNP NNP NNP
NP -> NP PP
NP-PRD -> NP PP
NP-SBJ -> -NONE-
NP-SBJ-1 -> NP , UCP ,
S -> NP-SBJ NP-PRD
S -> NP-SBJ-1 VP .
UCP -> ADJP CC NP
VBD -> 'was'
VBN -> 'named'
```

Висновок: на цій лабораторній роботі я ознайомлення з автоматичним синтаксичним аналізом в NLTK chart parser, дізналася про динамічне програмування, граматику залежностей.