

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ «ЛЬВІВСЬКА ПОЛІТЕХНІКА»
Кафедра САПР

ЗВІТ

до лабораторної роботи № 3

на тему:

ВИВЧЕННЯ БІБЛІОТЕКИ ПРИКЛАДНИХ ПРОГРАМ NLTK, ДЛЯ
ОПРАЦЮВАННЯ ТЕКСТІВ ПРИРОДНОЮ МОВОЮ.

ДОСТУП ТА РОБОТА З КОРПУСАМИ ТЕКСТІВ.

з дисципліни “Комп’ютерна лінгвістика”

Виконала:

Студентка групи ПРЛм-12

Рибчак Х. В.

Перевірив:

Асистент кафедри САПР

Дупак Б. П.

Львів 2015

МЕТА РОБОТИ

Вивчення основ програмування на мові Python. Вивчення методів доступу до корпусів текстів. Вивчення класу ConditionalFreqDist.

КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ

Вирішення задач обробки текстів природною мовою передбачає використання великих об'ємів лінгвістичних даних, або інішими словами передбачає роботу з корпусами текстів. Виконання даної лабораторної роботи допоможе знайти відповідь на наступні питання: які є відомі корпуси текстів та лексичні ресурси і як отримати до них доступ використовуючи Python; які корисні конструкції має Python для виконання цієї роботи.

Корпус текстів це великий набір текстів. Багато корпусів розроблені із збереженням балансу між текстами різних жанрів, або авторів. В попередній лабораторній роботі ми працювали з промовами президентів США, які є частиною корпусу US Presidential Inaugural Addresses. З промовами ми працювали, як з одним текстом не зважаючи на те, що кожна промова має окремого автора. Обробку ми здійснювали. При роботі з корпусами важливо мати засоби доступу як до окремих текстів так і до окремих частин текстів.

В NLTK входить невелика частина текстів з електронного архіву текстів Project Gutenberg, який містить 25000 безкоштовних електронних книжок різних авторів (<http://www.gutenberg.org/>). Тексти творів в окремих файлах. Для одержання назв файлів (ідентифікаторів файлів) в яких зберігаються текстів потрібно використати наступну функцію:

```
>>> import nltk
```

```
>>> nltk.corpus.gutenberg.fileids()
```

При програмуванні часто необхідно частину програми виконати (використати) декілька разів. Наприклад, потрібно написати програму, яка здійснює утворення множини з однини іменників і вона буде виконуватись в різних

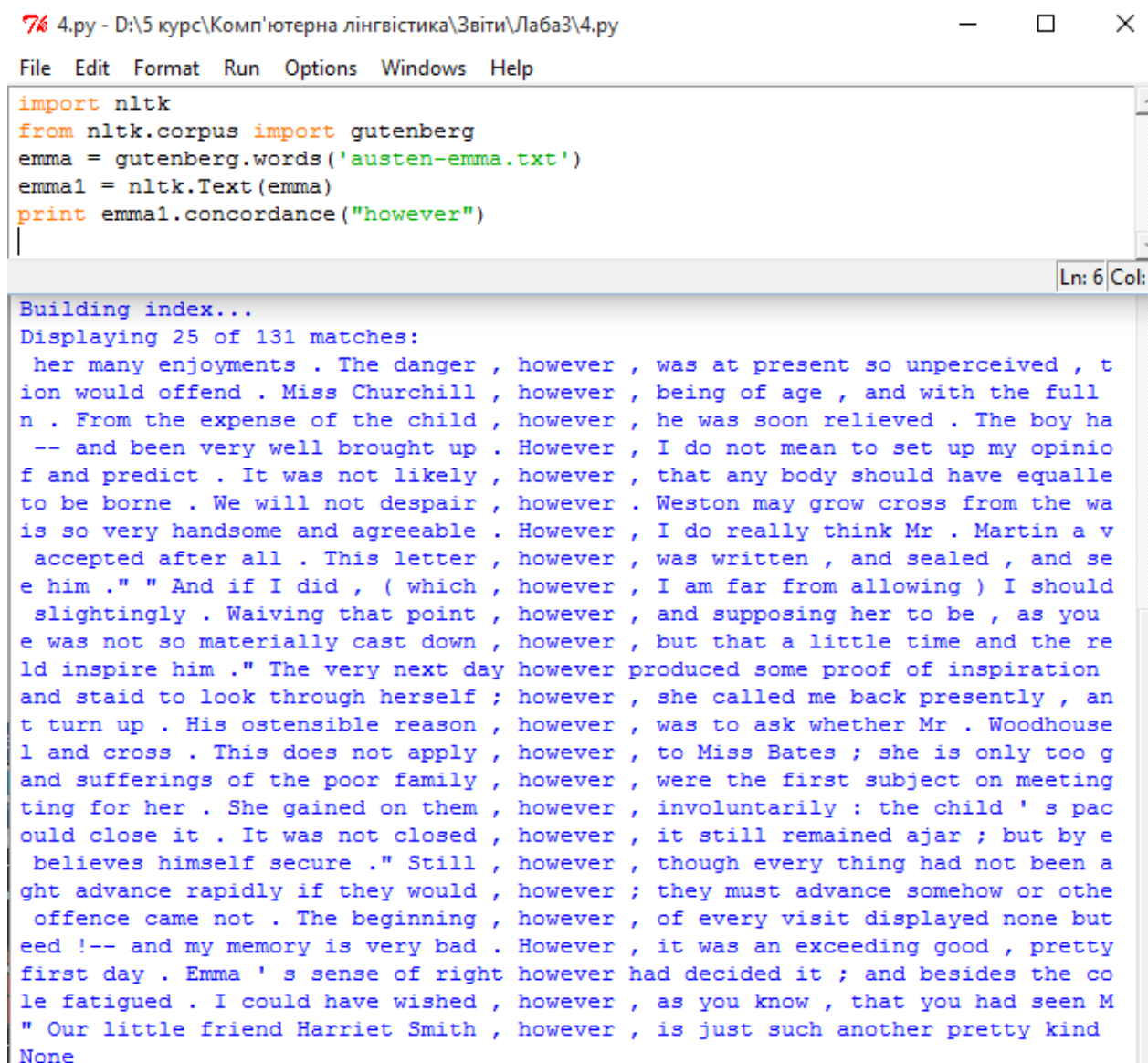
місцях програми. Швидше ніж повторювати той самий код декілька разів і більш ефективно і надійно організувати цю роботу через функцію. Функція - це програмна конструкція, яку можна викликати з одним або більше вхідними параметрами, і отримувати результат на виході. Визначаємо функцію, використовуючи 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.

ТЕКСТИ ПРОГРАМ НА МОВІ PYTHON

ВАРІАНТ №8

4. Використовуючи конкорданси поясніть відмінності у вживанні слова however на початку речення ("in whatever way", "to whatever extent", або "nevertheless").

Особливостей у вживанні слова however на початку речення не виявлено, воно завжди є вставним словом і виділяється комою/комами.



```
4.py - D:\5 курс\Комп'ютерна лінгвістика\Звіти\Лаба3\4.py
File Edit Format Run Options Windows Help

import nltk
from nltk.corpus import gutenberg
emma = gutenberg.words('austen-emma.txt')
emma1 = nltk.Text(emma)
print emma1.concordance("however")

Building index...
Displaying 25 of 131 matches:
her many enjoyments . The danger , however , was at present so unperceived , t
ion would offend . Miss Churchill , however , being of age , and with the full
n . From the expense of the child , however , he was soon relieved . The boy ha
-- and been very well brought up . However , I do not mean to set up my opinio
f and predict . It was not likely , however , that any body should have equalle
to be borne . We will not despair , however . Weston may grow cross from the wa
is so very handsome and agreeable . However , I do really think Mr . Martin a v
accepted after all . This letter , however , was written , and sealed , and se
e him . " " And if I did , ( which , however , I am far from allowing ) I should
slightly . Waiving that point , however , and supposing her to be , as you
e was not so materially cast down , however , but that a little time and the re
ld inspire him ." The very next day however produced some proof of inspiration
and staid to look through herself ; however , she called me back presently , an
t turn up . His ostensible reason , however , was to ask whether Mr . Woodhouse
l and cross . This does not apply , however , to Miss Bates ; she is only too g
and sufferings of the poor family , however , were the first subject on meeting
ting for her . She gained on them , however , involuntarily : the child ' s pac
ould close it . It was not closed , however , it still remained ajar ; but by e
believes himself secure ." Still , however , though every thing had not been a
ght advance rapidly if they would , however ; they must advance somehow or othe
offence came not . The beginning , however , of every visit displayed none but
eed !-- and my memory is very bad . However , it was an exceeding good , pretty
first day . Emma ' s sense of right however had decided it ; and besides the co
le fatigued . I could have wished , however , as you know , that you had seen M
" Our little friend Harriet Smith , however , is just such another pretty kind
None
```

Рис. 1. Текст програми №4.

6. Проаналізуйте таблицю частот модальних дієслів для різних жанрів. Спробуйте її пояснити. Знайдіть інші класи слів вживання яких також відрізняються в різних жанрах.

```
>>>
      can could  may might must will
adventure  46  151   5   58  27   50
belles_lettres 246 213 207 113 170 236
editorial  121  56   74  39  53 233
fiction    37 166   8   44  55  52
government 117  38 153  13 102 244
hobbies    268  58 131  22  83 264
humor      16  30   8    8   9  13
learned    365 159 324 128 202 340
lore       170 141 165  49  96 175
mystery    42 141  13  57  30  20
news       93  86  66  38  50 389
religion   82  59  78  12  54  71
reviews    45  40  45  26  19  58
romance    74 193  11  51  45  43
science_fiction 16  49   4  12   8  16
>>>
```

```
File Edit Format Run Options Windows Help
import nltk
from nltk.corpus import brown
cfd=nltk.ConditionalFreqDist(
    (genre, word)
    for genre in brown.categories()
    for word in brown.words(categories=genre))
genres = ['adventure','belles_lettres','editorial','fiction'
modals=['can','could','may','might','must','will']
cfd.tabulate(conditions=genres,samples=modals)
```

Рис. 2. Текст програми №6.

7. Напишіть програму для знаходження всіх слів в корпусі Brown, які зустрічаються не менш ніж три рази.

```
7.py - D:\5 курс\Комп'ютерна лінгвістика\Звіти\Лаба3\7.py
File Edit Format Run Options Windows Help
import nltk
from nltk.corpus import brown
texts = brown.words()
fdist = nltk.FreqDist ([w for w in texts])
k=sorted ([w for w in set (texts) if fdist[w]>=3 and w.isalpha()])
print (k[:20])

>>>
['A', 'ABO', 'ADC', 'AIA', 'AID', 'AIMO', 'AM', 'AP', 'AWOC', 'Aaron', 'Abbe', '
Abbey', 'Abe', 'Abel', 'Abolition', 'About', 'Above', 'Abraham', 'Abstract', 'Ab
straction']
>>>
```

Рис. 3. Текст програми №7.

8. Напишіть програму генерації таблиці відношень кількість слів/кількість оригінальних слів для всіх жанрів корпусу Brown. Проаналізуйте отримані результати та поясніть їх.

```
74 8.py - D:\5 курс\Комп'ютерна лінгвістика\Звіти\Лаба3\8.py
File Edit Format Run Options Windows Help

import nltk
from nltk.corpus import brown
for style in brown.categories():
    num_words = len(brown.words(categories = style))
    num_original = len(set(brown.words(categories = style)))
    print num_words, num_original, int(num_words/num_original), style

Ln: 7 Co

>>>
69342 8874 7 adventure
173096 18421 9 belles_lettres
61604 9890 6 editorial
68488 9302 7 fiction
70117 8181 8 government
82345 11935 6 hobbies
21695 5017 4 humor
181888 16859 10 learned
110299 14503 7 lore
57169 6982 8 mystery
100554 14394 6 news
39399 6373 6 religion
40704 8626 4 reviews
70022 8452 8 romance
14470 3233 4 science_fiction
>>>
```

Рис. 4. Текст програми №8.

10. Напишіть програму яка виводить на екран 50 найчастотніших біграмів тексту, за виключенням біграмів до складу яких входять незначущі слова.

```
74 10.py - D:\5 курс\Комп'ютерна лінгвістика\Звіти\Лаба3\10.py
File Edit Format Run Options Windows Help

import nltk
from nltk.book import *
from nltk import bigrams
from nltk.probability import FreqDist
m = bigrams(text6)
tags=".,!?:;-#''''[]"
new_bigrams = []
stopwords = nltk.corpus.stopwords.words('english')
new_bigrams = [bi for bi in m if bi[0].lower() not in stopwords \
               and bi[1].lower() not in stopwords and bi[0].lower() not in tags \
               and bi[1].lower() not in tags]
fdist = FreqDist(new_bigrams)
fdist.plot(50)
```

Рис. 5. Текст програми №10.

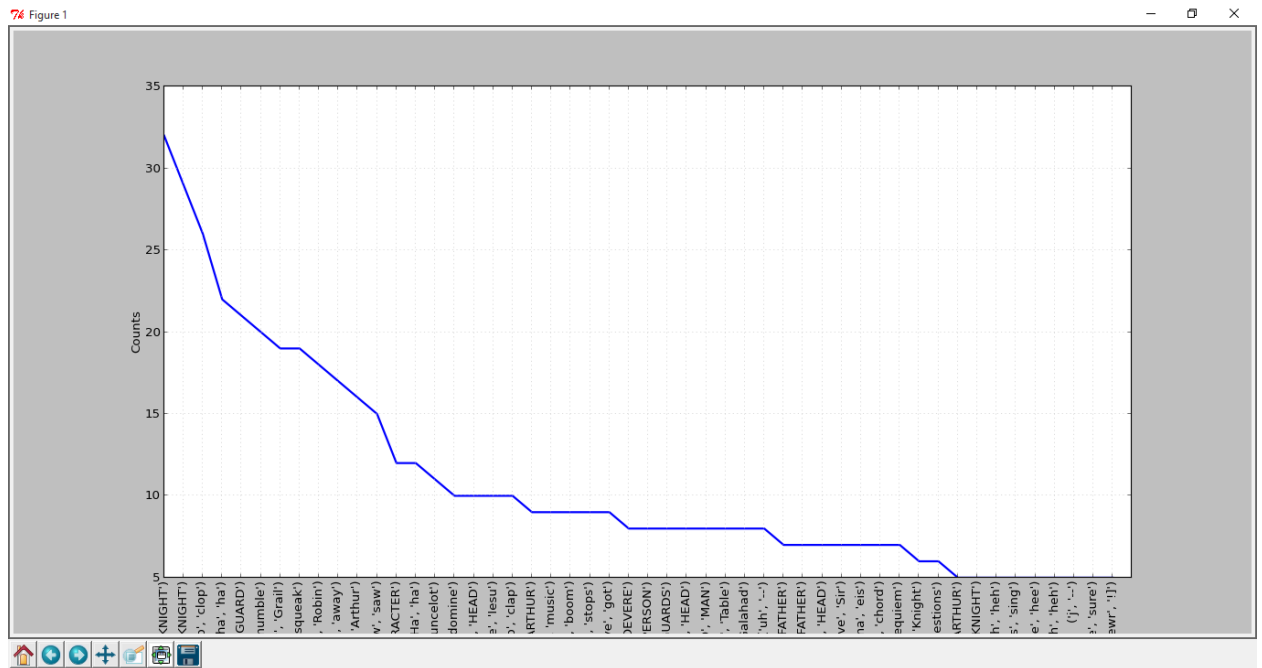


Рис. 6. Результат виконання програми №10.

13. Визначити функцію `hedge(text)`, яка обробляє текст і створює нову версію цього тексту додаючи слово 'like' перед кожним третім словом.

```

13.py - D:\5 курс\Комп'ютерна лінгвістика\Звіти\Лаба3\13.py
File Edit Format Run Options Windows Help

a = 'My name is Khrystyna. I am 21. I an doing my MA in AL.'
def hedge(text):
    b = []
    text = text.split()
    for x in text:
        if text.index(x) in range(2, len(text), 2):
            b.append('like ' + x)
        else:
            b.append(x)
    new_text = ''
    for word in b:
        new_text += word + ' '
    print new_text
hedge(a)

Ln: 16 Col:

>>>
My name like is Khrystyna. like I am like 21. like I like an doing like my MA li
ke in AL.
>>>

```

Рис. 7. Текст програми №13.

ВИСНОВОК

У цій лабораторній роботі я вивчила основи програмування на Python. Ознайомилася з прикладами корпусів текстів та методами доступу до них, навчилася будувати умовний частотний розподіл. Дізналася про поняття функції та модуля.