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

Кафедра “Системи автоматизованого проектування”

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-11

Липак О.В.

Прийняв:

викладач

Дупак Б.П.

Львів-2015

Мета роботи: вивчити основи програмування на мові *Python,* вивчити методи доступу до корпусів текстів, вивчити клас ConditionalFreqDist.

Теоретичні відомості.

Приклад використання функції	**Опис**
fileids()	Файли корпуса
fileids([categories])	Файли корпуса, що відповідають цій категорії
categories()	Категорії корпуса
categories([fileids])	Категорії корпуса, що відповідають цим файлам
raw()	Корпус, як послідовність символів
raw(fileids=[f1,f2,f3])	Послідовність символів з наступних файлів
raw(categories=[c1,c2])	Послідовність символів з наступних категорій
words()	Слова корпуса
words(fileids=[f1,f2,f3])	Слова з наступних файлів
words(categories=[c1,c2])	Слова з наступних категорій
sents()	Речення корпуса
sents(fileids=[f1,f2,f3])	Речення корпуса з наступних файлів
sents(categories=[c1,c2])	Речення корпуса з наступних категорій
abspath(fileid)	Місцезнаходження даного файлу на диску
encoding(fileid)	Кодування файлу (якщо відоме)
open(fileid)	Відкриття файла з корпуса для читання
root()	Шлях до місця де встановлено корпус

Приклад використання функції	Опис
readme()	Вміст файла README корпуса текстів

Використовуючи клас ConditionalFreqDist можна визначити частоту слів для різних жанрів. У випадку модальних дієслів програма буде виглядати наступним чином.

```
>>> cfd = nltk.ConditionalFreqDist(
...           (genre, word)
...           for genre in brown.categories()
...           for word in brown.words(categories=genre))
>>> genres = ['news', 'religion', 'hobbies', 'science_fiction', 'romance', 'humor']
>>> modals = ['can', 'could', 'may', 'might', 'must', 'will']
>>> cfd.tabulate(conditions=genres, samples=modals)
                  can could  may might must will
            news   93   86   66   38   50  389
        religion   82   59   78   12   54   71
         hobbies  268   58  131   22   83  264
science_fiction   16   49    4   12    8   16
         romance   74  193   11   51   45   43
           humor   16   30    8    8    9   13
```

Тексти програм на мові ***Python.***

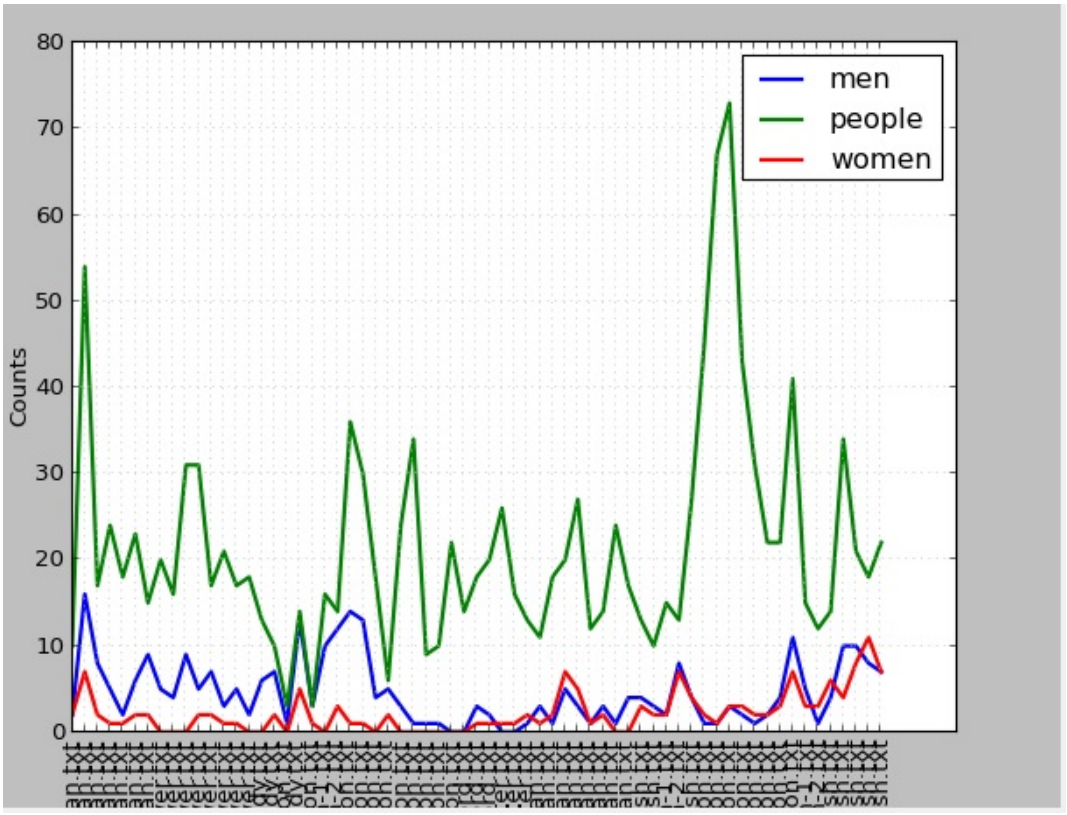
Варіант №7

3. Прочитайте тексти з корпусу State of the Union addresses використовуючи state_union модуль читання. Визначити частоту вживання слів men, women, people в кожному з документі. Як змінилася частота вживання цих слів з часом?

```
import nltk
from nltk.corpus import state_union
state_union.fileids()
word = state_union.words()
fdist = nltk.FreqDist([w for w in word])
vocab = ['men', 'women', 'people']
for i in vocab:
        print i + ':',fdist[i],

cfd = nltk.ConditionalFreqDist(
        (target, fileid)
        for fileid in state_union.fileids()
        for w in state_union.words(fileid)
        for target in ['men', 'women','people']
        if w.lower().startswith(target))
cfd.plot()
```

```
>>>
men: 228 women: 141 people: 1296
```



5. Виберіть пару текстів і дослідіть відмінності між ними (кількість оригінальних слів, багатство мови, жанр). Знайдіть слова, які мають різний зміст в цих текстах, подібно до слова monstrous в Moby Dick та у Sense and Sensibility.

```
import nltk
from nltk.corpus import brown
files = ['cr03','cr07']

for i in files:
    genre = brown.categories(fileids=i)
    words = brown.words(fileids=[i])
    set_i = len(set(words))
    len_i = len(words)/set_i
print 'filename: '+i+'.txt\n', 'genre: '+genre[0]+'\n', 'original words: '+str(set_i)+'\n', 'vocab: '+str(len_i)+'\n-------
```

```
filename: cr03.txt
genre: humor
original words: 1027
vocab: 2
-------
filename: cr07.txt
genre: humor
original words: 724
vocab: 3
-------
```

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

```
>>> import nltk
>>> from nltk.corpus import brown
>>> words = brown.words()
>>> fdist = nltk.FreqDist([w.lower() for w in words])
>>> finalwords=[w for w in words if fdist[w]>3]
>>> len(finalwords)
1000365
>>> finalwords[:20]
[u'said', u'an', u'investigation', u'of', u'recent', u'primary', u'election', u'
produced', u'``', u'no', u'evidence', u"''", u'that', u'any', u'irregularities',
 u'took', u'place', u'.', u'jury', u'further']
```

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

```
>>> import nltk
>>> from nltk.corpus import brown
>>> for category in brown.categories():
        num_words=len(brown.words(categories=category))
        num_vocab=len(set([w.lower() for w in brown.words (categories=category)]))
        print num_words, num_vocab,(num_words/num_vocab), category


69342 8289 8 adventure
173096 17058 10 belles_lettres
61604 9109 6 editorial
68488 8680 7 fiction
70117 7361 9 government
82345 10824 7 hobbies
21695 4755 4 humor
181888 15476 11 learned
110299 13403 8 lore
57169 6463 8 mystery
100554 13112 7 news
39399 5931 6 religion
40704 8069 5 reviews
70022 7883 8 romance
14470 3032 4 science_fiction
```

9. Напишіть програму для знаходження 50 найчастотніших слів в тексті, за виключенням незначущих слів.

```
>>> import nltk
>>> from nltk.corpus import gutenberg
>>> emma=gutenberg.words('austen-emma.txt')
>>> fdist=nltk.FreqDist([w.lower() for w in emma])
>>> vocabulary=fdist.keys()
>>> vocabulary [:50]
[',', '.', 'to', 'the', 'and', 'of', 'i', 'a', 'it', 'her'
, 'was', 'she', ';', 'in', 'not', '"', 'you', 'be', 'he',
'that', 'had', 'but', 'as', '--', 'for', 'have', 'is', 'wi
th', 'very', 'mr', 'his', '."', 'at', "'", 'so', 's', 'emm
a', 'all', 'could', 'would', 'been', 'him', 'no', 'my', 'm
rs', 'on', '.--', 'any', 'do', 'were']
```

12. Напишіть функцію word_freq(), яка приймає слово і назву частини корпуса Brown як аргументи і визначає частоту слова в заданій частині корпуса.

```
>>> import nltk
>>> from nltk.corpus import brown
>>> def word_freq (word,gender):
        fdist=nltk.FreqDist (nltk.corpus.brown.words (categories =gender))
        return word,fdist[word]

>>> word_freq('Russia', 'news')
('Russia', 11)
>>> word_freq('Ukraine', 'news')
('Ukraine', 0)
>>> word_freq('love', 'romance')
('love', 32)
```

Висновок: у цій лабораторній роботі я вивчила основи програмування на мові *Python*, вивчила методи доступу до корпусів текстів, вивчила клас ConditionalFreqDist.