**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

З дисципліни «Комп’ютерна лінгвістика»

на тему

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

ДОСТУП ТА РОБОТА З КОРПУСАМИ ТЕКСТІВ.»

Виконала

Ст.гр.ПРЛм-12

Гречух Ю.

Перевірив

Старший викладач кафедри САПР

Дупак П.

Львів-2015

**Мета роботи** - Вивчення основ програмування на мові *Python*, вивчення методів доступу до корпусів текстів, вивчення класу ConditionalFreqDist.

**Теоретичні відомості**

Вирішення задач обробки текстів природною мовою передбачає використання великих об’ємів лінгвістичних даних, або інишими словами передбачає роботу з корпусами текстів.

1. ***Доступ до корпусів текстів***.

Корпус текстів це великий набір текстів. Багато корпусів розроблені їз збереженням балансу між текстами різних жанрів, або авторів. При роботі з копусами важливо мати засоби доступу як до окремих тексті так і до окремих частин цих тексів а також і до окремих слів.

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()

За допомогою функції .words() можна отримати доступ до всіх слів тесту.

***1.2 Текст з Інтернету***.

Project Gutenberg включає тисячі книжок і він представляє літературну мову. Для роботи з менш формальною мовою NLTK містить набір текстів з Інтернету: тексти з форуму, тексти з фільму Пірати карибського моря, тексти особистих оголошень,телефонні розмови, огляд вин:

>>> from nltk.corpus import webtext

>>> for fileid in webtext.fileids():

... print fileid, webtext.raw(fileid)[:65], '...'

Також в NLTK входить корпус повідомлень з чатів, створений в Naval Postgraduate School для досліджень з метою автоматичного виявлення Інтернет злочинців.

***1.3 Корпус Brown***

Корпус Brown – це перший корпус англійської мови об’ємом один мільйон слів було створено в 1961-1964 роках в університеті Brown. Цей корпус містить тексти з 500 різних джерел, які відповідають різним жанрам, наприклад news, editorial, mystery і т.д. Використовуючи засоби NLTK можна отримати доступ до цього корпусу, як до списку слів або списку речень (кожне речення – список слів). Також доступна можливість вибору текстів окремої категорії або з окремого файлу. Корпус Brown – зручний ресурс для систематичного вивчення відмінностей між жанрами, або іншими словами для дослідження стилістики текстів.

***1.4 Корпус інформаційного агентства Рейтер***.

Корпус Reuters містить 10788 текстів новин загальним об’ємом 1.3 мільйона слів. Всі тексти поділені на категорії за 90 темами і поділені на два набори (тренування та тестування). Такий поділ необхідний для тренування та тестування алгоритмів автоматичного визначення тематики тексту.

***1.5 Корпус інаугураційних примов президентів США.***

Це набір 55 текстів, кожен з яких є промовою одного президента. Цікавою особливістю цього корпуса є можливість дослідити розподіл текстів за часовими проміжками. Назва кожного тексту містить рік проголошення промови і відповідно є можливість доступитися до цієї інформації , доступившись до перших чотирьох символів назви файлу [fileid[:4].

***1.6 Анотовані (розмічені) корпуси текстів.***

Більшість корпусів текстів є лінгвістично анотованими, тобто містять різного типу розмітку – морфологічну, синтаксичну, семантичну, в них можуть бути виділені власні назви, вказані семантичні ролі і т.п. NLTK забезпечує способи доступу до багатьох корпусів і розповсюджується з цими корпусами або їх фрагментами (при використанні NLTK всі корпуси по замовчуванню повинні зберігатися за наступним шляхом C:\nltk\_data\corpora). В Додатку В наведений перелік доступних корпусів текстів та їх короткий опис.

***1.9 Доступ до власних корпусів текстів***.

При наявності власного набору текстових файлів, до них також можна організувати доступ використовуючи вище перелічені методи попередньо використавши клас NLTK PlaintextCorpusReader .

1. ***Умовний частотний розподіл. Клас ConditionalFreqDist.***

Якщо тексти в корпусі поділені на різні категорії, (за жанром, тематикою, авторами) то можна побудувати частотні розподіли для кожної з категорій. Такі дані дозволяють досліджувати відмінності між жанрами. Умовний частотний розподіл це набір частотних розподілів, кожен з яких відповідає певній «умові». Такою умовою може бути категорія тексту.

1. ***Умови і події.***

Частотний розподіл визначає числові значення для кожної події (подіями можемо вважати вживання слів в тексті). Умовний частотний розподіл поєднує в пари кожну подію та умову. Замість обробки послідовності слів ( #1) обробляються послідовності пар (#2).

>>> text = ['The', 'Fulton', 'County', 'Grand', 'Jury', 'said', ...]  #1

>>> pairs = [('news', 'The'), ('news', 'Fulton'), ('news', 'County'), ...]  #2

Кожна пара відповідає шаблону (condition, event). Якщо розглядати корпус Brown за жанрами то отримаємо 15 умов (одна для жанру) і 1161192 подій (одна для слова).

Метод plot() а також метод tabulate() дозволяють визначати, які з умов будуть відображатися на екрані за допомогою параметра conditions= parameter. Так само визначають кількість прикладів для відображення за допомогою параметра samples= parameter.

1. ***Поняття функції та модуля***.

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.

Множина змінних і функцій збережених у файлі називаються в Python – модулем. Множина пов’язаних між собою модулів називають – пакетом. Програма обробки корпуса Brown це є приклад модуля, а множина програм для роботи зі всіма корпусами це є приклад пакету. NLTK це множина пакетів, яку називають бібліотекою.

**Практичні завдання**

**Завдання 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 документів. Як змінилася частота вживання цих слів з часом?.

*Програма:*

import nltk

from nltk.corpus import state\_union

cfd = nltk.ConditionalFreqDist(

(fileid, word)

for fileid in state\_union.fileids()

for word in state\_union.words(fileids=fileid))

fileid = state\_union.fileids()

word = ['men', 'women', 'people']

print(cfd.tabulate(conditions=fileid, samples=word))

*Результат:*

men women people

1945-Truman.txt 2 2 10

1946-Truman.txt 12 7 49

1947-Truman.txt 7 2 12

1948-Truman.txt 4 1 22

1949-Truman.txt 2 1 15

1950-Truman.txt 6 2 15

1951-Truman.txt 8 2 9

1953-Eisenhower.txt 3 0 17

1954-Eisenhower.txt 2 0 15

1955-Eisenhower.txt 4 0 26

1956-Eisenhower.txt 2 2 30

1957-Eisenhower.txt 5 2 11

1958-Eisenhower.txt 2 1 19

1959-Eisenhower.txt 4 1 11

1960-Eisenhower.txt 2 0 10

1961-Kennedy.txt 6 0 10

1962-Kennedy.txt 6 2 10

1963-Johnson.txt 0 0 3

1963-Kennedy.txt 8 5 12

1964-Johnson.txt 3 1 3

1965-Johnson-1.txt 7 0 16

1965-Johnson-2.txt 11 3 14

1966-Johnson.txt 12 1 35

1967-Johnson.txt 11 1 25

1968-Johnson.txt 4 0 17

1969-Johnson.txt 5 2 6

1970-Nixon.txt 2 0 23

1971-Nixon.txt 1 0 31

1972-Nixon.txt 1 0 7

1973-Nixon.txt 0 0 9

1974-Nixon.txt 0 0 19

1975-Ford.txt 0 0 13

1976-Ford.txt 3 1 18

1977-Ford.txt 2 1 17

1978-Carter.txt 0 1 26

1979-Carter.txt 0 1 15

1980-Carter.txt 1 2 11

1981-Reagan.txt 1 1 11

1982-Reagan.txt 1 1 17

1983-Reagan.txt 3 7 19

1984-Reagan.txt 3 5 23

1985-Reagan.txt 1 1 12

1986-Reagan.txt 2 2 14

1987-Reagan.txt 1 0 24

1988-Reagan.txt 1 0 16

1989-Bush.txt 2 3 13

1990-Bush.txt 3 2 9

1991-Bush-1.txt 2 2 13

1991-Bush-2.txt 7 7 13

1992-Bush.txt 4 4 26

1993-Clinton.txt 1 2 45

1994-Clinton.txt 1 1 63

1995-Clinton.txt 1 3 73

1996-Clinton.txt 2 3 40

1997-Clinton.txt 1 2 30

1998-Clinton.txt 2 2 22

1999-Clinton.txt 2 3 22

2000-Clinton.txt 5 6 41

2001-GWBush-1.txt 3 3 14

2001-GWBush-2.txt 1 2 12

2002-GWBush.txt 3 5 14

2003-GWBush.txt 6 4 33

2004-GWBush.txt 6 8 21

2005-GWBush.txt 8 11 18

2006-GWBush.txt 7 7 22

**Завдання 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.*

*Програма:*

import nltk

from nltk.corpus import state\_union

from nltk.corpus import gutenberg

from nltk.corpus import PlaintextCorpusReader

raw = gutenberg.raw("burgess-busterbrown.txt")

Johnson1963=nltk.corpus.state\_union.words('1963-Johnson.txt')

emma=nltk.corpus.gutenberg.words('austen-emma.txt')

num\_words1=len(emma)

num\_vocab1=len(set([w.lower() for w in emma]))

num\_words2=len(Johnson1963)

num\_vocab2=len(set([w.lower() for w in Johnson1963]))

print('Words in general from text 1:',num\_words1)

print('Unique words from text 1:', num\_vocab1 )

print ('Language richness of text 1:', int(num\_vocab1/num\_words1\*100),'%')

print('Words in general from text 2:', num\_words2)

print('Unique words from text 2:', num\_vocab2)

print ('Language richness of text 2:', int(num\_vocab2/num\_words2\*100), '%')

*Результат:*

Words in general from text 1: 192427

Unique words from text 1: 7344

Language richness of text 1: 3 %

Words in general from text 2: 1878

Unique words from text 2: 585

Language richness of text 2: 31 %

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

*Програма:*

import nltk

from nltk.corpus import brown

from nltk import FreqDist

words=brown.words()

words2=''

fdist=FreqDist(words)

for word in words:

if fdist[word]>=3:

print(word)

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

*Програма:*

import nltk

from nltk.corpus import brown

print ('genre','|', 'total words', '|', 'original words')

for genre in brown.categories():

num\_words = len(brown.words(categories=genre))

num\_vocab = len(set([w.lower() for w in brown.words(categories=genre)]))

print (genre,'|', num\_words, '|',num\_vocab)

*Результат:*

genre | total words | original words

adventure | 69342 | 8289

belles\_lettres | 173096 | 17058

editorial | 61604 | 9109

fiction | 68488 | 8680

government | 70117 | 7361

hobbies | 82345 | 10824

humor | 21695 | 4755

learned | 181888 | 15476

lore | 110299 | 13403

mystery | 57169 | 6463

news | 100554 | 13112

religion | 39399 | 5931

reviews | 40704 | 8069

romance | 70022 | 7883

science\_fiction | 14470 | 3032

**Задача 11:** Напишіть програму для створення таблиці частот слів для різних жанрів. Знайдіть слова чия присутність або відсутність є характерною для певних жанрів (подібно до модальних дієслів).

*Програма:*

import nltk

from nltk.corpus import brown

modals = ['can', 'could', 'may', 'might', 'must', 'will']

max\_fdist=0;

for genre in brown.categories():

text = brown.words(categories=genre)

fdist = nltk.FreqDist([w.lower() for w in text])

print(genre)

for m in modals:

print ("\t", m + ':',fdist[m],"\n")

for m in modals:

if fdist[m]>max\_fdist:

max\_fdist=fdist[m];

word=m;

print("The most frequent word or genre", genre," is ", word)

max\_fdist=0;

*Результат:*

adventure

can: 48

could: 154

may: 7

might: 59

must: 27

will: 51

The most frequent word or genre adventure is coul

belles\_lettres

can: 249

could: 216

may: 221

might: 113

must: 171

will: 246

The most frequent word or genre belles\_lettres is can

editorial

can: 124

could: 57

may: 79

might: 39

must: 55

will: 235

The most frequent word or genre editorial is will

…

**Задача 12.** Напишіть функцію word\_freq(), яка приймає слово і назву частини корпуса Brown як аргументи і визначає частоту слова в заданій частині корпуса.

*Програма:*

import nltk

from nltk import FreqDist

from nltk.corpus import brown

def word\_freq(word, part):

words=brown.words(fileids=part)

f=FreqDist(words)

return f[word]

print(word\_freq("city", "ca05"))

*Результат:* 13

**Висновок:** я ознайомилась з різними видами корпусів, а також їх представниками, а також з класом ConditionalFreqDist, та виконала індивідуальні завдання, застосувавши ці знання на практиці.