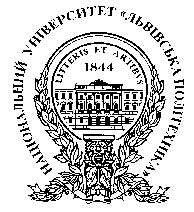
МІНІСТЕРСТВО ОСВІТИ І НАУКИ, МОЛОДІ ТА СПОРТУ УКРАЇНИ

Національний Університет “Львівська політехніка”



Лабораторна робота № 3

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

***ДОСТУП ТА РОБОТА З КОРПУСАМИ ТЕКСТІВ.***

Виконав:

студент групи ПРЛс-11

Форманюк А.А.

Прийняв:

асистент

Дупак Б.П.

Львів 2015

**Мета:**

* 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 інишими 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.

1. ***Доступ до корпусів текстів***.

Корпус текстів це великий набір текстів. Багато корпусів розроблені їз збереженням балансу між текстами різних жанрів, або авторів. В попередній лабораторній роботі ми працювали з промовами президентів США, які є частиною корпуса 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 копусами важливо мати засоби доступу як до окремих тексті так і до окремих частин цих тексів а також і до окремих слів.

***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], '...'

...

firefox.txt Cookie Manager: "Don't allow sites that set removed cookies to se...

grail.txt SCENE 1: [wind] [clop clop clop] KING ARTHUR: Whoa there! [clop...

overheard.txt White guy: So, do you have any plans for this evening? Asian girl...

pirates.txt PIRATES OF THE CARRIBEAN: DEAD MAN'S CHEST, by Ted Elliott & Terr...

singles.txt 25 SEXY MALE, seeks attrac older single lady, for discreet encoun...

wine.txt Lovely delicate, fragrant Rhone wine. Polished leather and strawb...

Також в NLTK входить корпус повідомлень з чатів, створений в Naval Postgraduate School для досліджень з метою автоматичного виявлення Інтернет злочинців. Цей корпус містить 10000 анонімних повідомлень в яких імена користувачів замінені за шаблоном "UserNNN" а також видалена інша персональна інформація, Корпус організований ,як 15 окремих файлів, кожен з яких містить декілька сотень повідомлень з певною датою створення та вікових даних авторів (підлітки, 20ти, 30ти та 40ка річні, дорослі). Назва файла містить інформацію про дату, вікову групу та кількість повідомлень, наприклад файл 10-19-20s\_706posts.xml містить 706 повідомлень двадцятирічних дописувачів від 19 жовтня 2006 року.

>>> from nltk.corpus import nps\_chat

>>> chatroom = nps\_chat.posts('10-19-20s\_706posts.xml')

>>> chatroom[123]

['i', 'do', "n't", 'want', 'hot', 'pics', 'of', 'a', 'female', ',',

'I', 'can', 'look', 'in', 'a', 'mirror', '.']

***1.3 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.

>>> from nltk.corpus import reuters

>>> reuters.fileids()

['test/14826', 'test/14828', 'test/14829', 'test/14832', ...]

>>> reuters.categories()

['acq', 'alum', 'barley', 'bop', 'carcass', 'castor-oil', 'cocoa',

'coconut', 'coconut-oil', 'coffee', 'copper', 'copra-cake', 'corn',

'cotton', 'cotton-oil', 'cpi', 'cpu', 'crude', 'dfl', 'dlr', ...]

На відміну від корпуса Brown, категорії текстів в цьому корпусі можуть накладатися одна на одну, оскільки тематика новин (газетних публікацій) переважно торкається багатьох тем. Засобами NLTK можна звернутися до тем, яких торкаються в одному або декількох текстах або навпаки дізнатися весь перелік текстів, які належать до певної категорії.

>>> reuters.categories('training/9865')

['barley', 'corn', 'grain', 'wheat']

>>> reuters.categories(['training/9865', 'training/9880'])

['barley', 'corn', 'grain', 'money-fx', 'wheat']

>>> reuters.fileids('barley')

['test/15618', 'test/15649', 'test/15676', 'test/15728', 'test/15871', ...]

>>> reuters.fileids(['barley', 'corn'])

['test/14832', 'test/14858', 'test/15033', 'test/15043', 'test/15106',

'test/15287', 'test/15341', 'test/15618', 'test/15618', 'test/15648', ...]

***1.4 Корпус інаугураційних примов президентів США.***

Знайомлячись з бібліотекою програм NLTK ми працювали з цим корпусом і розглядали весь корпус як один текст, що давало можливість знайти місце окремого слова в текстах промов починаючи від першого слова першої промови. Насправді корпус 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].

>>> from nltk.corpus import inaugural

>>> inaugural.fileids()

['1789-Washington.txt', '1793-Washington.txt', '1797-Adams.txt', ...]

>>> [fileid[:4] for fileid in inaugural.fileids()]

['1789', '1793', '1797', '1801', '1805', '1809', '1813', '1817', '1821', ...]

***1.5 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.

***1.6 Корпуси іншомовних текстів***.

NLTK включає та має засоби роботи з корпусами текстів іншими мовами крім англійської. Для роботи з цими корпусами потрібно попередньо ознайомитися з питаннями кодування символів в Python (Лабораторна робота №4).

>>> nltk.corpus.cess\_esp.words()

['El', 'grupo', 'estatal', 'Electricit\xe9\_de\_France', ...]

>>> nltk.corpus.floresta.words()

['Um', 'revivalismo', 'refrescante', 'O', '7\_e\_Meio', ...]

>>> nltk.corpus.indian.words('hindi.pos')

['\xe0\xa4\xaa\xe0\xa5\x82\xe0\xa4\xb0\xe0\xa5\x8d\xe0\xa4\xa3',

'\xe0\xa4\xaa\xe0\xa5\x8d\xe0\xa4\xb0\xe0\xa4\xa4\xe0\xa4\xbf\xe0\xa4\xac\xe0\xa4

\x82\xe0\xa4\xa7', ...]

>>> nltk.corpus.udhr.fileids()

['Abkhaz-Cyrillic+Abkh', 'Abkhaz-UTF8', 'Achehnese-Latin1', 'Achuar-Shiwiar-Latin1',

'Adja-UTF8', 'Afaan\_Oromo\_Oromiffa-Latin1', 'Afrikaans-Latin1', 'Aguaruna-Latin1',

'Akuapem\_Twi-UTF8', 'Albanian\_Shqip-Latin1', 'Amahuaca', 'Amahuaca-Latin1', ...]

>>> nltk.corpus.udhr.words('Javanese-Latin1')[11:]

[u'Saben', u'umat', u'manungsa', u'lair', u'kanthi', ...]

Останній з розглянутих в попередньому прикладів корпусів (udhr) це набір текстів різними мовами (300 мов) Декларації прав людини.

**Виконання роботи**

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

>>> import nltk

>>> from nltk.corpus import gutenberg

>>> sense = gutenberg.words ('austen-sense.txt')

>>> sense1 = nltk.Text(sense)

>>> print sense1.concordance('however')

Building index...

Displaying 25 of 155 matches:

hters . He meant not to be unkind , **however** , and , as a mark of his affection

e condition of visitors . As such , **however** , they were treated by her with qui

le ." His wife hesitated a little , **however** , in giving her consent to this pla

urned Mrs . John Dashwood . " But , **however** , ONE thing must be considered . Wh

can ever afford to live in . But , **however** , so it is . Your father thought on

ce inquiry or remark . Conversation **however** was not wanted , for Sir John was v

sary to the happiness of both ; for **however** dissimilar in temper and outward be

al engagements at home and abroad , **however** , supplied all the deficiencies of

s silent and grave . His appearance **however** was not unpleasing , in spite of hi

n their own house . One consolation **however** remained for them , to which the ex

in the country ? That is good news **however** ; I will ride over tomorrow , and a

ever so rich . I am glad to find , **however** , from what you say , that he is a

t to the excellence of such works , **however** disregarded before . Their taste wa

ly excited by her sister ; and that **however** a general resemblance of dispositio

d Marianne . " Do not boast of it , **however** ," said Elinor , " for it is injust

t will be any satisfaction to you , **however** , to be told , that I believe his c

wo wives , I know not . A few years **however** will settle her opinions on the rea

n his side impossible . His concern **however** was very apparent ; and after expre

d her husband and mother . The idea **however** started by her , was immediately pu

are determined on anything . But , **however** , I hope you will think better of i

I can guess what his business is , **however** ," said Mrs . Jennings exultingly .

o unfortunate an event ; concluding **however** by observing , that as they were al

r . Willoughby ." " Mr . Willoughby **however** is the only person who can have a r

sed in him . There is great truth , **however** , in what you have now urged of the

iced by him ." " Do not blame him , **however** , for departing from his character

None

>>>

Вживання слова **however** на початку речення не виявлено, зазвичай воно є відокремлене комами як вставне слово.

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

>>> 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', 'government', 'hobbies',

'humor', 'learned', 'lore', 'mystery', 'news', 'religion', 'reviews', 'romance',

'science\_fiction']

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

>>> cfd.tabulate(conditions=genres, samples=modals)

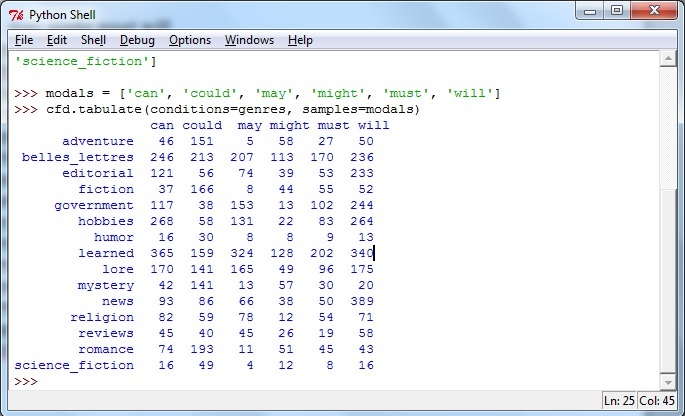


Рис.1 Результат виконання програми

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

>>> import nltk

>>> from nltk.corpus import brown

>>> text = brown.words()

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

>>> r = sorted ([w for w in set (text) if fdist[w]>=3 and w.isalpha()])

>>> print (r[:30])

['A', 'ABO', 'ADC', 'AIA', 'AID', 'AIMO', 'AM', 'AP', 'AWOC', 'Aaron', 'Abbe', 'Abbey', 'Abe', 'Abel', 'Abolition', 'About', 'Above', 'Abraham', 'Abstract', 'Abstraction', 'Academy', 'Acala', 'Accacia', 'According', 'Accordingly', 'Acey', 'Achievement', 'Acropolis', 'Across', 'Act']

>>>

1. 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 genre in brown.categories():

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

num\_original = len (set(brown.words (categories=genre)))

print num\_words, num\_original, int(num\_words/num\_original), genre

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

>>>

1. 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).

import nltk

from nltk.corpus import brown

months = ['september', 'october', 'november', 'december', 'january', 'fabruary', 'march', 'april', 'may', 'june', 'july', 'august']

genres = ['news', 'religion', 'hobbies', 'science\_fiction', 'romance', 'humor']

cfd = nltk.ConditionalFreqDist(

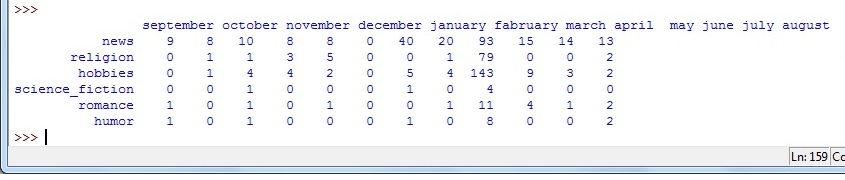
(genre, month)

for month in months

for genre in genres

for word in brown.words(categories=genre)if word.lower() ==month )

cfd.tabulate(conditions=genres, samples=months)



**Рис.2 Результат виконання програми**

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

text = 'This developed technique is used for intracranial surgery that does not require opening the skull, which is also significant'

def hedge(text):

r = []

text = text.split()

for i in text:

if text.index(i) in range(2, len(text), 2):

r.append('like ' + i)

else:

r.append(i)

w\_new = ''

for word in r:

w\_new+=word + ' '

print w\_new

hedge(text)

This developed like technique is like used for like intracranial surgery like that does like not require like opening the like skull, which is also like significant

>>>

**Висновок**: на лабораторній роботі я вивчив основ програмування на мові *Python*, методи доступу до корпусів текстів та клас ConditionalFreqDist.