МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ “ЛЬВІВСЬКА ПОЛІТЕХНІКА”

ІНСТИТУТ КОМП’ЮТЕРНИХ НАУК ТА ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ

*Кафедра “Системи автоматизованого проектування”*



Звіт

до лабораторної роботи №9

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

АВТОМАТИЧНИЙ МОРФОЛОГІЧНИЙ АНАЛІЗ (частина1)*»*

з дисципліни “Комп’ютерна лінгвістика”

Виконала:

студентка групи ПРЛм-11

Павлів І.О.

Прийняв:

викладач

Дупак Б.П.

Львів-2015

**Мета:**

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

**КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ**

***1. Вступ.***

Процес класифікації слів за їх приналежністю до частини мови і їх відповідне маркування називається морфологічним аналізом (tagging, POS tagging). В загальному для назв цих груп слів вживаються терміни – класи слів, лексичні категорії, частини мови.

Перелік тегів, який використовується для цієї специфічної задачі називається набором тегів. Наголос в цій та наступній лабораторних роботах буде зроблено на вивченні використання тегів і здійснення автоматичного морфологічного аналізу. Автоматичний морфологічний аналіз це дуже важливий та цінний етап опрацювання текстів природною мовою, результати якого мають широке застосування.

## *2.   Поняття автоматичного морфологічного а*налізатора

Морфологічний аналізатор ( POS-tagger), це програма, яка обробляє послідовність слів і ставить у відповідність до кожного з них, відповідний тег (тег відповідає певному набору морфологічних характеристик :

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | |  | >>>**import** **nltk**  >>> **text = nltk.word\_tokenize("And now for something completely different")**  **>>> nltk.pos\_tag(text)**  **[('And', 'CC'), ('now', 'RB'), ('for', 'IN'), ('something', 'NN'),**  **('completely', 'RB'), ('different', 'JJ')]** | |

Програма морфологічного аналізу автоматично встановила що and це CC, ( coordinating conjunction – сполучник сурядності); now та completely це RB, (adverbs – прислівник); for це IN, (preposition - прийменник); something це NN, ( noun- іменник); та different це JJ, (adjective – прикметник).

В NLTK задокументовано пояснення кожного з тегів і за допомогою простого запиту, наприклад nltk.help.upenn\_tagset('RB') (nltk.help.brown\_tagset('RB')), або регулярного виразу, наприклад nltk.help.upenn\_brown\_tagset('NN.\*') (nltk.help.brown\_tagset('NN.\*')) можна переглянути ці пояснення.

Розглянемо наступний приклад здійснення морфологічного аналізу омонімів:

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | |  | **>>> text = nltk.word\_tokenize("They refuse to permit us to obtain the refuse permit")**  **>>> nltk.pos\_tag(text)**  **[('They', 'PRP'), ('refuse', 'VBP'), ('to', 'TO'), ('permit', 'VB'), ('us', 'PRP'),('to', 'TO'), ('obtain', 'VB'), ('the', 'DT'), ('refuse', 'NN'), ('permit', 'NN')]** | |

Омоніми refuse та permit зустрічаються як дієслова теперішнього часу (VBP) та іменники (NN). Наприклад refUSE це дієслово заперечення ( "deny,") а REFuse це іменник в значенні "trash" (але вони не омофони). Отже, потрібно знати, яке з цих слів використовується для його правильної вимови в тексті. Не знаючи, яка це частина мови не можна бути впевненому у правильній вимові слова *(contest, insult, present, rebel, suspect)* . Наприклад *wing/nn* як в *wind blew*, вимовляється з коротким голосним, тоді як *wind/vb*, як в *wind a clock* вимовляється з довгим голосним.

***Виконати самостійно***. Знайти слова, які можуть бути і іменниками і дієсловами і не мають відмінностей у вимові (наприклад ski , race) .

Здійснивши простий аналіз розподілу слів у тексті, можна висунути гіпотезу про приналежність цього слова до певної лексичної категорії або поставити йому у відповідність певний тег. Розглянемо результати такого аналізу для слів woman (іменник), bought (дієслово), over (прийменник), та the (означальне слово). Метод text.similar() для заданого слова w, знаходить всі його контексти w1w w2, та знаходить всі слова w' в тексті, що зустрічаються в таких самих контекстах, w1w'w2.

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | |  | **>>> text = nltk.Text(word.lower() for word in nltk.corpus.brown.words())**  **>>> text.similar('woman')**  **Building word-context index...**  **man time day year car moment world family house country child boy**  **state job way war girl place room word**  **>>> text.similar('bought')**  **made said put done seen had found left given heard brought got been**  **was set told took in felt that**  **>>> text.similar('over')**  **in on to of and for with from at by that into as up out down through**  **is all about**  **>>> text.similar('the')**  **a his this their its her an that our any all one these my in your no**  **some other and** | |

Слова, які вживаються в аналогічному до woman контексті вказують на те що це іменник; bought – переважно дієслова; over - прийменники; the – означальні слова . Програма морфроргічного аналізу може коректно ідентифікувати теги цих слів при врахуванні їх контексту в реченні. Наприклад , The woman bought over $150,000 worth of clothes.

Автоматичний морфологічний аналіз також допомагає передбачити частину мови попередньо невідомих слів. Наприклад, якщо зустрічається слово *blogging* можна передбачити що це дієслово, з основою *blog* якщо воно вживається після допоміжного слова *to be* (*he was blogging)* scrobbling - scrobble, ( he was scrobbling).

## *3.   Морфологічно розмічені корпуси*

### *3.1 Представлення промаркованих слів*

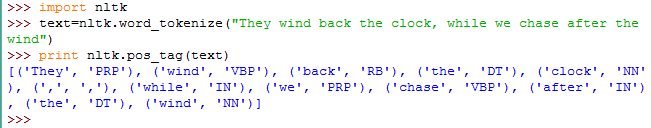
За домовленістю в NLTK промарковані слова (tokens) представляються з використанням типу даних – кортеж, Кортеж містить слово і тег. Для створення таких спеціальних кортежів з стандартної стрічки промаркованих слів потрібно використовувати функцію str2tuple():

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | |  | **>>> tagged\_token = nltk.tag.str2tuple('fly/NN')**  **>>> tagged\_token**  **('fly', 'NN')**  **>>> tagged\_token[0]**  **'fly'**  **>>> tagged\_token[1]**  **'NN'** | |

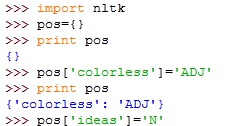
Отримати список промаркованих слів можна прямо зі стрічки. Перший крок , це поділ стрічки на окремі слова для отримання доступу до окремих стрічок word/tag а далі перетворення кожної з них в кортеж (використовуючи *str2tuple*()).

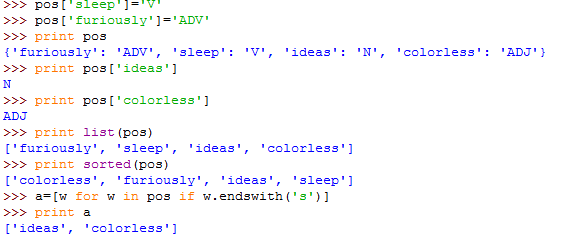
**Варіант №13**

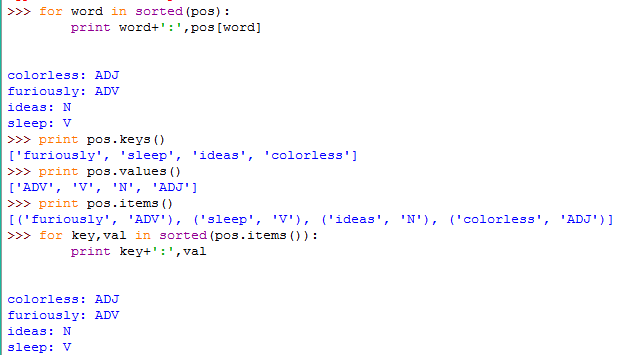
1. Токенізувати та здійснити морфологічний аналіз наступного речення: They wind back the clock, while we chase after the wind. Які відмінності у вимові слів пов’язані з їх морфологічними характеристиками.

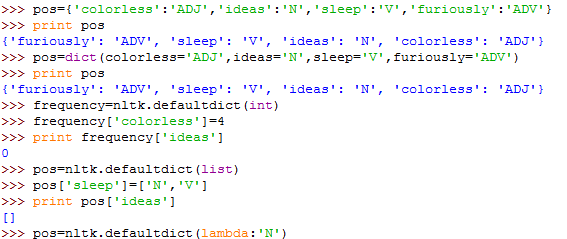


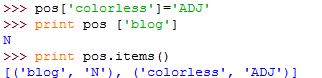
1. Опрацювати всі приклади з методичних вказівок по роботі зі словниками. Що станеться, якщо доступитися до неіснуючого запису звичайного словника та словника по замовчуванню?





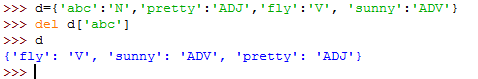




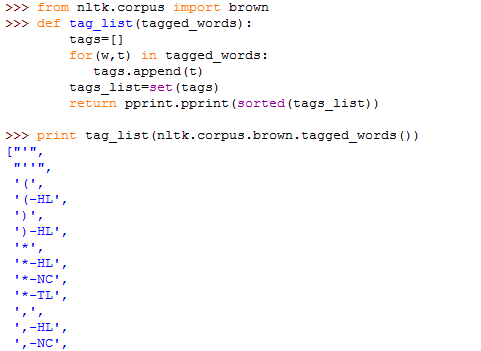


У випадку доступу до значення неіснуючого запису звичайного словника, отримуємо помилку. Якщо те саме зробимо з словником по замовчуванню, помилки не виявимо, повернеться значення за замовчуванням. Значення по замовчуванню це насправді функції, які перетворюють об’єкти у вказаний тип.

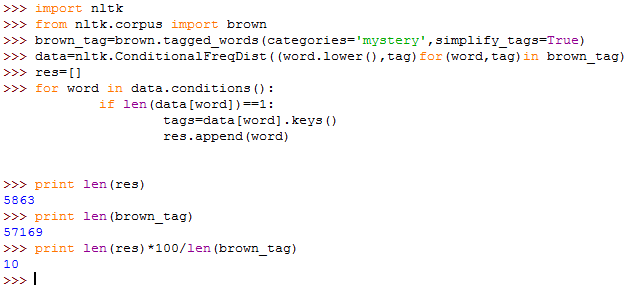
1. Спробуйте видалити запис зі словника d, використовуючи del d['abc'].



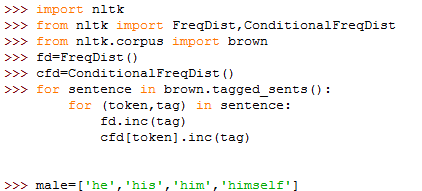
1. Використовуючи sorted() та set() отримайте відсортований список всіх тегів корпуса Brown без їх дублювання.

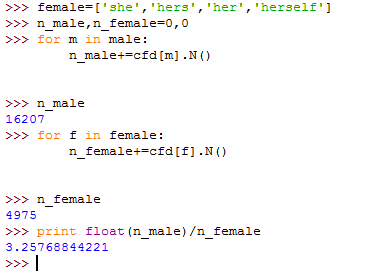


1. Напишіть програму для збору статистичних даних по розмічених корпусах і відповіді на наступне запитання: який відсоток типів слів (types) завжди маркуються тими самими тегами.

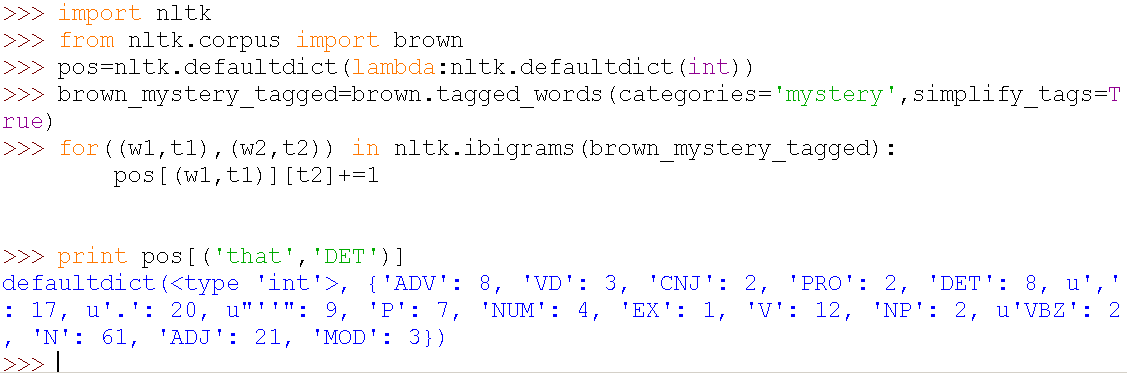


1. Напишіть програми для знаходження слів та словосполучень згідно відповідних їм тегів для відповіді на наступне питання: яке співвідношення між займенниками (чоловічими і жіночими).





1. Написати програму побудови словника, записами якого будуть набори словників. Використовуючи створений словник, збережіть у ньому набори можливих тегів, які зустрічаються після заданого слова з певним тегом, наприклад wordi → tagi → tagi+1.



**Висновок:**

я вивчила елементи бібліотеки прикладних програм nltk, для опрацювання текстів природною мовою; автоматичний морфологічний аналіз.