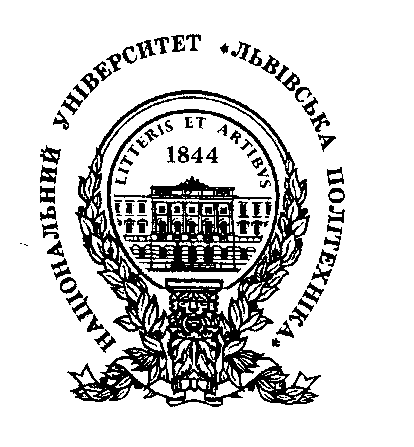
Міністерство освіти та науки України

Національний університет «Львівська Політехніка»

Інститут комп’ютерних наук та інформаційних технологій

Кафедра САПР



Лабораторна робота №4

на тему:

ВИВЧЕННЯ БІБЛІОТЕКИ ПРИКЛАДНИХ ПРОГРАМ NLTK, ДЛЯ ОПРАЦЮВАННЯ ТЕКСТІВ ПРИРОДНОЮ МОВОЮ.ДОСТУП ТА РОБОТА З ЛЕКСИЧНИМИ РЕСУРСАМИ.

Виконав:

студент ПРЛм-12

Білик Д.Г.

Перевірив:

Дупак Б.П.

Львів 2015

**МЕТА РОБОТИ**

* Вивчення основ програмування на мові *Python*.
* Вивчення методів доступу та роботи з лексичним ресурсами.
* Семантичний словник англійської мови WordNet.

**КОРОТКІ ТЕОРЕТИЧНІ ВІДОМОСТІ**

1. **Поняття функції та модуля.**

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.

1. **Генерація випадкового тексту за допомогою біграмів.**

Умовний частотний розподіл можна використати для побудови таблиці біграмів (пар слів). Функція NLTK bigrams() , як аргумент бере список слів і повертає список послідовних пар слів.

>>> sent = ['In', 'the', 'beginning', 'God', 'created', 'the', 'heaven',

... 'and', 'the', 'earth', '.']

>>> nltk.bigrams(sent)

[('In', 'the'), ('the', 'beginning'), ('beginning', 'God'), ('God', 'created'),

('created', 'the'), ('the', 'heaven'), ('heaven', 'and'), ('and', 'the'),

('the', 'earth'), ('earth', '.')]

В наступному прикладі кожне слово розглядається, як умова і для кожного з них будується частотний розподіл по словам, які слідують після нього. Функція generate\_model() містить простий цикл для генерації тексту. Коли ця функція викликається то одним з її аргументів є слово – початковий контекст (у прикладі 'living') . В циклі поточне значення змінної word виводиться на екран і її значення замінюється на слово, яке найчастіше є наступним словом (max()) . На наступному кроці циклу вже це слово буде наступним контекстом. Запропонований підхід генерації тексту швидко приводить до зациклювання, якого можна уникнути якщо вибирати наступні слова випадковим чином.

1. **Лексичні ресурси NLTK.**

Лексичний ресурс або просто словник це набір слів тa/або словосполучень, які асоціюються з такою інформацією, як частина мови та опис значення. Лексичні ресурси є вторинними по відношенню до текстів і зазвичай створюються і вдосконалюються з використанням текстів. Наприклад, якщо визначити текст my\_text тоді vocab = sorted(set(my\_text)) побудує словник тексту my\_text, word\_freq = FreqDist(my\_text) визначить частоту кожного слова в тексті. vocab та word\_freq – приклад простих лексичних ресурсів. Так само конкорданс дає інформацію про використання слів і ця інформація може бути використана при побудові словників. Стандартна термінологія для словників (англ. мова) представлена на Рис.1. Словникова стаття містить основне слово (лему), та відповідну інформацію (частина мови значення слова).

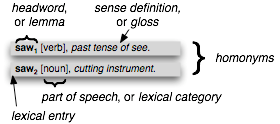
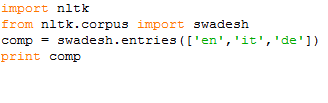


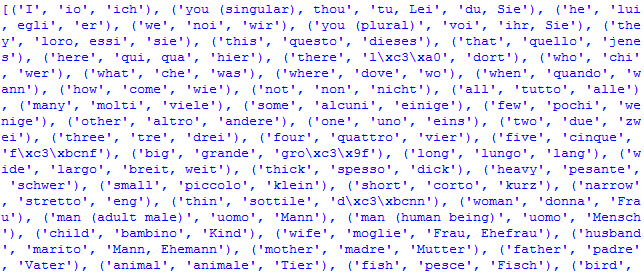
Рис.1. Термінологія англійської мови для записів словників.

Найпростіший словник це відсортований список слів. Досконаліші словники містять складну структуру записів та зв’язків між ними. В цій лабораторні роботі будуть розглянуті лексичні ресурси, які розповсюджуються разом з NLTK.

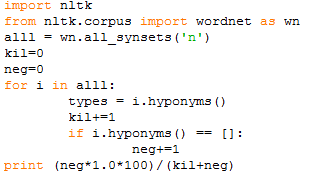
**ТЕКСТИ ПРОГРАМ НА МОВІ PYTHON**

**Варіант 13**

3.2. Використовуючи компаративний словник знайти близькі слова для німецької, італійської та англійської мов. Чи можуть отримані результати використовуватися для здійснення перекладу?

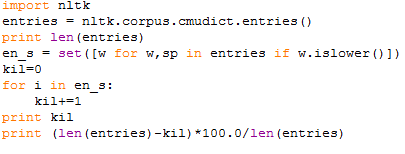


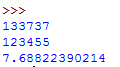
3.5. Який відсоток синсетів іменників не мають гіпонімів? До всіх синсетів можна доступитися за допомогою wn.all\_synsets('n').



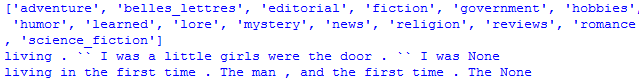
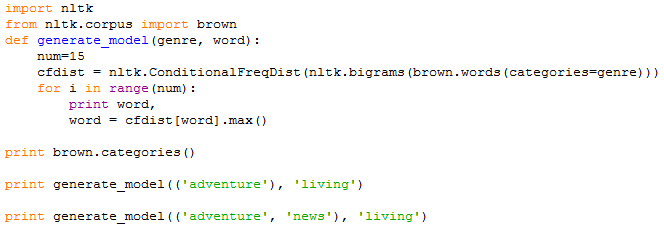


3.4. Здійснити аналіз словника вимов. Знайти скільки різних слів він містить. Який відсоток слів з цього словника можуть мати різну вимову?

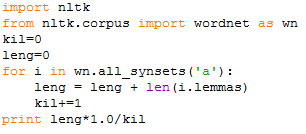




3.9. Модифікувати програму генерації випадкового тексту для виконання наступного: тренування програми на текстах двох різних жанрів та генерації тексту об’єднаного жанру.

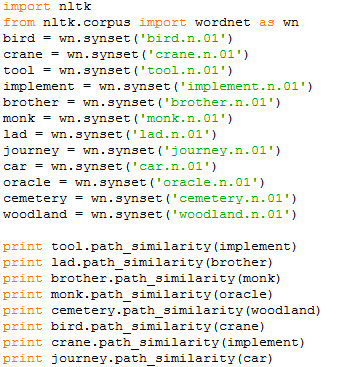


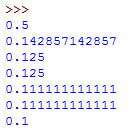
3.11. Полісемія - це явище коли одне слово має декілька значень ( іменник dog має 7 значень, кількість яких визначити можна як len(wn.synsets('dog', 'n'))). Знайдіть середнє значення полісемії для прикметників.





3.15. Використовуючи один з методів визначення подібності слів побудуйте відсортований по спаданню список значень подібності для наступних пар слів: bird-crane, tool-implement, brother-monk, lad-brother, crane-implement, journey-car, monk-oracle, cemetery-woodland.





**ВИСНОВОК**

На цій лабораторній роботі, ми вивчили основи програмування на мові Python, методи доступу та роботи з лексичним ресурсами та ознайомились з семантичним словником англійської мови WordNet.