Основы работы с библиотекой nltk (Natural Language Toolkit)

nltk – это великолепная библиотека для обработки текстов на естественных языках,
к ней прилагаются готовые загружаемые наборы исходных данных, т.н. «корпуса»,
а также программные интерфейсы для удобного доступа к этим данным.

1. Начало работы

Перед работой необходимо установить необходимые данные, пакеты, использовав метод download(). В появившемся окне выбрать необходимые пакеты, модули либо всё вместе. Можно также загружать модули и пакеты при помощи nltk.download('name'), где name - имя модуля.

```
>>> import nltk
>>> nltk.download()
>>>nltk.download('words')
```

2. Токенизация

Для поиска слова в тексте, для начала необходимо представить текст в виде списка слов. Данная операция имеет название токенизации. Для этого выполняем:

```
>>>s = 'We can modify an element of a list by assigning to one of its index values.'
>>>result_list = nltk.word_tokenize(s)
```

```
[out]:
['We', 'can', 'modify', 'an', 'element', 'of', 'a', 'list', 'by', 'assigning',
'to', 'one', 'of', 'its', 'index', 'values', '.']
```

3. Подсчет количества повторений

Для определения частоты повторения слов в тексте используется объект nltk.FreqDist
При создании экземпляра объекта в качестве аргумента передаётся токинезированный текст, т.е. список слов этого текста (иначе при передаче самого текста в виде строки, будет посчитана частота вхождения каждого символа). Пример:

```
>>>s = 'We can modify an element of a list by assigning to one of its index values.'
>>>text = nltk.word_tokenize(s)
>>>freq = nltk.FreqDist(text)
>>>#10 самых часто встречаемых слов
>>>print(freq.most_common(10))
```

```
[out]:
[('of', 2), ('its', 1), ('one', 1), ('values', 1), ('a', 1), ('to', 1),
('We', 1), ('by', 1), ('index', 1), ('list', 1)]
```

Метод nltk.FreqDist().hapaxes() возвращает слова встречающиеся лишь раз.

4. Поиск слов определенного размера

Для поиска слов определенного размера используется следующая методика:
[w for w in V if p(w)]

w - слово в тексте(словаре)
V - список слов
p - свойство для отбора
Пример:

```
>>>s = 'We can modify an element of a list by assigning to one of its index values.'
>>>text = nltk.word_tokenize(s)
>>>long_words = [w for w in text if len(w) > 5]
>>>print(long_words)
```

[out]:
['modify', 'element', 'assigning', 'values']

Для подсчета количества слов одного и того же размера можно использовать следующее:

```
>>>s = 'We can modify an element of a list by assigning to one of its index values.'
>>>text = nltk.word_tokenize(s)
>>>count_symb = nltk.FreqDist(len(w) for w in text)
>>>print(count_symb.most_common(10))
```

[out]:
[(2, 6), (3, 3), (1, 2), (6, 2), (4, 1), (5, 1), (7, 1), (9, 1)]

5. Словарь заданного текста

Для того чтобы получить словарь слов текста, преобразовываем список во множество, убирая повторяющиеся слова из текста.

```
>>>s = 'We can modify an element of a list by assigning to one of its index values.'
>>>text = nltk.word_tokenize(s)
>>>words = sorted(set(text))
>>>print(words)
```

[out]:
['.', 'We', 'a', 'an', 'assigning', 'by', 'can', 'element', 'index',
'its', 'list', 'modify', 'of', 'one', 'to', 'values']

6. Словарь без знаков препинания и разницы в регистре

Для того, чтобы получить словарь слов текста и устранить разницу в регистрах, убрать знаки пунктуации, используем следующий подход

```
set(w.lower() for w in text if w.isalpha())
```

7. Работа с корпусами (массивами текстовых данных)

Корпуса находятся в nltk.corpus
Наиболее распространенные words, brown, gutenberg
Данные корпуса являются наборами текстов (fileids)

```
>>>nltk.corpus.gutenberg.fileids()
```

```
[out]:
['austen-emma.txt', 'austen-persuasion.txt', 'austen-sense.txt',
'bible-kjv.txt', 'blake-poems.txt', 'bryant-stories.txt',
'burgess-busterbrown.txt', 'carroll-alice.txt', 'chesterton-ball.txt',
'chesterton-brown.txt', 'chesterton-thursday.txt', 'edgeworth-parents.txt',
'melville-moby_dick.txt', 'milton-paradise.txt', 'shakespeare-caesar.txt',
'shakespeare-hamlet.txt', 'shakespeare-macbeth.txt', 'whitman-leaves.txt']
```

Получить список слов корпуса или отдельного его текстового файла можно при помощи метода words()

```
>>>nltk.corpus.gutenberg.words()
[out]:
['[', 'Emma', 'by', 'Jane', 'Austen', '1816', ']', …]

>>>nltk.corpus.gutenberg.words('shakespeare-macbeth.txt')
[out]:
['[', 'The', 'Tragedie', 'of', 'Macbeth', 'by', …]
```

8. Загрузка своего корпуса

Загрузить свой собственный корпус можно посредством объекта PlaintextCorpusReader. При создании экземпляра этого класса, в качестве параметра передаётся директория, в которой будут находится файлы и шаблон, по которому будут отбираться файлы.

```
>>>path = '/home/my_folder'
>>>wordlists = PlaintextCorpusReader(path, '[a-z0-9]+.txt')
>>>print(wordlists.fileids())
```

```
[out]:
['file.txt', 'file5.txt']
```

9. Слова-синонимы

Для поиска слов-синонимов используется корпус wordnet

```
>>> nltk.wordnet.synsets('motorcar')
[Synset('car.n.01')]
>>> nltk.wordnet.synset('car.n.01').lemma_names()
```

['car', 'auto', 'automobile', 'machine', 'motorcar']

Или

```
>>>name = wordnet.synsets('motorcar')
>>>print(wordnet.synset(name[0].name()).lemma_names())
['car', 'auto', 'automobile', 'machine', 'motorcar']
```

Остальную информацию можно получить здесь:

http://www.nltk.org/book/