Lab 1: Cài đặt NLTK
và thực hành một số nội dung đơn giản

Nội dung

1. Cài đặt NLTK

Truy cập vào CMD của Window để tiến hành cài đặt NLTK.

```
pip3 install nltk
```

Kết quả hiện ra như sau:

```
Collecting nltk
  Downloading nltk-3.2.5.tar.gz (1.2MB)
```

```
    100% |████████████████████████████████| 1.2MB 765kB/s

Requirement already satisfied: six in /usr/local/lib/python3.6/site-
packages (from nltk)

Building wheels for collected packages: nltk

  Running setup.py bdist_wheel for nltk ... done

  Stored in directory:
/Users/jinohoang/Library/Caches/pip/wheels/18/9c/1f/276bc3f42161406246
8cb1c9d695e6086d0c73d67ea363c501

Successfully built nltk

Installing collected packages: nltk

Successfully installed nltk-3.2.5
```

Cài đặt các gói dữ liệu của NLTK. Truy cập Python console bằng cách gõ lệnh

```
python3
```

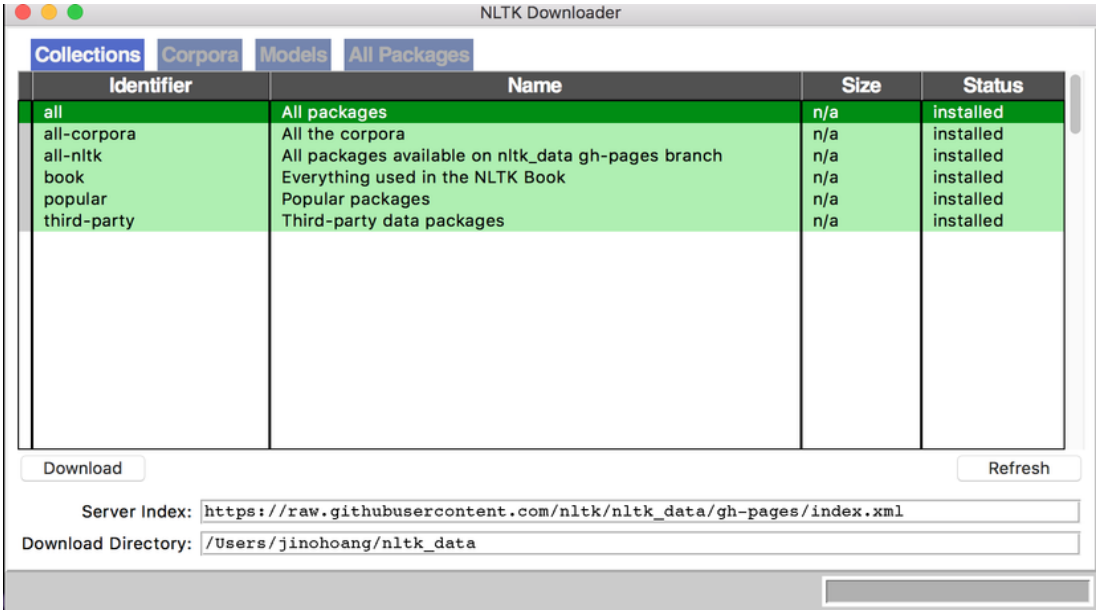
Kết quả

```
Python 3.6.2 (default, Jul 17 2017, 16:44:45)

[GCC 4.2.1 Compatible Apple LLVM 8.1.0 (clang-802.0.42)] on darwin

Type "help", "copyright", "credits" or "license" for more information.

>>>
```

Tại đây ta tiến hành cài đặt các gói dữ liệu của NLTK bằng cách gõ lệnh

```
>>> import nltk

>>> nltk.download()
```

Màn hình tải các gói dữ liệu của NLTK hiện ra, ở đây tôi đã cài đặt hết các gói.



Nhấn download. Vậy là NLTK đã sẵn sàng để sử dụng.

2. Import NLTK Book

- Sử dụng một package mà NLTK cung cấp sẵn đó là Book.
- Book chứa Text của các cuốn sách mẫu dành cho việc xử lý ngôn ngữ. Để sử dụng được Book, chúng ta cần import nó vào trong mã Python.

Ta tiến hành import book với lệnh "from nltk.book import *"

```
>>> from nltk.book import *
```

Kết quả:

```
*** Introductory Examples for the NLTK Book ***

Loading text1, ..., text9 and sent1, ..., sent9

Type the name of the text or sentence to view it.

Type: 'texts()' or 'sents()' to list the materials.

text1: Moby Dick by Herman Melville 1851

text2: Sense and Sensibility by Jane Austen 1811

text3: The Book of Genesis

text4: Inaugural Address Corpus
```

```
text5: Chat Corpus

text6: Monty Python and the Holy Grail

text7: Wall Street Journal

text8: Personals Corpus

text9: The Man Who Was Thursday by G . K . Chesterton 1908
```

Kết quả là ta đã có Text của 9 cuốn sách, được gán vào các biến tương ứng từ text1 -> text9.

Xem nội dung các biến thế nào:

```
>>> text1

<Text: Moby Dick by Herman Melville 1851>

>>> text2

<Text: Sense and Sensibility by Jane Austen 1811>

>>>
```

3. Tìm kiếm text

3.1. Tìm text ở trong đoạn văn bản

Ví dụ: để tìm từ **"monstrous"** trong văn bản được lưu trong biến text1, ta sử dụng hàm concordance().

```
>>> text1.concordance("monstrous")

Displaying 11 of 11 matches:

ong the former , one was of a most monstrous size . ... This came towards us ,

ON OF THE PSALMS . " Touching that monstrous bulk of the whale or ork we have r

ll over with a heathenish array of monstrous clubs and spears . Some were thick

d as you gazed , and wondered what monstrous cannibal and savage could ever hav

that has survived the flood ; most monstrous and most mountainous ! That Himmal

they might scout at Moby Dick as a monstrous fable , or still worse and more de
```

```
th of Radney .'" CHAPTER 55 Of the Monstrous Pictures of Whales . I shall ere l

ing Scenes . In connexion with the monstrous pictures of whales , I am strongly

ere to enter upon those still more monstrous stories of them which are to be fo

ght have been rummaged out of this monstrous cabinet there is no telling . But

of Whale - Bones ; for Whales of a monstrous size are oftentimes cast up dead u
```

Kết quả là chúng ta thấy từ **"monstrous"** được xuất hiện 11 lần trong văn bản.

3.2. Tìm các từ xuất hiện trong ngữ cảnh tương tự

Ta thấy từ **"monstrous"** sẽ đi đi cùng với ngữ cảnh như **"the .... pictures"** hay **"the ... size"**. Ta muốn tìm các từ mà chúng cũng được sử dụng trong các ngữ cảnh tương tự với từ **"monstrous"**. Để làm được điều đó, chúng ta sẽ sử dụng hàm similar(), cách sử dụng tương tự hàm concordance() phía trên.

```
>>> text1.similar("monstrous")

true contemptible christian abundant few part mean careful puzzled

mystifying passing curious loving wise doleful gamesome singular

delightfully perilous fearless
```

Thử với text2.

```
>>> text2.similar("monstrous")

very so exceedingly heartily a as good great extremely remarkably

sweet vast amazingly
```

Nếu xem xét về khía cạnh ngôn ngữ thì từ **"monstrous"** được sử dụng rất khác biệt về ý nghĩa. Với tác giả của đoạn text2, từ **"monstrous"** mang ý nghĩa tích cực hơn dựa vào ý nghĩa của các tính từ cùng ngữ cảnh như good, great, very, amazingly... .

4. Đếm số lượng từ trong văn bản.

4.1. Đếm tất cả lượng từ trong đoạn văn

Ta sử dụng hàm len() để đếm số từ trong đoạn văn

```
>>> len(text3)

44764
```

Kết quả là ta có 44.764 từ trong văn bản trên, bao gồm cả các dấu chấm câu.

Định nghĩa: mỗi một từ hay một dấu chấm câu sẽ được gọi là 1 "token".

4.2. Đếm số lượng từ sử dụng trong văn bản

Ta thực hiện bằng cách sử dụng hàm len() và hàm set() lồng nhau.

```
>>> len(set(text3))

2789
```

Để xem những từ đó là từ gì và sắp xếp chúng theo thứ tự, ta vẫn dùng hàm set() và bao bên ngoài là hàm sorted() để sắp xếp chúng theo thứ tự A-Z.

```
>>> sorted(set(text3))

['!', "'", '(', ')', ',', ',)', '.', '.)', ':', ';', ';)', '?', '?)', 'A', 'Abel',
'Abelmizraim', 'Abidah', 'Abide', 'Abimael', 'Abimelech', 'Abr', 'Abrah', 'Abraham',
'Abram', 'Accad', 'Achbor', 'Adah', 'Adam', 'Adbeel', 'Admah', 'Adullamite', 'After',
'Aholibamah', 'Ahuzzath', 'Ajah', 'Akan', 'All', 'Allonbachuth', 'Almighty',
'Almodad', 'Also', 'Alvah', 'Alvan', 'Am', 'Amal', 'Amalek', 'Amalekites', 'Ammon',
'Amorite', 'Amorites', 'Amraphel', 'An', 'Anah', 'Anamim', 'And', 'Aner', 'Angel',
'Appoint', 'Aram', 'Aran', 'Ararat', 'Arbah', 'Ard', 'Are', 'Areli', 'Arioch',
'Arise', 'Arkite', 'Arodi', 'Arphaxad', 'Art'....
```

4.3. Đếm số lần xuất hiện của 1 từ trong văn bản

Sử dụng hàm count(). Ví dụ: xác định số lần từ "lol" xuất hiện trong đoạn văn bản số 5.

```
>>> text5.count('lol')

704
```

Như vậy từ "lol" xuất hiện tới 704 lần trong văn bản thứ 5.

4.4. Tính tỉ lệ phần trăm của một từ trong văn bản

Tính số phần trăm của từ "lol" trong văn bản số 5:

```
>>> 100 * text5.count("lol") / len(text5)

1.5640968673628082
```

5. Lists of Words

5.1. Danh sách

NLTK Book ngoài cung cấp các văn bản text từ 1-9, còn cung cấp cho chúng ta các sentence, tức là các câu. Ví dụ: "sent1".

```
>>> sent1

['Call', 'me', 'Ishmael', '.']
```

Ta thấy "sent1" được cung cấp dưới dạng 1 Python lists, chứa các từ và dấu câu của 1 sentence. Và vì nó là 1 list, cho nên ta có thể đếm số từ và dấu câu trong list.

```
>>> len(sent1)

4
```

Tạo một List gồm các từ của một câu:

```
>>> my_sent = ["Donald", "Trump", "is", "president", "45th", "of","USA"]

>>> len(my_sent)

7

>>> sorted(set(my_sent))

['45th', 'Donald', 'Trump', 'USA', 'is', 'of', 'president']

>>> my_sent.count("is")

1
```

Cộng hai list:

```
>>> my_sent + sent1

['Donald', 'Trump', 'is', 'president', '45th', 'of', 'USA', 'Call', 'me', 'Ishmael',
'.']
```

Thêm 1 từ vào list:

```
>>> my_sent.append("America")

>>> my_sent

['Donald', 'Trump', 'is', 'president', '45th', 'of', 'USA', 'America']
```

5.2. Book cũng là list

Vì Book cũng là List nên ta có thể truy cập tới từng phần từ thông qua chỉ số

```
>>> text4[1005]

'Heaven'
```

Ta cũng có thể truy ngược vị trị của phần tử bằng hàm index()

```
>>> text4.index("Heaven")

1005
```

Lấy từ các trong một khoảng index nào đó (slicing)

```
>>> text5[16715:16735]

['U86', 'thats', 'why', 'something', 'like', 'gamefly', 'is', 'so', 'good',
'because', 'you', 'can', 'actually', 'play', 'a', 'full', 'game', 'without',
'buying', 'it']
```

5.3. Strings

Về cơ bản, string cũng tương tự như một Lists.

```
>>> my_str = "Donald Trump is president 45th of US"
```

```
>>> my_str[5]

'd'

>>> my_str.index("i")

13

>>> my_str + " and very rich"

'Donald Trump is president 45th of US and very rich'

>>> my_str * 2

'Donald Trump is president 45th of USDonald Trump is president 45th of US'
```

6. Thống kê căn bản

6.1. Frequency Distributions - Phân bố theo tần suất

NLTK cung cấp cho chúng ta 1 hàm để tính phân bố tần suất, đó là FreqDist().

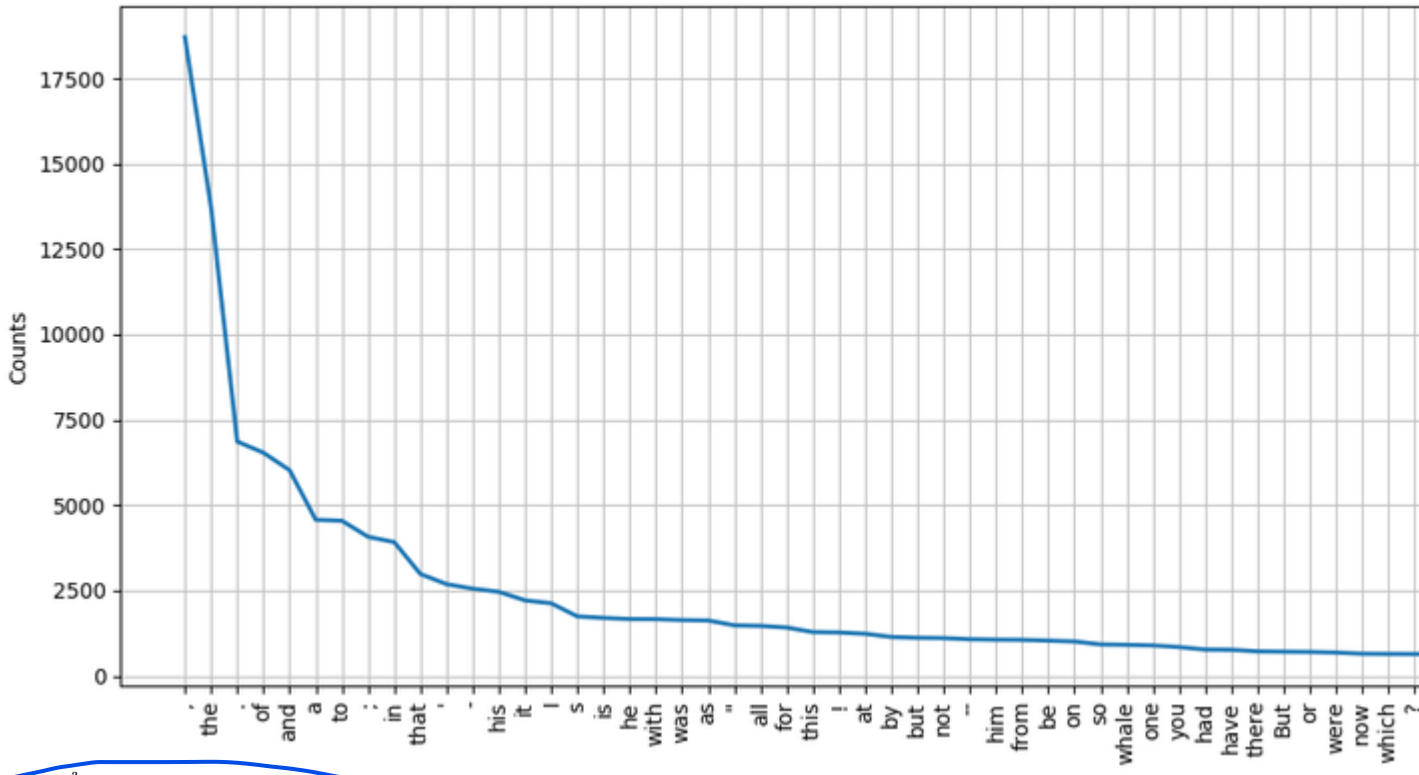
```
>>> fdist1 = FreqDist(text1)

>>> vocabulary1 = list(fdist1.keys())

>>> vocabulary1[:50]

['[', 'Moby', 'Dick', 'by', 'Herman', 'Melville', '1851', ']', 'ETYMOLOGY', '.', '(',
'Supplied', 'a', 'Late', 'Consumptive', 'Usher', 'to', 'Grammar', 'School', ')',
'The', 'pale', '--', 'threadbare', 'in', 'coat', ',', 'heart', 'body', 'and',
'brain', ';', 'I', 'see', 'him', 'now', 'He', 'was', 'ever', 'dusting', 'his', 'old',
'lexicons', 'grammars', 'with', 'queer', 'handkerchief', 'mockingly', 'embellished',
'all']
```

Chú ý: Để lấy Key của Dict trong Python 3, ta phải convert nó sang dạng List để truy cập được bằng Indexing.

Vẽ biểu đồ về tần xuất của 50 từ được dùng nhiều nhất trong văn bản text1 (sử dụng **matplotlib**).

```
fdist1 = FreqDist(text1)

fdist1.plot(50)
```

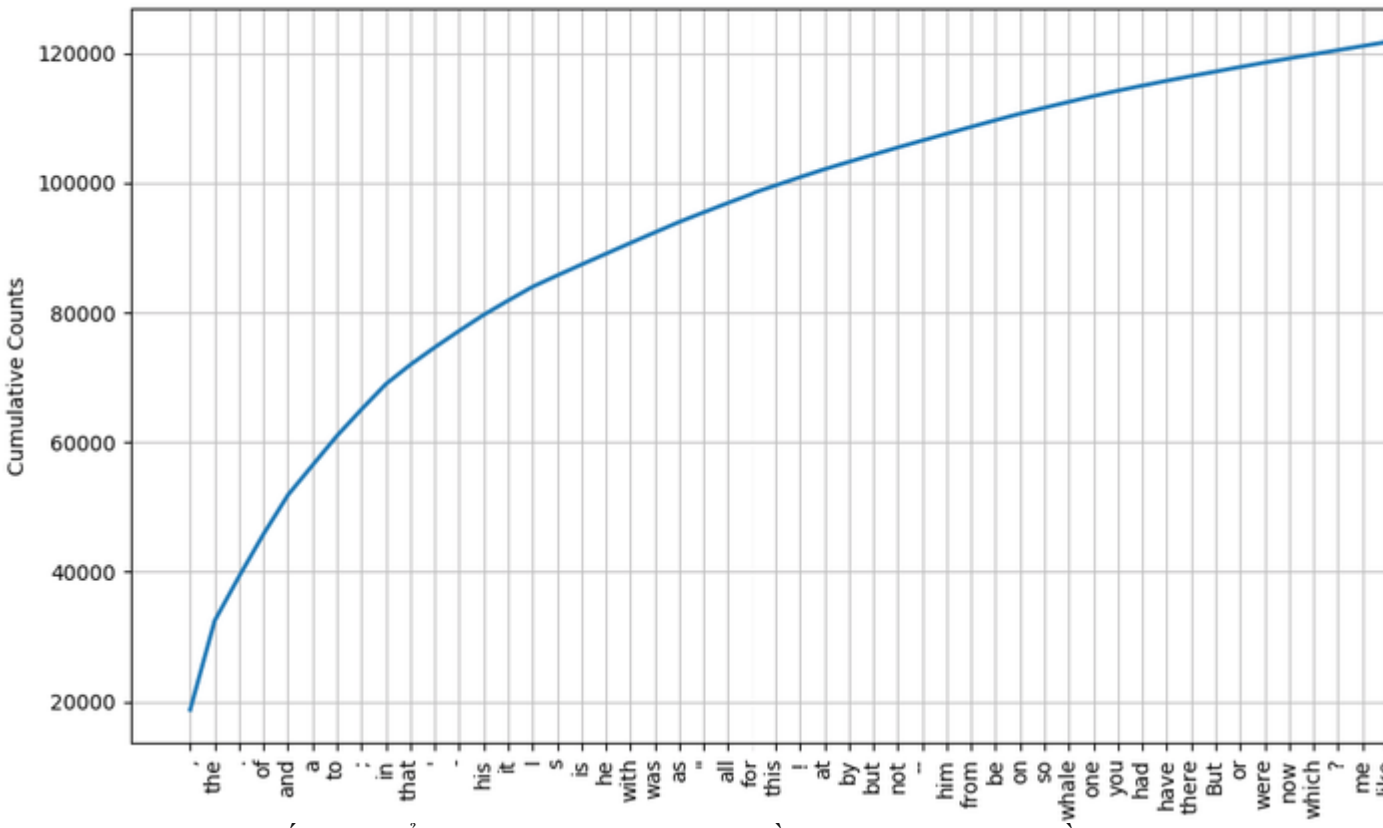
Kết quả



Tính tổng luỹ tích của các từ:

```
len(text1)

fdist1 = FreqDist(text1)

fdist1.plot(50, cumulative=True)
```

Kết quả:



Nhận xét: 50 từ này xuất hiện tổng cộng hơn 120 ngàn lần trong văn bản. Bằng cách lấy len() của text1, ta thấy 50 từ chiếm khoảng một nửa số từ trong văn bản text1!

6.2. Lựa chọn các từ theo điều kiện

Bài toán: Lấy các từ có nhiều hơn 15 ký tự trong văn bản.

```
V = set(text1)

long_words = [w for w in V if len(w) > 15]

sorted(long_words)
```

```
['CIRCUMNAVIGATION',  'Physiognomically',  'apprehensiveness',  'cannibalistically',
'characteristically',  'circumnavigating',  'circumnavigation',  'circumnavigations',
'comprehensiveness',  'hermaphroditical',  'indiscriminately',  'indispensableness',
'irresistibleness',  'physiognomically',  'preternaturalness',  'responsibilities',
'simultaneousness',  'subterraneousness',  'supernaturalness',  'superstitiousness',
'uncomfortableness', 'uncompromisedness', 'undiscriminating', 'uninterpenetratingly']
```

Bài toán: Lấy các từ có nhiều hơn 7 ký tự và xuất hiện nhiều hơn 7 lần trong văn bản.

```
>>> fdist5 = FreqDist(text5)

>>> sorted([w for w in set(text5) if len(w) > 7 and fdist5[w] > 7])

['#14-19teens', '#talkcity_adults', '((((((((((', '........', 'Question', 'actually',
'anything', 'computer', 'cute.-ass', 'everyone', 'football', 'innocent', 'listening',
'remember', 'seriously', 'something', 'together', 'tomorrow', 'watching']
```

6.3. Collocations và Bigrams

Collocations là sự kết hợp của từ để tạo thành một từ mới hoặc 1 câu có ý nghĩa. Ví dụ từ "thiên nga đen" là một Collocation, hay "chó trắng" cũng là một Collocation.

Có nhiều từ đơn lẻ sẽ không có ý nghĩa khi đứng một mình, nên trong 1 câu ra phải tìm ra các cặp Collocations để máy tính có thể hiểu được ý nghĩa của một câu. Bây giờ chúng ta làm thế nào để tách được riêng các cặp Collocations trong 1 câu đây?. Câu trả lời là sử dụng Bigrams. Ví dụ với câu "Donald Trump is president 45th of America and very rich". Ta sử dụng hàm bigrams()

```
>>> import nltk

>>> bigrm = list(nltk.bigrams("Donald Trump is president 45th of America and very
rich".split()))

>>> bigrm

[('Donald', 'Trump'), ('Trump', 'is'), ('is', 'president'), ('president', '45th'),
('45th', 'of'), ('of', 'America'), ('America', 'and'), ('and', 'very'), ('very',
'rich')]

>>>
```

Tuy nhiên, khi sử dụng Bigrams thì sẽ có những cụm mà không có nghĩa, do Bigrams đơn giản chỉ là tách từng từ và kết hợp với từ trước và sau của nó. Ngoài ra có những từ đứng 1 mình cũng có ý nghĩa, nhưng nếu sử dụng Bigrams thì nó sẽ không còn ý nghĩa nữa do bị kết hợp với một từ khác. Ta sử dụng hàm collocations() để lấy các collocations trong các đoạn văn mẫu.

```
>>> text4.collocations()

United States; fellow citizens; four years; years ago; Federal

Government; General Government; American people; Vice President; Old

World; Almighty God; Fellow citizens; Chief Magistrate; Chief Justice;
```

```
God bless; every citizen; Indian tribes; public debt; one another;

foreign nations; political parties
```

Ta thấy kết quả chính xác hơn. Việc tách các collocations là một thao tác cơ bản và khá quan trọng trong NLP. Sau này chúng ta sẽ cùng tìm hiểu kỹ hơn.

7. Bài tập

1. Hãy tính độ dài của 50 từ đầu tiên trong văn bản text1.
2. Hãy tính độ dài của 50 từ xuất hiện nhiều nhất trong văn bản text5.
3. Vẽ biểu đồ về sự phân bố độ dài của 50 từ xuất hiện nhiều nhất trong văn bản text2.

Tính phần trăm các từ có 5 ký tự trong văn bản text3.