Introdução à Biblioteca NLTK

1 - Importar biblioteca

In [2]:
```
import nltk
# nltk.download()
from nltk.book import *
```

```
*** 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
```

2 - text(), sents(), textn, sentn

In [3]:
```
texts()
```

```
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
```

In [4]:
```
sents()
```

```
sent1: Call me Ishmael .
sent2: The family of Dashwood had long been settled in Sussex .
sent3: In the beginning God created the heaven and the earth .
sent4: Fellow - Citizens of the Senate and of the House of Representatives :
sent5: I have a problem with people PMing me to lol JOIN
sent6: SCENE 1 : [ wind ] [ clop clop clop ] KING ARTHUR : Whoa there !
sent7: Pierre Vinken , 61 years old , will join the board as a nonexecutive director Nov.
29 .
sent8: 25 SEXY MALE , seeks attrac older single lady , for discreet encounters .
sent9: THE suburb of Saffron Park lay on the sunset side of London , as red and ragged as
a cloud of sunset .
```

3 - Comprimento de uma sentença do texto 7

In [5]:
```
len(sent7)
```

Out[5]:
```
18
```

4 - Quantidade de palavras no corpus do Jornal de Wall Street

In [6]:
```
len(text7)
```

Out[6]:
```
100676
```

5 - Quantidade de palavras únicas no corpus do Jornal de Wall Street

In [7]:
```
len(set(text7))
```

Out[7]:
```
12408
```

In [8]:
```
list(set(text7))[:10]
```

Out[8]:
```
['state-owned',
 'comes',
 '*T*-25',
 '*T*-123',
 'Amin',
 'retractable',
 'quarter',
 'Starting',
 'worrying',
 'Swiss']
```

6 - Frequência de palavras

In [9]:
```
dist = FreqDist(text7)
len(dist)
```

Out[9]:
```
12408
```

7 - A função keys() da frequência de palavras

In [10]:
```
vocab1 = dist.keys()
list(vocab1)[:100]
```

Out[10]:
```
['Pierre',
 'Vinken',
 ',',
 '61',
 'years',
 'old',
 'will',
 'join',
 'the',
 'board',
 'as',
 'a',
 'nonexecutive',
 'director',
 'Nov.',
 '29',
 '.',
 'Mr.',
 'is',
 'chairman',
 'of',
 'Elsevier',
 'N.V.',
 'Dutch',
 'publishing',
 'group',
 'Rudolph',
```

```
'Agnew',
'55',
'and',
'former',
'Consolidated',
'Gold',
'Fields',
'PLC',
'was',
'named',
'*-1',
'this',
'British',
'industrial',
'conglomerate',
'A',
'form',
'asbestos',
'once',
'used',
'*',
'to',
'make',
'Kent',
'cigarette',
'filters',
'has',
'caused',
'high',
'percentage',
'cancer',
'deaths',
'among',
'workers',
'exposed',
'it',
'more',
'than',
'30',
'ago',
'researchers',
'reported',
'0',
'*T*-1',
'The',
'fiber',
'crocidolite',
'unusually',
'resilient',
'enters',
'lungs',
'with',
'even',
'brief',
'exposures',
'causing',
'symptoms',
'that',
'show',
'up',
'decades',
'later',
'said',
'*T*-2',
'Lorillard',
'Inc.',
```

```
 'unit',
 'New',
 'York-based',
 'Loews',
 'Corp.',
 'makes',
 'cigarettes']
```

8 - Frequência que uma palavra foi utilizada

In [11]:
```
dist['president']
```

Out[11]:
```
133
```

9 - Quantidade de vezes que uma palavra ocorre + condição de comprimento

In [12]:
```
[w for w in vocab1 if len(w) > 5 and dist[w] > 100]
```

Out[12]:
```
['billion',
 'company',
 'president',
 'because',
 'market',
 'million',
 'shares',
 'trading',
 'program']
```

10 - Normalização

In [13]:
```
input1 = "List listed lists listing listings"
words1 = input1.lower().split(' ')
words1
```

Out[13]:
```
['list', 'listed', 'lists', 'listing', 'listings']
```

11 - Stemming

In [15]:
```
porter = nltk.PorterStemmer()
[porter.stem(t) for t in words1]
```

Out[15]:
```
['list', 'list', 'list', 'list', 'list']
```

12 - Lematização (Lemmatization)

A lematização é o processo, efetivamente, de deflexionar uma palavra para determinar o seu lema (as flexões chamam-se lexemas).

In [16]:
```
udhr = nltk.corpus.udhr.words('English-Latin1')
udhr[:20]
```

Out[16]:
```
['Universal',
 'Declaration',
 'of',
 'Human',
 'Rights',
 'Preamble',
 'Whereas',
 'recognition',
```

```
 'of',
 'the',
 'inherent',
 'dignity',
 'and',
 'of',
 'the',
 'equal',
 'and',
 'inalienable',
 'rights',
 'of']
```

In [17]:
```
[porter.stem(t) for t in udhr[:20]]
```

Out[17]:
```
['univers',
 'declar',
 'of',
 'human',
 'right',
 'preambl',
 'wherea',
 'recognit',
 'of',
 'the',
 'inher',
 'digniti',
 'and',
 'of',
 'the',
 'equal',
 'and',
 'inalien',
 'right',
 'of']
```

In [18]:
```
WNlemma = nltk.WordNetLemmatizer()
[WNlemma.lemmatize(t) for t in udhr[:20]]
```

Out[18]:
```
['Universal',
 'Declaration',
 'of',
 'Human',
 'Rights',
 'Preamble',
 'Whereas',
 'recognition',
 'of',
 'the',
 'inherent',
 'dignity',
 'and',
 'of',
 'the',
 'equal',
 'and',
 'inalienable',
 'right',
 'of']
```

13 - Tokenização (tokenization)

In [21]:
```
text11 = "Children shouldn't drink a better sugary drink before bed."
[WNlemma.lemmatize(t) for t in nltk.word_tokenize(text11)]
```

Out[21]:
```
['Children',
 'should',
 "n't",
 'drink',
 'a',
 'better',
 'sugary',
 'drink',
 'before',
 'bed',
 '.']
```

In [22]:
```
[porter.stem(t) for t in nltk.word_tokenize(text11)]
```

Out[22]:
```
['children',
 'should',
 "n't",
 'drink',
 'a',
 'better',
 'sugari',
 'drink',
 'befor',
 'bed',
 '.']
```

In [23]:
```
nltk.word_tokenize(text11)
```

Out[23]:
```
['Children',
 'should',
 "n't",
 'drink',
 'a',
 'better',
 'sugary',
 'drink',
 'before',
 'bed',
 '.']
```

In [24]:
```
text12 = "This is the first sentence. A gallon of milk in the U.S. costs $2.99. Is this th
nltk.sent_tokenize(text12)
```

Out[24]:
```
['This is the first sentence.',
 'A gallon of milk in the U.S. costs $2.99.',
 'Is this the third sentence?',
 'Yes, it is!']
```

In [25]:
```
len(nltk.sent_tokenize(text12))
```

Out[25]:
```
4
```

14 - Sobre análise sintática

In [28]:
```
nltk.help.upenn_tagset('RB')
```

```
RB: adverb
    occasionally unabatingly maddeningly adventurously professedly
    stirringly prominently technologically magisterially predominately
    swiftly fiscally pitilessly ...
```

15 - Part-of-speech nltk.pos_tag()

In [31]:
```
text13 = nltk.word_tokenize(text11)
nltk.pos_tag(text13)
```

Out[31]:
```
[('Children', 'NNP'),
 ('should', 'MD'),
 ("n't", 'RB'),
 ('drink', 'VB'),
 ('a', 'DT'),
 ('better', 'JJR'),
 ('sugary', 'JJ'),
 ('drink', 'NN'),
 ('before', 'IN'),
 ('bed', 'NN'),
 ('.', '.')]
```

16 - Ambiguidade POS tagging

In [32]:
```
text14 = nltk.word_tokenize("Visiting aunts can be a nuisance")
nltk.pos_tag(text14)

# 'Visiting' é 'JJ' na forma alternativa
```

Out[32]:
```
[('Visiting', 'VBG'),
 ('aunts', 'NNS'),
 ('can', 'MD'),
 ('be', 'VB'),
 ('a', 'DT'),
 ('nuisance', 'NN')]
```

17 - Part-of-speech nltk.pos_tag()

In [33]:
```
text15 = nltk.word_tokenize("Alice loves Bob")
grammar = nltk.CFG.fromstring("""
S -> NP VP
VP -> V NP
NP -> 'Alice' | 'Bob'
V -> 'loves'
""")

parser = nltk.ChartParser(grammar)
trees = parser.parse_all(text15)
for tree in trees:
    print(tree)
```

```
(S (NP Alice) (VP (V loves) (NP Bob)))
```

18 - Treebank (coleção de árvores de análise sintática)

In [34]:
```
from nltk.corpus import treebank
text17 = treebank.parsed_sents('wsj_0001.mrg')[0]
print(text17)
```

```
(S
  (NP-SBJ
    (NP (NNP Pierre) (NNP Vinken))
    (, ,)
    (ADJP (NP (CD 61) (NNS years)) (JJ old))
    (, ,))
  (VP
    (MD will)
```

```
    (VP
      (VB join)
      (NP (DT the) (NN board))
      (PP-CLR (IN as) (NP (DT a) (JJ nonexecutive) (NN director)))
      (NP-TMP (NNP Nov.) (CD 29))))
  (. .))
```

19 - Outras ambiguidades

In [35]:
```
text18 = nltk.word_tokenize("The old man the boat")
nltk.pos_tag(text18)
```

Out[35]:
```
[('The', 'DT'), ('old', 'JJ'), ('man', 'NN'), ('the', 'DT'), ('boat', 'NN')]
```

In [36]:
```
text19 = nltk.word_tokenize("Colorless green ideas sleep furiously")
nltk.pos_tag(text19)
```

Out[36]:
```
[('Colorless', 'NNP'),
 ('green', 'JJ'),
 ('ideas', 'NNS'),
 ('sleep', 'VBP'),
 ('furiously', 'RB')]
```

In [ ]:
```

```