Getting_started_with_NLTK

December 22, 2023

1 Getting started with NLTK

[ ]:
```
pip install nltk
```

[3]:
```
import nltk
```

[4]:
```
nltk.download()
```

```
showing info https://raw.githubusercontent.com/nltk/nltk_data/gh-pages/index.xml
```

[4]:
```
True
```

[6]:
```
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
```

[7]:
```
text1.concordance("monster")
```

```
Displaying 25 of 49 matches:
des cometh within the chaos of this monster ' s mouth , be it beast , boat , or
nter into the dreadful gulf of this monster ' s ( whale ' s ) mouth , are immed
time with a lance ; but the furious monster at length rushed on the boat ; hims
 . Such a portentous and mysterious monster roused all my curiosity . Then the
and flank with the most exasperated monster . Long usage had , for this Stubb ,
ACK ).-- Under this head I reckon a monster which , by the various names of Fin
arned the history of that murderous monster against whom I and all the others h
```

```
ocity , cunning , and malice in the monster attacked ; therefore it was , that
iathan is restricted to the ignoble monster primitively pursued in the North ;
 and incontestable character of the monster to strike the imagination with unwo
mberment . Then , in darting at the monster , knife in hand , he had but given
e rock ; instead of this we saw the monster sailing off with the utmost gravity
e at Constantinople , a great sea - monster was captured in the neighboring Pro
 Of what precise species this sea - monster was , is not mentioned . But as he
man reasoning , Procopius ' s sea - monster , that for half a century stove the
hale ," as he called the fictitious monster which he declared to be incessantly
d his intention to hunt that mortal monster in person . But such a supposition
ng us on and on , in order that the monster might turn round upon us , and rend
d famous , and most deadly immortal monster , Don ;-- but that would be too lon
oluntarily lifted his voice for the monster , though for some little time past
s rescuing Andromeda from the sea - monster or whale . Where did Guido get the
 huge corpulence of that Hogarthian monster undulates on the surface , scarcely
nd is drawn just balancing upon the monster ' s spine ; and standing in that pr
 of cutting - in ) hove over to the monster as if to a quay ; and a boat , hurr
eet in length . They fancy that the monster to which these arms belonged ordina
```

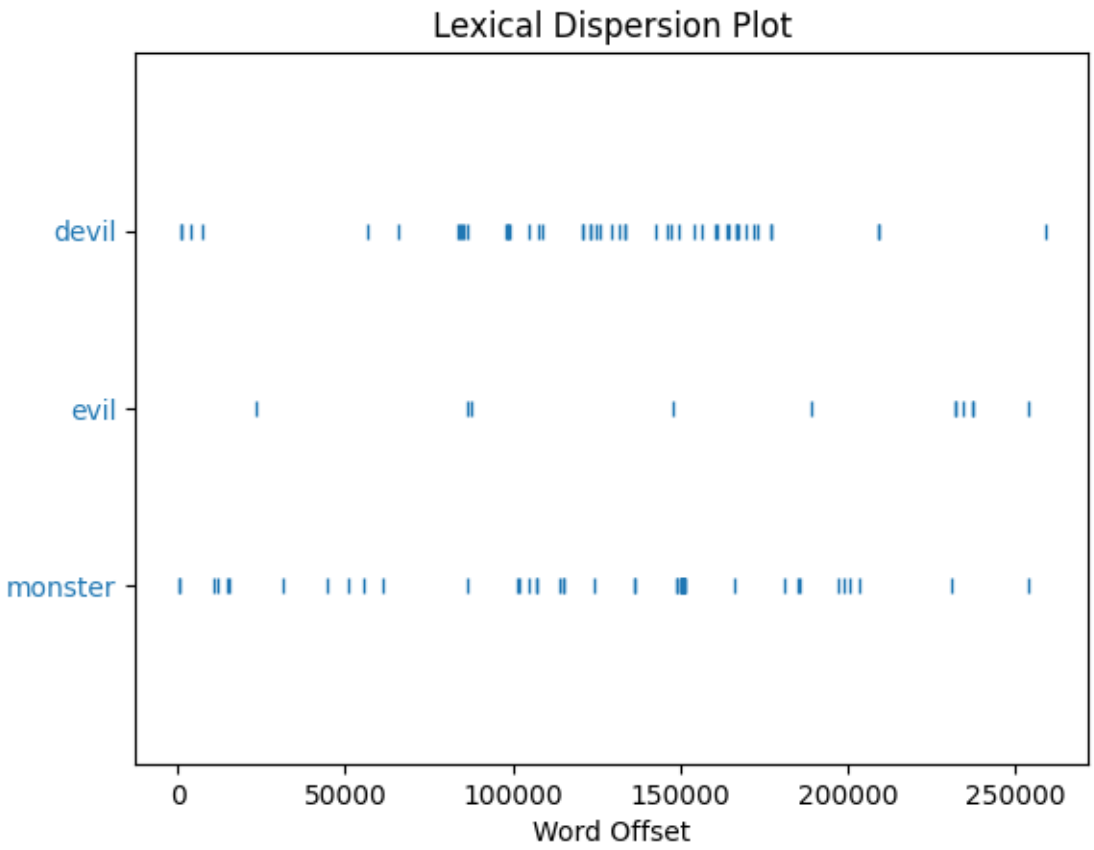
[8]:
```
text1.similar("monster")
```

```
whale ship world sea whales boat pequod other sun leviathan thing king
water head captain air crew cabin body more
```

[9]:
```
text1.common_contexts(["monster", "person"])
```

```
the_that
```

[10]:
```
text1.dispersion_plot(["monster", "evil", "devil"])
```



[ ]:
```
set(text1)
```

[ ]:
```
sorted(set(text1))
```

[13]:
```
len(text3)
```

[13]: 44764

[14]:
```
texts = [text1,text2,text3,text4,text5,text6,text7,text8,text9]
```

[15]:
```
print(texts)
```

```
[<Text: Moby Dick by Herman Melville 1851>, <Text: Sense and Sensibility by Jane
Austen 1811>, <Text: The Book of Genesis>, <Text: Inaugural Address Corpus>,
<Text: Chat Corpus>, <Text: Monty Python and the Holy Grail>, <Text: Wall Street
Journal>, <Text: Personals Corpus>, <Text: The Man Who Was Thursday by G . K .
Chesterton 1908>]
```

[20]:
```
for index, name in enumerate(texts):
    print(f"Corpus Number {index + 1}: {len(name)}")
```

```
Corpus Number 1: 260819
Corpus Number 2: 141576
Corpus Number 3: 44764
Corpus Number 4: 152901
Corpus Number 5: 45010
Corpus Number 6: 16967
Corpus Number 7: 100676
Corpus Number 8: 4867
Corpus Number 9: 69213
```

[21]:
```
for index, name in enumerate(texts):
    print(f"Corpus Number {index + 1}: {len(set(name))}")
```

```
Corpus Number 1: 19317
Corpus Number 2: 6833
Corpus Number 3: 2789
Corpus Number 4: 10025
Corpus Number 5: 6066
Corpus Number 6: 2166
Corpus Number 7: 12408
Corpus Number 8: 1108
Corpus Number 9: 6807
```

[22]:
```
for index, name in enumerate(texts):
    print(f"Corpus Number {index + 1}: {len(set(name))/len(name)}")
```

```
Corpus Number 1: 0.07406285585022564
Corpus Number 2: 0.04826383002768831
Corpus Number 3: 0.06230453042623537
Corpus Number 4: 0.06556530042314962
Corpus Number 5: 0.13477005109975562
Corpus Number 6: 0.1276595744680851
Corpus Number 7: 0.12324685128531129
Corpus Number 8: 0.22765564002465585
Corpus Number 9: 0.0983485761345412
```

[23]:
```
def TK(word):
    return f"TF of Word {word}: {text1.count(word) / len(text1) * 100}"
```

[24]:
```
print(TK('monster'), TK('evil'), TK('devil'), TK('the'))
```

```
TF of Word monster: 0.018786974875296663 TF of Word evil: 0.004217484155678842
TF of Word devil: 0.01955379017632918 TF of Word the: 5.260736372733581
```

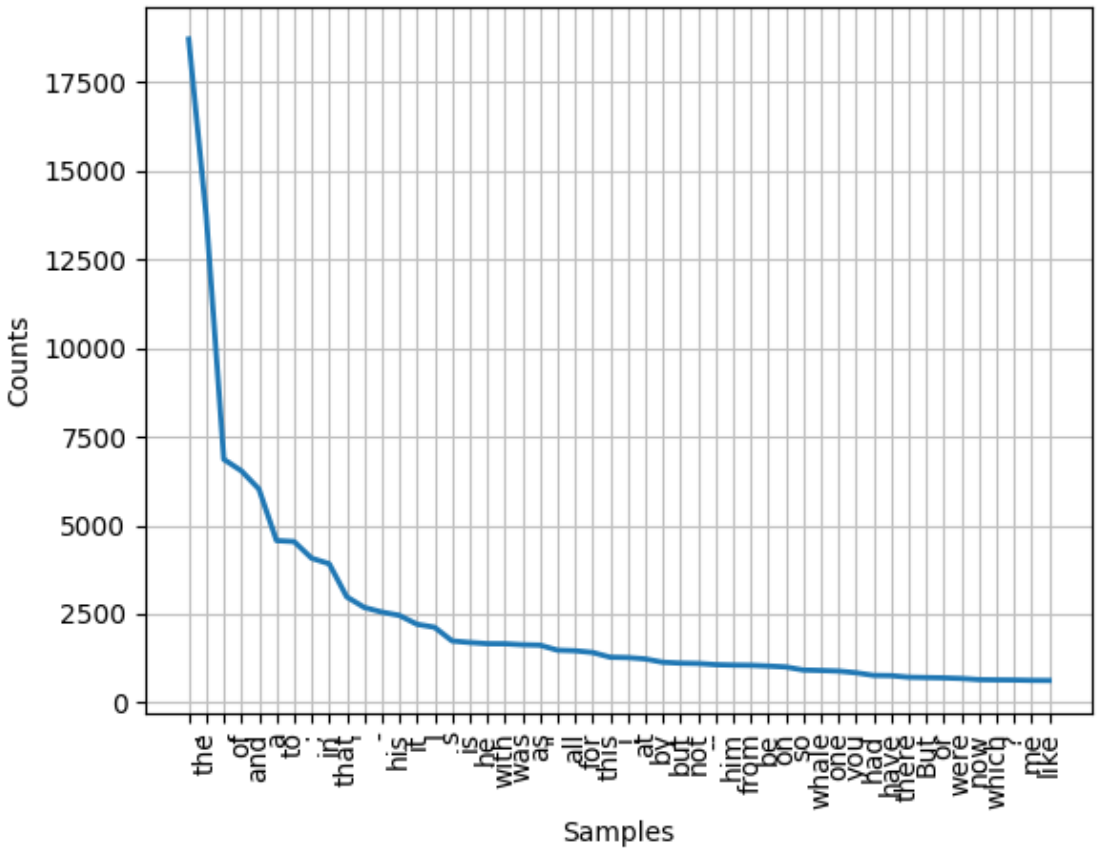
[26]:
```
fdist1 = FreqDist(text1)
fdist1.most_common(3)
```

[26]:
```
[(',', 18713), ('the', 13721), ('.', 6862)]
```

[27]:
```
print("Common word 1 ", TK(','), "Common word 2 ", TK('the'),"Common word 3 ",␣
↪TK('.'))
```

```
Common word 1  TF of Word ,: 7.174707364110744 Common word 2  TF of Word the:
5.260736372733581 Common word 3  TF of Word .: 2.630943297842565
```

[28]:
```
fdist1.plot(50)
```



[28]:
```
<Axes: xlabel='Samples', ylabel='Counts'>
```

[31]:
```
import math
```

[32]:
```
def LOGTK(word):
    return f"LOGTF of Word {word}: {math.log(text1.count(word)+1,10)}"
```

[33]:
```
print(LOGTK('monster'), LOGTK('evil'), LOGTK('devil'), LOGTK('the'),"Common␣
↪word 1 ", LOGTK(','), "Common word 2 ", LOGTK('the'),"Common word 3 ",␣
↪LOGTK('.'))
```

```
LOGTF of Word monster: 1.6989700043360185 LOGTF of Word evil: 1.0791812460476247
```

```
LOGTF of Word devil: 1.716003343634799 LOGTF of Word the: 4.137417414990392
Common word 1  LOGTF of Word ,: 4.272166625140787 Common word 2  LOGTF of Word
the: 4.137417414990392 Common word 3  LOGTF of Word .: 3.836513998890671
```

[34]:
```
def IDF(word):
    return f"LOGTF of Word {word}: {math.log(9 / text1.count(word), 10)}"
```

[35]:
```
print(IDF('monster'), IDF('evil'), IDF('devil'), IDF('the'),"Common word 1 ",␣
 ↪IDF(','), "Common word 2 ", IDF('the'),"Common word 3 ", IDF('.'))
```

```
LOGTF of Word monster: -0.7359535705891886 LOGTF of Word evil:
-0.08715017571890013 LOGTF of Word devil: -0.7533276666586114 LOGTF of Word the:
-3.1831432548946452 Common word 1  LOGTF of Word ,: -3.3179009081517252 Common
word 2  LOGTF of Word the: -3.1831432548946452 Common word 3  LOGTF of Word .:
-2.8822082042808295
```

[36]:
```
def IDF(word):
    return f"IDF of Word {word}: {math.log(9 / text1.count(word), 10)}"
```

[37]:
```
print(IDF('monster'), IDF('evil'), IDF('devil'), IDF('the'),"Common word 1 ",␣
 ↪IDF(','), "Common word 2 ", IDF('the'),"Common word 3 ", IDF('.'))
```

```
IDF of Word monster: -0.7359535705891886 IDF of Word evil: -0.08715017571890013
IDF of Word devil: -0.7533276666586114 IDF of Word the: -3.1831432548946452
Common word 1  IDF of Word ,: -3.3179009081517252 Common word 2  IDF of Word
the: -3.1831432548946452 Common word 3  IDF of Word .: -2.8822082042808295
```

[38]:
```
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to /home/pheonix/nltk_data…
[nltk_data]   Package punkt is already up-to-date!
```

[38]:
```
True
```

[39]:
```
text = "NLTK is a powerful library for natural language processing."
words = nltk.word_tokenize(text)
sentences = nltk.sent_tokenize(text)
print(words)
print(sentences)
```

```
['NLTK', 'is', 'a', 'powerful', 'library', 'for', 'natural', 'language',
'processing', '.']
['NLTK is a powerful library for natural language processing.']
```

[40]:
```
nltk.download('averaged_perceptron_tagger')
tags = nltk.pos_tag(words)
print(tags)
```

```
[('NLTK', 'NNP'), ('is', 'VBZ'), ('a', 'DT'), ('powerful', 'JJ'), ('library',
'NN'), ('for', 'IN'), ('natural', 'JJ'), ('language', 'NN'), ('processing',
'NN'), ('.', '.')]
```

```
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /home/pheonix/nltk_data…
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
```

[41]:
```
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     /home/pheonix/nltk_data…
[nltk_data]   Package stopwords is already up-to-date!
```

[41]:
```
True
```

[42]:
```
from nltk.corpus import stopwords
```

[43]:
```
filtered_words =[word for word in words if word.lower() not in stopwords.
↪words('english')]
```

[44]:
```
print(filtered_words)
```

```
['NLTK', 'powerful', 'library', 'natural', 'language', 'processing', '.']
```

[ ]:
```

```