Exploring NLTK

Download the required library and data

```
import nltk
nltk.download("stopwords")
nltk.download("wordnet")
nltk.download("punkt")
nltk.download("omw-1.4")
nltk.download("book")
from nltk.book import *
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package omw-1.4 to /root/nltk_data...
[nltk_data] Downloading collection 'book'
[nltk_data]    |
[nltk_data]    | Downloading package abc to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/abc.zip.
[nltk_data]    | Downloading package brown to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/brown.zip.
[nltk_data]    | Downloading package chat80 to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/chat80.zip.
[nltk_data]    | Downloading package cmudict to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/cmudict.zip.
[nltk_data]    | Downloading package conll2000 to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/conll2000.zip.
[nltk_data]    | Downloading package conll2002 to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/conll2002.zip.
[nltk_data]    | Downloading package dependency_treebank to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/dependency_treebank.zip.
[nltk_data]    | Downloading package genesis to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/genesis.zip.
[nltk_data]    | Downloading package gutenberg to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/gutenberg.zip.
[nltk_data]    | Downloading package ieer to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/ieer.zip.
[nltk_data]    | Downloading package inaugural to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/inaugural.zip.
[nltk_data]    | Downloading package movie_reviews to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/movie_reviews.zip.
[nltk_data]    | Downloading package nps_chat to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/nps_chat.zip.
[nltk_data]    | Downloading package names to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/names.zip.
```

```
[nltk_data]    | Downloading package ppattach to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/ppattach.zip.
[nltk_data]    | Downloading package reuters to /root/nltk_data...
[nltk_data]    | Downloading package senseval to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/senseval.zip.
[nltk_data]    | Downloading package state_union to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/state_union.zip.
[nltk_data]    | Downloading package stopwords to /root/nltk_data...
[nltk_data]    |   Package stopwords is already up-to-date!
[nltk_data]    | Downloading package swadesh to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/swadesh.zip.
[nltk_data]    | Downloading package timit to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/timit.zip.
[nltk_data]    | Downloading package treebank to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/treebank.zip.
[nltk_data]    | Downloading package toolbox to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/toolbox.zip.
[nltk_data]    | Downloading package udhr to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/udhr.zip.
[nltk_data]    | Downloading package udhr2 to /root/nltk_data
```

tokens()

The code below uses the tokens() method, built in the NLTK Text Object, to print the first 20 tokens of text1.

Things I learned

1- To create a Text Object, we need to use the Text() method. For example: abc = Text([‘abc’,'def']) will create a Text Object called 'abc' with tokens ['abc','def'].

2- textObject.tokens returns a list of all the tokens in the provided text.

```
text1.tokens[0:20]
```

```
['[',
 'Moby',
 'Dick',
 'by',
 'Herman',
 'Melville',
 '1851',
 ']',
 'ETYMOLOGY',
 '.',
 '(',
 'Supplied',
 'by',
 'a',
 'Late',
 'Consumptive',
 'Usher',
```

```
 'to',
 'a',
 'Grammar']
```

concordance()

The code below uses the concordance() method to look for the first five lines that mention the word 'sea', in text1.

```
text1.concordance('sea',30,5)
```

```
Displaying 5 of 455 matches:
at is in the sea ." -- ISAIAH
" The Indian Sea breedeth the
 days on the sea , when about
sters of the sea , appeared .
 beating the sea before him i
```

count()

The code below uses the count() method to count how many times the given word has been mentioned in the calling text object.

count(): API vs. Python

The count() method built in Python and the one used for NLTK Text Object have the same syntax: text.count(word). They also return the same thing: the number of times the given word has been mentioned in the text calling the method.

```
text1.count('Moby')
```

```
84
```

word_tokenize()

The code below uses the word_tokenize() method to print the first 10 tokens of the provided text.

Note: The text used below is taken from The Great Gatsby.

```
raw_text = 'In my younger and more vulnerable years my father gave me some advice that I’ve b
from nltk import word_tokenize
tokens = word_tokenize(raw_text)
tokens[0:10]
```

```
['In',
 'my',
 'younger',
 'and',
 'more',
 'vulnerable',
 'years',
 'my',
 'father',
 'gave']
```

sent_tokenize

The code below uses the sent_tokenize() method to print each sentence of the provided text, as a separate token.

```
from nltk import sent_tokenize
sentences = sent_tokenize(raw_text)
sentences
```

```
['In my younger and more vulnerable years my father gave me some advice that I’ve been
turning over in my mind ever since.',
 '“Whenever you feel like criticizing any one,” he told me, “just remember that all
the people in this world haven’t had the advantages that you’ve had”.',
 'He didn’t say any more, but we’ve always been unusually communicative in a reserved
way, and I understood that he meant a great deal more than that.',
 'In consequence, I’m inclined to reserve all judgments, a habit that has opened up
many curious natures to me and also made me the victim of not a few veteran bores.',
 'The abnormal mind is quick to detect and attach itself to this quality when it
appears in a normal person, and so it came about that in college I was unjustly
accused of being a politician, because I was privy to the secret griefs of wild,
unknown men.']
```

PorterStemmmer()

The code below uses the PorterStemmer to stem the tokened text from an earlier step

```
from nltk.stem.porter import *
stemmer = PorterStemmer()
newlist = [stemmer.stem(i) for i in tokens]
newlist
```

```
['in',
 'my',
 'younger',
 'and',
 'more',
 'vulner',
```

```
'year',
'my',
'father',
'gave',
'me',
'some',
'advic',
'that',
'i',
''',
've',
'been',
'turn',
'over',
'in',
'my',
'mind',
'ever',
'sinc',
'.',
'“',
'whenev',
'you',
'feel',
'like',
'critic',
'ani',
'one',
',',
'”',
'he',
'told',
'me',
',',
'“',
'just',
'rememb',
'that',
'all',
'the',
'peopl',
'in',
'thi',
'world',
'haven',
''',
't',
'had',
'the',
'advantag',
'that',
'you',
```

▾ WordNetLemmatizer()

The code below uses the WordNetLemmatizer to lemmatize the tokened text from an earlier step

Stem vs. Lemmas

vulner - vulnerable

advic - advice

sinc - since

whenev - Whenever

critic - criticizing

```
from nltk.stem import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
newlist = [lemmatizer.lemmatize(i) for i in tokens]
newlist
```

```
['In',
 'my',
 'younger',
 'and',
 'more',
 'vulnerable',
 'year',
 'my',
 'father',
 'gave',
 'me',
 'some',
 'advice',
 'that',
 'I',
 "'",
 've',
 'been',
 'turning',
 'over',
 'in',
 'my',
 'mind',
 'ever',
 'since',
 '.',
 '“',
 'Whenever',
 'you',
 'feel',
 'like',
 'criticizing',
 'any',
 'one',
```

```
',',
'„',
'he',
'told',
'me',
',',
'“',
'just',
'remember',
'that',
'all',
'the',
'people',
'in',
'this',
'world',
'haven',
'’',
't',
'had',
'the',
'advantage',
'that',
'you',
```

Comments

After executing several methods from NLTK library, I see how this library can be very useful for processing texts for different types of applications. The methods built in this library can be used to tokenize, parse, stem, lemmatize, and a lot more. Also, the code is easy to understand and easy to use. I believe I will be using this library for my future projects whenever I would need to process some given text. Not only that, but I will be using NLTK to help the computer analyze the written text, for my Machine Learning course.

Colab paid products - Cancel contracts here

✓ 0s completed at 7:09 PM