Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.

Downloading

Below, we are downloading all the necessary libraries from NLTK

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

```
[nltk_data]    |   Package senseval is already up-to-date!
[nltk_data]    | Downloading package state_union to /root/nltk_data...
[nltk_data]    |   Package state_union is already up-to-date!
[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]    |   Package swadesh is already up-to-date!
[nltk_data]    | Downloading package timit to /root/nltk_data...
[nltk_data]    |   Package timit is already up-to-date!
[nltk_data]    | Downloading package treebank to /root/nltk_data...
[nltk_data]    |   Package treebank is already up-to-date!
[nltk_data]    | Downloading package toolbox to /root/nltk_data...
[nltk_data]    |   Package toolbox is already up-to-date!
[nltk_data]    | Downloading package udhr to /root/nltk_data...
[nltk_data]    |   Package udhr is already up-to-date!
[nltk_data]    | Downloading package udhr2 to /root/nltk_data...
[nltk_data]    |   Package udhr2 is already up-to-date!
[nltk_data]    | Downloading package unicode_samples to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Package unicode_samples is already up-to-date!
[nltk_data]    | Downloading package webtext to /root/nltk_data...
[nltk_data]    |   Package webtext is already up-to-date!
[nltk_data]    | Downloading package wordnet to /root/nltk_data...
[nltk_data]    |   Package wordnet is already up-to-date!
[nltk_data]    | Downloading package wordnet_ic to /root/nltk_data...
[nltk_data]    |   Package wordnet_ic is already up-to-date!
[nltk_data]    | Downloading package words to /root/nltk_data...
[nltk_data]    |   Package words is already up-to-date!
[nltk_data]    | Downloading package maxent_treebank_pos_tagger to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Package maxent_treebank_pos_tagger is already up-
[nltk_data]    |       to-date!
[nltk_data]    | Downloading package maxent_ne_chunker to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Package maxent_ne_chunker is already up-to-date!
[nltk_data]    | Downloading package universal_tagset to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Package universal_tagset is already up-to-date!
[nltk_data]    | Downloading package punkt to /root/nltk_data...
[nltk_data]    |   Package punkt is already up-to-date!
```

```
[nltk_data]    |   Package punkt is already up-to-date!
[nltk_data]    | Downloading package book_grammars to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Package book_grammars is already up-to-date!
[nltk_data]    | Downloading package city_database to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Package city_database is already up-to-date!
[nltk_data]    | Downloading package tagsets to /root/nltk_data...
[nltk_data]    |   Package tagsets is already up-to-date!
[nltk_data]    | Downloading package panlex_swadesh to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Package panlex_swadesh is already up-to-date!
[nltk_data]    | Downloading package averaged_perceptron_tagger to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Package averaged_perceptron_tagger is already up-
[nltk_data]    |       to-date!
[nltk_data]    |
[nltk_data]  Done downloading collection book
True
```

TOKENS

We are importing text1 from nltk.book

The output should be the first 20 tokens of the text1

```
from nltk.book import text1
print(" ".join(text1.tokens[0:20]))
```

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

Concordance

The output should be 5 instances within text1 where the string "sea" is present

```
text1.concordance("sea", 20, 5)
```

```
Displaying 5 of 455 matches:
 in the sea ." -- I
 Indian Sea breedet
 on the sea , when
 of the sea , appea
ing the sea before
```

Difference between built-in count() and nltk count()

Python's built-in count() works text. It goes through the string and sees if any substring matches the input

NLTK's count() takes text, tokenizes it, and then runs Python's built-in count() on the resulting list

Below, we make a string variable called "normalText" by taking the tokens of text1 and concatening them together into a string. We call the built-in count() on normalText and we pass in the string "sea"

After that, we call NLTK's count() on text1 and pass in the string "sea" as well.

The output displays how the different count()'s arrive at different counts of the string "sea"

```
normalText = ""
for token in text1.tokens:
  normalText+=token + " "

print("Built-in count(): " +str(normalText.count("sea")) + "\nNLTK count(): " + str(text1.cou
```

```
Built-in count(): 702
NLTK count(): 433
```

WORD_TOKENIZE

Below, we are using word_tokenize on a string variable called "raw_text" to get a list of tokens. We then print out the first 10 of those tokens.

We were free to pick what raw_text could be. Since we are dealing with words in this class, I figured "Why not pick a passage all about the Word" and so I chose John 1:1-5. But of course, this Word isn't just a noise or a sound but the full expression and revelation of God. This Word, we find out in John 1:14, became flesh and dwelt among us.

This Word is Jesus Christ the Lord

```
raw_text = """In the beginning was the Word, and the Word was with God, and the Word was God.
He was in the beginning with God.
All things came into being through Him, and apart from Him nothing came into being that has c
In Him was life, and the life was the Light of men.
The Light shines in the darkness, and the darkness did not comprehend it."""
tokens = word_tokenize(raw_text)
tokens[0:10]
```

```
['In', 'the', 'beginning', 'was', 'the', 'Word', ',', 'and', 'the', 'Word']
```

SENT_TOKENIZE

Above, we used word_tokenize() on raw_text to get the tokens

Below we now use sent_tokenize() on raw_text to get the sentences of raw_text and output them

```
from nltk.tokenize import sent_tokenize
sent_tokenize(raw_text)
```

```
['In the beginning was the Word, and the Word was with God, and the Word was God.',
 'He was in the beginning with God.',
 'All things came into being through Him, and apart from Him nothing came into being
that has come into being.',
 'In Him was life, and the life was the Light of men.',
 'The Light shines in the darkness, and the darkness did not comprehend it.']
```

PorterStemmer

After importing the needed methods, we make a PorterStemmer() object and name it stemmer. Using a list comprehension, we stem each of the tokens of raw_text and print out the resulting list to the screen

```
from nltk.stem import *
stemmer = PorterStemmer()
stemmedTokens = [stemmer.stem(token) for token in tokens]
stemmedTokens
```

```
['in',
 'the',
 'begin',
 'wa',
 'the',
 'word',
 ',',
 'and',
 'the',
 'word',
 'wa',
 'with',
 'god',
 ',',
 'and',
 'the',
 'word',
 'wa',
 'god',
 '.',
 'he',
 'wa',
 'in',
 'the',
```

```
 'begin',
 'with',
 'god',
 '.',
 'all',
 'thing',
 'came',
 'into',
 'be',
 'through',
 'him',
 ',',
 'and',
 'apart',
 'from',
 'him',
 'noth',
 'came',
 'into',
 'be',
 'that',
 'ha',
 'come',
 'into',
 'be',
 '.',
 'in',
 'him',
 'wa',
 'life',
 ',',
 'and',
 'the',
 'life',
```

WordNetLemmatizer

After importing the needed methods, we make a WordNetLemmatizer() object and name it wnl. Using a list comprehension, we lemmatize each of the tokens of raw_text and print out the resulting list to the screen

```
from nltk.stem import WordNetLemmatizer
wnl = WordNetLemmatizer()
lemmatizedTokens = [wnl.lemmatize(token) for token in tokens]
lemmatizedTokens
```

```
['In',
 'the',
 'beginning',
 'wa',
 'the',
 'Word',
```

```
        ',',
        'and',
        'the',
        'Word',
        'wa',
        'with',
        'God',
        ',',
        'and',
        'the',
        'Word',
        'wa',
        'God',
        '.',
        'He',
        'wa',
        'in',
        'the',
        'beginning',
        'with',
        'God',
        '.',
        'All',
        'thing',
        'came',
        'into',
        'being',
        'through',
        'Him',
        ',',
        'and',
        'apart',
        'from',
        'Him',
        'nothing',
        'came',
        'into',
        'being',
        'that',
        'ha',
        'come',
        'into',
        'being',
        '.',
        'In',
        'Him',
        'wa',
        'life',
        ',',
        'and',
        'the',
        'life',
```

Conclusion

Functionality of the NLTK library

The NLTK Library is huge! As such, there is a lot of functionality inside of it. You can tokenize, stem, and lemmaize and that is just the surface level. You have access to a large body of text to work with and test out as well as functions such as concordance

Code Quality of the NLTK library

I am not really in a position to judge code quality since all my code is pretty awful. But I do wish the documentation online was a bit better. For example, if they gave mroe examples on how various functions worked, that would make learning NLTK more intutivite. I also had a lot of difficulty accessing the functions I wanted since I needed to keep importing or downloading various things. If there was a way to streamline things, I think that could be a great help

Potential Usages for NLTK

In terms of projects, I think being able to tokenize text will be quite helpful for the future of the class. In terms of personal use, I would like to see if someone has text of the New Testament in Greek. Being able to use a concordance on that would save so much time when studying the Bible!

Could not connect to the reCAPTCHA service. Please check your internet connection and reload to get a reCAPTCHA challenge.