CS4395 Portfolio Assignment 2: Exploring NLTK

Shane Arwood

September 11, 2022

This notebook uses sample texts from nltk as well as a separate exerpt to examine nltk's tokenize, lemmatize, stem, and concordance functionalities among others.

```
import nltk
nltk.download('book')
nltk.download('omw-1.4')
```

```
[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]     | Downloading package book_grammars to
[nltk_data]     |     /root/nltk_data...
[nltk_data]     |   Package book_grammars is already up-to-date!
```

```
[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


[nltk_data] Downloading package omw-1.4 to /root/nltk_data...
True
```

```
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
```

The code block uses the tokens() method to obtain the first 20 tokens of text1. Two things learned:

1) Text objects wrap tokens of strings that create a document, which can be extracted into a list via the tokens() method.

2) The tokens() method extracts punctuation as well as individual words from the Text object.

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

```
['[',
 'Moby',
 'Dick',
 'by',
 'Herman',
 'Melville',
 '1851',
 ']',
```

```
 'ETYMOLOGY',
 '.',
 '(',
 'Supplied',
 'by',
 'a',
 'Late',
 'Consumptive',
 'Usher',
 'to',
 'a',
 'Grammar']
```

This code block prints a concordance for the word 'sea' in text1.

```
text1.concordance('sea', 80, 5)
```

```
Displaying 5 of 455 matches:
 shall slay the dragon that is in the sea ." -- ISAIAH " And what thing soever
 S PLUTARCH ' S MORALS . " The Indian Sea breedeth the most and the biggest fis
cely had we proceeded two days on the sea , when about sunrise a great many Wha
many Whales and other monsters of the sea , appeared . Among the former , one w
 waves on all sides , and beating the sea before him into a foam ." -- TOOKE '
```

The API's count method takes a value of any type as a parameter and returns the number of times that value occurs in the Text object by tokenizing the calling object into a list first. Python's count method is already performed on a list or string, so it just takes the value and returns the nuber of times that value occurs.

```
# Python count() - performed on a list
textSample = ['Woodchuck', 'Chuck', 'chucks', 'wood']
textSample.count('chucks')
```

```
1
```

```
# nltk count() - performed on a text object
text1.count('sea')
```

```
433
```

The code block saves the following exerpt into a variable then tokenizes the text and prints the first 10 tokens.

"She had lost a portion of her ease. She cleared her throat again, contemplated the gray birdbath, which was supported by a number of carved figures, gnomes or elves, with patient bearded faces, who seemed in the act of bearing it away. Cloud sighed. She glanced at a tiny gold watch that was

pinned to her bosom. It had little curling wings on either side. Time flies. She looked at Smoky and smiled apologetically."

Excerpt From Little, Big John Crowley https://books.apple.com/us/book/little-big/id451592203 This material may be protected by copyright.

```
raw_text = 'She had lost a portion of her ease. She cleared her throat again, contempl
from nltk.tokenize import word_tokenize
tokens = word_tokenize(raw_text)
tokens[:10]
```

```
['She', 'had', 'lost', 'a', 'portion', 'of', 'her', 'ease', '.', 'She']
```

Using the same raw text, the code block tokenizes the text into sentences and displays them.

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

```
['She had lost a portion of her ease.',
 'She cleared her throat again, contemplated the gray birdbath, which was
supported by a number of carved figures, gnomes or elves, with patient bearded
faces, who seemed in the act of bearing it away.',
 'Cloud sighed.',
 'She glanced at a tiny gold watch that was pinned to her bosom.',
 'It had little curling wings on either side.',
 'Time flies.',
 'She looked at Smoky and smiled apologetically.']
```

The code block uses nltk's porter stemmer and a list comprehension to stem the text. The list is displayed.

```
from nltk.stem.porter import *
token_stemmer = PorterStemmer()
stemmed_tokens = [token_stemmer.stem(t) for t in tokens]
print(stemmed_tokens)
```

```
time', 'fli', '.', 'she', 'look', 'at', 'smoki', 'and', 'smile', 'apologet', '.']
```

The code block uses nltk's WordNetLemmatizer and a list comprehension to lemmatize the text and display the list.

Differences between stems and lemmas, listed as stem-lemma:

1. clear-cleared
2. contempl-contemplated
3. support-supported
4. fli-fly
5. apologet-apologetically

```
from nltk.stem import WordNetLemmatizer
token_lemmatizer = WordNetLemmatizer()
lemmatized_tokens = [token_lemmatizer.lemmatize(t) for t in tokens]
print(lemmatized_tokens)
```

```
ly', '.', 'She', 'looked', 'at', 'Smoky', 'and', 'smiled', 'apologetically', '.']
```

Final comment cell: reflection

I am impressed by the functionality of the nltk library. Very specific tasks, such as lemmatizing a text, can easy be accomplished using a simple function on a text object. I think the API for text objects is very useful as well, especially with reagrd to how simple it is to analyse specific portions of a text using colons. The code qualiy of the nltk library seems to be very high quality. The API is well documented, and the code is written in such a way that it is easy to follow how different functions are implemented. When writing code using the nltk library, the syntax is specific and intuitive, making it an effective library for coding in not just in functionality but also convenience. NLTK can be used in future projects as a simple means to parse and analyze text. In other languages, such tasks can be daunting. Using python's data structures in conjunction with the nltk library can make the first step of a project much more effective so time can be focused on higher level tasks. In particular, I beleive the tokens and concordances will be useful, along with the lemmatizing functions.

Colab paid products - Cancel contracts here

0s completed at 11:17 PM