Harshith Nanda

CS 4395.001 - Professor Mazidi

09/11/2022

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

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Package stopwords 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 punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package omw-1.4 to /root/nltk_data...
[nltk_data]   Package omw-1.4 is already up-to-date!
True
```

```
nltk.download('book')
```

```
[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...
[nltk_data]    |   Unzipping corpora/udhr2.zip.
[nltk_data]    | Downloading package unicode_samples to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping corpora/unicode_samples.zip.
[nltk_data]    | Downloading package webtext to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/webtext.zip.
```

```
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
```

Question 3

1. The tokens method contains a list with every word (token) in the text object.
2. The method has no parameters, it simply returns the contents of the list parameter in the text object.

```
tokens = text1.tokens
print(tokens[0:20])
```

```
['[', 'Moby', 'Dick', 'by', 'Herman', 'Melville', '1851', ']', 'ETYMOLOGY', '.', '(', 'S
```

Question 4 - This code cell demonstrates the concordance method.

```
text1.concordance('sea', lines=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 '
```

Question 5 -

Its a wrapper method for the python count method. It returns a call to the python count method with the list of tokens itself as the parameter.

```
count = text1.count('sea')
print(count)

count = text1.tokens.count('sea')
print(count)
```

```
433
433
```

Question 6 -

The passage is from "The Possibility of an Island" by Michael Houellebecq.

The code demonstrates the word tokenizer, which tokenizes text and turns it into a list.

```
from nltk.tokenize import word_tokenize
raw_text = 'Your only chance of survival, if you are sincerely smitten, lies in hiding this f
tokens = word_tokenize(raw_text)
print(tokens[0:10])
```

```
['Your', 'only', 'chance', 'of', 'survival', ',', 'if', 'you', 'are', 'sincerely']
```

Question 7 -

Demonstrates the sentence tokenizer. Splits the text into sentences.

```
from nltk.tokenize import sent_tokenize
sent_tokens = sent_tokenize(raw_text)
print("\n".join(sent_tokens))
```

```
Your only chance of survival, if you are sincerely smitten, lies in hiding this fact fro
What sadness there is in this simple observation!
What an accusation against man!
However, it had never occurred to me to contest this law, nor to imagine disobeying it:
```

Question 8 -

Demonstrates the porterstemmer, which stems tokens.

```
from nltk.stem import *
stemmer = PorterStemmer()
food = 'pomogranate radish wasabi Loganberry Lychee Magellan Barberry Apple Mamey Sapote'
stem_tokens = [stemmer.stem(word) for word in food.split(" ")]
print(stem_tokens)
```

```
['pomogran', 'radish', 'wasabi', 'loganberri', 'lyche', 'magellan', 'barberri', 'appl',
```

Question 9 -

Demonstrates the WordNetLemmatizer. Lemmatizes tokens.

Some differences include: pomogran - pomogranate loganberri - Loganberry lyche - Lychee barberri - Barberry appl - Apple sapot - Sapote

```
from nltk.stem.porter import *
from nltk.stem import WordNetLemmatizer
wnl = WordNetLemmatizer()
lemma_tokens = [wnl.lemmatize(token, pos="n") for token in food.split(" ")]
print(lemma_tokens)
```

```
['pomogranate', 'radish', 'wasabi', 'Loganberry', 'Lychee', 'Magellan', 'Barberry', 'App
```

Question 10 -

1. The NLTK library is a useful library for text processing with Python. The library has a diverse array of objects and methods to help make text processing easier and build machine learning models to analyze text. However, the methods aren't perfect, since you can sometimes see

incorrect stemming and lemmatizing of certain words, but this isn't that big of a deal in the grand scheme of things, and should not significantly affect the usefulness of the library.

2. The code is simple enough. The methods are somewhat difficult to use at first for beginners, but you get the hang of it quickly. I gets easier to use as you learn. There is very detailed documentation and tutorials available to learning isn't particularly difficult. The code for the nltk library is simple enough to interpret.
3. I think nltk is a must have for any machine learning model that processes text. I have worked on sentiment analysis models in the past and have used nltk for them. I think nltk will be useful for building bots that can interpret and respond to text, such as chat bots, etc. It is a must have for tools such as moderation of social media domains such as twitter, facebook, etc.