Assignment 2: Exploring NLTK

Charith Muppidi - srm190013

```
import nltk
nltk.download('stopwords')
nltk.download('wordnet')
nltk.download('punkt')
nltk.download('omw-1.4')
nltk.download('genesis')
nltk.download('inaugural')
nltk.download('nps_chat')
nltk.download('webtext')
nltk.download('treebank')
nltk.download('gutenberg')
```

```
[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!
[nltk_data] Downloading package genesis to /root/nltk_data...
[nltk_data]   Package genesis is already up-to-date!
[nltk_data] Downloading package inaugural to /root/nltk_data...
[nltk_data]   Package inaugural is already up-to-date!
[nltk_data] Downloading package nps_chat to /root/nltk_data...
[nltk_data]   Package nps_chat 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 treebank to /root/nltk_data...
[nltk_data]   Package treebank is already up-to-date!
[nltk_data] Downloading package gutenberg to /root/nltk_data...
[nltk_data]   Package gutenberg is already up-to-date!
True
```

The tokens() method functions more as an attribute caller which is why to call it you have to write like .tokens. Throughtout the object class, attributes are private instance variables, as denotes by the underscores, to method calls are used to access them.

```
from nltk.book import text1
text1.tokens[:20]
```

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

From nltk.book we are getting text1 "Moby Dick" by Herman Melville, and displaying the tokens from indexes 0 to 19.

```
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 '
```

Here, we use concordance to find the first 5 lines that contain the word 'sea' and display them, aligning the 'sea's with each other.

There really is not much of a difference between the two Count() methods since the Text objects count() method works by making the tokens attribute call Python's count() method.

```
a = 'Sea'
print(text1.count('sea'))
print(text1.count("Sea"))
print(text1.count(a))
print(text1.tokens.count('sea'))
print(text1.tokens.count("Sea"))
print(text1.tokens.count(a))

    433
    22
    22
    433
    22
    22
```

I am using Python's and Text's count() method on the word sea to see if there is any difference between the two. The count() methods are returning the number of times the token appears in the text.

```
# Franklin Delano Roosevelt's Inaugural Address, March 04, 1933
raw_text = """I am certain that my fellow Americans expect that on my induction into the Pres
I will address them with a candor and a decision which the present situation of our nation im
This is preeminently the time to speak the truth, the whole truth, frankly and boldly.
Nor need we shrink from honestly facing conditions in our country today.
This great nation will endure as it has endured, will revive and will prosper.
So, first of all, let me assert my firm belief that the only thing we have to fear is fear it
nameless, unreasoning unjustified terror which paralyzes needed efforts to convert retreat in
In every dark hour of our national life a leadership of frankness and vigor has met with that
support of the people themselves, which is essential to victory.
I am convinced that you will again give that support to leadership in these critical days."""
tokens = nltk.word_tokenize(raw_text)
print(tokens[:10])
```

```
    ['I', 'am', 'certain', 'that', 'my', 'fellow', 'Americans', 'expect', 'that', 'on']
```

I took the first five sentences in FDR's first Inaugural addres, then I tokenized each words and printed the first 10 of them.

```
sentences = nltk.sent_tokenize(raw_text)
[t for t in sentences]
```

```
    ['I am certain that my fellow Americans expect that on my induction into the Presidency
    \nI will address them with a candor and a decision which the present situation of our
    nation impels.',
     'This is preeminently the time to speak the truth, the whole truth, frankly and
    boldly.',
     'Nor need we shrink from honestly facing conditions in our country today.',
     'This great nation will endure as it has endured, will revive and will prosper.',
     'So, first of all, let me assert my firm belief that the only thing we have to fear is
    fear itself -- \nnameless, unreasoning unjustified terror which paralyzes needed
    efforts to convert retreat into advance.',
     'In every dark hour of our national life a leadership of frankness and vigor has met
    with that understanding and \nsupport of the people themselves, which is essential to
    victory.',
     'I am convinced that you will again give that support to leadership in these critical
    days.']
```

This time I am tokenizing and printing the sentences.

```
stemer = nltk.PorterStemmer()
[stemer.stem(t) for t in tokens]
```

```
      'paralyz',
      'need',
      'effort',
      'to',
      'convert',
      'retreat',
      'into',
      'advanc',
      '.',
      'in',
      'everi',
      'dark',
      'hour',
      'of',
      'our',
      'nation',
      'life',
      'a',
      'leadership',
      'of',
      'frank',
      'and',
      'vigor',
      'ha',
      'met',
      'with',
      'that',
      'understand',
      'and',
      'support',
      'of',
      'the',
      'peopl',
      'themselv',
      ',',
      'which',
      'is',
      'essenti',
      'to',
      'victori',
      '.',
      'i',
      'am',
      'convinc',
      'that',
      'you',
      'will',
      'again',
      'give',
      'that',
      'support',
      'to',
      'leadership',
      'in',
      'these',
      'critic',
      'day',
      '.']
```

We are using the PorterStemmer to stem the words in the tokens list.

i-I

presid-Presidency

decis-decision

situat-situation

impel-impels

```
lmr = nltk.WordNetLemmatizer()
[lmr.lemmatize(t) for t in tokens]
```

```
      'it',
      'ha',
      'endured',
      ',',
      'will',
      'revive',
      'and',
      'will',
      'prosper',
      '.',
      'So',
      ',',
      'first',
      'of',
      'all',
      ',',
      'let',
      'me',
      'assert',
      'my',
      'firm',
      'belief',
      'that',
      'the',
      'only',
      'thing',
      'we',
      'have',
      'to',
      'fear',
      'is',
      'itself',
      '--',
      'nameless',
      ',',
      'unreasoning',
      'unjustified',
      'terror',
      'which',
      'paralyzes',
      'needed',
      'effort',
      'to',
      'convert',
      'retreat',
      'into',
      'advance',
      '.',
      'In',
      'every',
      'dark',
      'hour',
      'of',
      'our',
      'national',
      'life',
      'a',
```

Double-click (or enter) to edit

Now we are using WordNetLemmatizer to lemmatize the words.

Since this a Python library, the code is top tier. The NLTK library is very functional: there are so many uses for this. I am thinking of using this library to stemming words to find root words. I will also use a lemmatizer to find words that are found in the dictionary. With these to processes, I can compare root/base words with versions with prefixes and suffixes, and I can analyze sound changes in that occur in specific letters.