Portfolio Assignment 2: Exploring NLTK

Instruction Steps

Step 1 - Create a Python Notebook

Python Notebook created by Jupyter Notebook

Step 2 - Import NLTK and libraries

NLTK imported and libraries installed.

Step 3 - NLTK Text Object

Import NLTK Text Object

```
In [1]: 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
```

Two things about tokens() method and Text object:

1. Each token in tokens() method is a word or a punctuation.
2. Each text in Text object is text document and already tokenized

The code below extract the first 20 tokens from text1 and save it to first20_token variable. Then print out the result

```
In [2]: first20_token = []
        for i in range(0,20):
          first20_token.append(text1.tokens[i])
        print(first20_token)
```

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

Step 4 - The concordance() method in API

- The code below search for the word 'sea' inside the text1. Word matching is not case sensetive.
- Then print out 5 lines which contain the word 'sea'.
- Each line will contain 80 character (0-79)

In [3]:
```
text1.concordance('sea', 79, 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 '
```

Step 5 - The count() method in API

- The count() method in the API work by counting the number of times this word appear in the text. The word here is a string.
- It works the same as Python's count method because in Python the count() method return the count of how many time a object occurs in list.

The count() method in API below count the number of time the word 'sea' appear in text1

In [4]:
```
text1.count('sea')
```

Out[4]:
```
433
```

The count() method in Python below count the integer 1 inside a list name mixList which contain different type of data type.

In [5]:
```
lst = [7,8,9]
mixList = ['str1', 'str2', 'str2', 1, 2.2, lst, 1, 1]
mixList.count(1)
```

Out[5]:
```
3
```

Step 6 - NLTK's word tokenizer

The raw_text input is from "Route 66 California: The End of the Trail by VOA"

'When Route 66 travelers cross the Colorado River into the state of California, it is easy to begin to feel like the journey is almost over. After all, California is the eighth and final state you visit. But, the end of the road in Los Angeles is still a 500-kilometer drive away. And much of that drive goes through the treacherous Mojave Desert. Eastern California’s Route 66 is broken, hot and deserted. The road passes crumbled buildings of bypassed towns. Needles is first city in California after crossing the Colorado River. It was once a major stop along Route 66 and the Old National Trails Highway. That road was built before Route 66, in 1913.'

The code below tokenize the raw text into words and save them into variable tokens using NLTK's word tokenizer. Then print out the first 10 tokens

```
In [6]: from nltk import word_tokenize
        raw_text = 'When Route 66 travelers cross the Colorado River into the state of California, it is easy to begi
        n to feel like the journey is almost over. After all, California is the eighth and final state you visit. Bu
        t, the end of the road in Los Angeles is still a 500-kilometer drive away. And much of that drive goes throug
        h the treacherous Mojave Desert. Eastern California’s Route 66 is broken, hot and deserted. The road passes c
        rumbled buildings of bypassed towns. Needles is first city in California after crossing the Colorado River. I
        t was once a major stop along Route 66 and the Old National Trails Highway. That road was built before Route
         66, in 1913.'
        tokens = word_tokenize(raw_text)
        for i in range(0,10):
            print(tokens[i])
```

```
When
Route
66
travelers
cross
the
Colorado
River
into
the
```

Step 7 - NLTK's sentence tokenizer

The code below tokenize the raw text into sentences and save them into variable sentences using NLTK's sentence tokenizer. Then print out each sentences.

In [7]:
```
from nltk import sent_tokenize
sentences = sent_tokenize(raw_text)
for sentence in sentences:
    print(sentence)
```

```
When Route 66 travelers cross the Colorado River into the state of California, it is easy to begin to feel li
ke the journey is almost over.
After all, California is the eighth and final state you visit.
But, the end of the road in Los Angeles is still a 500-kilometer drive away.
And much of that drive goes through the treacherous Mojave Desert.
Eastern California’s Route 66 is broken, hot and deserted.
The road passes crumbled buildings of bypassed towns.
Needles is first city in California after crossing the Colorado River.
It was once a major stop along Route 66 and the Old National Trails Highway.
That road was built before Route 66, in 1913.
```

Step 8 - NLTK's PorterStemmer

Stemming means removing affixes. The result may be tokens that are not actually words or have different meanings

The code below using list comprehension to stem the text and display the list.

- The raw text was tokenizing into words and save them into variable tokens using NLTK's word tokenizer.
- Then the variable tokens was stemmed into variable stemmed using NLTK's PorterStemmer

```
In [8]: from nltk.stem.porter import PorterStemmer
        stemmer = PorterStemmer()
        tokens = word_tokenize(raw_text)
        stemmed = [stemmer.stem(t) for t in tokens]
        print(stemmed)
```

```
['when', 'rout', '66', 'travel', 'cross', 'the', 'colorado', 'river', 'into', 'the', 'state', 'of', 'californ
ia', ',', 'it', 'is', 'easi', 'to', 'begin', 'to', 'feel', 'like', 'the', 'journey', 'is', 'almost', 'over',
'.', 'after', 'all', ',', 'california', 'is', 'the', 'eighth', 'and', 'final', 'state', 'you', 'visit', '.',
'but', ',', 'the', 'end', 'of', 'the', 'road', 'in', 'lo', 'angel', 'is', 'still', 'a', '500-kilomet', 'driv
e', 'away', '.', 'and', 'much', 'of', 'that', 'drive', 'goe', 'through', 'the', 'treacher', 'mojav', 'deser
t', '.', 'eastern', 'california', ''', 's', 'rout', '66', 'is', 'broken', ',', 'hot', 'and', 'desert', '.',
'the', 'road', 'pass', 'crumbl', 'build', 'of', 'bypass', 'town', '.', 'needl', 'is', 'first', 'citi', 'in',
'california', 'after', 'cross', 'the', 'colorado', 'river', '.', 'it', 'wa', 'onc', 'a', 'major', 'stop', 'al
ong', 'rout', '66', 'and', 'the', 'old', 'nation', 'trail', 'highway', '.', 'that', 'road', 'wa', 'built', 'b
efor', 'rout', '66', ',', 'in', '1913', '.']
```

Step 9 - NLTK's WordNetLemmatizer

Lemmatization is similar to stemming but considers the context and converts the word to its meaningful base form, which is called Lemma.

The code below using list comprehension to lemmatize the text and display the list.

- The raw text was tokenizing into words and save them into variable tokens using NLTK's word tokenizer.
- Then the variable tokens was lemmatized into variable lemmas using NLTK's WordNetLemmatizer.

```
In [9]: from nltk.stem import WordNetLemmatizer
        tokens = word_tokenize(raw_text)
        wnl = WordNetLemmatizer()
        lemmas = [wnl.lemmatize(t) for t in tokens]
        print(lemmas)
```

```
['When', 'Route', '66', 'traveler', 'cross', 'the', 'Colorado', 'River', 'into', 'the', 'state', 'of', 'Calif
ornia', ',', 'it', 'is', 'easy', 'to', 'begin', 'to', 'feel', 'like', 'the', 'journey', 'is', 'almost', 'ove
r', '.', 'After', 'all', ',', 'California', 'is', 'the', 'eighth', 'and', 'final', 'state', 'you', 'visit',
'.', 'But', ',', 'the', 'end', 'of', 'the', 'road', 'in', 'Los', 'Angeles', 'is', 'still', 'a', '500-kilomete
r', 'drive', 'away', '.', 'And', 'much', 'of', 'that', 'drive', 'go', 'through', 'the', 'treacherous', 'Mojav
e', 'Desert', '.', 'Eastern', 'California', ''', 's', 'Route', '66', 'is', 'broken', ',', 'hot', 'and', 'dese
rted', '.', 'The', 'road', 'pass', 'crumbled', 'building', 'of', 'bypassed', 'town', '.', 'Needles', 'is', 'f
irst', 'city', 'in', 'California', 'after', 'crossing', 'the', 'Colorado', 'River', '.', 'It', 'wa', 'once',
'a', 'major', 'stop', 'along', 'Route', '66', 'and', 'the', 'Old', 'National', 'Trails', 'Highway', '.', 'Tha
t', 'road', 'wa', 'built', 'before', 'Route', '66', ',', 'in', '1913', '.']
```

Five differences in the stems verses the lemmas. (stem - lemma)

- rout - Route
- travel - traveler
- treacher - treacherous
- crumbl - crumbled
- needl - Needles
- citi - city

Step 10 - Comment on NLTK libraries

Functionality of NLTK libraries

- The NLTK libraries are very usefull when doing the text processing. It offers a variety methods such as word_tokenize(), sent_tokenize(), stemming, lemmatization, etc...These methods provide efficent ways to normalize and break raw text into smaller part such as word, sentence, grouping word with same root.
- The NLTK libraries also include a lot built-in text which user can test and work right away.

Code quality of NLTK libraries

The NLTK libraries generate a better output on text processing compare to similar Python's method(). For example:

- The word_tokenizer() can seperated word and punctuation better then split() method in Python without the separator specification
- The sent_tokenizer() can perform sentence segmentation very well. NLTK will not end a sentence on just any '.'. It can recognize the different the dot in "Dr." and the dot which end the sentences.

Future project using NLTK

- NLTK can be used in combination of speech recognization to analyzing a conversation.
- NLTK can be used for sentiment analysis and evaluation.