Portfolio Assignment 3: Exploring NLTK

Objectives:

- Practice using features of NLTK
- Examine a professional-level NLP API

Turn in:

- Upload your pdf document to eLearning for grading
- Also upload the pdf document to your portfolio and link to it from your index page

Instructions:

1. Create a Python notebook (Jupyter or Colab) with appropriate headings. You will later print-to-pdf for uploading. ***Note***: intersperse all the code cells below with text cells that use markdown to describe what the code is doing and its output. Make sure that your notebook displays the code output.
2. If you use Jupyter notebook with NLTK and libraries installed plus the nltk book download, you are good to go. If you use Colab, insert a code chunk at the top of your notebook to install these items:

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

3. Code cell: Each of the built-in 9 texts is an NLTK Text object. Look at the code for the Text object at this link: https://www.nltk.org/_modules/nltk/text.html. Look at the tokens() method. Extract the first 20 tokens from text1. List two things you learned about the tokens() method or Text objects in the text cell above this code cell.
4. Look at the concordance() method in the API. Using the documentation to guide you, in code, print a concordance for text1 word 'sea', selecting only 5 lines.
5. Code cell: Look at the count() method in the API. How does this work, and how is it different or the same as Python's count method? Write your commentary above the code cell. In the code cells, experiment with both count() methods.
6. Code cell: Using raw text of at least 5 sentences of your choice from any source (cite the source), save the text into a variable called raw_text. Using NLTK's word tokenizer, tokenize the text into variable 'tokens'. Print the first 10 tokens.
7. Code cell: Using the same raw text, and NLTK's sentence tokenizer sent_tokenize(), perform sentence segmentation and display the sentences.
8. Code cell: Using NLTK's PorterStemmer(), write a list comprehension to stem the text. Display the list.

Caution: All course work is run through plagiarism detection software comparing students' work as well as work from previous semesters and other sources.

9. Code cell: Using NLTK's WordNetLemmatizer, write a list comprehension to lemmatize the text. Display the list. In the text cell above this code cell, list at least 5 differences you see in the stems verses the lemmas. You can just write them each on a line, like this:
   stem-lemma

10. Comment cell: Write a paragraph outlining:
    a. your opinion of the functionality of the NLTK library
    b. your opinion of the code quality of the NLTK library
    c. a list of ways you may use NLTK in future projects

Grading Rubric:

- 10 points per step