Exploring NLTK Michael Petrey

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')

```
from nltk import *
```

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.

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

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

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

```
text1.concordance('sea', 75,5)
```

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

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.

In the NLTK API, the count function works by counting the number of tokens in the text that equal the desired word. In Python, the count function counts the number of elements in the text that equal the search term.

```
sample_text = 'Hi this is a sample text'
```

```
sample_tokens = word_tokenize(sample_text)

# nltk
print(sample_text.count('a'))


# python
print(sample_tokens.count('a'))
```

```
    2
    1
```

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.

Source: Herbert, Frank. Dune. Hodder Paperback, 2006.

```
from nltk.tokenize import word_tokenize

raw_text = "A beginning is the time for taking the most delicate care that the balances are correct. This every sister of the Bene

tokens = word_tokenize(raw_text)

tokens[:20]
```

```
    ['A',
     'beginning',
     'is',
     'the',
     'time',
     'for',
     'taking',
     'the',
     'most',
     'delicate',
     'care',
     'that',
     'the',
     'balances',
     'are',
     'correct',
     '.',
     'This',
     'every',
     'sister']
```

7. Code cell: Using the same raw text, and NLTK's sentence tokenizer sent_tokenize(), perform sentence segmentation and display the sentences.

```
from nltk.corpus.reader.tagged import sent_tokenize

sent_tokenize(raw_text)
```

```
    ['A beginning is the time for taking the most delicate care that the balances are correct.',
     'This every sister of the Bene Gesserit knows.',
     'To begin your study of the life of MuadDib, then, take care that you first place him in his time: born in the 57th year of
    the Padishah Emperor, Shaddam IV.',
     "And take the most special care that you locate Muad'Dib in his place: the planet Arrakis.",
     'Do not be deceived by the fact that he was born on Caladan and lived his first fifteen years there.',
     'Arrakis, the planet known as Dune, is forever his place.']
```

8. Code cell: Using NLTK's PorterStemmer(), write a list comprehension to stem the text. Display the list.

```
from nltk.stem import *
from nltk.stem.porter import *
from nltk.stem.snowball import SnowballStemmer

stemmer = PorterStemmer()

stem = [stemmer.stem(plural) for plural in tokens]
print(stem)
```

```
['a', 'begin', 'is', 'the', 'time', 'for', 'take', 'the', 'most', 'delic', 'care', 'that', 'the', 'balanc', 'are', 'correct',
```

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

```
token_lemma = [ WordNetLemmatizer().lemmatize(token) for token in tokens ]
print(token_lemma)
```

```
['A', 'beginning', 'is', 'the', 'time', 'for', 'taking', 'the', 'most', 'delicate', 'care', 'that', 'the', 'balance', 'are',
```

Differences:

1. a-A
2. begin-beginning
3. take-taking
4. delic-delicate
5. balanc-balance

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

So far, nltk has been relatively straight forward. All of the functions are intuitive and well documented. NLTK would be useful for projects using sentiment analysis and speech recognition.

Colab paid products - Cancel contracts here