```
1 import nltk
2 nltk.download('stopwords')
3 nltk.download('wordnet')
4 nltk.download('punkt')
5 nltk.download('omw-1.4')
6 nltk.download('book')
```

```
[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.
[nltk_data]    | Downloading package wordnet to /root/nltk_data...
[nltk_data]    |   Package wordnet is already up-to-date!
[nltk_data]    | Downloading package wordnet_ic to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/wordnet_ic.zip.
[nltk_data]    | Downloading package words to /root/nltk_data...
[nltk_data]    |   Unzipping corpora/words.zip.
[nltk_data]    | Downloading package maxent_treebank_pos_tagger to
[nltk_data]    |     /root/nltk_data...
[nltk_data]    |   Unzipping taggers/maxent_treebank_pos_tagger.zip.
```

## Step 3 - Tokens Method

I learned that tokens() returns a list object, and that the tokens list is created upon instantiation of the Text object.

```
1 from nltk.book import text1
2 step3Ans = text1.tokens[0:20]
3 step3Ans
```

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

## Step 4 - Concordance

```
1 from nltk import ConcordanceIndex
2 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 '
```

## Step 5 - Count()

The count method in the API returns the number of instances of the given token in the text object provided. The count method from Python returns the number of objects that match the given object in the given list.

```
1 #NLTK count
2 print(text1.count("sea"))
3 #API Count
4 print(step3Ans.count("Moby"))
```

```
433
1
```

## Step 6 - Tokens

```
1 from nltk.corpus.reader.tagged import word_tokenize
2 import nltk.corpus
3 from nltk.text import Text
4 #Source - Harry Potter and the Half Blood Prince
5 raw_text = 'Voldemort himself created his worst enemy, just as tyrants everywhere do! H
6 tokens = word_tokenize(raw_text)
7 tokens[0:10]
```

```
['Voldemort',
 'himself',
 'created',
 'his',
 'worst',
 'enemy',
 ',',
 'just',
 'as',
 'tyrants']
```

## Step 7 - Sentence Segmentation

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

```
from nltk.text import Text
tokens = sent_tokenize(raw_text)
tokens
```

```
['Voldemort himself created his worst enemy, just as tyrants everywhere do!',
 'Have you any idea how much tyrants fear the people they oppress?',
 'All of them realize that, one day, amongst their many victims, there is sure to be
one who rises against them and strikes back!']
```

## Step 8 - List Comprehension

```
from nltk.stem import *
from nltk.corpus.reader.tagged import word_tokenize
import nltk.corpus
tokens = word_tokenize(raw_text)
stemmer = PorterStemmer()

stemTokens = [stemmer.stem(word) for word in tokens]
stemTokens
```

```
['voldemort',
 'himself',
 'creat',
 'hi',
 'worst',
 'enemi',
 ',',
 'just',
 'as',
 'tyrant',
 'everywher',
 'do',
 '!',
 'have',
 'you',
 'ani',
 'idea',
 'how',
 'much',
 'tyrant',
 'fear',
 'the',
 'peopl',
 'they',
 'oppress',
 '?',
 'all',
 'of',
 'them',
 'realiz',
 'that',
```

```
 ',',
 'one',
 'day',
 ',',
 'amongst',
 'their',
 'mani',
 'victim',
 ',',
 'there',
 'is',
 'sure',
 'to',
 'be',
 'one',
 'who',
 'rise',
 'against',
 'them',
 'and',
 'strike',
 'back',
 '!']
```

## Step 9 - WordNetLemmatize

Removes all s's at end of tokens – only removes s's where they most likely create plurals

Removes suffixes other than s – only removes s

Swaps y at end of tokens with I – only changes s at end of tokens

Changes letters – only removes letters

Changes uppercase to lowercase – does not change case

```
1 nltk.download('omw-1.4')
2 from nltk.stem import WordNetLemmatizer
3 from nltk.corpus.reader.tagged import word_tokenize
4 import nltk.corpus
5 tokens = word_tokenize(raw_text)
6 lemma = WordNetLemmatizer()
7
8 lemTokens = [lemma.lemmatize(word) for word in tokens]
9 lemTokens
```

```
[nltk_data] Downloading package omw-1.4 to /root/nltk_data...
[nltk_data]   Package omw-1.4 is already up-to-date!
['Voldemort',
 'himself',
 'created',
 'his',
```

```
 'worst',
 'enemy',
 ',',
 'just',
 'a',
 'tyrant',
 'everywhere',
 'do',
 '!',
 'Have',
 'you',
 'any',
 'idea',
 'how',
 'much',
 'tyrant',
 'fear',
 'the',
 'people',
 'they',
 'oppress',
 '?',
 'All',
 'of',
 'them',
 'realize',
 'that',
 ',',
 'one',
 'day',
 ',',
 'amongst',
 'their',
 'many',
 'victim',
 ',',
 'there',
 'is',
 'sure',
 'to',
 'be',
 'one',
 'who',
 'rise',
 'against',
 'them',
 'and',
 'strike',
 'back',
 '!']
```

## Step 10

In my opinion, the NLTK library is extremely useful. The variety of ways it offers for users to be able to process text allows the library to be applicable to a wide range of text processing tasks. As I have yet to encounter any bugs in the code, I can only assume that the code quality of the NLTK library is great. For me, the main way I would use NLTK in any future projects would be converting any text I need to handle into a simpler format to make it easier to work with in my program, allowing me to have whatever text recognition software I'm using look for the base forms of words instead of having to check every form of every word, vastly simplifying my projects.