

To Implement Basic Function in NLTK packages using python.

Procedure:

1. Install and import NLTK package

```
[1]: pip install nltk

Collecting nltk
  Downloading nltk-3.9.1-py3-none-any.whl.metadata (2.9 kB)
Collecting click (from nltk)
  Using cached click-8.1.7-py3-none-any.whl.metadata (3.0 kB)
Requirement already satisfied: joblib in c:\users\hema\appdata\roaming\python\python311\site-packages (from nltk) (1.2.0)
Collecting regex>=2021.8.3 (from nltk)
  Downloading regex-2024.11.6-cp311-cp311-win_amd64.whl.metadata (41 kB)
----- 0.0/41.5 kB ? eta -:-:--
----- 10.2/41.5 kB ? eta -:-:--
----- 20.5/41.5 kB 320.0 kB/s eta 0:00:01
----- 41.0/41.5 kB 326.8 kB/s eta 0:00:01
----- 41.5/41.5 kB 250.5 kB/s eta 0:00:00
Requirement already satisfied: tqdm in c:\users\hema\miniconda3\lib\site-packages (from nltk) (4.65.0)
Requirement already satisfied: colorama in c:\users\hema\appdata\roaming\python\python311\site-packages (from click->nltk) (0.4.6)
Downloading nltk-3.9.1-py3-none-any.whl (1.5 MB)
----- 0.0/1.5 MB ? eta -:-:--
----- 0.0/1.5 MB 1.4 MB/s eta 0:00:02
----- 0.1/1.5 MB 1.1 MB/s eta 0:00:02

[2]: import nltk
```

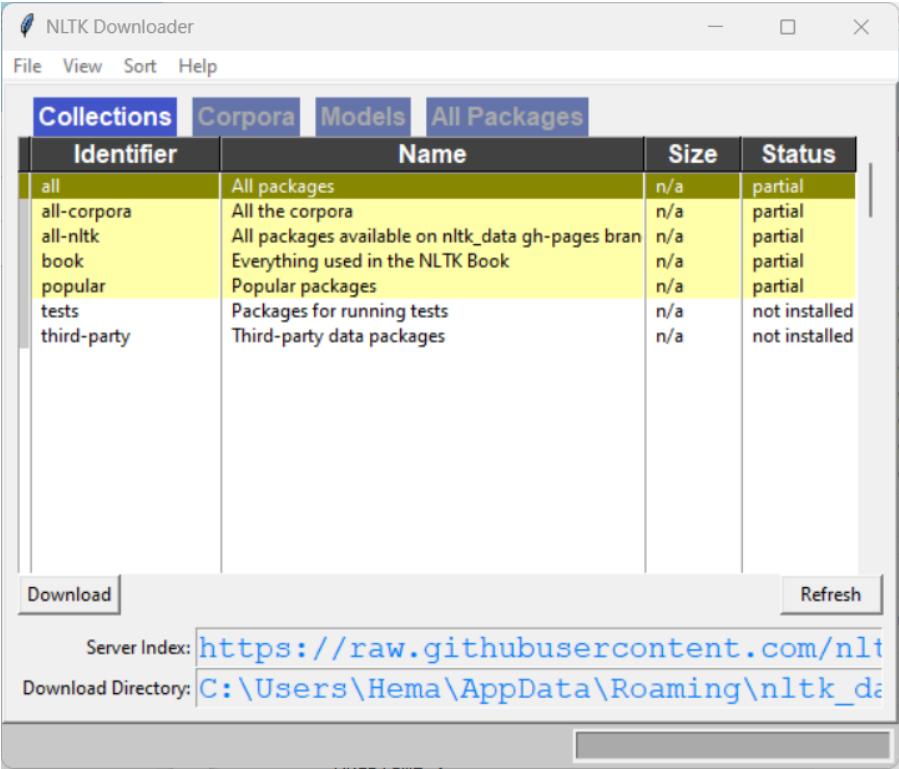
2. Explore all packages

```
[2]: import nltk

[3]: nltk.download()

showing info https://raw.githubusercontent.com/nltk/nltk_data/gh-pages/index.xml

[3]: True
```



3. Using dir function list all the functions

```
[4]: dir(nltk)

[4]: ['ARLSTem',
      'ARLSTem2',
      'AbstractLazySequence',
      'AffixTagger',
      'AlignedSent',
      'Alignment',
      'AnnotationTask',
      'ApplicationExpression',
      'Assignment',
      'BigramAssocMeasures',
      'BigramCollocationFinder',
      'BigramTagger',
      'BinaryMaxentFeatureEncoding',
      'BlanklineTokenizer',
      'BllipParser',
      'BottomUpChartParser',
      'BottomUpLeftCornerChartParser',
      'BottomUpProbabilisticChartParser']
```

4. Make use of any five functions

a. Tokenization

```
[39]: # Tokenization

text = "This is a Natural Language Processing concept. These are the basic function in NLP."
words = text.split()
print (words)

['This', 'is', 'a', 'Natural', 'Language', 'Processing', 'concept.', 'These', 'are', 'the', 'basic', 'function', 'in', 'NLP.']
```

b. Stemming

```
[40]: # Stemming

from nltk.stem.porter import PorterStemmer
stemmed = [PorterStemmer().stem(w) for w in words]
print (stemmed)

['thi', 'is', 'a', 'natur', 'languag', 'process', 'concept.', 'these', 'are', 'the', 'basic', 'function', 'in', 'nlp.']
```

c. Lemmatization

```
[41]: # Lemmatization

from nltk.stem.wordnet import WordNetLemmatizer
lemmed = [WordNetLemmatizer().lemmatize(w) for w in words]
print (lemmed)

['This', 'is', 'a', 'Natural', 'Language', 'Processing', 'concept.', 'These', 'are', 'the', 'basic', 'function', 'in', 'NLP.']
```

d. POS Tagging

```
[45]: # POS tagging

from nltk import pos_tag
from nltk import word_tokenize

text = "This is a Natural Language Processing concept. These are the basic function in NLP."
tokenized_text = word_tokenize(text)
tags = pos_tag(tokenized_text)
tags

[45]: [('This', 'DT'),
      ('is', 'VBZ'),
      ('a', 'DT'),
      ('Natural', 'JJ'),
      ('Language', 'NNP'),
      ('Processing', 'NNP'),
      ('concept', 'NN'),
      ('.', '.'),
      ('These', 'DT'),
      ('are', 'VBP'),
      ('the', 'DT'),
      ('basic', 'JJ'),
      ('function', 'NN'),
      ('in', 'IN'),
      ('NLP', 'NNP'),
      ('.', '.')]

🔍 ⬆ ⬇ ⬇ ⬇ ⬇
```

e. Frequency Distribution

```
[43]: # Frequency Distribution

from nltk.probability import FreqDist
from nltk.tokenize import word_tokenize

text = "This is a sample sentence. This sentence is for testing."
words = word_tokenize(text)
freq_dist = FreqDist(words)
most_common = freq_dist.most_common(5)
print (most_common)

[('This', 2), ('is', 2), ('sentence', 2), ('.', 2), ('a', 1)]
```