Untitled3

July 15, 2025

[3]:
```
from urllib import request
```

[4]:
```
url = "https://www.gutenberg.org/browse/scores/top#books-last1"
```

[5]:
```
response = request.urlopen(url)
```

[6]:
```
raw = response.read().decode('utf8')
```

[7]:
```
import nltk
from nltk.tokenize import word_tokenize
url = "https://www.gutenberg.org/browse/scores/top#books-last1"
response = request.urlopen(url)
tokens = word_tokenize(raw)
print(tokens[:200])
```

```
---------------------------------------------------------------------------
LookupError                               Traceback (most recent call last)
Cell In[7], line 5
      3 url = "https://www.gutenberg.org/browse/scores/top#books-last1"
      4 response = request.urlopen(url)
----> 5 tokens = word_tokenize(raw)
      6 print(tokens[:200])

File /opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages/nltk/
 ↪tokenize/__init__.py:129, in word_tokenize(text, language, preserve_line)
    114 def word_tokenize(text, language="english", preserve_line=False):
    115     """
    116     Return a tokenized copy of *text*,
    117     using NLTK's recommended word tokenizer
   (…)
    127     :type preserve_line: bool
    128     """
--> 129     sentences = [text] if preserve_line else␣
 ↪sent_tokenize(text, language)
    130     return [
    131         token for sent in sentences for token in␣
 ↪_treebank_word_tokenizer.tokenize(sent)
```

```
    132     ]

File /opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages/nltk/tokenize/__init__.py:106, in sent_tokenize(text, language)
     96 def sent_tokenize(text, language="english"):
     97     """
     98     Return a sentence-tokenized copy of *text*,
     99     using NLTK's recommended sentence tokenizer
   (…)
    104     :param language: the model name in the Punkt corpus
    105     """
--> 106     tokenizer = load(f"tokenizers/punkt/{language}.pickle")
    107     return tokenizer.tokenize(text)

File /opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages/nltk/data.py:750, in load(resource_url, format, cache, verbose, logic_parser, fstruct_reader, encoding)
    747     print(f"<<Loading {resource_url}>>")
    749 # Load the resource.
--> 750 opened_resource = _open(resource_url)
    752 if format == "raw":
    753     resource_val = opened_resource.read()

File /opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages/nltk/data.py:876, in _open(resource_url)
    873 protocol, path_ = split_resource_url(resource_url)
    875 if protocol is None or protocol.lower() == "nltk":
--> 876     return find(path_, path + [""]).open()
    877 elif protocol.lower() == "file":
    878     # urllib might not use mode='rb', so handle this one ourselves:
    879     return find(path_, [""]).open()

File /opt/conda/envs/anaconda-2024.02-py310/lib/python3.10/site-packages/nltk/data.py:583, in find(resource_name, paths)
    581 sep = "*" * 70
    582 resource_not_found = f"\n{sep}\n{msg}\n{sep}\n"
--> 583 raise LookupError(resource_not_found)

LookupError:
**********************************************************************
  Resource punkt not found.
  Please use the NLTK Downloader to obtain the resource:

  >>> import nltk
  >>> nltk.download('punkt')

  For more information see: https://www.nltk.org/data.html
  Attempted to load tokenizers/punkt/PY3/english.pickle

  Searched in:
    - '/home/e9f2e6f2-2728-40a7-9938-d80ebfe53d70/nltk_data'
    - '/opt/conda/envs/anaconda-2024.02-py310/nltk_data'
    - '/opt/conda/envs/anaconda-2024.02-py310/share/nltk_data'
    - '/opt/conda/envs/anaconda-2024.02-py310/lib/nltk_data'
    - '/usr/share/nltk_data'
    - '/usr/local/share/nltk_data'
    - '/usr/lib/nltk_data'
    - '/usr/local/lib/nltk_data'
    - ''
**********************************************************************
```

[8]:

```
import nltk
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     /home/e9f2e6f2-2728-40a7-9938-
[nltk_data]     d80ebfe53d70/nltk_data…
[nltk_data]   Unzipping tokenizers/punkt.zip.
```

[8]: True

[9]:

```
import nltk
from nltk.tokenize import word_tokenize
url = "https://www.gutenberg.org/browse/scores/top#books-last1"
response = request.urlopen(url)
tokens = word_tokenize(raw)
print(tokens[:200])
```

```
['<', '!', 'DOCTYPE', 'html', '>', '<', 'html', 'class=', "''", 'client-nojs',
"''", 'lang=', "''", 'en', "''", 'dir=', "''", 'ltr', "''", '>', '<', 'head',
'>', '<', 'meta', 'charset=', "''", 'UTF-8', "''", '>', '<', 'title', '>',
'Top', '100', '|', 'Project', 'Gutenberg', '<', '/title', '>', '<', 'link',
'rel=', "''", 'stylesheet', "''", 'href=', "''", '/gutenberg/style2.css', '?',
'v=1.5', "''", '>', '<', 'link', 'rel=', "''", 'stylesheet', "''", 'href=',
"''", '/gutenberg/collapsible.css', '?', '1.3', "''", '>', '<', 'link', 'rel=',
"''", 'stylesheet', "''", 'href=', "''", '/gutenberg/new_nav.css', '?', 'v=1.6',
"''", '>', '<', 'link', 'rel=', "''", 'stylesheet', "''", 'href=', "''",
'/gutenberg/pg-desktop-one.css', '?', 'v=1.1', "''", '>', '<', 'meta', 'name=',
"''", 'viewport', "''", 'content=', "''", 'width=device-width', ',', 'initial-
scale=1', "''", '>', '<', 'meta', 'name=', "''", 'keywords', "''", 'content=',
"''", 'books', ',', 'ebooks', ',', 'free', ',', 'kindle', ',', 'android', ',',
'iphone', ',', 'ipad', "''", '>', '<', 'meta', 'name=', "''", 'google-site-
verification', "'", 'content=', "'", 'wucOEvSnj5kP3Ts_36OfP64laakK-1mVTg-
ptrGC9io', "'", '>', '<', 'meta', 'name=', "'", 'alexaVerifyID', "'",
'content=', "'", '4WNaCljsE-A82vP_ih2H_UqXZvM', "'", '>', '<', 'link', 'rel=',
"'", 'copyright', "'", 'href=', "'", 'https', ':',
'//www.gnu.org/copyleft/fdl.html', "'", '>', '<', 'link', 'rel=', "'", 'icon',
"'", 'type=', "'", 'image/png', "'", 'href=', "'", '/gutenberg/favicon.ico',
"'", 'sizes=', "'", '16x16', "'", '>', '<', 'meta', 'property=', "'", 'og',
':', 'title', "'", 'content=', "'", 'Project', 'Gutenberg', "'", '>', '<',
'meta', 'property=']
```

[ ]: