

# Session 6 – Workshop – Sentiment Analysis – 50 pts

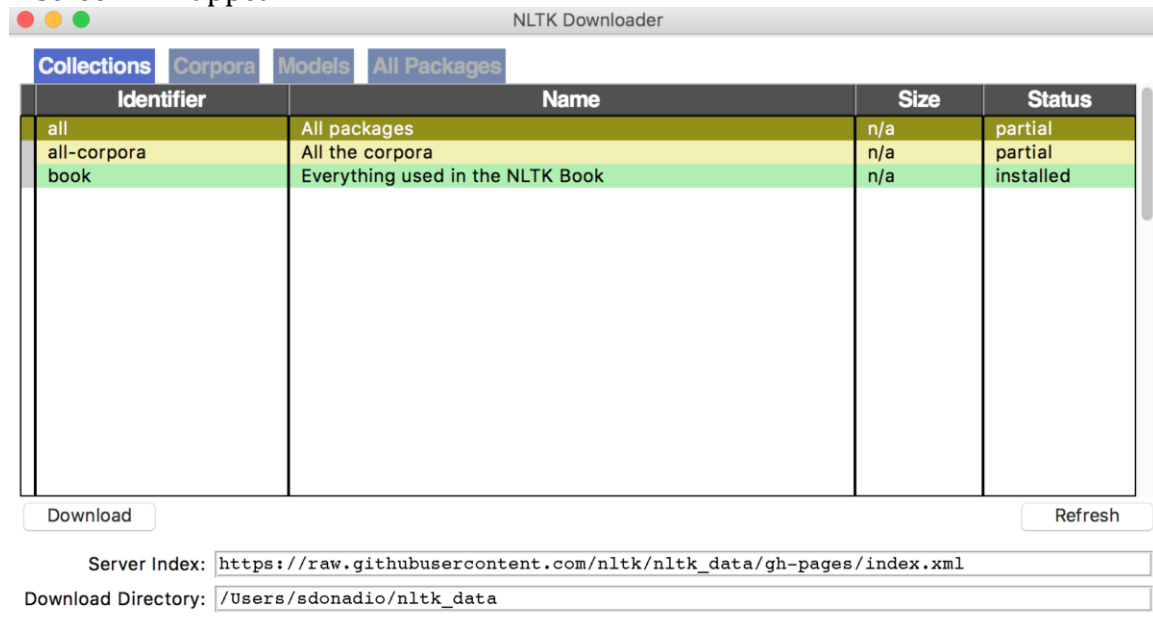
---

## Part I: Install NLTK library

Install the library nltk.  
Then run the following code:

```
In [*]: import nltk  
        nltk.download()
```

A screen will appear:



The screenshot shows the NLTK Downloader window. At the top, there are tabs for 'Collections', 'Corpora', 'Models', and 'All Packages'. Below the tabs is a table with the following data:

Identifier	Name	Size	Status
all	All packages	n/a	partial
all-corpora	All the corpora	n/a	partial
book	Everything used in the NLTK Book	n/a	installed

Below the table, there are two buttons: 'Download' and 'Refresh'. At the bottom, there are two input fields: 'Server Index: [https://raw.githubusercontent.com/nltk/nltk\\_data/gh-pages/index.xml](https://raw.githubusercontent.com/nltk/nltk_data/gh-pages/index.xml)' and 'Download Directory: /Users/sdonadio/nltk\_data'.

Choose to download "book" for all packages, and then click 'download.'

If you download "all" you will need a lot of space:  
This will give you all of the tokenizers, chunkers, other algorithms, and all of the corpora.

Reminder

Corpus - Body of text, singular.  
Example: A collection of data analytics paper.

Lexicon - Words and their meanings. Example: English dictionary.

Token: Words

## Part II: Tokenize by sentences/words

You will use the following functions:

```
from nltk.tokenize import sent_tokenize, word_tokenize

# Create a few sentences and store them into one variable:
a='The computing world today is in the middle of a
revolution: mobile clients and cloud computing have emerged
as the dominant paradigms driving programming and hardware
innovation today. The Fifth Edition of Computer Architecture
focuses on this dramatic shift, exploring the ways in which
software and technology in the cloud are accessed by cell
phones, tablets, laptops, and other mobile computing
devices. Each chapter includes two real-world examples, one
mobile and one datacenter, to illustrate this revolutionary
change.'
```

What do you obtain?

## Part III: Useless words or stop words

You will use the function `stopwords.words` with the argument 'english' to remove all the useless words.

Give the code to display all the useless words from the string you selected.  
They all need to be different (you need to use a built-in Python data structure).

Now you can remove all the useless words from your original work tokenized sentence.

## Part IV: Reduce words with different declination

Some words have numerous declinations but they mean the same.

Birds, bird  
Hammer, hammers, hammering

Using the code below you will simplify the code:

```
from nltk.stem import PorterStemmer
from nltk.tokenize import sent_tokenize, word_tokenize

ps = PorterStemmer()

words = {Tokenized words}

for w in words:
    print(ps.stem(w))
```

## Part V: Part of speech tagging

We are going to assign for each word a function in the sentence using the following command:

```
import nltk
from nltk.corpus import state_union
from nltk.tokenize import PunktSentenceTokenizer
tokenized =
PunktSentenceTokenizer(EXAMPLE_TEXT).tokenize(EXAMPLE_TEXT)
def process_content():
    try:
        for i in tokenized:
            words = nltk.word_tokenize(i)
            tagged = nltk.pos_tag(words)
            print(tagged)

    except Exception as e:
        print(str(e))
```

## Part VI: Regrouping words into meaningful chunk

```
def process_content():
    try:
        for i in tokenized:
            words = nltk.word_tokenize(i)
            tagged = nltk.pos_tag(words)
            chunkGram = r"""Chunk: {<RB.?*>*<VB.?*>*<NNP>+<NN>?}"""
            chunkParser = nltk.RegexpParser(chunkGram)
            chunked = chunkParser.parse(tagged)
            for subtree in chunked.subtrees():
                print(subtree)
            #chunked.draw()

    except Exception as e:
        print(str(e))
```

```
from nltk.tokenize import sent_tokenize, PunktSentenceTokenizer
from nltk.corpus import gutenberg
sample = gutenberg.raw("bible-kjv.txt")
tok = sent_tokenize(sample)
for x in range(5):
    print(tok[x])
```

[The King James Bible]

The Old Testament of the King James Bible

The First Book of Moses: Called Genesis

1:1 In the beginning God created the heaven and the earth.

1:2 And the earth was without form, and void; and darkness was upon the face of the deep.

And the Spirit of God moved upon the face of the waters.

1:3 And God said, Let there be light: and there was light.

1:4 And God saw the light, that it was good: and God divided the light from the darkness.

## Part VII: Wordnet

Pick a word XXX and YYY and a verb VVV

```
from nltk.corpus import wordnet
syns = wordnet.synsets(XXX)
print(syns[0].name())

print(syns[0].lemmas()[0].name())

syns = wordnet.synsets(YYY)
print(syns[0].lemmas()[0].name())

print(syns[0].lemmas()[0].name())

syns = wordnet.synsets(VVV)
print(syns[0].lemmas()[0].name())

print(syns[0].lemmas()[0].name())
```

```
synonyms = []
antonyms = []
for syn in wordnet.synsets(XXX):
    for l in syn.lemmas():
        synonyms.append(l.name())
        if l.antonyms():
            antonyms.append(l.antonyms()[0].name())

print(set(synonyms))
print(set(antonyms))
```

```
synonyms = []
antonyms = []

for syn in wordnet.synsets("bad"):
    for l in syn.lemmas():
        synonyms.append(l.name())
        if l.antonyms():
            antonyms.append(l.antonyms()[0].name())

print(set(synonyms))
print(set(antonyms))
```

## Part XIX: Text Classification

```
import nltk
import random
from nltk.corpus import movie_reviews

documents = [(list(movie_reviews.words(fileid)), category)
              for category in movie_reviews.categories()
              for fileid in movie_reviews.fileids(category)]

random.shuffle(documents)

print(documents[1])

all_words = []
for w in movie_reviews.words():
    all_words.append(w.lower())

all_words = nltk.FreqDist(all_words)
print(all_words.most_common(15))
print(all_words["stupid"])
```

## Tweeter Sentiment Analysis with NLTK

<https://pythonprogramming.net/twitter-sentiment-analysis-nltk-tutorial/>