Data Processing
– Natural Language Processing (NPL)

Jung PARK, PhD
Research Fellow in Data Science

Version 2018 Aug 05

Data Processing

- Data wrangling
  - Merging the unstructured data
  - Handling text data
  - Handling images
  - Handling audios

- Contents analysis
  - NPL (Natural Language Processing)

Words frequency using Natural Language Processing (NPL)

```
def plot_word_freq(url):
    """Takes a url (from Project Gutenberg) and plots a word frequency
    distribution"""
    # Make the request and check object type
    r = requests.get(url)
    # Extract HTML from Response object and print
    html = r.text
    # Create a BeautifulSoup object from the HTML
    soup = BeautifulSoup(html, "html5lib")
    # Get the text out of the soup and print it
    text = soup.get_text()
    # Create tokenizer
    tokenizer = RegexpTokenizer('\w+')
    # Create tokens
    tokens = tokenizer.tokenize(text)
    # Initialize new list
    words = []
    # Loop through list tokens and make lower case
    for word in tokens:
        words.append(word.lower())
    # Get English stopwords and print some of them
    sw = nltk.corpus.stopwords.words('english')
    # Initialize new list
    words_ns = []
    # Add to words_ns all words that are in words but not in sw
    for word in words:
        if word not in sw:
            words_ns.append(word)
    # Create freq dist and plot
    freqdist1 = nltk.FreqDist(words_ns)
    freqdist1.plot(25)
```

[Steps]

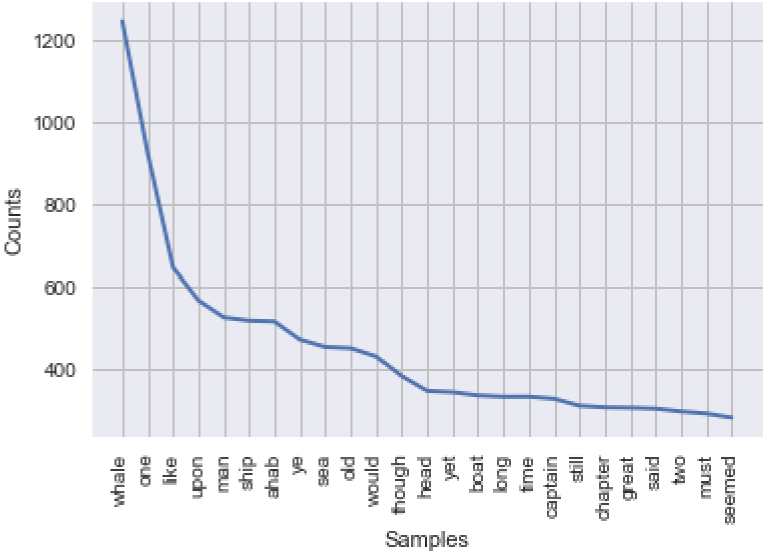
1. Get the data from web
2. Extract text from html
3. Tokenize the text
4. Lower the case
5. Remove stop words
6. Count the words frequency

Required Python Packages

```
import requests
from bs4 import BeautifulSoup

import re from nltk.tokenize
import RegexpTokenizer
import nltk

import matplotlib.pyplot as plt
import seaborn as sns
```



The most frequent words is “whale”

References

Word Frequency in Moby Dick
https://www.datacamp.com/projects/38

https://github.com/datacamp/datacamp_facebook_live_nlp/blob/master/NLP_FB_live_coding_soln_verbose.ipynb