## Lab 2 - Stemming and Part of Speech Tagging with NLTK

Today we will build upon what we learned last week concerning frequency distributions to utilize stemming and part of speech (POS) taggers.

Stemming is the process for reducing words to their word stem, base or root form. For example, the result of stemming the words “kicker,” “kicked” and “kicking” is “kick.”

The process of classifying words into their parts of speech and labeling them accordingly is known as part-of-speech tagging, POS-tagging, or simply tagging. Parts of speech are also known as word classes or lexical categories. The collection of tags used for a particular task is known as a tagset.

A part-of-speech tagger, or POS-tagger, processes a sequence of words, and attaches a part of speech tag to each word.

We’ll be tagging words in the input file as these types of nouns, among other tags:

NN: Noun, singular or mass

NNP: Proper noun, singular

NNS: Noun, plural

NNPS: Proper noun, plural

You can find a comprehensive list of POS abbreviations and their meanings here: (<https://cs.nyu.edu/grishman/jet/guide/PennPOS.html)>.

**Preliminary Steps**

#import modules

import nltk

from nltk.corpus import stopwords

from nltk import FreqDist

#read input data

f = open('input\_new.txt','r')

raw = f.read()

#replace change line operator ‘\n’ into space

raw = raw.replace('\n',' ')

#Tokenization

tokens = nltk.word\_tokenize(raw)

#Stopwords Removal and only keep text data then change to lowercase

mystopwords = stopwords.words('english')

words = [w.lower() for w in tokens if w.isalpha() if w.lower()not in mystopwords]

**Stemming**

Here we use 3 different stemmers, you can examine the result and decide which one to use in your project.

1. Porter Stemmer

#Use Porter Stemmer

porter = nltk.PorterStemmer()

stem1 = [porter.stem(w) for w in words]

#Encode with utf-8

stem1 = [w.encode('utf8') for w in stem1]

#Get the frequency distribution

freq1 = FreqDist(stem1)

#Sort the result

sorted\_freq1 = sorted(freq1.items(),key = lambda k: k[1], reverse = True)

1. Lancaster Stemmer

#Steps are the same with the Porter Stemmer

lancaster = nltk.LancasterStemmer()

stem2 = [lancaster.stem(w) for w in words]

stem2 = [w.encode('utf8') for w in stem2]

freq2 = FreqDist(stem2)

sorted\_freq2 = sorted(freq2.items(),key = lambda k: k[1], reverse = True)

1. WordNet Lemmatizer

#Steps are the same with the previous two stemmers

wnl = nltk.WordNetLemmatizer()

stem3 = [lancaster.stem(w) for w in words]

stem3 = [w.encode('utf8') for w in stem3]

freq3 = FreqDist(stem3)

sorted\_freq3 = sorted(freq3.items(),key = lambda k: k[1], reverse = True)

#write steming result into .txt file

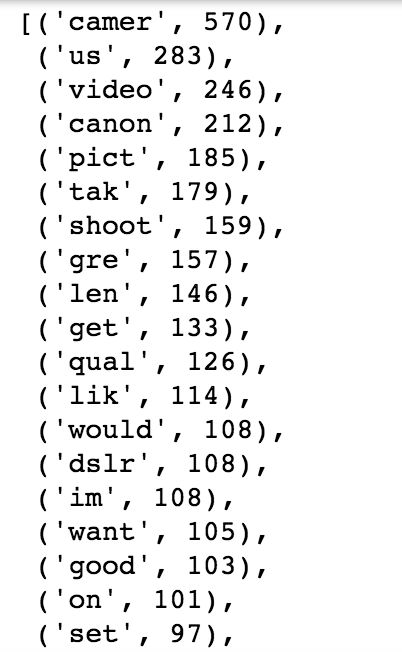
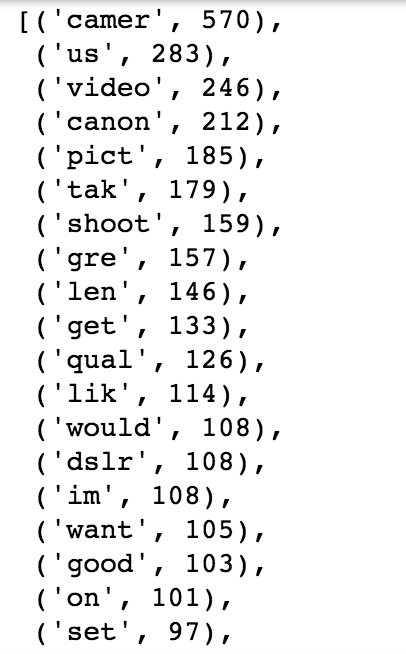
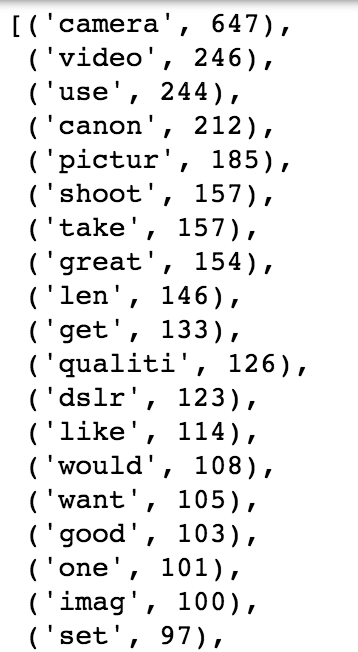
with open('stemming\_output.txt','w') as f:

for word, frequency in sorted\_freq1: #here you can change to sorted\_freq2 or 3

f.write(str(word)+'\t'+str(frequency)+'\n')

Output:

Poster Stemmer Lancaster Stemmer WordNet Lemmatizer



*Compare the three result, find the one that you want to use for your own project.*

**Part of Speech Tagging**

Here we want to extract 4 POS tags, NN, NNP, NNS, NNPS.

#use the NLTK built-in POS tag function

POS\_tags = nltk.pos\_tag(tokens) #use unprocessed 'tokens', not 'words'

#Generate a list of POS tags

POS\_tag\_list = [(word,tag) for (word,tag) in POS\_tags if tag.startswith('N')]

#Generate a frequency distribution of all the POS tags

tag\_freq = nltk.FreqDist(POS\_tag\_list)

#Sort the result

sorted\_tag\_freq = sorted(tag\_freq.items(), key = lambda k:k[1], reverse = True)

#write result into .txt file

with open('POS\_output.txt','w') as f:

for (word,tag),frequency in sorted\_tag\_freq

f.write(str(word)+'\t'+str(tag)+'\t'+str(frequency)+'\n')

Output:

