Chapter 6 - Data Sourcing via Web

Segment 5 - Introduction to NLP

```
import nltk
```

```
text = "On Wednesday, the Association for Computing Machinery, the world’s largest society of computing professionals, announced that
```

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

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
True
```

Sentence Tokenizer

```
from nltk.tokenize import sent_tokenize
sent_tk = sent_tokenize(text)
print("Sentence tokenizing the text: \n")
print(sent_tk)
```

```
Sentence tokenizing the text:

['On Wednesday, the Association for Computing Machinery, the world’s largest society of computing professionals, announced that H
```

Word Tokenizer

```
from nltk.tokenize import word_tokenize
word_tk = word_tokenize(text)
print("Word tokenizing the text: \n")
print(word_tk)
```

```
Word tokenizing the text:

['On', 'Wednesday', ',', 'the', 'Association', 'for', 'Computing', 'Machinery', ',', 'the', 'world', '’', 's', 'largest', 'societ
```

Removing stop words

```
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to /root/nltk_data...
[nltk_data]   Unzipping corpora/stopwords.zip.
True
```

```
from nltk.corpus import stopwords

sw = set(stopwords.words("english"))
print("Stop words in English language are: \n")
print(sw)
```

```
Stop words in English language are:

{'such', 'a', 'with', 'out', "haven't", 't', 'more', 'only', 'wasn', 's', 'themselves', 'as', 'can', 'once', 'i', 'above', 'very
```

```
filtered_words = [w for w in word_tk if not w in sw]

print("The text after removing stop words \n")
print(filtered_words)
```

```
The text after removing stop words

['On', 'Wednesday', ',', 'Association', 'Computing', 'Machinery', ',', 'world', '’', 'largest', 'society', 'computing', 'professi
```

Stemming

```
from nltk.stem import PorterStemmer
```

```
from nltk.tokenize import sent_tokenize, word_tokenize

port_stem = PorterStemmer()


stemmed_words = []

for w in filtered_words:
    stemmed_words.append(port_stem.stem(w))

print("Filtered Sentence: \n", filtered_words, "\n")
print("Stemmed Sentence: \n", stemmed_words)
```

```
Filtered Sentence:
 ['On', 'Wednesday', ',', 'Association', 'Computing', 'Machinery', ',', 'world', '’', 'largest', 'society', 'computing', 'profess

Stemmed Sentence:
 ['On', 'wednesday', ',', 'associ', 'comput', 'machineri', ',', 'world', '’', 'largest', 'societi', 'comput', 'profession', ',',
```

Lemmatizing

```
nltk.download('wordnet')
#nltk.download('wordnet')
```

```
[nltk_data] Downloading package wordnet to /root/nltk_data...
[nltk_data]   Unzipping corpora/wordnet.zip.
True
```

```
from nltk.stem.wordnet import WordNetLemmatizer

lem = WordNetLemmatizer()

from nltk.stem.porter import PorterStemmer
stem = PorterStemmer()

lemm_words = []

for i in filtered_words:
```

```
    lemm_words.append(lem.lemmatize(i))

print(lemm_words)
''' from nltk.stem.wordnet import WordNetLemmatizer
lem = WordNetLemmatizer()
l=[]
for i in fil:
  l.append(lem.lemmatize(i))'''
```

```
['On', 'Wednesday', ',', 'Association', 'Computing', 'Machinery', ',', 'world', '’', 'largest', 'society', 'computing', 'professi
' from nltk.stem.wordnet import WordNetLemmatizer\nlem = WordNetLemmatizer()\nl=[]\nfor i in fil:\n  l.append(lem.lemmatize(i))'
```

Parts of Speech Tagging

```
nltk.download('averaged_perceptron_tagger')
#nltk.download('averaged_perceptron_tagger')
```

```
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     /root/nltk_data...
[nltk_data]   Unzipping taggers/averaged_perceptron_tagger.zip.
True
```

```
from nltk import pos_tag
pos_tagged_words = pos_tag(word_tk)

print(pos_tagged_words)
''' from nltk import pos_tag
p=pos_tag(word_tk)'''
```

```
[('On', 'IN'), ('Wednesday', 'NNP'), (',', ','), ('the', 'DT'), ('Association', 'NNP'), ('for', 'IN'), ('Computing', 'VBG'), ('Ma
```

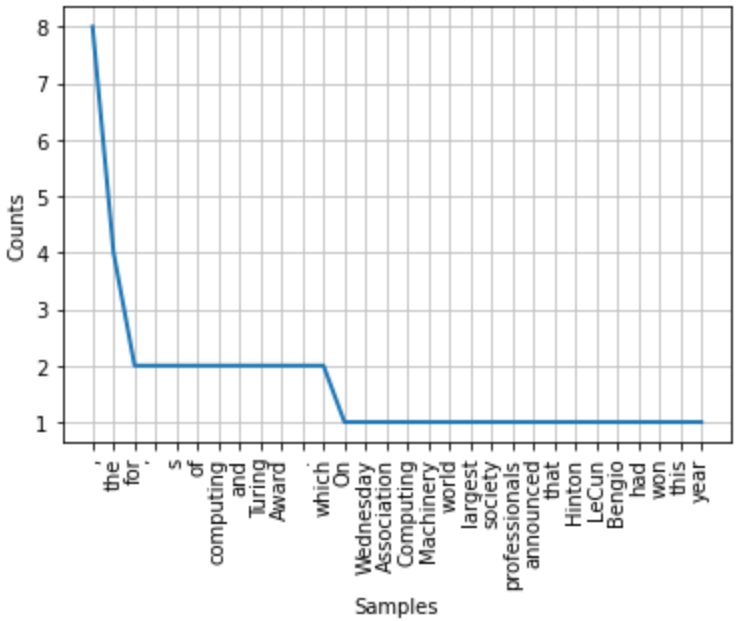
Frequency Distribution Plots

```
from nltk.probability import FreqDist
fd = FreqDist(word_tk)
print(fd)
```

```
''' from nltk.probability import FreqDist
f= FreqDist(word_tk)'''
```

```
<FreqDist with 56 samples and 76 outcomes>
```

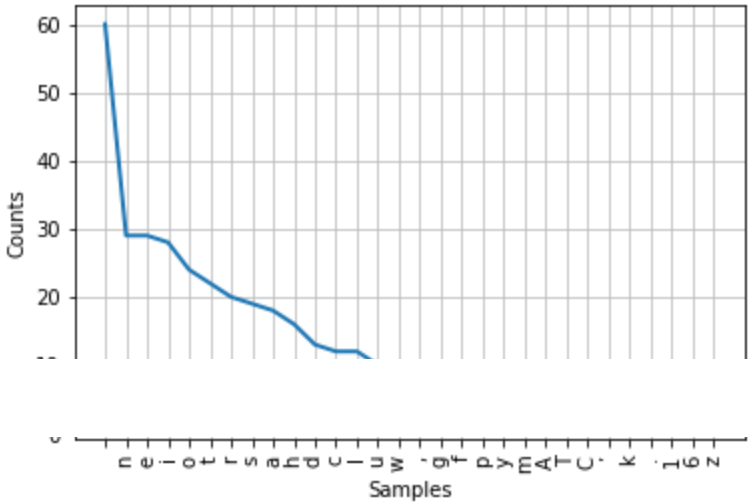
```
import matplotlib.pyplot as plt
fd.plot(30, cumulative=False)
plt.show()
''' import matplotlib.pyplot as plt
fd.plot(30, cumulative= False)
plt.show()'''
```




```
' import matplotlib.pyplot as plt\nfd.plot(30, cumulative= False)\nplt.show()'
```

```
fd_alpha = FreqDist(text)
print(fd_alpha)
fd_alpha.plot(30, cumulative=False)
'''f= FreqDist(text)
f.plot(30, cumulative= False)'''
```

```
<FreqDist with 41 samples and 387 outcomes>
```




```
'f= FreqDist(text)\nf.plot(30, cumulative= False)'
```