```
Chapter 6 - Data Sourcing via Web

Part 5 - Introduction to NLP
```

In [1]:
```
import nltk
```

In [2]:
```
n 1966, is often called the Nobel Prize of computing, and it includes a $1 millio
```

In [3]:
```
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\danal\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping tokenizers\punkt.zip.
```

Out[3]:
```
True
```

```
Sentence Tokenizer
```

In [5]:
```
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 so
ciety of computing professionals, announced that Hinton, LeCun and Bengio had w
on this year’s Turing Award for their work on neural networks.', 'The Turing Aw
ard, which was introduced in 1966, is often called the Nobel Prize of computin
g, and it includes a $1 million prize, which the three scientists will share.']
```

```
Word Tokenizer
```

In [6]:
```
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', 'society', 'of', 'computing', 'profes
sionals', ',', 'announced', 'that', 'Hinton', ',', 'LeCun', 'and', 'Bengio', 'h
ad', 'won', 'this', 'year', '’', 's', 'Turing', 'Award', 'for', 'their', 'wor
k', 'on', 'neural', 'networks', '.', 'The', 'Turing', 'Award', ',', 'which', 'w
as', 'introduced', 'in', '1966', ',', 'is', 'often', 'called', 'the', 'Nobel',
'Prize', 'of', 'computing', ',', 'and', 'it', 'includes', 'a', '$', '1', 'milli
on', 'prize', ',', 'which', 'the', 'three', 'scientists', 'will', 'share', '.']
```

```
Removing Stop Words
```

In [7]:
```
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\danal\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping corpora\stopwords.zip.
```

Out[7]: True

In [8]:
```
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:

{'it', 'didn', 'hasn', "haven't", 'the', 'are', 'an', 'most', "hasn't", "might
n't", 'not', 'theirs', 'and', 'which', 'shouldn', 've', 'his', 'were', 'wasn',
'during', 'for', 'now', 'is', 'your', 'needn', 'o', 'me', "it's", "isn't", 'wha
t', "wouldn't", 'been', 'those', 'same', 't', 'to', 'so', 'at', "should've", 'u
p', 'he', 'with', 'through', 'hadn', 'there', 'our', "she's", 'then', 'other',
'i', 'than', 'm', 'myself', 'herself', 'such', 'or', "you'll", 'until', 'in',
'this', 'himself', 'all', 'don', 'when', 'any', 'hers', 'very', 'below', 'ourse
lves', 'as', 'couldn', 'him', 'where', 'she', 'these', 's', 'wouldn', 'whom',
'down', 'we', 'why', 'you', 'while', 'over', 'y', 'their', "won't", 'some', 'yo
urs', "don't", 'nor', 'mustn', 'own', 'who', 'no', 'doesn', 'because', 'each',
"doesn't", 'of', "needn't", 'above', "you'd", 'between', 'before', 'll', 'on',
'had', 're', 'did', "hadn't", 'once', "you've", 'doing', 'isn', 'about', 'wil
l', 'having', "couldn't", 'can', 'just', 'being', 'both', "aren't", 'ma', 'your
self', 'from', 'off', 'after', 'if', 'am', 'but', 'her', 'into', 'only', "were
n't", 'ours', "that'll", 'here', 'does', 'more', 'how', "shouldn't", 'has', 'th
at', "mustn't", 'again', "wasn't", 'too', 'weren', 'was', 'do', 'them', 'themse
lves', 'be', 'have', 'won', 'under', 'aren', "you're", 'd', 'a', 'shan', 'the
y', 'yourselves', 'by', 'its', "didn't", 'out', 'itself', 'against', 'my', "sha
n't", 'mightn', 'ain', 'further', 'few', 'haven', 'should'}
```

In [10]:
```
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', 'professionals', ',', 'announced', 'Hin
ton', ',', 'LeCun', 'Bengio', 'year', '’', 'Turing', 'Award', 'work', 'neural',
'networks', '.', 'The', 'Turing', 'Award', ',', 'introduced', '1966', ',', 'oft
en', 'called', 'Nobel', 'Prize', 'computing', ',', 'includes', '$', '1', 'milli
on', 'prize', ',', 'three', 'scientists', 'share', '.']
```

```
Stemming
```

In [11]:
```
from nltk.stem import PorterStemmer
from nltk.tokenize import sent_tokenize, word_tokenize

port_stem = PorterStemmer()
```

In [12]:
```
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', ',', 'worl
d', '’', 'largest', 'society', 'computing', 'professionals', ',', 'announced',
'Hinton', ',', 'LeCun', 'Bengio', 'year', '’', 'Turing', 'Award', 'work', 'neur
al', 'networks', '.', 'The', 'Turing', 'Award', ',', 'introduced', '1966', ',',
'often', 'called', 'Nobel', 'Prize', 'computing', ',', 'includes', '$', '1', 'm
illion', 'prize', ',', 'three', 'scientists', 'share', '.']

Stemmed Sentence:
 ['On', 'wednesday', ',', 'associ', 'comput', 'machineri', ',', 'world', '’',
'largest', 'societi', 'comput', 'profession', ',', 'announc', 'hinton', ',', 'l
ecun', 'bengio', 'year', '’', 'ture', 'award', 'work', 'neural', 'network',
'.', 'the', 'ture', 'award', ',', 'introduc', '1966', ',', 'often', 'call', 'no
bel', 'prize', 'comput', ',', 'includ', '$', '1', 'million', 'prize', ',', 'thr
ee', 'scientist', 'share', '.']
```

Lemmatizing

In [13]:
```
nltk.download('wordnet')
```

```
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\danal\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping corpora\wordnet.zip.
```

Out[13]:
```
True
```

In [22]:
```
from nltk.stem.wordnet import WordNetLemmatizer

lem = WordNetLemmatizer()

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

lemm_words = []

for i in range(len(filtered_words)):
    lemm_words.append(lem.lemmatize(filtered_words[i]))

print(lemm_words)
```

```
['On', 'Wednesday', ',', 'Association', 'Computing', 'Machinery', ',', 'world',
'’', 'largest', 'society', 'computing', 'professional', ',', 'announced', 'Hint
on', ',', 'LeCun', 'Bengio', 'year', '’', 'Turing', 'Award', 'work', 'neural',
'network', '.', 'The', 'Turing', 'Award', ',', 'introduced', '1966', ',', 'ofte
n', 'called', 'Nobel', 'Prize', 'computing', ',', 'includes', '$', '1', 'millio
n', 'prize', ',', 'three', 'scientist', 'share', '.']
```

Parts of Speech Tagging

In [26]:
```
nltk.download('averaged_perceptron_tagger')
```

```
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\danal\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping taggers\averaged_perceptron_tagger.zip.
```

Out[26]:
```
True
```

In [27]:
```
from nltk import pos_tag
pos_tagged_words = pos_tag(word_tk)

print(pos_tagged_words)
```

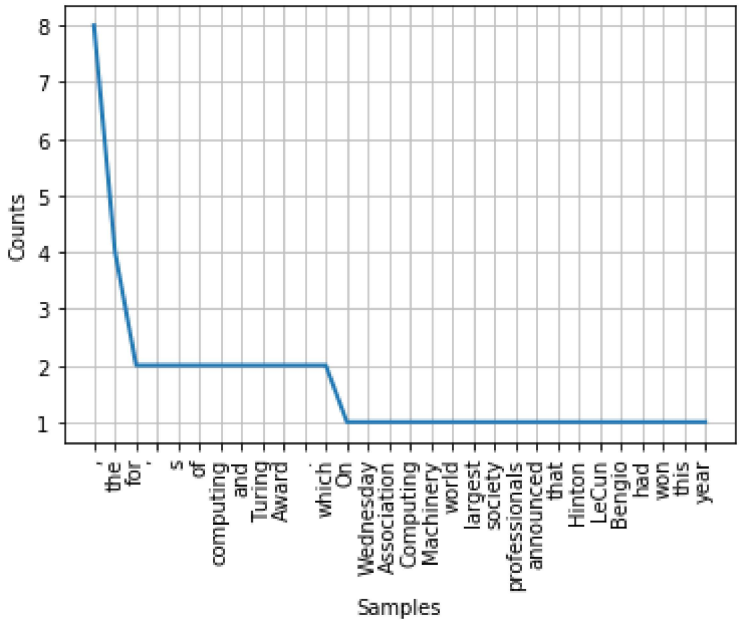
```
[('On', 'IN'), ('Wednesday', 'NNP'), (',', ','), ('the', 'DT'), ('Association',
'NNP'), ('for', 'IN'), ('Computing', 'VBG'), ('Machinery', 'NNP'), (',', ','),
('the', 'DT'), ('world', 'NN'), ('’', 'NNP'), ('s', 'RB'), ('largest', 'JJS'),
('society', 'NN'), ('of', 'IN'), ('computing', 'VBG'), ('professionals', 'NN
S'), (',', ','), ('announced', 'VBD'), ('that', 'IN'), ('Hinton', 'NNP'), (',',
','), ('LeCun', 'NNP'), ('and', 'CC'), ('Bengio', 'NNP'), ('had', 'VBD'), ('wo
n', 'VBN'), ('this', 'DT'), ('year', 'NN'), ('’', 'VBZ'), ('s', 'JJ'), ('Turin
g', 'NNP'), ('Award', 'NNP'), ('for', 'IN'), ('their', 'PRP$'), ('work', 'NN'),
('on', 'IN'), ('neural', 'JJ'), ('networks', 'NNS'), ('.', '.'), ('The', 'DT'),
('Turing', 'NNP'), ('Award', 'NNP'), (',', ','), ('which', 'WDT'), ('was', 'VB
D'), ('introduced', 'VBN'), ('in', 'IN'), ('1966', 'CD'), (',', ','), ('is', 'V
BZ'), ('often', 'RB'), ('called', 'VBN'), ('the', 'DT'), ('Nobel', 'NNP'), ('Pr
ize', 'NNP'), ('of', 'IN'), ('computing', 'NN'), (',', ','), ('and', 'CC'), ('i
t', 'PRP'), ('includes', 'VBZ'), ('a', 'DT'), ('$', '$'), ('1', 'CD'), ('millio
n', 'CD'), ('prize', 'NN'), (',', ','), ('which', 'WDT'), ('the', 'DT'), ('thre
e', 'CD'), ('scientists', 'NNS'), ('will', 'MD'), ('share', 'NN'), ('.', '.')]
```

Frequency Distribution Plots

In [29]:
```
from nltk.probability import FreqDist
fd = FreqDist(word_tk)
print(fd)
```

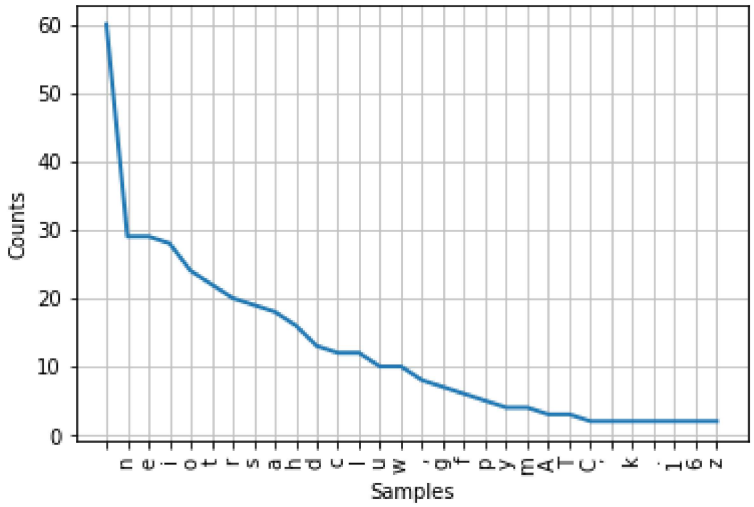
```
<FreqDist with 56 samples and 76 outcomes>
```

In [30]:
```
import matplotlib.pyplot as plt
fd.plot(30, cumulative=False)
plt.show()
```



In [31]:
```
fd_alpha = FreqDist(text)
print(fd_alpha)
fd_alpha.plot(30, cumulative=False)
```

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



Out[31]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x1ffc5cbdc88>
```