Nlp

In [1]:
```
# 1 - incude libraries
```

In [2]:
```
import nltk
nltk.download('punkt')
nltk.download('popular')
from nltk.tokenize import RegexpTokenizer
from nltk.corpus import stopwords
from nltk import FreqDist
from __future__ import division
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\DrRizk\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading collection 'popular'
[nltk_data]    |
[nltk_data]    | Downloading package cmudict to
[nltk_data]    |     C:\Users\DrRizk\AppData\Roaming\nltk_data...
[nltk_data]    |   Package cmudict is already up-to-date!
[nltk_data]    | Downloading package gazetteers to
[nltk_data]    |     C:\Users\DrRizk\AppData\Roaming\nltk_data...
[nltk_data]    |   Package gazetteers is already up-to-date!
[nltk_data]    | Downloading package genesis to
[nltk_data]    |     C:\Users\DrRizk\AppData\Roaming\nltk_data...
[nltk_data]    |   Package genesis is already up-to-date!
[nltk_data]    | Downloading package gutenberg to
[nltk_data]    |     C:\Users\DrRizk\AppData\Roaming\nltk_data...
[nltk_data]    |   Package gutenberg is already up-to-date!
[nltk_data]    | Downloading package inaugural to
[nltk_data]    |     C:\Users\DrRizk\AppData\Roaming\nltk_data...
[nltk_data]    |   Package inaugural is already up-to-date!
[nltk_data]    | Downloading package movie_reviews to
[nltk_data]    |     C:\Users\DrRizk\AppData\Roaming\nltk_data...
[nltk_data]    |   Package movie_reviews is already up-to-date!
[nltk_data]    | Downloading package names to
[nltk_data]    |     C:\Users\DrRizk\AppData\Roaming\nltk_data...
[nltk_data]    |   Package names is already up-to-date!
[nltk_data]    | Downloading package shakespeare to
[nltk_data]    |     C:\Users\DrRizk\AppData\Roaming\nltk_data...
[nltk_data]    |   Package shakespeare is already up-to-date!
[nltk_data]    | Downloading package stopwords to
[nltk_data]    |     C:\Users\DrRizk\AppData\Roaming\nltk_data...
[nltk_data]    |   Package stopwords is already up-to-date!
[nltk_data]    | Downloading package treebank to
[nltk_data]    |     C:\Users\DrRizk\AppData\Roaming\nltk_data...
[nltk_data]    |   Package treebank is already up-to-date!
[nltk_data]    | Downloading package twitter_samples to
```

In [3]:
```
#2-after cleaning the text read it
```

In [4]:
```
storytext = open("cleverfox.txt").read()
```

In [5]:
```
%pwd
```

Out[5]:
```
'C:\\Users\\DrRizk'
```

In [6]:
```
#The output is an object that consists of all the words found in the story as word “tokens“.
def preprocess(sentence):
    sentence = sentence.lower()
    tokenizer = RegexpTokenizer(r'\w+')
    tokens = tokenizer.tokenize(sentence)
    return " ".join(tokens)
preprocessedStory = preprocess(storytext)
tokens = nltk.word_tokenize(preprocessedStory)
print(tokens[0:20])
```

```
['this', 'short', 'story', 'the', 'clever', 'fox', 'is', 'quite', 'interesting', 'to', 'all', 'the', 'people', 'enjoy', 'readin
g', 'this', 'story', 'there', 'once', 'lived']
```

In [7]:
```
#function to identify an introductory metric for our story.
#The Lexical Diversity represents the ratio of unique words used to the total number of words in the story.
def lexical_diversity(text):
    return len(set(text)) / len(text)

lexical_diversity(tokens)
```

Out[7]:
```
0.4662756598240469
```

In [8]:
```
#nb of tokens
len(tokens)
```

Out[8]:
```
341
```

In [9]:
```
len(set(tokens))
```

Out[9]:
```
159
```

In [10]:
```
#The FreqDist() function turns our set of tokens into a Frequency Distribution object, giving us the frequencies of all tokens in
fdist1 = FreqDist(tokens)
print(fdist1)
```

```
<FreqDist with 159 samples and 341 outcomes>
```

In [11]:
```
#plot frequency distribution
fdist1.plot(50, cumulative=True)
```

```
<Figure size 640x480 with 1 Axes>
```

In [12]:
```
#not many of those words are useful for analysis. “The”, “and”, “to”, “a”, and “of” are all used in the English language to provi
from nltk.corpus import stopwords
stop = stopwords.words('english')
remstop = [i for i in tokens if i not in stop]
remstop[0:20]
```

Out[12]:
```
['short',
 'story',
 'clever',
 'fox',
 'quite',
 'interesting',
 'people',
 'enjoy',
 'reading',
 'story',
 'lived',
 'crow',
 'one',
 'day',
 'hungry',
 'able',
 'get',
 'food',
 'previous',
 'day']
```

In [13]:
```
#nb of word left
len(remstop)
```

Out[13]:
```
164
```

In [14]:
```
#how many unique words
len(set(remstop))
```

Out[14]:
```
103
```

In [15]:
```
#turn our stopword-free list of tokens into a Frequency Distribution object
fdist2 = FreqDist(remstop)
print(fdist2)
```

```
<FreqDist with 103 samples and 164 outcomes>
```

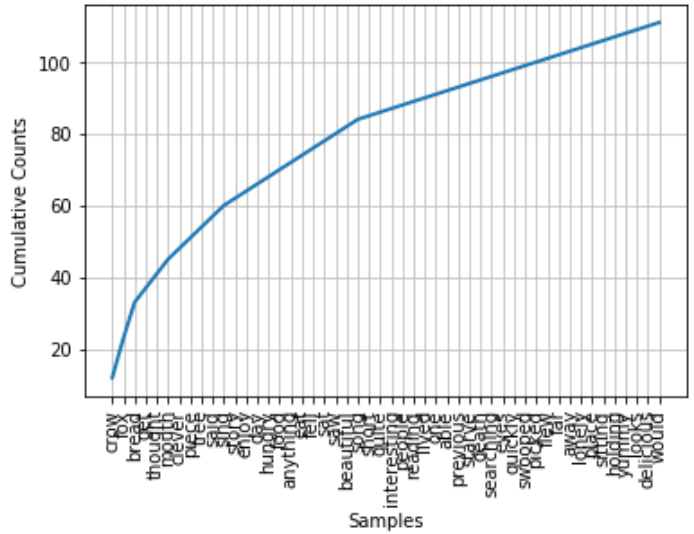
In [16]:
```
fdist2.items()
```

Out[16]:
```
dict_items([('short', 1), ('story', 2), ('clever', 3), ('fox', 11), ('quite', 1), ('interesting', 1), ('people', 1), ('enjoy',
2), ('reading', 1), ('lived', 1), ('crow', 12), ('one', 1), ('day', 2), ('hungry', 2), ('able', 1), ('get', 4), ('food', 2),
('previous', 1), ('anything', 2), ('eat', 2), ('starve', 1), ('death', 1), ('thought', 4), ('searching', 1), ('eyes', 1), ('fel
l', 2), ('piece', 3), ('bread', 10), ('quickly', 1), ('swooped', 1), ('picked', 1), ('flew', 1), ('far', 1), ('away', 1), ('lon
ely', 1), ('place', 1), ('sat', 2), ('tree', 3), ('saw', 2), ('sitting', 1), ('holding', 1), ('mouth', 4), ('yummy', 1), ('look
s', 1), ('delicious', 1), ('would', 1), ('give', 1), ('decided', 1), ('use', 1), ('cunning', 1), ('means', 1), ('guess', 1),
('wants', 1), ('shall', 1), ('hold', 1), ('carefully', 1), ('held', 1), ('even', 1), ('tightly', 1), ('spoke', 1), ('politely',
1), ('said', 3), ('hello', 1), ('friend', 1), ('say', 1), ('crows', 1), ('lovely', 1), ('birds', 1), ('charming', 1), ('flatter
ing', 1), ('heard', 1), ('besides', 1), ('beautiful', 2), ('also', 1), ('sweet', 1), ('voice', 1), ('please', 1), ('sing', 3),
('song', 2), ('started', 1), ('believe', 1), ('saying', 1), ('knows', 1), ('true', 1), ('beauty', 1), ('must', 1), ('bird', 1),
('whole', 1), ('world', 1), ('soon', 1), ('foolish', 1), ('opened', 1), ('beak', 1), ('ground', 1), ('waiting', 1), ('moment',
1), ('caught', 1), ('gulped', 1), ('throat', 1), ('paid', 1), ('heavy', 1), ('price', 1), ('foolishness', 1)])
```

In [17]:
```
# list of the top 20 most common non-stopwords in my short story.
fdist2.items()
```

Out[17]:
```
dict_items([('short', 1), ('story', 2), ('clever', 3), ('fox', 11), ('quite', 1), ('interesting', 1), ('people', 1), ('enjoy',
2), ('reading', 1), ('lived', 1), ('crow', 12), ('one', 1), ('day', 2), ('hungry', 2), ('able', 1), ('get', 4), ('food', 2),
('previous', 1), ('anything', 2), ('eat', 2), ('starve', 1), ('death', 1), ('thought', 4), ('searching', 1), ('eyes', 1), ('fel
l', 2), ('piece', 3), ('bread', 10), ('quickly', 1), ('swooped', 1), ('picked', 1), ('flew', 1), ('far', 1), ('away', 1), ('lon
ely', 1), ('place', 1), ('sat', 2), ('tree', 3), ('saw', 2), ('sitting', 1), ('holding', 1), ('mouth', 4), ('yummy', 1), ('look
s', 1), ('delicious', 1), ('would', 1), ('give', 1), ('decided', 1), ('use', 1), ('cunning', 1), ('means', 1), ('guess', 1),
('wants', 1), ('shall', 1), ('hold', 1), ('carefully', 1), ('held', 1), ('even', 1), ('tightly', 1), ('spoke', 1), ('politely',
1), ('said', 3), ('hello', 1), ('friend', 1), ('say', 1), ('crows', 1), ('lovely', 1), ('birds', 1), ('charming', 1), ('flatter
ing', 1), ('heard', 1), ('besides', 1), ('beautiful', 2), ('also', 1), ('sweet', 1), ('voice', 1), ('please', 1), ('sing', 3),
('song', 2), ('started', 1), ('believe', 1), ('saying', 1), ('knows', 1), ('true', 1), ('beauty', 1), ('must', 1), ('bird', 1),
```

In [18]:
```
#plot this frequency distribution:
fdist2.plot(50, cumulative=True)
```



In [19]:
```
# measure of how rich our vocabulary was with the stopwords removed.
lexical_diversity(remstop)
```

Out[19]:
```
0.6280487804878049
```

In [20]:
```
#turn that data into a list, which we'll then use to write out a .csv file.
array = fdist2.items()
mylist = [list(i) for i in array]
print(mylist)
```

```
[['short', 1], ['story', 2], ['clever', 3], ['fox', 11], ['quite', 1], ['interesting', 1], ['people', 1], ['enjoy', 2], ['readi
ng', 1], ['lived', 1], ['crow', 12], ['one', 1], ['day', 2], ['hungry', 2], ['able', 1], ['get', 4], ['food', 2], ['previous',
1], ['anything', 2], ['eat', 2], ['starve', 1], ['death', 1], ['thought', 4], ['searching', 1], ['eyes', 1], ['fell', 2], ['pie
ce', 3], ['bread', 10], ['quickly', 1], ['swooped', 1], ['picked', 1], ['flew', 1], ['far', 1], ['away', 1], ['lonely', 1], ['p
lace', 1], ['sat', 2], ['tree', 3], ['saw', 2], ['sitting', 1], ['holding', 1], ['mouth', 4], ['yummy', 1], ['looks', 1], ['del
icious', 1], ['would', 1], ['give', 1], ['decided', 1], ['use', 1], ['cunning', 1], ['means', 1], ['guess', 1], ['wants', 1],
['shall', 1], ['hold', 1], ['carefully', 1], ['held', 1], ['even', 1], ['tightly', 1], ['spoke', 1], ['politely', 1], ['said',
3], ['hello', 1], ['friend', 1], ['say', 1], ['crows', 1], ['lovely', 1], ['birds', 1], ['charming', 1], ['flattering', 1], ['h
eard', 1], ['besides', 1], ['beautiful', 2], ['also', 1], ['sweet', 1], ['voice', 1], ['please', 1], ['sing', 3], ['song', 2],
['started', 1], ['believe', 1], ['saying', 1], ['knows', 1], ['true', 1], ['beauty', 1], ['must', 1], ['bird', 1], ['whole',
1], ['world', 1], ['soon', 1], ['foolish', 1], ['opened', 1], ['beak', 1], ['ground', 1], ['waiting', 1], ['moment', 1], ['caug
ht', 1], ['gulped', 1], ['throat', 1], ['paid', 1], ['heavy', 1], ['price', 1], ['foolishness', 1]]
```

In [21]:
```
print(type(mylist))
```

```
<class 'list'>
```

In [22]:
```
import csv
with open('newfilefox.csv', 'w') as result:
    writer = csv.writer(result, dialect='excel')
    writer.writerows(array)

with open('tokensfox.csv', 'w') as result:
    writer = csv.writer(result, dialect='excel')
    writer.writerow(remstop)
```