In [1]:
```
import re, pprint, string
import pandas as pd
from nltk import word_tokenize, sent_tokenize
import nltk
from nltk.corpus import stopwords
nltk.download('stopwords')
nltk.download('punkt')
from nltk.util import ngrams
from nltk.corpus import stopwords
stop_words = set(stopwords.words('english'))
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Admin\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\Admin\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
```

In [2]:
```
string.punctuation = string.punctuation +'“'+'”'+'-'+'’'+'‘'+'—'
string.punctuation = string.punctuation.replace('.', '')
```

In [6]:
```
corpus = pd.read_csv('C:/Users/Admin/Downloads/corpus.csv')
print(corpus)
file_nl_removed = ""
for line in corpus:
  line_nl_removed = line.replace("\n", " ")      #removes newlines
  file_nl_removed += line_nl_removed
file_p = "".join([char for char in file_nl_removed if char not in string.punctuation])   #removes all

print(file_p)
```

```
                                                text        label  \
0   Stuning even for the non-gamer: This sound tra...  __label__2
1   The best soundtrack ever to anything.: I'm rea...   __label__2
2   " Amazing!: This soundtrack is my favorite mus...   __label__2
3   Excellent Soundtrack: I truly like this soundt...   __label__2
4   " Excellent Soundtrack: I truly like this soun...   __label__2
5   " Remember, Pull Your Jaw Off The Floor After ...   __label__2
6   " an absolute masterpiece: I am quite sure any...   __label__1
7   " Buyer beware: This is a self-published book,...   __label__1
8   " Glorious story: I loved Whisper of the wicke...   __label__2
9   " A FIVE STAR BOOK: I just finished reading Wh...   __label__2
10  " Disappointed Romanian!: This book in my opin...   __label__2
11  " Not the best: I bought both this and lonely ...   __label__2
12   Good but received defective book: I bought th...   __label__2
13  " From The Label:: From Memphis, TN. comes The...   __label__1
14  " Either 1 or 5 Stars. Depends on how you look...   __label__1

    Unnamed: 2  Unnamed: 3  Unnamed: 4  Unnamed: 5  Unnamed: 6  Unnamed: 7  \
0          NaN         NaN         NaN         NaN         NaN         NaN
1          NaN         NaN         NaN         NaN         NaN         NaN
2          NaN         NaN         NaN         NaN         NaN         NaN
3          NaN         NaN         NaN         NaN         NaN         NaN
4          NaN         NaN         NaN         NaN         NaN         NaN
5          NaN         NaN         NaN         NaN         NaN         NaN
6          NaN         NaN         NaN         NaN         NaN         NaN
7          NaN         NaN         NaN         NaN         NaN         NaN
8          NaN         NaN         NaN         NaN         NaN         NaN
9          NaN         NaN         NaN         NaN         NaN         NaN
10         NaN         NaN         NaN         NaN         NaN         NaN
11         NaN         NaN         NaN         NaN         NaN         NaN
12         NaN         NaN         NaN         NaN         NaN         NaN
13         NaN         NaN         NaN         NaN         NaN         NaN
14         NaN         NaN         NaN         NaN         NaN         NaN

    Unnamed: 8  Unnamed: 9  ...  Unnamed: 12  Unnamed: 13  Unnamed: 14  \
0          NaN         NaN  ...          NaN          NaN          NaN
1          NaN         NaN  ...          NaN          NaN          NaN
2          NaN         NaN  ...          NaN          NaN          NaN
3          NaN         NaN  ...          NaN          NaN          NaN
4          NaN         NaN  ...          NaN          NaN          NaN
5          NaN         NaN  ...          NaN          NaN          NaN
6          NaN         NaN  ...          NaN          NaN          NaN
7          NaN         NaN  ...          NaN          NaN          NaN
8          NaN         NaN  ...          NaN          NaN          NaN
9          NaN         NaN  ...          NaN          NaN          NaN
10         NaN         NaN  ...          NaN          NaN          NaN
11         NaN         NaN  ...          NaN          NaN          NaN
12         NaN         NaN  ...          NaN          NaN          NaN
13         NaN         NaN  ...          NaN          NaN          NaN
14         NaN         NaN  ...          NaN          NaN          NaN

    Unnamed: 15  Unnamed: 16  Unnamed: 17  Unnamed: 18  Unnamed: 19  \
0           NaN          NaN          NaN          NaN          NaN
1           NaN          NaN          NaN          NaN          NaN
2           NaN          NaN          NaN          NaN          NaN
3           NaN          NaN          NaN          NaN          NaN
4           NaN          NaN          NaN          NaN          NaN
5           NaN          NaN          NaN          NaN          NaN
6           NaN          NaN          NaN          NaN          NaN
7           NaN          NaN          NaN          NaN          NaN
8           NaN          NaN          NaN          NaN          NaN
9           NaN          NaN          NaN          NaN          NaN
10          NaN          NaN          NaN          NaN          NaN
11          NaN          NaN          NaN          NaN          NaN
12          NaN          NaN          NaN          NaN          NaN
13          NaN          NaN          NaN          NaN          NaN
```

```
14          NaN          NaN          NaN          NaN          NaN

    Unnamed: 20  Unnamed: 21
0           NaN          NaN
1           NaN          NaN
2           NaN          NaN
3           NaN          NaN
4           NaN          NaN
5           NaN          NaN
6           NaN          NaN
7           NaN          NaN
8           NaN          NaN
9           NaN          NaN
10          NaN          NaN
11          NaN          NaN
12          NaN          NaN
13          NaN          NaN
14          NaN          NaN

[15 rows x 22 columns]
textlabelUnnamed 2Unnamed 3Unnamed 4Unnamed 5Unnamed 6Unnamed 7Unnamed 8Unnamed 9Unnamed 10Unname
d 11Unnamed 12Unnamed 13Unnamed 14Unnamed 15Unnamed 16Unnamed 17Unnamed 18Unnamed 19Unnamed 20Unn
amed 21
```

In [7]:
```
sents = nltk.sent_tokenize(file_p)
print(sents)
print("The number of sentences is", len(sents))
#prints the number of sentences

words = nltk.word_tokenize(file_p)
print(words)
print("The number of tokens is", len(words))
#prints the number of tokens

average_tokens = round(len(words)/len(sents))
print("The average number of tokens per sentence is",average_tokens)
#prints the average number of tokens per sentence

unique_tokens = set(words)
print("The number of unique tokens are", len(unique_tokens))
#prints the number of unique tokens
```

```
['textlabelUnnamed 2Unnamed 3Unnamed 4Unnamed 5Unnamed 6Unnamed 7Unnamed 8Unnamed 9Unnamed 10Unnamed
11Unnamed 12Unnamed 13Unnamed 14Unnamed 15Unnamed 16Unnamed 17Unnamed 18Unnamed 19Unnamed 20Unnamed
21']
The number of sentences is 1
['textlabelUnnamed', '2Unnamed', '3Unnamed', '4Unnamed', '5Unnamed', '6Unnamed', '7Unnamed', '8Unnam
ed', '9Unnamed', '10Unnamed', '11Unnamed', '12Unnamed', '13Unnamed', '14Unnamed', '15Unnamed', '16Un
named', '17Unnamed', '18Unnamed', '19Unnamed', '20Unnamed', '21']
The number of tokens is 21
The average number of tokens per sentence is 21
The number of unique tokens are 21
```

In [8]:
```
unigram=[]
bigram=[]
trigram=[]
fourgram=[]
tokenized_text = []
for sentence in sents:
    sentence = sentence.lower()
    sequence = word_tokenize(sentence)
    for word in sequence:
        if (word =='.'):
            sequence.remove(word)
        else:
            unigram.append(word)
    tokenized_text.append(sequence)
    bigram.extend(list(ngrams(sequence, 2)))
#unigram, bigram, trigram, and fourgram models are created
    trigram.extend(list(ngrams(sequence, 3)))
    fourgram.extend(list(ngrams(sequence, 4)))
```

In [9]:
```
def removal(x):
#removes ngrams containing only stopwords
    y = []
    for pair in x:
        count = 0
        for word in pair:
            if word in stop_words:
                count = count or 0
            else:
                count = count or 1
        if (count==1):
            y.append(pair)
    return(y)
```

In [10]:
```
bigram = removal(bigram)
trigram = removal(trigram)
fourgram = removal(fourgram)
freq_bi = nltk.FreqDist(bigram)
freq_tri = nltk.FreqDist(trigram)
freq_four = nltk.FreqDist(fourgram)
print("Most common n-grams without stopword removal and without add-1 smoothing: \n")
print ("Most common bigrams: ", freq_bi.most_common(5))
print ("\nMost common trigrams: ", freq_tri.most_common(5))
print ("\nMost common fourgrams: ", freq_four.most_common(5))
```

```
Most common n-grams without stopword removal and without add-1 smoothing:

Most common bigrams:  [(('textlabelunnamed', '2unnamed'), 1), (('2unnamed', '3unnamed'), 1), (('3unn
amed', '4unnamed'), 1), (('4unnamed', '5unnamed'), 1), (('5unnamed', '6unnamed'), 1)]

Most common trigrams:  [(('textlabelunnamed', '2unnamed', '3unnamed'), 1), (('2unnamed', '3unnamed',
'4unnamed'), 1), (('3unnamed', '4unnamed', '5unnamed'), 1), (('4unnamed', '5unnamed', '6unnamed'),
1), (('5unnamed', '6unnamed', '7unnamed'), 1)]

Most common fourgrams:  [(('textlabelunnamed', '2unnamed', '3unnamed', '4unnamed'), 1), (('2unname
d', '3unnamed', '4unnamed', '5unnamed'), 1), (('3unnamed', '4unnamed', '5unnamed', '6unnamed'), 1),
(('4unnamed', '5unnamed', '6unnamed', '7unnamed'), 1), (('5unnamed', '6unnamed', '7unnamed', '8unnam
ed'), 1)]
```