In [24]:

```
import nltk

porter=nltk.PorterStemmer()
lancaster=nltk.LancasterStemmer()

file0=nltk.corpus.gutenberg.fileids()[0]
emmatext=nltk.corpus.gutenberg.raw(file0)
emmatokens=nltk.wordpunct_tokenize(emmatext)
emmawords=[w.lower() for w in emmatokens]
```

In [14]:

```
emmaregstem = [porter.stem(t) for t in emmatokens]
emmaregstem[1:100]
```

```
Out[14]: ['emma',
          'by',
          'jane',
          'austen',
          '1816',
          ']',
          'volum',
          'i',
          'chapter',
          'i',
          'emma',
          'woodhous',
          ',',
          'handsom',
          ',',
          'clever',
          ',',
          'and',
          'rich',
          ',',
          'with',
          'a',
          'comfort',
          'home',
          'and',
          'happi',
          'disposit',
          ',',
          'seem',
          'to',
          'unit',
          'some',
          'of',
          'the',
          'best',
          'bless',
          'of',
          'exist',
          ';',
          'and',
          'had',
          'live',
          'nearli',
          'twenti',
          '-',
          'one',
          'year',
          'in',
          'the',
          'world',
          'with',
          'veri',
          'littl',
          'to',
          'distress',
          'or',
          'vex',
          'her',
          '.',
          'she',
 'wa',
 'the',
 'youngest',
 'of',
 'the',
 'two',
 'daughter',
 'of',
 'a',
 'most',
 'affection',
 ',',
 'indulg',
 'father',
 ';',
 'and',
 'had',
 ',',
 'in',
 'consequ',
 'of',
 'her',
 'sister',
 "'",
 's',
 'marriag',
 ',',
 'been',
 'mistress',
 'of',
 'hi',
 'hous',
 'from',
 'a',
 'veri',
 'earli',
 'period',
 '.',
 'her']
```

In [18]:

```
emmalowerstem=[porter.stem(t) for t in emmawords]
emmalowerstem[1:100]
```

```
Out[18]: ['emma',
          'by',
          'jane',
          'austen',
          '1816',
          ']',
          'volum',
          'i',
          'chapter',
          'i',
          'emma',
          'woodhous',
          ',',
          'handsom',
          ',',
          'clever',
          ',',
          'and',
          'rich',
          ',',
          'with',
          'a',
          'comfort',
          'home',
          'and',
          'happi',
          'disposit',
          ',',
          'seem',
          'to',
          'unit',
          'some',
          'of',
          'the',
          'best',
          'bless',
          'of',
          'exist',
          ';',
          'and',
          'had',
          'live',
          'nearli',
          'twenti',
          '-',
          'one',
          'year',
          'in',
          'the',
          'world',
          'with',
          'veri',
          'littl',
          'to',
          'distress',
          'or',
          'vex',
          'her',
          '.',
          'she',
 'wa',
 'the',
 'youngest',
 'of',
 'the',
 'two',
 'daughter',
 'of',
 'a',
 'most',
 'affection',
 ',',
 'indulg',
 'father',
 ';',
 'and',
 'had',
 ',',
 'in',
 'consequ',
 'of',
 'her',
 'sister',
 "'",
 's',
 'marriag',
 ',',
 'been',
 'mistress',
 'of',
 'hi',
 'hous',
 'from',
 'a',
 'veri',
 'earli',
 'period',
 '.',
 'her']
```

In [25]:
```
emmaregstem = [lancaster.stem(t) for t in emmatokens]
emmaregstem[1:100]
```

```
Out[25]: ['emm',
          'by',
          'jan',
          'aust',
          '1816',
          ']',
          'volum',
          'i',
          'chapt',
          'i',
          'emm',
          'woodh',
          ',',
          'handsom',
          ',',
          'clev',
          ',',
          'and',
          'rich',
          ',',
          'with',
          'a',
          'comfort',
          'hom',
          'and',
          'happy',
          'disposit',
          ',',
          'seem',
          'to',
          'unit',
          'som',
          'of',
          'the',
          'best',
          'bless',
          'of',
          'ex',
          ';',
          'and',
          'had',
          'liv',
          'near',
          'twenty',
          '-',
          'on',
          'year',
          'in',
          'the',
          'world',
          'with',
          'very',
          'littl',
          'to',
          'distress',
          'or',
          'vex',
          'her',
          '.',
          'she',
 'was',
 'the',
 'youngest',
 'of',
 'the',
 'two',
 'daught',
 'of',
 'a',
 'most',
 'affect',
 ',',
 'indulg',
 'fath',
 ';',
 'and',
 'had',
 ',',
 'in',
 'consequ',
 'of',
 'her',
 'sist',
 "'",
 's',
 'marry',
 ',',
 'been',
 'mistress',
 'of',
 'his',
 'hous',
 'from',
 'a',
 'very',
 'ear',
 'period',
 '.',
 'her']
```

```
In [26]: emmalowerstem=[lancaster.stem(t) for t in emmawords]
         emmalowerstem[1:100]
```

```
Out[26]: ['emm',
          'by',
          'jan',
          'aust',
          '1816',
          ']',
          'volum',
          'i',
          'chapt',
          'i',
          'emm',
          'woodh',
          ',',
          'handsom',
          ',',
          'clev',
          ',',
          'and',
          'rich',
          ',',
          'with',
          'a',
          'comfort',
          'hom',
          'and',
          'happy',
          'disposit',
          ',',
          'seem',
          'to',
          'unit',
          'som',
          'of',
          'the',
          'best',
          'bless',
          'of',
          'ex',
          ';',
          'and',
          'had',
          'liv',
          'near',
          'twenty',
          '-',
          'on',
          'year',
          'in',
          'the',
          'world',
          'with',
          'very',
          'littl',
          'to',
          'distress',
          'or',
          'vex',
          'her',
          '.',
          'she',
 'was',
 'the',
 'youngest',
 'of',
 'the',
 'two',
 'daught',
 'of',
 'a',
 'most',
 'affect',
 ',',
 'indulg',
 'fath',
 ';',
 'and',
 'had',
 ',',
 'in',
 'consequ',
 'of',
 'her',
 'sist',
 "'",
 's',
 'marry',
 ',',
 'been',
 'mistress',
 'of',
 'his',
 'hous',
 'from',
 'a',
 'very',
 'ear',
 'period',
 '.',
 'her']
```

In [28]:

```
def stem(word):
    for suffix in ['ing','ly','ed','ious','ies','ive','es','s']:
        if word.endswith(suffix):
            return word[:-len(suffix)]
    return word


stemmedword=stem('friends')
stemmedword
```

Out[28]:

```
'friend'
```

In [29]:

```
wnl=nltk.WordNetLemmatizer()
emmalemma=[wnl.lemmatize(t) for t in emmawords]
emmalemma[1:100]
```

```
Out[29]: ['emma',
          'by',
          'jane',
          'austen',
          '1816',
          ']',
          'volume',
          'i',
          'chapter',
          'i',
          'emma',
          'woodhouse',
          ',',
          'handsome',
          ',',
          'clever',
          ',',
          'and',
          'rich',
          ',',
          'with',
          'a',
          'comfortable',
          'home',
          'and',
          'happy',
          'disposition',
          ',',
          'seemed',
          'to',
          'unite',
          'some',
          'of',
          'the',
          'best',
          'blessing',
          'of',
          'existence',
          ';',
          'and',
          'had',
          'lived',
          'nearly',
          'twenty',
          '-',
          'one',
          'year',
          'in',
          'the',
          'world',
          'with',
          'very',
          'little',
          'to',
          'distress',
          'or',
          'vex',
          'her',
          '.',
          'she',
 'wa',
 'the',
 'youngest',
 'of',
 'the',
 'two',
 'daughter',
 'of',
 'a',
 'most',
 'affectionate',
 ',',
 'indulgent',
 'father',
 ';',
 'and',
 'had',
 ',',
 'in',
 'consequence',
 'of',
 'her',
 'sister',
 "'",
 's',
 'marriage',
 ',',
 'been',
 'mistress',
 'of',
 'his',
 'house',
 'from',
 'a',
 'very',
 'early',
 'period',
 '.',
 'her']
```

In [40]:

```
type(emmatext)
len(emmatext)
shorttext=emmatext[:150]
```

In [33]:

```
for char in shorttext[:10]:
    print(char)
```

```
[
E
m
m
a

b
y

J
```

In [41]:

```
newemmatext=emmatext.replace('\n',' ')
shorttext=newemmatext[:150]
```

```
shorttext
```

Out[41]:

```
'[Emma by Jane Austen 1816]  VOLUME I  CHAPTER I   Emma Woodhouse, handsome, clever,
and rich, with a comfortable home and happy disposition, seemed to'
```

In [45]:

```
import re
pword=re.compile('\w+')
re.findall(pword,shorttext)
```

Out[45]:

```
['Emma',
 'by',
 'Jane',
 'Austen',
 '1816',
 'VOLUME',
 'I',
 'CHAPTER',
 'I',
 'Emma',
 'Woodhouse',
 'handsome',
 'clever',
 'and',
 'rich',
 'with',
 'a',
 'comfortable',
 'home',
 'and',
 'happy',
 'disposition',
 'seemed',
 'to']
```

In [46]:

```
specialtext = 'U.S.A. poster-print costs $12.40, with 10% off.'
re.findall(pword, specialtext)
```

Out[46]:

```
['U', 'S', 'A', 'poster', 'print', 'costs', '12', '40', 'with', '10', 'off']
```

In [48]:

```
ptoken=re.compile('(\w+(-\w+)*)')
re.findall(ptoken,specialtext)
```

Out[48]:

```
[('U', ''),
 ('S', ''),
 ('A', ''),
 ('poster-print', '-print'),
 ('costs', ''),
 ('12', ''),
 ('40', ''),
 ('with', ''),
 ('10', ''),
 ('off', '')]
```

In [49]:

```
pabbrev=re.compile('(([A-Z]\.)+)')
re.findall(pabbrev,specialtext)
```

Out[49]:

```
[('U.S.A.', 'A.')]
```

In [51]:

```
ptoken=re.compile('(\w+(-\w+)*|([A-Z]\.)+)')
re.findall(ptoken,specialtext)
```

Out[51]:
```
[('U', '', ''),
 ('S', '', ''),
 ('A', '', ''),
 ('poster-print', '-print', ''),
 ('costs', '', ''),
 ('12', '', ''),
 ('40', '', ''),
 ('with', '', ''),
 ('10', '', ''),
 ('off', '', '')]
```

In [52]:
```
ptoken = re.compile('(([A-Z]\.)+|\w+(-\w+)*)')
re.findall(ptoken,specialtext)
```

Out[52]:
```
[('U.S.A.', 'A.', ''),
 ('poster-print', '', '-print'),
 ('costs', '', ''),
 ('12', '', ''),
 ('40', '', ''),
 ('with', '', ''),
 ('10', '', ''),
 ('off', '', '')]
```

In [53]:
```
ptoken = re.compile(r'(([A-Z]\.)+|\w+(-\w+)*|\$?\d+(\.\d+)?)')
re.findall(ptoken,specialtext)
```

Out[53]:
```
[('U.S.A.', 'A.', '', ''),
 ('poster-print', '', '-print', ''),
 ('costs', '', '', ''),
 ('$12.40', '', '', '.40'),
 ('with', '', '', ''),
 ('10', '', '', ''),
 ('off', '', '', '')]
```

In [54]:
```
ptoken = re.compile(r'''([A-Z]\.)+
| \w+(-\w+)*
| \$?\d+(\.\d+)?
''', re.X)
```

In [59]:
```
pattern =r''' (?x)
    ([A-Z]\.)+
    | \w+(-\w+)*
    | \$?\d+(\.\d+)?%?
    | \.\.\.
    | [][.,;"'?():-_']
    | '''
```

In [60]:
```
nltk.regexp_tokenize(shorttext,pattern)
```

```
Out[60]: [('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', ''),
          ('', '', '')]
```

In [61]:

```
nltk.regexp_tokenize(specialtext,pattern)
```

Out[61]:

```
[('A.', '', ''),
 ('', '', ''),
 ('', '-print', ''),
 ('', '', ''),
 ('', '', ''),
 ('', '', ''),
 ('', '', '.40'),
 ('', '', ''),
 ('', '', ''),
 ('', '', ''),
 ('', '', ''),
 ('', '', ''),
 ('', '', ''),
 ('', '', ''),
 ('', '', ''),
 ('', '', ''),
 ('', '', '')]
```

In [63]:

```
tweetPattern = r''' (?x)
| (:-\)|;-\))
| &(amp|lt|gt|quot);
| \d+:\d+
| (\d+,)+?\d{3}(?=([^,]|$))
| ([A-Z]\.)+
| (--+)
| \w+(-\w+)*
| ['\".?!,:;]+
'''
```

In [65]:

```
tweet1 = "@natalieohayre I agree #hc09 needs reform- but not by crooked politicians wh
tweet2 = "To Sen. Roland Burris: Affordable, quality health insurance can't wait http:
tweet3 = "RT @karoli: RT @Seriou: .@whitehouse I will stand w/ Obama on #healthcare, I
```

In [68]:

```
nltk.regexp_tokenize(tweet3,tweetPattern)
```

Out[68]:

```
[('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', ''),
 ('', '', '', '', '', '', '')]
```

```
In [ ]:
```