In [1]:
```
import os
import nltk
import nltk.corpus
```

In [8]:
```
y="NLTK is intended to support research and teaching in NLP or closely related are
```

In [9]:
```
y
```

Out[9]:
```
'NLTK is intended to support research and teaching in NLP or closely related area
s,including empirical linguistics, cognitive science, artificial intelligence, inf
ormation retrieval, and machine learning.NLTK has been used successfully as a teac
hing tool, as an individual study tool, and as a platform for prototyping and buil
ding research systems. There are 32 universities in the US and 25 countries using
NLTK in their courses. NLTK supports classification, tokenization, stemming, taggi
ng, parsing, and semantic reasoning functionalities'
```

In [10]:
```
type(y)
```

Out[10]:
```
str
```

In [11]:
```
from nltk.tokenize import word_tokenize
import nltk
```

In [12]:
```
ytokens=word_tokenize(y)
ytokens
```

Out[12]:
```
['NLTK',
 'is',
 'intended',
 'to',
 'support',
 'research',
 'and',
 'teaching',
 'in',
 'NLP',
 'or',
 'closely',
 'related',
 'areas',
 ',',
 'including',
 'empirical',
 'linguistics',
 ',',
 'cognitive',
 'science',
 ',',
 'artificial',
 'intelligence',
 ',',
 'information',
 'retrieval',
 ',',
 'and',
```

```
 'machine',
 'learning.NLTK',
 'has',
 'been',
 'used',
 'successfully',
 'as',
 'a',
 'teaching',
 'tool',
 ',',
 'as',
 'an',
 'individual',
 'study',
 'tool',
 ',',
 'and',
 'as',
 'a',
 'platform',
 'for',
 'prototyping',
 'and',
 'building',
 'research',
 'systems',
 '.',
 'There',
 'are',
 '32',
 'universities',
 'in',
 'the',
 'US',
 'and',
 '25',
 'countries',
 'using',
 'NLTK',
 'in',
 'their',
 'courses',
 '.',
 'NLTK',
 'supports',
 'classification',
 ',',
 'tokenization',
 ',',
 'stemming',
 ',',
 'tagging',
 ',',
 'parsing',
 ',',
 'and',
 'semantic',
 'reasoning',
 'functionalities']
```

In [13]:

```
len(ytokens)
```

Out[13]:
```
89
```

In [14]:
```
from nltk.probability import FreqDist
fdist=FreqDist()
```

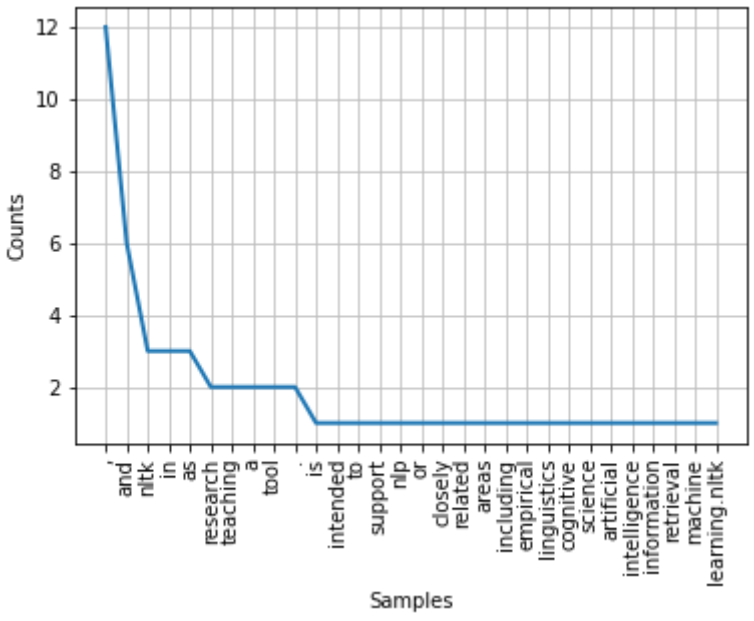
In [16]:
```
for word in ytokens:
    fdist[word.lower()]+=1
fdist
```

Out[16]:
```
FreqDist({',': 12, 'and': 6, 'nltk': 3, 'in': 3, 'as': 3, 'research': 2, 'teaching
': 2, 'a': 2, 'tool': 2, '.': 2, ...})
```

In [20]:
```
fdist_10=fdist.most_common(10)
fdist_10
```

Out[20]:
```
[(',', 12),
 ('and', 6),
 ('nltk', 3),
 ('in', 3),
 ('as', 3),
 ('research', 2),
 ('teaching', 2),
 ('a', 2),
 ('tool', 2),
 ('.', 2)]
```

In [39]:
```
import matplotlib.pyplot as plt
fdist.plot(30,cumulative=False)
plt.show()
```



In [26]:
```
from nltk.util import bigrams,trigrams,ngrams
```

In [27]:
```
str_bigrams=list(nltk.bigrams(ytokens))
str_bigrams
```

Out[27]:
```
[('NLTK', 'is'),
```

```
('is', 'intended'),
('intended', 'to'),
('to', 'support'),
('support', 'research'),
('research', 'and'),
('and', 'teaching'),
('teaching', 'in'),
('in', 'NLP'),
('NLP', 'or'),
('or', 'closely'),
('closely', 'related'),
('related', 'areas'),
('areas', ','),
(',', 'including'),
('including', 'empirical'),
('empirical', 'linguistics'),
('linguistics', ','),
(',', 'cognitive'),
('cognitive', 'science'),
('science', ','),
(',', 'artificial'),
('artificial', 'intelligence'),
('intelligence', ','),
(',', 'information'),
('information', 'retrieval'),
('retrieval', ','),
(',', 'and'),
('and', 'machine'),
('machine', 'learning.NLTK'),
('learning.NLTK', 'has'),
('has', 'been'),
('been', 'used'),
('used', 'successfully'),
('successfully', 'as'),
('as', 'a'),
('a', 'teaching'),
('teaching', 'tool'),
('tool', ','),
(',', 'as'),
('as', 'an'),
('an', 'individual'),
('individual', 'study'),
('study', 'tool'),
('tool', ','),
(',', 'and'),
('and', 'as'),
('as', 'a'),
('a', 'platform'),
('platform', 'for'),
('for', 'prototyping'),
('prototyping', 'and'),
('and', 'building'),
('building', 'research'),
('research', 'systems'),
('systems', '.'),
('.', 'There'),
('There', 'are'),
('are', '32'),
('32', 'universities'),
('universities', 'in'),
('in', 'the'),
('the', 'US'),
('US', 'and'),
```

```
('and', '25'),
('25', 'countries'),
('countries', 'using'),
('using', 'NLTK'),
('NLTK', 'in'),
('in', 'their'),
('their', 'courses'),
('courses', '.'),
('.', 'NLTK'),
('NLTK', 'supports'),
('supports', 'classification'),
('classification', ','),
(',', 'tokenization'),
('tokenization', ','),
(',', 'stemming'),
('stemming', ','),
(',', 'tagging'),
('tagging', ','),
(',', 'parsing'),
('parsing', ','),
(',', 'and'),
('and', 'semantic'),
('semantic', 'reasoning'),
```

In [28]:

```
str_trigrams=list(nltk.trigrams(ytokens))
str_trigrams
```

Out[28]:

```
[('NLTK', 'is', 'intended'),
 ('is', 'intended', 'to'),
 ('intended', 'to', 'support'),
 ('to', 'support', 'research'),
 ('support', 'research', 'and'),
 ('research', 'and', 'teaching'),
 ('and', 'teaching', 'in'),
 ('teaching', 'in', 'NLP'),
 ('in', 'NLP', 'or'),
 ('NLP', 'or', 'closely'),
 ('or', 'closely', 'related'),
 ('closely', 'related', 'areas'),
 ('related', 'areas', ','),
 ('areas', ',', 'including'),
 (',', 'including', 'empirical'),
 ('including', 'empirical', 'linguistics'),
 ('empirical', 'linguistics', ','),
 ('linguistics', ',', 'cognitive'),
 (',', 'cognitive', 'science'),
 ('cognitive', 'science', ','),
 ('science', ',', 'artificial'),
 (',', 'artificial', 'intelligence'),
 ('artificial', 'intelligence', ','),
 ('intelligence', ',', 'information'),
 (',', 'information', 'retrieval'),
 ('information', 'retrieval', ','),
 ('retrieval', ',', 'and'),
 (',', 'and', 'machine'),
 ('and', 'machine', 'learning.NLTK'),
 ('machine', 'learning.NLTK', 'has'),
 ('learning.NLTK', 'has', 'been'),
 ('has', 'been', 'used'),
 ('been', 'used', 'successfully'),
 ('used', 'successfully', 'as'),
 ('successfully', 'as', 'a'),
```

```
('as', 'a', 'teaching'),
('a', 'teaching', 'tool'),
('teaching', 'tool', ','),
('tool', ',', 'as'),
(',', 'as', 'an'),
('as', 'an', 'individual'),
('an', 'individual', 'study'),
('individual', 'study', 'tool'),
('study', 'tool', ','),
('tool', ',', 'and'),
(',', 'and', 'as'),
('and', 'as', 'a'),
('as', 'a', 'platform'),
('a', 'platform', 'for'),
('platform', 'for', 'prototyping'),
('for', 'prototyping', 'and'),
('prototyping', 'and', 'building'),
('and', 'building', 'research'),
('building', 'research', 'systems'),
('research', 'systems', '.'),
('systems', '.', 'There'),
('.', 'There', 'are'),
('There', 'are', '32'),
('are', '32', 'universities'),
('32', 'universities', 'in'),
('universities', 'in', 'the'),
('in', 'the', 'US'),
('the', 'US', 'and'),
('US', 'and', '25'),
('and', '25', 'countries'),
('25', 'countries', 'using'),
('countries', 'using', 'NLTK'),
('using', 'NLTK', 'in'),
('NLTK', 'in', 'their'),
('in', 'their', 'courses'),
('their', 'courses', '.'),
('courses', '.', 'NLTK'),
('.', 'NLTK', 'supports'),
('NLTK', 'supports', 'classification'),
('supports', 'classification', ','),
('classification', ',', 'tokenization'),
(',', 'tokenization', ','),
('tokenization', ',', 'stemming'),
(',', 'stemming', ','),
('stemming', ',', 'tagging'),
(',', 'tagging', ','),
('tagging', ',', 'parsing'),
(',', 'parsing', ','),
('parsing', ',', 'and'),
(',', 'and', 'semantic'),
('and', 'semantic', 'reasoning'),
('semantic', 'reasoning', 'functionalities')]
```

In [29]:

```
str_ngrams=list(nltk.ngrams(ytokens,7))
str_ngrams
```

Out[29]:

```
[('NLTK', 'is', 'intended', 'to', 'support', 'research', 'and'),
 ('is', 'intended', 'to', 'support', 'research', 'and', 'teaching'),
 ('intended', 'to', 'support', 'research', 'and', 'teaching', 'in'),
 ('to', 'support', 'research', 'and', 'teaching', 'in', 'NLP'),
 ('support', 'research', 'and', 'teaching', 'in', 'NLP', 'or'),
 ('research', 'and', 'teaching', 'in', 'NLP', 'or', 'closely'),
 ('and', 'teaching', 'in', 'NLP', 'or', 'closely', 'related'),
```

```
('teaching', 'in', 'NLP', 'or', 'closely', 'related', 'areas'),
('in', 'NLP', 'or', 'closely', 'related', 'areas', ','),
('NLP', 'or', 'closely', 'related', 'areas', ',', 'including'),
('or', 'closely', 'related', 'areas', ',', 'including', 'empirical'),
('closely', 'related', 'areas', ',', 'including', 'empirical', 'linguistics'),
('related', 'areas', ',', 'including', 'empirical', 'linguistics', ','),
('areas', ',', 'including', 'empirical', 'linguistics', ',', 'cognitive'),
(',', 'including', 'empirical', 'linguistics', ',', 'cognitive', 'science'),
('including', 'empirical', 'linguistics', ',', 'cognitive', 'science', ','),
('empirical', 'linguistics', ',', 'cognitive', 'science', ',', 'artificial'),
('linguistics',
 ',',
 'cognitive',
 'science',
 ',',
 'artificial',
 'intelligence'),
(',', 'cognitive', 'science', ',', 'artificial', 'intelligence', ','),
('cognitive',
 'science',
 ',',
 'artificial',
 'intelligence',
 ',',
 'information'),
('science',
 ',',
 'artificial',
 'intelligence',
 ',',
 'information',
 'retrieval'),
(',', 'artificial', 'intelligence', ',', 'information', 'retrieval', ','),
('artificial', 'intelligence', ',', 'information', 'retrieval', ',', 'and'),
('intelligence', ',', 'information', 'retrieval', ',', 'and', 'machine'),
(',', 'information', 'retrieval', ',', 'and', 'machine', 'learning.NLTK'),
('information', 'retrieval', ',', 'and', 'machine', 'learning.NLTK', 'has'),
('retrieval', ',', 'and', 'machine', 'learning.NLTK', 'has', 'been'),
(',', 'and', 'machine', 'learning.NLTK', 'has', 'been', 'used'),
('and', 'machine', 'learning.NLTK', 'has', 'been', 'used', 'successfully'),
('machine', 'learning.NLTK', 'has', 'been', 'used', 'successfully', 'as'),
('learning.NLTK', 'has', 'been', 'used', 'successfully', 'as', 'a'),
('has', 'been', 'used', 'successfully', 'as', 'a', 'teaching'),
('been', 'used', 'successfully', 'as', 'a', 'teaching', 'tool'),
('used', 'successfully', 'as', 'a', 'teaching', 'tool', ','),
('successfully', 'as', 'a', 'teaching', 'tool', ',', 'as'),
('as', 'a', 'teaching', 'tool', ',', 'as', 'an'),
('a', 'teaching', 'tool', ',', 'as', 'an', 'individual'),
('teaching', 'tool', ',', 'as', 'an', 'individual', 'study'),
('tool', ',', 'as', 'an', 'individual', 'study', 'tool'),
(',', 'as', 'an', 'individual', 'study', 'tool', ','),
('as', 'an', 'individual', 'study', 'tool', ',', 'and'),
('an', 'individual', 'study', 'tool', ',', 'and', 'as'),
('individual', 'study', 'tool', ',', 'and', 'as', 'a'),
('study', 'tool', ',', 'and', 'as', 'a', 'platform'),
('tool', ',', 'and', 'as', 'a', 'platform', 'for'),
(',', 'and', 'as', 'a', 'platform', 'for', 'prototyping'),
('and', 'as', 'a', 'platform', 'for', 'prototyping', 'and'),
('as', 'a', 'platform', 'for', 'prototyping', 'and', 'building'),
('a', 'platform', 'for', 'prototyping', 'and', 'building', 'research'),
('platform', 'for', 'prototyping', 'and', 'building', 'research', 'systems'),
('for', 'prototyping', 'and', 'building', 'research', 'systems', '.'),
('prototyping', 'and', 'building', 'research', 'systems', '.', 'There'),
```

```
('and', 'building', 'research', 'systems', '.', 'There', 'are'),
('building', 'research', 'systems', '.', 'There', 'are', '32'),
('research', 'systems', '.', 'There', 'are', '32', 'universities'),
('systems', '.', 'There', 'are', '32', 'universities', 'in'),
('.', 'There', 'are', '32', 'universities', 'in', 'the'),
('There', 'are', '32', 'universities', 'in', 'the', 'US'),
('are', '32', 'universities', 'in', 'the', 'US', 'and'),
('32', 'universities', 'in', 'the', 'US', 'and', '25'),
('universities', 'in', 'the', 'US', 'and', '25', 'countries'),
('in', 'the', 'US', 'and', '25', 'countries', 'using'),
('the', 'US', 'and', '25', 'countries', 'using', 'NLTK'),
('US', 'and', '25', 'countries', 'using', 'NLTK', 'in'),
('and', '25', 'countries', 'using', 'NLTK', 'in', 'their'),
('25', 'countries', 'using', 'NLTK', 'in', 'their', 'courses'),
('countries', 'using', 'NLTK', 'in', 'their', 'courses', '.'),
('using', 'NLTK', 'in', 'their', 'courses', '.', 'NLTK'),
('NLTK', 'in', 'their', 'courses', '.', 'NLTK', 'supports'),
('in', 'their', 'courses', '.', 'NLTK', 'supports', 'classification'),
('their', 'courses', '.', 'NLTK', 'supports', 'classification', ','),
('courses', '.', 'NLTK', 'supports', 'classification', ',', 'tokenization'),
('.', 'NLTK', 'supports', 'classification', ',', 'tokenization', ','),
('NLTK', 'supports', 'classification', ',', 'tokenization', ',', 'stemming'),
('supports', 'classification', ',', 'tokenization', ',', 'stemming', ','),
('classification', ',', 'tokenization', ',', 'stemming', ',', 'tagging'),
(',', 'tokenization', ',', 'stemming', ',', 'tagging', ','),
('tokenization', ',', 'stemming', ',', 'tagging', ',', 'parsing'),
(',', 'stemming', ',', 'tagging', ',', 'parsing', ','),
('stemming', ',', 'tagging', ',', 'parsing', ',', 'and'),
(',', 'tagging', ',', 'parsing', ',', 'and', 'semantic'),
('tagging', ',', 'parsing', ',', 'and', 'semantic', 'reasoning'),
('.', 'parsing', '.', 'and', 'semantic', 'reasoning', 'functionalities')]
```

In [31]:

```
from nltk.corpus import stopwords
stop_words = set(stopwords.words("english"))
print(stop_words)
```

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

In [33]:
```
filtered_sent=[]
for w in ytokens:
    if w not in stop_words:
        filtered_sent.append(w)
print("Tokenized Sentence:",ytokens)
print("Filterd Sentence:",filtered_sent)

print(len(ytokens))
print(len(filtered_sent))
```

```
Tokenized Sentence: ['NLTK', 'is', 'intended', 'to', 'support', 'research', 'and',
'teaching', 'in', 'NLP', 'or', 'closely', 'related', 'areas', ',', 'including', 'e
mpirical', 'linguistics', ',', 'cognitive', 'science', ',', 'artificial', 'intelli
gence', ',', 'information', 'retrieval', ',', 'and', 'machine', 'learning.NLTK', '
has', 'been', 'used', 'successfully', 'as', 'a', 'teaching', 'tool', ',', 'as', 'a
n', 'individual', 'study', 'tool', ',', 'and', 'as', 'a', 'platform', 'for', 'prot
otyping', 'and', 'building', 'research', 'systems', '.', 'There', 'are', '32', 'un
iversities', 'in', 'the', 'US', 'and', '25', 'countries', 'using', 'NLTK', 'in', '
their', 'courses', '.', 'NLTK', 'supports', 'classification', ',', 'tokenization',
',', 'stemming', ',', 'tagging', ',', 'parsing', ',', 'and', 'semantic', 'reasonin
g', 'functionalities']
Filterd Sentence: ['NLTK', 'intended', 'support', 'research', 'teaching', 'NLP', '
closely', 'related', 'areas', ',', 'including', 'empirical', 'linguistics', ',', '
cognitive', 'science', ',', 'artificial', 'intelligence', ',', 'information', 'ret
rieval', ',', 'machine', 'learning.NLTK', 'used', 'successfully', 'teaching', 'too
l', ',', 'individual', 'study', 'tool', ',', 'platform', 'prototyping', 'building
', 'research', 'systems', '.', 'There', '32', 'universities', 'US', '25', 'countri
es', 'using', 'NLTK', 'courses', '.', 'NLTK', 'supports', 'classification', ',', '
tokenization', ',', 'stemming', ',', 'tagging', ',', 'parsing', ',', 'semantic', '
reasoning', 'functionalities']
89
65
```

In [36]:
```
from nltk.stem import PorterStemmer
ps = PorterStemmer()
stemmed_words=[]
for w in filtered_sent:
    stemmed_words.append(ps.stem(w))

print("Filtered Sentence:",filtered_sent)
print("Stemmed Sentence:",stemmed_words)
```

```
Filtered Sentence: ['NLTK', 'intended', 'support', 'research', 'teaching', 'NLP',
'closely', 'related', 'areas', ',', 'including', 'empirical', 'linguistics', ',',
'cognitive', 'science', ',', 'artificial', 'intelligence', ',', 'information', 're
trieval', ',', 'machine', 'learning.NLTK', 'used', 'successfully', 'teaching', 'to
ol', ',', 'individual', 'study', 'tool', ',', 'platform', 'prototyping', 'building
', 'research', 'systems', '.', 'There', '32', 'universities', 'US', '25', 'countri
es', 'using', 'NLTK', 'courses', '.', 'NLTK', 'supports', 'classification', ',', '
tokenization', ',', 'stemming', ',', 'tagging', ',', 'parsing', ',', 'semantic', '
reasoning', 'functionalities']
Stemmed Sentence: ['nltk', 'intend', 'support', 'research', 'teach', 'nlp', 'close
', 'relat', 'area', ',', 'includ', 'empir', 'linguist', ',', 'cognit', 'scienc',
',', 'artifici', 'intellig', ',', 'inform', 'retriev', ',', 'machin', 'learning.nl
tk', 'use', 'success', 'teach', 'tool', ',', 'individu', 'studi', 'tool', ',', 'pl
atform', 'prototyp', 'build', 'research', 'system', '.', 'there', '32', 'univers',
'us', '25', 'countri', 'use', 'nltk', 'cours', '.', 'nltk', 'support', 'classif',
',', 'token', ',', 'stem', ',', 'tag', ',', 'pars', ',', 'semant', 'reason', 'func
tion']
```

In [45]:
```
from nltk.stem.wordnet import WordNetLemmatizer
import pandas as pd
lem = WordNetLemmatizer()
stem = PorterStemmer()
word = "flying"
print("Lemmatized Word:",lem.lemmatize(word,"v"))
print("Stemmed Word:",stem.stem(word))
```

```
Lemmatized Word: fly
Stemmed Word: fli
```

In [37]:
```
POS_tagging = nltk.pos_tag(filtered_sent)
print(POS_tagging)
```

```
[('NLTK', 'NNP'), ('intended', 'VBD'), ('support', 'NN'), ('research', 'NN'), ('te
aching', 'VBG'), ('NLP', 'NNP'), ('closely', 'RB'), ('related', 'JJ'), ('areas', '
NNS'), (',', ','), ('including', 'VBG'), ('empirical', 'JJ'), ('linguistics', 'NNS
'), (',', ','), ('cognitive', 'JJ'), ('science', 'NN'), (',', ','), ('artificial',
'JJ'), ('intelligence', 'NN'), (',', ','), ('information', 'NN'), ('retrieval', 'N
N'), (',', ','), ('machine', 'NN'), ('learning.NLTK', 'NN'), ('used', 'VBN'), ('su
ccessfully', 'RB'), ('teaching', 'VBG'), ('tool', 'NN'), (',', ','), ('individual
', 'JJ'), ('study', 'NN'), ('tool', 'NN'), (',', ','), ('platform', 'NN'), ('proto
typing', 'VBG'), ('building', 'NN'), ('research', 'NN'), ('systems', 'NNS'), ('.',
'.'), ('There', 'EX'), ('32', 'CD'), ('universities', 'NNS'), ('US', 'PRP'), ('25
', 'CD'), ('countries', 'NNS'), ('using', 'VBG'), ('NLTK', 'NNP'), ('courses', 'NN
S'), ('.', '.'), ('NLTK', 'NNP'), ('supports', 'VBZ'), ('classification', 'NN'),
(',', ','), ('tokenization', 'NN'), (',', ','), ('stemming', 'VBG'), (',', ','),
('tagging', 'VBG'), (',', ','), ('parsing', 'VBG'), (',', ','), ('semantic', 'JJ
'), ('reasoning', 'NN'), ('functionalities', 'NNS')]
```

In [40]:
```
from nltk.sentiment.vader import SentimentIntensityAnalyzer
hotel_rev=['Great place to be when you are in banglore.',
           'The place was being renovated when I visited so the seating was limited
           'Loved the ambience,loved the food.',
            'Mushroom fried rice was tasty',
           'The food was delicious but not over the top',
           'The place is not easy to locate']
sid=SentimentIntensityAnalyzer()
for sentence in hotel_rev:
    print(sentence)
    ss=sid.polarity_scores(sentence)
    for k in ss:
        print('{0}:{1}, '.format(k,ss[k]),end='')
        print()
```

```
Great place to be when you are in banglore.
neg:0.0,
neu:0.661,
pos:0.339,
compound:0.6249,
The place was being renovated when I visited so the seating was limited.
neg:0.147,
neu:0.853,
pos:0.0,
compound:-0.2263,
Loved the ambience,loved the food.
neg:0.0,
neu:0.506,
pos:0.494,
compound:0.5994,
```

```
Mushroom fried rice was tasty
neg:0.0,
neu:1.0,
pos:0.0,
compound:0.0,
The food was delicious but not over the top
neg:0.168,
neu:0.623,
pos:0.209,
compound:0.1184,
The place is not easy to locate
neg:0.286,
neu:0.714,
pos:0.0,
```

In [44]:

```
#count vectorizer
from sklearn.feature_extraction.text import CountVectorizer
new_text=['This is the first document.',
          'This document is the second document.',
          'And this is the third one.',
          'Is this the first document']
vectorizer = CountVectorizer()
X = vectorizer.fit_transform(new_text).toarray()
X
```

Out[44]:

```
array([[0, 1, 1, 1, 0, 0, 1, 0, 1],
       [0, 2, 0, 1, 0, 1, 1, 0, 1],
       [1, 0, 0, 1, 1, 0, 1, 1, 1],
       [0, 1, 1, 1, 0, 0, 1, 0, 1]], dtype=int64)
```

In [ ]: