In [14]:
```
import nltk
```

In [15]:
```
t="""Hello Mr. Smith, how are you doing today? The weather is great, and city is
 sky is pinkish-blue. You shouldn't eat cardboard. Exemple : nouvelles images à v
```

In [16]:
```
text.split(' ')
```

Out[16]:
```
['Hello',
 'Mr.',
 'Smith,',
 'how',
 'are',
 'you',
 'doing',
 'today?',
 'The',
 'weather',
 'is',
 'great,',
 'and',
 'city',
 'is',
 'awesome.\nThe',
 'sky',
 'is',
 'pinkish-blue.',
 'You',
 "shouldn't",
 'eat',
 'cardboard.',
 'Exemple\xa0:',
 'nouvelles',
 'images',
 'à',
 'venir',
 'demain',
 'nouvelles',
 'images',
 'à',
 'venir',
 'demain.']
```

In [17]:
```
!pip install nltk
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: nltk in c:\programdata\anaconda3\lib\site-packag
es (3.7)
Requirement already satisfied: regex>=2021.8.3 in c:\programdata\anaconda3\lib
\site-packages (from nltk) (2022.3.15)
Requirement already satisfied: joblib in c:\programdata\anaconda3\lib\site-pack
ages (from nltk) (1.1.0)
Requirement already satisfied: tqdm in c:\programdata\anaconda3\lib\site-packag
es (from nltk) (4.64.0)
Requirement already satisfied: click in c:\programdata\anaconda3\lib\site-packa
ges (from nltk) (8.0.4)
Requirement already satisfied: colorama in c:\programdata\anaconda3\lib\site-pa
ckages (from click->nltk) (0.4.4)
```

In [55]:
```
text="""Hello Mr. Smith, how are you doing today? The weather is great, and city
The sky is pinkish-blue. You shouldn't eat cardboard.Merhaba Bay Smith, bugün nas
Gökyüzü pembemsi-mavidir. karton yememelisin.Bonjour M. Smith, comment allez-vous
Le ciel est bleu rosé. Tu ne devrais pas manger de carton."""
```

Sentence Tokenization

In [56]:
```
nltk.download('punkt')
```

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

Out[56]:
```
True
```

In [89]:
```
from nltk.tokenize import sent_tokenize
```

In [90]:
```
tokenized_text=sent_tokenize(text)
print(tokenized_text)
```

```
['Hello Mr. Smith, how are you doing today?', 'The weather is great, and city i
s awesome.', 'The sky is pinkish-blue.', "You shouldn't eat cardboard.Merhaba B
ay Smith, bugün nasılsınız?", 'Hava harika ve şehir harika.', 'Gökyüzü pembemsi
-mavidir.', "karton yememelisin.Bonjour M. Smith, comment allez-vous aujourd'hu
i ?", 'Il fait beau et la ville est géniale.', 'Le ciel est bleu rosé.', 'Tu ne
devrais pas manger de carton.']
```

In [ ]:
```

```

Word Tokenization

```
In [91]: from nltk.tokenize import word_tokenize
```

```
In [92]: tokenized_word=word_tokenize(text)
```

```
In [93]: print(tokenized_word)
```

```
['Hello', 'Mr.', 'Smith', ',', 'how', 'are', 'you', 'doing', 'today', '?', 'Th
e', 'weather', 'is', 'great', ',', 'and', 'city', 'is', 'awesome', '.', 'The',
'sky', 'is', 'pinkish-blue', '.', 'You', 'should', "n't", 'eat', 'cardboard.Mer
haba', 'Bay', 'Smith', ',', 'bugün', 'nasılsınız', '?', 'Hava', 'harika', 've',
'şehir', 'harika', '.', 'Gökyüzü', 'pembemsi-mavidir', '.', 'karton', 'yememeli
sin.Bonjour', 'M.', 'Smith', ',', 'comment', 'allez-vous', "aujourd'hui", '?',
'Il', 'fait', 'beau', 'et', 'la', 'ville', 'est', 'géniale', '.', 'Le', 'ciel',
'est', 'bleu', 'rosé', '.', 'Tu', 'ne', 'devrais', 'pas', 'manger', 'de', 'cart
on', '.']
```

Frequency Distribution

```
In [94]: from nltk.probability import FreqDist
```

```
In [95]: fdist = FreqDist(tokenized_word)
```

```
In [96]: print(fdist)
```

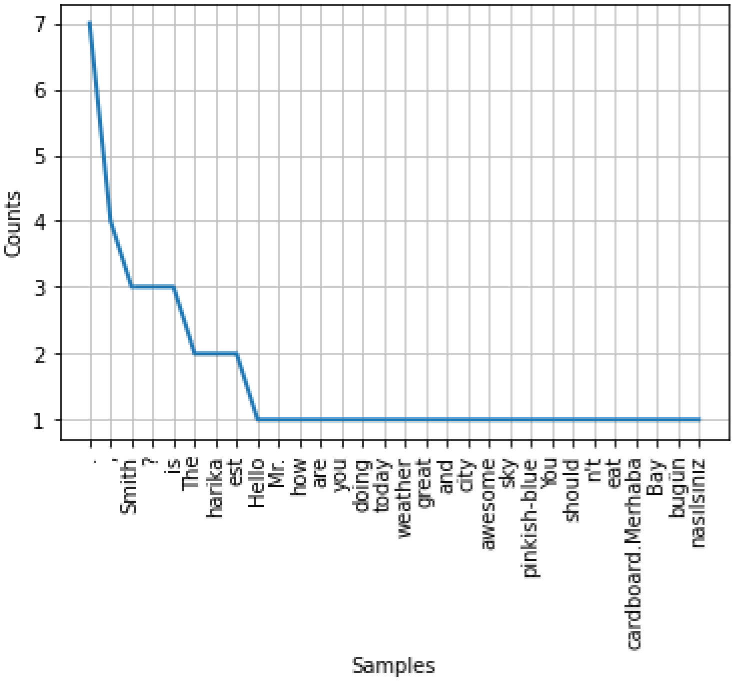
```
<FreqDist with 59 samples and 77 outcomes>
```

```
In [97]: fdist.most_common(2)
```

```
Out[97]: [('.', 7), (',', 4)]
```

```
In [98]: # Frequency Distribution Plot
         import matplotlib.pyplot as plt
```

In [99]:
```
fdist.plot(30,cumulative=False)
plt.show()
```



Stopwords

In [100]:
```
from nltk.corpus import stopwords
```

In [101]:
```
nltk.download('stopwords')
```

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

Out[101]:
```
True
```

In [102]:
```
stop_words=set(stopwords.words("Turkish"))
```

In [71]:
```
print(stop_words)
```

```
{'veya', 'aslında', 'yani', 'biz', 'acaba', 'ya', 'her', 'bu', 'şu', 'bazı', 'k
im', 'ki', 'tüm', 'niçin', 'hem', 'sanki', 'birşey', 'hep', 'mü', 'en', 'ile',
'eğer', 'o', 'da', 'az', 'belki', 'hepsi', 'siz', 'çünkü', 'nasıl', 'ise', 'v
e', 'de', 'nerede', 'nerde', 'birkaç', 'ne', 'hiç', 'biri', 'mu', 'çok', 'gib
i', 'kez', 'daha', 'niye', 'şey', 'neden', 'mı', 'nereye', 'diye', 'defa', 'içi
n', 'ama'}
```

In [72]:
```
stop_words=set(stopwords.words("French"))
```

In [73]:
```
print(stop_words)
```

```
{'ayant', 'aurait', 'étiez', 'fûmes', 'serai', 'ont', 'que', 'ton', 'pour', 'eu
ssent', 'des', 'moi', 'c', 'étantes', 'eues', 'soyons', 'ou', 'auriez', 'l',
'y', 'eut', 'eusses', 'toi', 'n', 'sur', 'étais', 'aurai', 'à', 'eus', 'eue',
'étaient', 'notre', 'de', 'ne', 'avez', 'serez', 'auraient', 'par', 'aie', 'éti
ons', 'aux', 'fus', 'ait', 'les', 'au', 'leur', 'serions', 'j', 'vos', 'ayons',
'un', 'on', 'qui', 'vous', 'aura', 'seriez', 'avaient', 'ayante', 'me', 'nos',
'te', 'auras', 'avais', 'ma', 'lui', 'étante', 'eusse', 'été', 'seront', 'soien
t', 'ta', 'ils', 'serons', 'pas', 'le', 'aient', 'je', 'ayez', 'eussiez', 'fû
t', 'votre', 'suis', 'es', 'm', 'sommes', 'eûmes', 'ces', 't', 'fussiez', 'il',
'soyez', 'étant', 'fûtes', 'avons', 'avions', 'fussions', 'mes', 'as', 'sois',
'son', 'mon', 'se', 'seraient', 'eûtes', 'fusses', 'aurons', 'en', 'ayants', 'a
urions', 'tes', 'étée', 'aurais', 'nous', 'aurez', 'étants', 'mais', 'du', 'es
t', 'auront', 'une', 'soit', 'eût', 'tu', 'aies', 'même', 'et', 'ce', 'eu', 'a
i', 'avait', 'sa', 'dans', 'sont', 'étés', 'avec', 'fussent', 'fut', 'êtes', 'l
a', 'étées', 'sera', 'eux', 'd', 's', 'ayantes', 'était', 'ses', 'seras', 'ell
e', 'qu', 'fusse', 'eurent', 'furent', 'serais', 'aviez', 'eussions', 'serait'}
```

In [74]:
```
stop_words=set(stopwords.words("english"))
```

In [75]:
```
print(stop_words)
```

```
{'to', 'than', 'needn', 'you', 'who', 'did', 'mightn', 'she', 'some', 'where',
'too', 'aren', 'an', 'these', 'yours', 'hasn', 'here', 'shouldn', 'yourself',
'y', 'mustn', "you've", "don't", 'what', 'don', 'be', 'by', 've', "won't", 'whe
n', 'during', "mustn't", 'should', 'having', 'most', 'over', "haven't", 'his',
'very', 'its', 'the', 'wasn', 'haven', 'shan', 'for', "aren't", 'own', 'from',
'my', "you're", 'all', "hadn't", "it's", 'doesn', 'again', 'same', 'further',
'at', "needn't", 'on', "you'd", 'is', 'me', 'of', 'been', 'ourselves', "could
n't", 'up', 'ma', 'while', 'yourselves', 'weren', 'with', 're', 'hadn', 'just',
'then', 'had', 'ours', 'll', 'myself', "weren't", 'because', "she's", 'into',
'out', 'how', 'm', "doesn't", 'won', 'before', 't', 'both', "didn't", "might
n't", 'their', 'whom', 'but', 'theirs', 'wouldn', "you'll", 'those', 'him', 'un
til', 'so', 'as', 'themselves', 'them', 'few', 'only', 'that', 'this', 'under',
'down', 'hers', 'which', 'not', 'will', 'and', 'was', 'such', 'itself', 'am',
'each', 'couldn', "should've", 'herself', 'or', 'if', 'has', 'a', "that'll", 'n
o', 'didn', 'other', 'now', 'her', 'there', 'our', 'through', 'nor', 'isn', 'yo
ur', 'below', 'after', "shouldn't", 'ain', 'are', 'being', 'were', 'i', 'abou
t', 'he', 'off', 'o', "isn't", 'between', 'himself', 'have', 'they', 'why',
's', 'against', 'd', 'more', 'does', 'can', 'above', 'any', "wasn't", 'in', 'w
e', 'doing', "hasn't", 'do', "shan't", "wouldn't", 'it', 'once'}
```

Removing Stopwords

In [76]:

```
if 'our' in stop_words:print("fount it")
```

```
fount it
```

In [77]:

```
filtered_sent = []
for word in tokenized_sent:
    if word not in stop_words:
        filtered_sent.append(word)
print("Tokenized Sentence:",tokenized_sent)
print("Filterd Sentence:",filtered_sent)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
Input In [77], in <cell line: 2>()
      1 filtered_sent = []
----> 2 for word in tokenized_sent:
      3     if word not in stop_words:
      4         filtered_sent.append(word)

NameError: name 'tokenized_sent' is not defined
```

In [79]:

```
# download stpwords
import nltk
nltk.download('stopwords')

# import nltk for stopwords
from nltk.corpus import stopwords
stop_words = set(stopwords.words('english'))
print(stop_words)

# assign string
no_wspace_string='python released in was a major revision of the language that is

# convert string to list of words
lst_string = [no_wspace_string][0].split()
print(lst_string)

# remove stopwords
no_stpwords_string=""
for i in lst_string:
    if not i in stop_words:
        no_stpwords_string += i+' '

        # removing last space
no_stpwords_string = no_stpwords_string[:-1]
print(no_stpwords_string)
```

```
{'to', 'than', 'needn', 'you', 'who', 'did', 'mightn', 'she', 'some', 'where',
'too', 'aren', 'an', 'these', 'yours', 'hasn', 'here', 'shouldn', 'yourself',
'y', 'mustn', "you've", "don't", 'what', 'don', 'be', 'by', 've', "won't", 'whe
n', 'during', "mustn't", 'should', 'having', 'most', 'over', "haven't", 'his',
'very', 'its', 'the', 'wasn', 'haven', 'shan', 'for', "aren't", 'own', 'from',
'my', "you're", 'all', "hadn't", "it's", 'doesn', 'again', 'same', 'further',
'at', "needn't", 'on', "you'd", 'is', 'me', 'of', 'been', 'ourselves', "could
n't", 'up', 'ma', 'while', 'yourselves', 'weren', 'with', 're', 'hadn', 'just',
'then', 'had', 'ours', 'll', 'myself', "weren't", 'because', "she's", 'into',
'out', 'how', 'm', "doesn't", 'won', 'before', 't', 'both', "didn't", "might
n't", 'their', 'whom', 'but', 'theirs', 'wouldn', "you'll", 'those', 'him', 'un
til', 'so', 'as', 'themselves', 'them', 'few', 'only', 'that', 'this', 'under',
'down', 'hers', 'which', 'not', 'will', 'and', 'was', 'such', 'itself', 'am',
'each', 'couldn', "should've", 'herself', 'or', 'if', 'has', 'a', "that'll", 'n
o', 'didn', 'other', 'now', 'her', 'there', 'our', 'through', 'nor', 'isn', 'yo
ur', 'below', 'after', "shouldn't", 'ain', 'are', 'being', 'were', 'i', 'abou
t', 'he', 'off', 'o', "isn't", 'between', 'himself', 'have', 'they', 'why',
's', 'against', 'd', 'more', 'does', 'can', 'above', 'any', "wasn't", 'in', 'w
e', 'doing', "hasn't", 'do', "shan't", "wouldn't", 'it', 'once'}
['python', 'released', 'in', 'was', 'a', 'major', 'revision', 'of', 'the', 'lan
guage', 'that', 'is', 'not', 'completely', 'backward', 'compatible', 'and', 'mu
ch', 'python', 'code', 'does', 'not', 'run', 'unmodified', 'on', 'python', 'wit
h', 'python', 's', 'endoflife', 'only', 'python', 'x', 'and', 'later', 'are',
'supported', 'with', 'older', 'versions', 'still', 'supporting', 'eg', 'window
s', 'and', 'old', 'installers', 'not', 'restricted', 'to', 'bit', 'windows']
python released major revision language completely backward compatible much pyt
hon code run unmodified python python endoflife python x later supported older
versions still supporting eg windows old installers restricted bit windows
```

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

Lexicon Normalization

Stemming

In [80]:

```
# Stemming
from nltk.stem import PorterStemmer
from nltk.tokenize import sent_tokenize, word_tokenize
```

In [81]:

```
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: []
Stemmed Sentence: []
```

Lemmatization

In [82]:

```
nltk.download('wordnet')
```

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

Out[82]:

```
True
```

In [83]:

```
nltk.download('omw-1.4')
```

```
[nltk_data] Downloading package omw-1.4 to
[nltk_data]     C:\Users\oumar\AppData\Roaming\nltk_data...
[nltk_data]   Package omw-1.4 is already up-to-date!
```

Out[83]:

```
True
```

In [84]:

```
#Lexicon Normalization
#performing stemming and Lemmatization

from nltk.stem.wordnet import WordNetLemmatizer
lem = WordNetLemmatizer()

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

word = "flying"
print("Lemmatized Word:",lem.lemmatize(word,"v"))
print("Stemmed Word:",stem.stem(word))
```

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

POS Tagging

In [85]:

```
sent = "Albert Einstein was born in Ulm, Germany in 1879."
```

In [86]:

```
tokens=nltk.word_tokenize(sent)
print(tokens)
```

```
['Albert', 'Einstein', 'was', 'born', 'in', 'Ulm', ',', 'Germany', 'in', '187
9', '.']
```

In [87]:

```
nltk.download('averaged_perceptron_tagger')
```

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

Out[87]:

```
True
```

In [88]:

```
nltk.pos_tag(tokens)
```

Out[88]:

```
[('Albert', 'NNP'),
 ('Einstein', 'NNP'),
 ('was', 'VBD'),
 ('born', 'VBN'),
 ('in', 'IN'),
 ('Ulm', 'NNP'),
 (',', ','),
 ('Germany', 'NNP'),
 ('in', 'IN'),
 ('1879', 'CD'),
 ('.', '.')]
```

In [ ]:

In [ ]: