In [2]:
```
pip install nltk
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: nltk in c:\users\vrushabh\appdata\roaming\pyth
on\python311\site-packages (3.8.1)
Requirement already satisfied: click in c:\users\vrushabh\appdata\roaming\pyt
hon\python311\site-packages (from nltk) (8.1.3)
Requirement already satisfied: joblib in c:\users\vrushabh\appdata\roaming\py
thon\python311\site-packages (from nltk) (1.2.0)
Requirement already satisfied: regex>=2021.8.3 in c:\users\vrushabh\appdata\r
oaming\python\python311\site-packages (from nltk) (2023.5.5)
Requirement already satisfied: tqdm in c:\users\vrushabh\appdata\roaming\pyth
on\python311\site-packages (from nltk) (4.65.0)
Requirement already satisfied: colorama in c:\users\vrushabh\appdata\roaming
\python\python311\site-packages (from click->nltk) (0.4.6)
Note: you may need to restart the kernel to use updated packages.
```

In [3]:
```
pip install textblob
```

```
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: textblob in c:\users\vrushabh\appdata\roaming
\python\python311\site-packages (0.17.1)
Requirement already satisfied: nltk>=3.1 in c:\users\vrushabh\appdata\roaming
\python\python311\site-packages (from textblob) (3.8.1)
Requirement already satisfied: click in c:\users\vrushabh\appdata\roaming\pyt
hon\python311\site-packages (from nltk>=3.1->textblob) (8.1.3)
Requirement already satisfied: joblib in c:\users\vrushabh\appdata\roaming\py
thon\python311\site-packages (from nltk>=3.1->textblob) (1.2.0)
Requirement already satisfied: regex>=2021.8.3 in c:\users\vrushabh\appdata\r
oaming\python\python311\site-packages (from nltk>=3.1->textblob) (2023.5.5)
Requirement already satisfied: tqdm in c:\users\vrushabh\appdata\roaming\pyth
on\python311\site-packages (from nltk>=3.1->textblob) (4.65.0)
Requirement already satisfied: colorama in c:\users\vrushabh\appdata\roaming
\python\python311\site-packages (from click->nltk>=3.1->textblob) (0.4.6)
Note: you may need to restart the kernel to use updated packages.
```

In [4]:
```
from textblob import TextBlob
```

In [5]:
```
t1= TextBlob("i havv abs")
print("incorrect statement",t1)
```

```
incorrect statement i havv abs
```

In [6]:
```
t1.correct()
```

Out[6]:
```
TextBlob("i have as")
```

In [7]:
```
import nltk
nltk.download('punkt')
sentence1=TextBlob("hi how are you all")
sentence1.words
```

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

Out[7]:
```
WordList(['hi', 'how', 'are', 'you', 'all'])
```

In [8]:
```
t2=TextBlob("baby who cares i know Who cares")
print(t2.words.pluralize())
```

```
['babies', 'whoes', 'caress', 'is', 'knows', 'Whoes', 'caress']
```

In [9]:
```
t2.words[2].singularize()
```

Out[9]:
```
'care'
```

In [10]:
```
t2.words.count("cares")
```

Out[10]:
```
2
```

In [11]:
```
t2.word_counts["who"]
```

Out[11]:
```
2
```

In [12]:
```
t2.words.count("who",case_sensitive=True)
```

Out[12]:
```
1
```

In [13]:
```
print(t2.parse())
```

```
baby/NN/B-NP/O who/WP/O/O cares/VBZ/B-VP/O i/PRP/B-NP/O know/VB/B-VP/O Who/W
P/O/O cares/VBZ/B-VP/O
```

In [14]:
```
zen=TextBlob("Beautiful is Better")
zen[0:19]
```

Out[14]:
```
TextBlob("Beautiful is Better")
```

In [15]:
```
zen.upper()
```

Out[15]:
```
TextBlob("BEAUTIFUL IS BETTER")
```

In [16]:
```
zen.lower()
```

Out[16]:
```
TextBlob("beautiful is better")
```

In [17]:
```
zen.find("is")
```

Out[17]:
```
10
```

In [18]:
```
blob=TextBlob("Beautiful is Better")
blob.ngrams(n=3)
```

Out[18]:
```
[WordList(['Beautiful', 'is', 'Better'])]
```

In [19]:
```
##tokenize the word


import nltk
from nltk.tokenize import sent_tokenize
text = "i will clear all the practicals with the flying colours "
print(text)
```

```
i will clear all the practicals with the flying colours
```

In [20]:
```
##word tokenize

from nltk.tokenize import word_tokenize
wordTokenize= word_tokenize(text)
print(wordTokenize)
```

```
['i', 'will', 'clear', 'all', 'the', 'practicals', 'with', 'the', 'flying',
'colours']
```

In [21]:
```
##frequency distribution

from nltk.probability import FreqDist

freqdist=FreqDist(text)

print(freqdist)
```

```
<FreqDist with 18 samples and 56 outcomes>
```

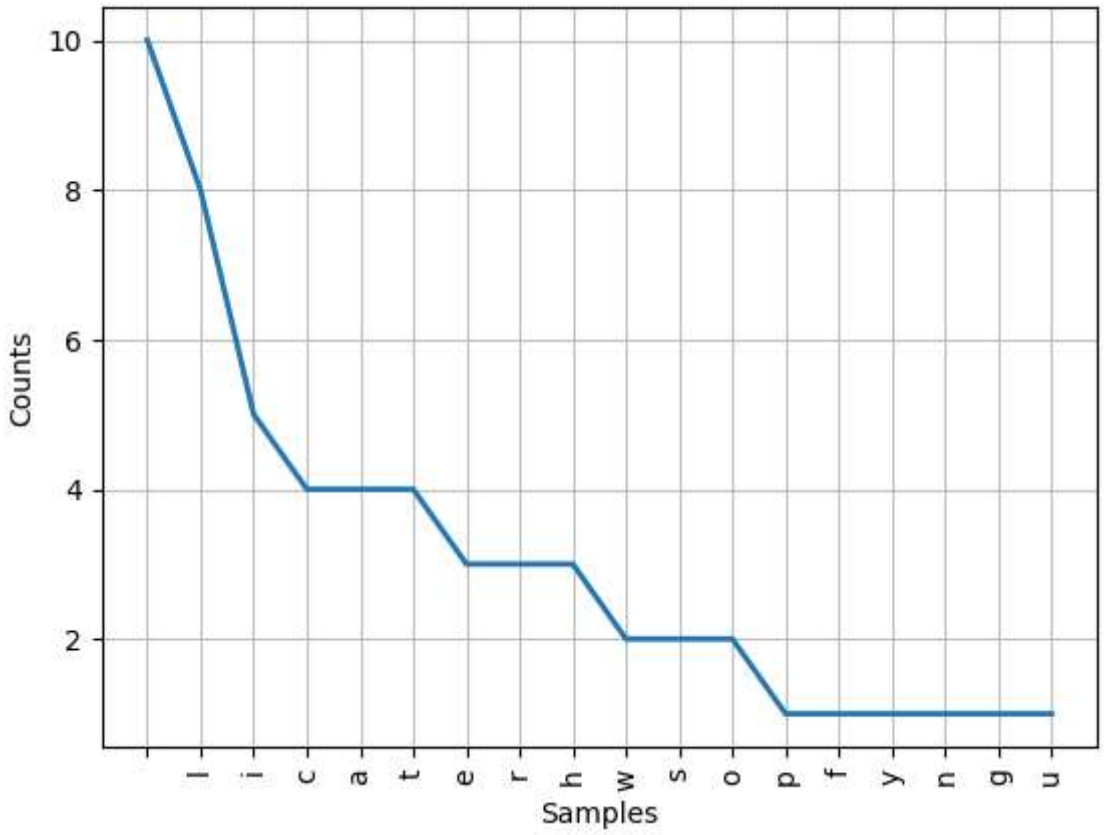
In [22]:
```
freqdist.most_common(3)
```

Out[22]:
```
[(' ', 10), ('l', 8), ('i', 5)]
```

In [23]:
```
import matplotlib.pyplot as plt

freqdist.plot(30,cumulative=False)

plt.show()
```



In [24]:
```
nltk.download('stopwords')
```

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

Out[24]: True

In [25]:
```
#stopwords
from nltk.corpus import stopwords
stop_words=set(stopwords.words('english'))
print(stop_words)
```

```
{'and', "isn't", "she's", 'did', 'whom', "shouldn't", 'doesn', 'just', 'm',
'there', 'aren', 'isn', 'too', 'the', 'because', 'yourselves', 'myself', 'oth
er', 'she', 'ours', "you're", 'was', 'more', 'by', 'y', "weren't", 'why', 'ha
ving', "shan't", 'they', 'be', 'haven', 'your', 'here', 'than', 'himself', 'd
idn', 'his', 'until', 'before', 'o', 'this', 'again', "mightn't", 'some', 'ha
dn', "mustn't", 're', 'into', 'itself', "hasn't", 'herself', "needn't", 'betw
een', "haven't", 'theirs', 'its', 'when', "hadn't", 'what', "you've", 'he',
'an', 'if', 'after', 'from', 'can', 'themselves', 'mightn', 'as', 'no', 'ar
e', 'hers', 'out', 'each', 'few', 'couldn', 'how', 's', 'ma', 'her', "won't",
'wouldn', 'i', 'who', 'am', 'you', 'while', 'once', 'own', 'that', 'ain', 'wi
ll', 'so', 'down', 'at', 'is', 'ourselves', 'these', 'same', 'weren', 'belo
w', "aren't", 'yourself', 'to', 'on', 'him', "doesn't", 'for', "you'll", 'dur
ing', 'shan', 'them', 'shouldn', 'doing', 'but', "wasn't", 'a', 'up', 'hasn',
'where', 'not', 'it', "couldn't", 'been', 'has', 'me', 'nor', 'have', 'whic
h', 'with', 'do', 'most', 'under', 'any', 'their', 'won', 'or', 'mustn', 'd',
'we', 'such', 'does', 'very', 'now', 'our', "didn't", 'about', 'off', 'of',
'both', 't', 've', "wouldn't", 'against', 'should', 'don', 'further', 'need
n', "don't", 'in', 'were', 'had', 'only', 'wasn', 'll', "should've", 'all',
'through', 'yours', 'above', 'my', 'then', "you'd", "it's", "that'll", 'thos
e', 'being', 'over'}
```

In [26]:
```
#stem and lemmatization

nltk.download('wordnet')
```

```
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\Vrushabh\AppData\Roaming\nltk_data...
```

Out[26]:
```
True
```

In [33]:
```
from nltk.stem.wordnet import WordNetLemmatizer
lem=WordNetLemmatizer()
from nltk.stem.porter import PorterStemmer
ps = PorterStemmer()
word="flying"
print("lemmatized word:",lem.lemmatize(word))
print("stem word:",ps.stem(word))
```

```
lemmatized word: flying
stem word: fli
```

In [35]:
```
from nltk.tokenize import word_tokenize
token=word_tokenize("hi i am vrushabh")
print(token)
```

```
['hi', 'i', 'am', 'vrushabh']
```

In [38]:
```
nltk.download('averaged_perceptron_tagger')
```

```
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\Vrushabh\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping taggers\averaged_perceptron_tagger.zip.
```

Out[38]:
```
True
```

In [40]:
```
nltk.pos_tag(token)
```

Out[40]:
```
[('hi', 'NN'), ('i', 'NN'), ('am', 'VBP'), ('vrushabh', 'NN')]
```

In [41]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
```

In [42]:
```
corpus = [
    "This is the first document.",
    "This document is the second document.",
    "And this is the third one.",
    "Is this the first document?",
]
```

In [43]:
```
vectorizer = TfidfVectorizer()
```

In [44]:
```
tfidf_matrix = vectorizer.fit_transform(corpus)
```

In [45]:
```
tfidf_matrix
```

Out[45]:
```
<4x9 sparse matrix of type '<class 'numpy.float64'>'
        with 21 stored elements in Compressed Sparse Row format>
```

In [46]:
```
feature_names = vectorizer.get_feature_names_out()

feature_names
```

Out[46]:
```
array(['and', 'document', 'first', 'is', 'one', 'second', 'the', 'third',
       'this'], dtype=object)
```

In [47]:
```
for doc_index, doc in enumerate(corpus):
    print(f"\nDocument {doc_index + 1}: ")
    for term_index, term in enumerate(feature_names):
        tfidf_value = tfidf_matrix[doc_index, term_index]
        if tfidf_value > 0:
            print(f"{term}: {tfidf_value:.2f}")
```

```
Document 1:
document: 0.47
first: 0.58
is: 0.38
the: 0.38
this: 0.38

Document 2:
document: 0.69
is: 0.28
second: 0.54
the: 0.28
this: 0.28

Document 3:
and: 0.51
is: 0.27
one: 0.51
the: 0.27
third: 0.51
this: 0.27

Document 4:
document: 0.47
first: 0.58
is: 0.38
the: 0.38
this: 0.38
```

In [ ]:
```
```