In [14]:

**from** textblob **import** TextBlob

**import** nltk

In [4]:

b **=** TextBlob("I havv good spelling")

In [5]:

b.correct()

Out[5]: TextBlob("I have good spelling")

In [22]:

**import** nltk nltk.download('punkt')

b1 **=** TextBlob("Beautiful is better than ugly."

"Explicit is better than implicit." "Simple is better than complex.")

b1.words

[nltk\_data] Downloading package punkt to

[nltk\_data] C:\Users\ADMIN\AppData\Roaming\nltk\_data... [nltk\_data] Package punkt is already up-to-date!

Out[22]: WordList(['Beautiful', 'is', 'better', 'than', 'ugly.Explicit', 'is', 'better

', 'than', 'implicit.Simple', 'is', 'better', 'than', 'complex'])

In [23]:

b1.sentences

Out[23]: [Sentence("Beautiful is better than ugly.Explicit is better than implicit.Sim ple is better than complex.")]

In [24]:

sentence **=** TextBlob("Use 4 spaces per indentation level") sentence.words

Out[24]: WordList(['Use', '4', 'spaces', 'per', 'indentation', 'level'])

In [27]:

sentence.words[2].singularize()

Out[27]: 'space'

In [28]:

sentence.words[5].pluralize()

Out[28]: 'levels'

In [30]:

animals **=** TextBlob("cat dog octopus") animals.words

Out[30]: WordList(['cat', 'dog', 'octopus'])

In [31]:

animals.words.pluralize()

Out[31]: WordList(['cats', 'dogs', 'octopodes'])

In [32]:

sen **=** TextBlob("We are no longer the knights who say Ni." "We are now the knig sen.word\_counts['ekki']

Out[32]: 3

In [33]:

sen.words.count('ekki')

Out[33]: 3

In [34]:

sen.words.count('ekki', case\_sensitive**=True**)

Out[34]: 2

In [35]:

b **=** TextBlob("And now for something completely different.") print(b.parse())

And/CC/O/O now/RB/B-ADVP/O for/IN/B-PP/B-PNP something/NN/B-NP/I-PNP complete ly/RB/B-ADJP/O different/JJ/I-ADJP/O ././O/O

In [36]:

b1[0:19]

TextBlob("Beautiful is better")

Out[36]: TextBlob("Beautiful is better")

In [37]:

b1.upper()

Out[37]: TextBlob("BEAUTIFUL IS BETTER THAN UGLY.EXPLICIT IS BETTER THAN IMPLICIT.SIMP LE IS BETTER THAN COMPLEX.")

In [39]:

b1.find("Simple")

Out[39]: 63

In [40]:

apple\_blob **=** TextBlob('apples') banana\_blob **=** TextBlob('bananas') apple\_blob **<** banana\_blob

Out[40]: True

In [41]:

blob **=** TextBlob("Now is better then never.") blob.ngrams(n**=**3)

Out[41]: [WordList(['Now', 'is', 'better']),

WordList(['is', 'better', 'then']),

WordList(['better', 'then', 'never'])]

In [42]:

**import** nltk

**from** nltk **import** tokenize

**from** nltk.tokenize **import** sent\_tokenize

text **=** """ Good Day everyone, how are you all today? Its fun learning data ana text

Out[42]: ' Good Day everyone, how are you all today? Its fun learning data analysis. H ope you all are practicing well.'

In [43]:

tokenized\_text **=** sent\_tokenize(text)

In [45]:

print(tokenized\_text)

[' Good Day everyone, how are you all today?', 'Its fun learning data analysi s.', 'Hope you all are practicing well.']

In [47]:

**from** nltk.tokenize **import** word\_tokenize tokenizer\_word **=** word\_tokenize(text) print(tokenizer\_word)

['Good', 'Day', 'everyone', ',', 'how', 'are', 'you', 'all', 'today', '?', 'I

ts', 'fun', 'learning', 'data', 'analysis', '.', 'Hope', 'you', 'all', 'are', 'practicing', 'well', '.']

In [48]:

**from** nltk.probability **import** FreqDist fdist **=** FreqDist(tokenizer\_word) print(fdist)

<FreqDist with 19 samples and 23 outcomes>

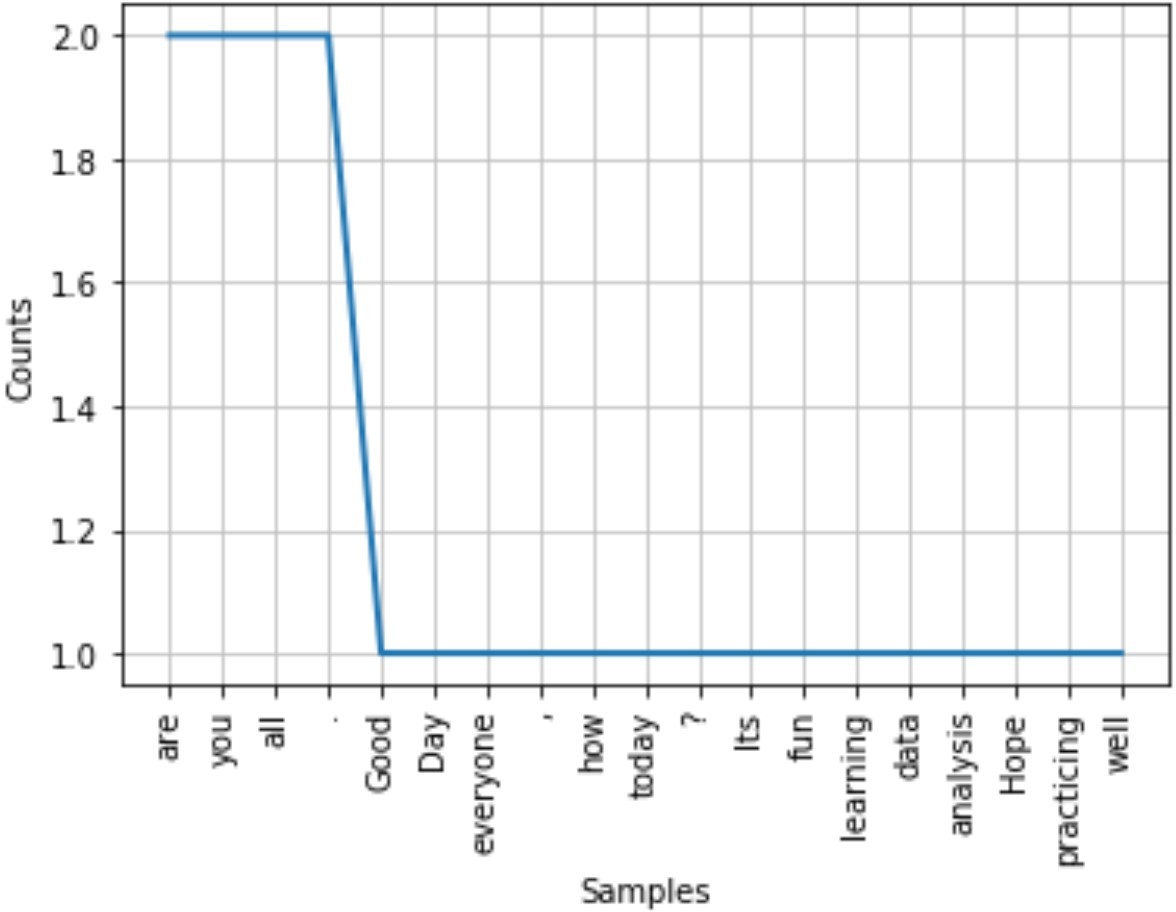
In [49]:

fdist.most\_common(4)

Out[49]: [('are', 2), ('you', 2), ('all', 2), ('.', 2)]

In [50]:

**import** matplotlib.pyplot **as** plt fdist.plot(30,cumulative**=False**) plt.show()



In [52]:

nltk.download('stopwords')

[nltk\_data] Downloading package stopwords to

[nltk\_data] C:\Users\ADMIN\AppData\Roaming\nltk\_data... [nltk\_data] Unzipping corpora\stopwords.zip.

Out[52]: True

In [57]:

**from** nltk.corpus **import** stopwords stop\_words **=** set(stopwords.words("english")) print(stop\_words)

{'wasn', 'how', 'herself', 'couldn', 'up', 'during', 'itself', 'than', 't', '

can', 'was', 'mustn', 'yours', 'did', 'didn', 'my', 'at', 'he', 'again', "wou

ldn't", 'myself', "isn't", 'it', "you'll", 'whom', 'any', "you're", "couldn'

t", 'shouldn', 'all', 'its', 'just', "hasn't", 'won', 'o', "mightn't", 'which

', 'hasn', 'that', 'theirs', 'over', 'i', 'such', 'am', 'until', 'on', 'throu

gh', 's', 'about', 'being', "mustn't", 'under', 'same', 'has', "should've", '

each', 'we', 'them', 'were', 'nor', 'here', "shan't", 'does', 'below', 'why',

'her', 'mightn', 'be', "it's", 'of', 'above', 'aren', 'doesn', 'by', 'wouldn

', 'then', 'him', "she's", 'while', 'with', 'for', 'their', 'to', 'out', 'if

', 'don', 'ourselves', 'in', 'themselves', 'ours', 'his', 'off', "wasn't", 't

he', 'very', 'so', 'are', 'what', 'from', 'there', 'will', 'yourselves', "tha

t'll", 'she', 'some', 'between', "aren't", 'shan', 'who', 'few', 'before', 'w

eren', 'this', 'because', 'both', 'd', 'hers', 'll', 'but', "won't", "don't",

'is', 'and', 'where', 'too', "shouldn't", 'ma', 'these', "hadn't", 'you', "yo

u've", 'down', "needn't", 'or', 'have', 'after', 'only', 'ain', 'own', 'once

', "you'd", 've', 'against', 'y', 'as', 'had', 'haven', 'more', "haven't", 'f

urther', 'no', 'those', "weren't", 'should', 'been', 'when', 'your', 'me', 'a

n', 'having', 'isn', 'm', 'doing', "doesn't", 'our', 'do', "didn't", 're', 'h

adn', 'they', 'into', 'not', 'himself', 'needn', 'most', 'other', 'now', 'a', 'yourself'}

In [60]:

filtered\_sent**=**[]

**for** w **in** tokenizer\_word:

**if** w **not in** stop\_words: filtered\_sent.append(w)

print("Tokenized sentence : ",tokenizer\_word) print("Filtered sentence : ", filtered\_sent)

Tokenized sentence : ['Good', 'Day', 'everyone', ',', 'how', 'are', 'you', '

all', 'today', '?', 'Its', 'fun', 'learning', 'data', 'analysis', '.', 'Hope

', 'you', 'all', 'are', 'practicing', 'well', '.']

Filtered sentence : ['Good', 'Day', 'everyone', ',', 'today', '?', 'Its', 'f

un', 'learning', 'data', 'analysis', '.', 'Hope', 'practicing', 'well', '.']

In [61]:

**from** nltk.stem **import** PorterStemmer

**from** nltk.tokenize **import** sent\_tokenize, word\_tokenize 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 : ['Good', 'Day', 'everyone', ',', 'today', '?', 'Its', 'f

un', 'learning', 'data', 'analysis', '.', 'Hope', 'practicing', 'well', '.']

Stemmed Sentence : ['good', 'day', 'everyon', ',', 'today', '?', 'it', 'fun

', 'learn', 'data', 'analysi', '.', 'hope', 'practic', 'well', '.']

In [62]:

nltk.download('wordnet')

[nltk\_data] Downloading package wordnet to

[nltk\_data] C:\Users\ADMIN\AppData\Roaming\nltk\_data... [nltk\_data] Unzipping corpora\wordnet.zip.

Out[62]: True

In [64]:

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

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

word **=** "flying"

print("Lemmatizer Word : ", lem.lemmatize(word, "v")) print("Stemmed Word : ", stem.stem(word))

Lemmatizer Word : fly Stemmed Word : fli

In [65]:

sent **=** "Albert Einstien was born in Ulm, Germany in 1879." tokens **=** nltk.word\_tokenize(sent)

print(tokens)

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

In [66]:

nltk.download('averaged\_perceptron\_tagger')

[nltk\_data] Downloading package averaged\_perceptron\_tagger to [nltk\_data] C:\Users\ADMIN\AppData\Roaming\nltk\_data... [nltk\_data] Unzipping taggers\averaged\_perceptron\_tagger.zip.

Out[66]: True

In [67]:

nltk.pos\_tag(tokens)

Out[67]: [('Albert', 'NNP'),

('Einstien', 'NNP'),

('was', 'VBD'),

('born', 'VBN'),

('in', 'IN'),

('Ulm', 'NNP'),

(',', ','),

('Germany', 'NNP'),

('in', 'IN'),

('1879', 'CD'),

('.', '.')]

In [68]:

**from** collections **import** Counter

sentence **=** "Texas A&M University is located in Texas" term\_frequency **=** Counter(sentence.split())

In [69]:

term\_frequency

Out[69]: Counter({'Texas': 2,

'A&M': 1,

'University': 1,

'is': 1,

'located': 1,

'in': 1})