In [1]:
```
import os
import nltk
nltk.download()
```

```
showing info https://raw.githubusercontent.com/nltk/nltk_data/gh-pages/index.xml
```

Out[1]: True

In [2]:
```
import nltk.corpus
```

In [3]:
```
print(os.listdir(nltk.data.find('corpora')))
```

```
['abc', 'abc.zip', 'alpino', 'alpino.zip', 'bcp47.zip', 'biocreative_ppi', 'biocr
eative_ppi.zip', 'brown', 'brown.zip', 'brown_tei', 'brown_tei.zip', 'cess_cat',
'cess_cat.zip', 'cess_esp', 'cess_esp.zip', 'chat80', 'chat80.zip', 'city_databas
e', 'city_database.zip', 'cmudict', 'cmudict.zip', 'comparative_sentences', 'comp
arative_sentences.zip', 'comtrans.zip', 'conll2000', 'conll2000.zip', 'conll200
2', 'conll2002.zip', 'conll2007.zip', 'crubadan', 'crubadan.zip', 'dependency_tre
ebank', 'dependency_treebank.zip', 'dolch', 'dolch.zip', 'english_wordnet', 'engl
ish_wordnet.zip', 'europarl_raw', 'europarl_raw.zip', 'extended_omw.zip', 'flores
ta', 'floresta.zip', 'framenet_v15', 'framenet_v15.zip', 'framenet_v17', 'framene
t_v17.zip', 'gazetteers', 'gazetteers.zip', 'genesis', 'genesis.zip', 'gutenber
g', 'gutenberg.zip', 'ieer', 'ieer.zip', 'inaugural', 'inaugural.zip', 'indian',
'indian.zip', 'jeita.zip', 'kimmo', 'kimmo.zip', 'knbc.zip', 'lin_thesaurus', 'li
n_thesaurus.zip', 'machado.zip', 'mac_morpho', 'mac_morpho.zip', 'masc_tagged.zi
p', 'mock_corpus.zip', 'movie_reviews', 'movie_reviews.zip', 'mte_teip5', 'mte_te
ip5.zip', 'names', 'names.zip', 'nombank.1.0.zip', 'nonbreaking_prefixes', 'nonbr
eaking_prefixes.zip', 'nps_chat', 'nps_chat.zip', 'omw-1.4.zip', 'omw.zip', 'opin
ion_lexicon', 'opinion_lexicon.zip', 'panlex_swadesh.zip', 'paradigms', 'paradigm
s.zip', 'pe08', 'pe08.zip', 'pil', 'pil.zip', 'pl196x', 'pl196x.zip', 'ppattach',
'ppattach.zip', 'problem_reports', 'problem_reports.zip', 'product_reviews_1', 'p
roduct_reviews_1.zip', 'product_reviews_2', 'product_reviews_2.zip', 'propbank.zi
p', 'pros_cons', 'pros_cons.zip', 'ptb', 'ptb.zip', 'qc', 'qc.zip', 'reuters.zi
p', 'rte', 'rte.zip', 'semcor.zip', 'senseval', 'senseval.zip', 'sentence_polarit
y', 'sentence_polarity.zip', 'sentiwordnet', 'sentiwordnet.zip', 'shakespeare',
'shakespeare.zip', 'sinica_treebank', 'sinica_treebank.zip', 'smultron', 'smultro
n.zip', 'state_union', 'state_union.zip', 'stopwords', 'stopwords.zip', 'subjecti
vity', 'subjectivity.zip', 'swadesh', 'swadesh.zip', 'switchboard', 'switchboard.
zip', 'timit', 'timit.zip', 'toolbox', 'toolbox.zip', 'treebank', 'treebank.zip',
'twitter_samples', 'twitter_samples.zip', 'udhr', 'udhr.zip', 'udhr2', 'udhr2.zi
p', 'unicode_samples', 'unicode_samples.zip', 'universal_treebanks_v20.zip', 'ver
bnet', 'verbnet.zip', 'verbnet3', 'verbnet3.zip', 'webtext', 'webtext.zip', 'word
net.zip', 'wordnet2021.zip', 'wordnet2022', 'wordnet2022.zip', 'wordnet31.zip',
'wordnet_ic', 'wordnet_ic.zip', 'words', 'words.zip', 'ycoe', 'ycoe.zip']
```

In [4]:
```
AI = '''Artificial Intelligence refers to the intelligence of machines. This is
humans and animals. With Artificial Intelligence, machines perform functions suc
problem-solving. Most noteworthy, Artificial Intelligence is the simulation of h
It is probably the fastest-growing development in the World of technology and in
AI could solve major challenges and crisis situations.'''
```

In [5]:
```
AI
```

Out[5]:
```
'Artificial Intelligence refers to the intelligence of machines. This is in con
trast to the natural intelligence of\nhumans and animals. With Artificial Intel
ligence, machines perform functions such as learning, planning, reasoning and\n
problem-solving. Most noteworthy, Artificial Intelligence is the simulation of
human intelligence by machines.\nIt is probably the fastest-growing development
in the World of technology and innovation. Furthermore, many experts believe\nA
I could solve major challenges and crisis situations.'
```

In [6]:
```
type(AI)
```

Out[6]:
```
str
```

In [7]:
```
from nltk.tokenize import word_tokenize
```

In [8]:
```
AI_tokens= word_tokenize(AI)
AI_tokens
```

Out[8]:
```
['Artificial',
 'Intelligence',
 'refers',
 'to',
 'the',
 'intelligence',
 'of',
 'machines',
 '.',
 'This',
 'is',
 'in',
 'contrast',
 'to',
 'the',
 'natural',
 'intelligence',
 'of',
 'humans',
 'and',
 'animals',
 '.',
 'With',
 'Artificial',
 'Intelligence',
 ',',
 'machines',
 'perform',
 'functions',
 'such',
 'as',
 'learning',
 ',',
 'planning',
 ',',
 'reasoning',
 'and',
 'problem-solving',
 '.',
 'Most',
 'noteworthy',
 ',',
 'Artificial',
 'Intelligence',
 'is',
 'the',
 'simulation',
 'of',
 'human',
 'intelligence',
 'by',
 'machines',
 '.',
 'It',
 'is',
 'probably',
 'the',
 'fastest-growing',
 'development',
 'in',
```

```
 'the',
 'World',
 'of',
 'technology',
 'and',
 'innovation',
 '.',
 'Furthermore',
 ',',
 'many',
 'experts',
 'believe',
 'AI',
 'could',
 'solve',
 'major',
 'challenges',
 'and',
 'crisis',
 'situations',
 '.']
```

In [9]:
```
len(AI_tokens)
```

Out[9]:
```
81
```

In [10]:
```
AI
```

Out[10]:
```
'Artificial Intelligence refers to the intelligence of machines. This is in con
trast to the natural intelligence of\nhumans and animals. With Artificial Intel
ligence, machines perform functions such as learning, planning, reasoning and\n
problem-solving. Most noteworthy, Artificial Intelligence is the simulation of
human intelligence by machines.\nIt is probably the fastest-growing development
in the World of technology and innovation. Furthermore, many experts believe\nA
I could solve major challenges and crisis situations.'
```

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

In [12]:
```
AI_sent=sent_tokenize(AI)
AI_sent
```

Out[12]:
```
['Artificial Intelligence refers to the intelligence of machines.',
 'This is in contrast to the natural intelligence of\nhumans and animals.',
 'With Artificial Intelligence, machines perform functions such as learning, pl
anning, reasoning and\nproblem-solving.',
 'Most noteworthy, Artificial Intelligence is the simulation of human intellige
nce by machines.',
 'It is probably the fastest-growing development in the World of technology and
innovation.',
 'Furthermore, many experts believe\nAI could solve major challenges and crisis
situations.']
```

In [13]:
```
len(AI_sent)
```

Out[13]:
```
6
```

In [14]:
```
AI
```

Out[14]:
```
'Artificial Intelligence refers to the intelligence of machines. This is in con
trast to the natural intelligence of\nhumans and animals. With Artificial Intel
ligence, machines perform functions such as learning, planning, reasoning and\n
problem-solving. Most noteworthy, Artificial Intelligence is the simulation of
human intelligence by machines.\nIt is probably the fastest-growing development
in the World of technology and innovation. Furthermore, many experts believe\nA
I could solve major challenges and crisis situations.'
```

In [15]:
```
from nltk.tokenize import blankline_tokenize
AI_blank=blankline_tokenize(AI)
AI_blank
```

Out[15]:
```
['Artificial Intelligence refers to the intelligence of machines. This is in co
ntrast to the natural intelligence of\nhumans and animals. With Artificial Inte
lligence, machines perform functions such as learning, planning, reasoning and
\nproblem-solving. Most noteworthy, Artificial Intelligence is the simulation o
f human intelligence by machines.\nIt is probably the fastest-growing developme
nt in the World of technology and innovation. Furthermore, many experts believe
\nAI could solve major challenges and crisis situations.']
```

In [16]:
```
len(AI_blank)
```

Out[16]:
```
1
```

In [17]:
```
from nltk.tokenize import WhitespaceTokenizer
wt=WhitespaceTokenizer().tokenize(AI)
wt
```

Out[17]:
```
['Artificial',
 'Intelligence',
 'refers',
 'to',
 'the',
 'intelligence',
 'of',
 'machines.',
 'This',
 'is',
 'in',
 'contrast',
 'to',
 'the',
 'natural',
 'intelligence',
 'of',
 'humans',
 'and',
 'animals.',
 'With',
 'Artificial',
 'Intelligence,',
 'machines',
 'perform',
 'functions',
 'such',
 'as',
 'learning,',
 'planning,',
 'reasoning',
 'and',
 'problem-solving.',
 'Most',
 'noteworthy,',
 'Artificial',
 'Intelligence',
 'is',
 'the',
 'simulation',
 'of',
 'human',
 'intelligence',
 'by',
 'machines.',
 'It',
 'is',
 'probably',
 'the',
 'fastest-growing',
 'development',
 'in',
 'the',
 'World',
 'of',
 'technology',
 'and',
 'innovation.',
 'Furthermore,',
 'many',
```

```
 'experts',
 'believe',
 'AI',
 'could',
 'solve',
 'major',
 'challenges',
 'and',
 'crisis',
 'situations.']
```

In [18]:
```
print(len(wt))
```

```
70
```

In [19]:
```
len(AI_tokens)
```

Out[19]:
```
81
```

In [20]:
```
s='Good apple cost $3.338 in hyderabad. Please buy two of them. Thanks.'
s
```

Out[20]:
```
'Good apple cost $3.338 in hyderabad. Please buy two of them. Thanks.'
```

In [21]:
```
from nltk.tokenize import wordpunct_tokenize
wordpunct_tokenize(s)
```

Out[21]:
```
['Good',
 'apple',
 'cost',
 '$',
 '3',
 '.',
 '338',
 'in',
 'hyderabad',
 '.',
 'Please',
 'buy',
 'two',
 'of',
 'them',
 '.',
 'Thanks',
 '.']
```

In [22]:
```
w_p= wordpunct_tokenize(AI)
w_p
```

Out[22]:
```
['Artificial',
 'Intelligence',
 'refers',
 'to',
 'the',
 'intelligence',
 'of',
 'machines',
 '.',
 'This',
 'is',
 'in',
 'contrast',
 'to',
 'the',
 'natural',
 'intelligence',
 'of',
 'humans',
 'and',
 'animals',
 '.',
 'With',
 'Artificial',
 'Intelligence',
 ',',
 'machines',
 'perform',
 'functions',
 'such',
 'as',
 'learning',
 ',',
 'planning',
 ',',
 'reasoning',
 'and',
 'problem',
 '-',
 'solving',
 '.',
 'Most',
 'noteworthy',
 ',',
 'Artificial',
 'Intelligence',
 'is',
 'the',
 'simulation',
 'of',
 'human',
 'intelligence',
 'by',
 'machines',
 '.',
 'It',
 'is',
 'probably',
 'the',
 'fastest',
```

```
 '-',
 'growing',
 'development',
 'in',
 'the',
 'World',
 'of',
 'technology',
 'and',
 'innovation',
 '.',
 'Furthermore',
 ',',
 'many',
 'experts',
 'believe',
 'AI',
 'could',
 'solve',
 'major',
 'challenges',
 'and',
 'crisis',
 'situations',
 '.']
```

In [23]:
```
len(w_p)
```

Out[23]: 85

In [24]:
```
import nltk
```

In [25]:
```
from nltk.util import bigrams,trigrams,ngrams
```

In [26]:
```
string = 'we are learner of prakash senapathi from 10.30am batch'
quotes_tokens= nltk.word_tokenize(string)
quotes_tokens
```

Out[26]:
```
['we',
 'are',
 'learner',
 'of',
 'prakash',
 'senapathi',
 'from',
 '10.30am',
 'batch']
```

In [27]:
```
string
```

Out[27]: 'we are learner of prakash senapathi from 10.30am batch'

In [28]:
```
quotes_tokens
```

Out[28]:
```
['we',
 'are',
 'learner',
 'of',
 'prakash',
 'senapathi',
 'from',
 '10.30am',
 'batch']
```

In [29]:
```
len(quotes_tokens)
```

Out[29]:
```
9
```

In [30]:
```
quotes_bigrams=list(nltk.bigrams(quotes_tokens))
quotes_bigrams
```

Out[30]:
```
[('we', 'are'),
 ('are', 'learner'),
 ('learner', 'of'),
 ('of', 'prakash'),
 ('prakash', 'senapathi'),
 ('senapathi', 'from'),
 ('from', '10.30am'),
 ('10.30am', 'batch')]
```

In [31]:
```
quotes_trigrams=list(nltk.trigrams(quotes_tokens))
quotes_trigrams
```

Out[31]:
```
[('we', 'are', 'learner'),
 ('are', 'learner', 'of'),
 ('learner', 'of', 'prakash'),
 ('of', 'prakash', 'senapathi'),
 ('prakash', 'senapathi', 'from'),
 ('senapathi', 'from', '10.30am'),
 ('from', '10.30am', 'batch')]
```

In [32]:
```
quotes_ngrams=list(nltk.ngrams(quotes_tokens))
quotes_ngrams
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[32], line 1
----> 1 quotes_ngrams=list(nltk.ngrams(quotes_tokens))
      2 quotes_ngrams

TypeError: ngrams() missing 1 required positional argument: 'n'
```

In [33]:
```
quotes_ngrams=list(nltk.ngrams(quotes_tokens,5))
quotes_ngrams
```

Out[33]:
```
[('we', 'are', 'learner', 'of', 'prakash'),
 ('are', 'learner', 'of', 'prakash', 'senapathi'),
 ('learner', 'of', 'prakash', 'senapathi', 'from'),
 ('of', 'prakash', 'senapathi', 'from', '10.30am'),
 ('prakash', 'senapathi', 'from', '10.30am', 'batch')]
```

In [34]:
```
len(quotes_tokens)
```

Out[34]: 9

In [35]:
```
quotes_ngrams=list(nltk.ngrams(quotes_tokens,8))
quotes_ngrams
```

Out[35]:
```
[('we', 'are', 'learner', 'of', 'prakash', 'senapathi', 'from', '10.30am'),
 ('are', 'learner', 'of', 'prakash', 'senapathi', 'from', '10.30am', 'batch')]
```

In [36]:
```
quotes_ngrams=list(nltk.ngrams(quotes_tokens,10))
quotes_ngrams
```

Out[36]: []

In [37]:
```
from nltk.stem import PorterStemmer
pst= PorterStemmer()
```

In [38]:
```
pst.stem('affection')
```

Out[38]: 'affect'

In [39]:
```
pst.stem('playing')
```

Out[39]: 'play'

In [40]:
```
pst.stem('maximum')
```

Out[40]: 'maximum'

In [41]:
```
words_to_stem=['give','giving','given','gave']
for words in words_to_stem:
    print(words+ ' : ' +pst.stem(words))
```

```
give : give
giving : give
given : given
gave : gave
```

In [42]:
```
words_to_stem=['give','giving','given','gaved','thinking','loving','maximum','pr
for words in words_to_stem:
    print(words+ ' : ' +pst.stem(words))
```

```
give : give
giving : give
given : given
gaved : gave
thinking : think
loving : love
maximum : maximum
priyanka sulaganti : priyanka sulaganti
```

In [43]:
```
from nltk.stem import LancasterStemmer
lst= LancasterStemmer()
for words in words_to_stem:
    print(words+ ' : ' +lst.stem(words))
```

```
give : giv
giving : giv
given : giv
gaved : gav
thinking : think
loving : lov
maximum : maxim
priyanka sulaganti : priyanka sulaganti
```

In [44]:
```
from nltk.stem import SnowballStemmer
sbst=SnowballStemmer('english')
for words in words_to_stem:
    print(words+ ' : ' +sbst.stem(words))
```

```
give : give
giving : give
given : given
gaved : gave
thinking : think
loving : love
maximum : maximum
priyanka sulaganti : priyanka sulaganti
```

In [45]:
```
stemmer=SnowballStemmer("german")
>>> stemmer.stem("Autobahnen")
```

Out[45]: 'autobahn'

In [46]:
```
from nltk.stem import wordnet
from nltk.stem import WordNetLemmatizer
word_lem= WordNetLemmatizer()
```

words_to_stem

In [47]:
```
for words in words_to_stem:
    print(words+ ' : ' +word_lem.lemmatize(words))
```

```
give : give
giving : giving
given : given
gaved : gaved
thinking : thinking
loving : loving
maximum : maximum
priyanka sulaganti : priyanka sulaganti
```

In [48]:
```
from nltk.corpus import stopwords
```

In [49]:
```
stopwords.words('english')
```

Out[49]:
```
['a',
 'about',
 'above',
 'after',
 'again',
 'against',
 'ain',
 'all',
 'am',
 'an',
 'and',
 'any',
 'are',
 'aren',
 "aren't",
 'as',
 'at',
 'be',
 'because',
 'been',
 'before',
 'being',
 'below',
 'between',
 'both',
 'but',
 'by',
 'can',
 'couldn',
 "couldn't",
 'd',
 'did',
 'didn',
 "didn't",
 'do',
 'does',
 'doesn',
 "doesn't",
 'doing',
 'don',
 "don't",
 'down',
 'during',
 'each',
 'few',
 'for',
 'from',
 'further',
 'had',
 'hadn',
 "hadn't",
 'has',
 'hasn',
 "hasn't",
 'have',
 'haven',
 "haven't",
 'having',
 'he',
 "he'd",
```

```
"he'll",
'her',
'here',
'hers',
'herself',
"he's",
'him',
'himself',
'his',
'how',
'i',
"i'd",
'if',
"i'll",
"i'm",
'in',
'into',
'is',
'isn',
"isn't",
'it',
"it'd",
"it'll",
"it's",
'its',
'itself',
"i've",
'just',
'll',
'm',
'ma',
'me',
'mightn',
"mightn't",
'more',
'most',
'mustn',
"mustn't",
'my',
'myself',
'needn',
"needn't",
'no',
'nor',
'not',
'now',
'o',
'of',
'off',
'on',
'once',
'only',
'or',
'other',
'our',
'ours',
'ourselves',
'out',
'over',
'own',
```

```
're',
's',
'same',
'shan',
"shan't",
'she',
"she'd",
"she'll",
"she's",
'should',
'shouldn',
"shouldn't",
"should've",
'so',
'some',
'such',
't',
'than',
'that',
"that'll",
'the',
'their',
'theirs',
'them',
'themselves',
'then',
'there',
'these',
'they',
"they'd",
"they'll",
"they're",
"they've",
'this',
'those',
'through',
'to',
'too',
'under',
'until',
'up',
've',
'very',
'was',
'wasn',
"wasn't",
'we',
"we'd",
"we'll",
"we're",
'were',
'weren',
"weren't",
"we've",
'what',
'when',
'where',
'which',
'while',
'who',
```

```
 'whom',
 'why',
 'will',
 'with',
 'won',
 "won't",
 'wouldn',
 "wouldn't",
 'y',
 'you',
 "you'd",
 "you'll",
 'your',
 "you're",
 'yours',
 'yourself',
 'yourselves',
 "you've"]
```

In [50]:
```
len(stopwords.words('english'))
```

Out[50]: 198

In [ ]:
```

```