In [1]:
```
pip install nltk
```

```
Requirement already satisfied: nltk in c:\users\samua\anaconda3\lib\site-packages (3.9.1)
Requirement already satisfied: click in c:\users\samua\anaconda3\lib\site-packages (from nltk) (8.1.8)
Requirement already satisfied: joblib in c:\users\samua\anaconda3\lib\site-packages (from nltk) (1.4.2)
Requirement already satisfied: regex>=2021.8.3 in c:\users\samua\anaconda3\lib\site-packages (from nltk) (2024.11.6)
Requirement already satisfied: tqdm in c:\users\samua\anaconda3\lib\site-packages (from nltk) (4.67.1)
Requirement already satisfied: colorama in c:\users\samua\anaconda3\lib\site-packages (from click->nltk) (0.4.6)
Note: you may need to restart the kernel to use updated packages.
```

In [2]:
```
import os
import nltk
nltk.download()
```

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

Out[2]:
```
True
```

In [3]:
```
#import nltk
#nltk.data.path.append("/path/to/nltk_data")
#nltk.download('punkt', download_dir="/path/to/nltk_data")
```

In [4]:
```
import nltk.corpus
```

In [5]:
```
# we will see what is mean by corpora and what all are availabel in nltk python library
print(os.listdir(nltk.data.find('corpora')))

#you get a lot of file, some of have some textual document, different function associated with that function
#for our example i will lets take consideration as brown & we will understand what exactly nlp can do
```

```
['abc', 'abc.zip', 'alpino', 'alpino.zip', 'bcp47.zip', 'biocreative_ppi', 'biocreative_ppi.zip', 'brown', 'brown.zip', 'brown_
tei', 'brown_tei.zip', 'cess_cat', 'cess_cat.zip', 'cess_esp', 'cess_esp.zip', 'chat80', 'chat80.zip', 'city_database', 'city_d
atabase.zip', 'cmudict', 'cmudict.zip', 'comparative_sentences', 'comparative_sentences.zip', 'comtrans.zip', 'conll2000', 'con
ll2000.zip', 'conll2002', 'conll2002.zip', 'conll2007.zip', 'crubadan', 'crubadan.zip', 'dependency_treebank', 'dependency_tree
bank.zip', 'dolch', 'dolch.zip', 'english_wordnet', 'english_wordnet.zip', 'europarl_raw', 'europarl_raw.zip', 'extended_omw.zi
p', 'floresta', 'floresta.zip', 'framenet_v15', 'framenet_v15.zip', 'framenet_v17', 'framenet_v17.zip', 'gazetteers', 'gazettee
rs.zip', 'genesis', 'genesis.zip', 'gutenberg', 'gutenberg.zip', 'ieer', 'ieer.zip', 'inaugural', 'inaugural.zip', 'indian', 'i
ndian.zip', 'jeita.zip', 'kimmo', 'kimmo.zip', 'knbc.zip', 'lin_thesaurus', 'lin_thesaurus.zip', 'machado.zip', 'mac_morpho',
'mac_morpho.zip', 'masc_tagged.zip', 'mock_corpus', 'mock_corpus.zip', 'movie_reviews', 'movie_reviews.zip', 'mte_teip5', 'mte_
teip5.zip', 'names', 'names.zip', 'nombank.1.0.zip', 'nonbreaking_prefixes', 'nonbreaking_prefixes.zip', 'nps_chat', 'nps_chat.
zip', 'omw-1.4.zip', 'omw.zip', 'opinion_lexicon', 'opinion_lexicon.zip', 'panlex_swadesh.zip', 'paradigms', 'paradigms.zip',
'pe08', 'pe08.zip', 'pil', 'pil.zip', 'pl196x', 'pl196x.zip', 'ppattach', 'ppattach.zip', 'problem_reports', 'problem_reports.z
ip', 'product_reviews_1', 'product_reviews_1.zip', 'product_reviews_2', 'product_reviews_2.zip', 'propbank.zip', 'pros_cons',
'pros_cons.zip', 'ptb', 'ptb.zip', 'qc', 'qc.zip', 'reuters.zip', 'rte', 'rte.zip', 'semcor.zip', 'senseval', 'senseval.zip',
'sentence_polarity', 'sentence_polarity.zip', 'sentiwordnet', 'sentiwordnet.zip', 'shakespeare', 'shakespeare.zip', 'sinica_tre
ebank', 'sinica_treebank.zip', 'smultron', 'smultron.zip', 'state_union', 'state_union.zip', 'stopwords', 'stopwords.zip', 'sub
jectivity', 'subjectivity.zip', 'swadesh', 'swadesh.zip', 'switchboard', 'switchboard.zip', 'timit', 'timit.zip', 'toolbox', 't
oolbox.zip', 'treebank', 'treebank.zip', 'twitter_samples', 'twitter_samples.zip', 'udhr', 'udhr.zip', 'udhr2', 'udhr2.zip', 'u
nicode_samples', 'unicode_samples.zip', 'universal_treebanks_v20.zip', 'verbnet', 'verbnet.zip', 'verbnet3', 'verbnet3.zip', 'w
ebtext', 'webtext.zip', 'wordnet.zip', 'wordnet2021.zip', 'wordnet2022', 'wordnet2022.zip', 'wordnet31.zip', 'wordnet_ic', 'wor
dnet_ic.zip', 'words', 'words.zip', 'ycoe', 'ycoe.zip']
```

In [6]:
```python
from nltk.corpus import brown
brown.words()
```

Out[6]:
```
['The', 'Fulton', 'County', 'Grand', 'Jury', 'said', ...]
```

In [7]:
```python
nltk.corpus.brown.fileids()
```

```
Out[7]: ['ca01',
         'ca02',
         'ca03',
         'ca04',
         'ca05',
         'ca06',
         'ca07',
         'ca08',
         'ca09',
         'ca10',
         'ca11',
         'ca12',
         'ca13',
         'ca14',
         'ca15',
         'ca16',
         'ca17',
         'ca18',
         'ca19',
         'ca20',
         'ca21',
         'ca22',
         'ca23',
         'ca24',
         'ca25',
         'ca26',
         'ca27',
         'ca28',
         'ca29',
         'ca30',
         'ca31',
         'ca32',
         'ca33',
         'ca34',
         'ca35',
         'ca36',
         'ca37',
         'ca38',
         'ca39',
         'ca40',
                 'ca41',
                 'ca42',
                 'ca43',
                 'ca44',
                 'cb01',
                 'cb02',
                 'cb03',
                 'cb04',
                 'cb05',
                 'cb06',
                 'cb07',
                 'cb08',
                 'cb09',
                 'cb10',
                 'cb11',
                 'cb12',
                 'cb13',
                 'cb14',
                 'cb15',
                 'cb16',
                 'cb17',
                 'cb18',
                 'cb19',
                 'cb20',
                 'cb21',
                 'cb22',
                 'cb23',
                 'cb24',
                 'cb25',
                 'cb26',
                 'cb27',
                 'cc01',
                 'cc02',
                 'cc03',
                 'cc04',
                 'cc05',
                 'cc06',
                 'cc07',
                 'cc08',
                 'cc09',
                 'cc10',
                 'cc11',
                 'cc12',
                 'cc13',
                 'cc14',
                 'cc15',
                 'cc16',
                 'cc17',
                 'cd01',
                 'cd02',
                 'cd03',
                 'cd04',
                 'cd05',
                 'cd06',
                 'cd07',
                 'cd08',
                 'cd09',
                 'cd10',
                 'cd11',
                 'cd12',
                 'cd13',
                 'cd14',
                 'cd15',
                 'cd16',
                 'cd17',
                 'ce01',
                 'ce02',
                 'ce03',
                 'ce04',
                 'ce05',
                 'ce06',
                 'ce07',
                 'ce08',
                 'ce09',
                 'ce10',
                 'ce11',
                 'ce12',
                 'ce13',
                 'ce14',
                 'ce15',
                 'ce16',
                 'ce17',
                 'ce18',
                 'ce19',
                 'ce20',
                 'ce21',
                 'ce22',
                 'ce23',
                 'ce24',
                 'ce25',
                 'ce26',
                 'ce27',
                 'ce28',
                 'ce29',
                 'ce30',
                 'ce31',
                 'ce32',
                 'ce33',
                 'ce34',
                 'ce35',
                 'ce36',
                 'cf01',
                 'cf02',
                 'cf03',
                 'cf04',
                 'cf05',
                 'cf06',
                 'cf07',
                 'cf08',
                 'cf09',
                 'cf10',
                 'cf11',
                 'cf12',
                 'cf13',
                 'cf14',
                 'cf15',
                 'cf16',
                 'cf17',
                 'cf18',
                 'cf19',
                 'cf20',
                 'cf21',
                 'cf22',
                 'cf23',
                 'cf24',
                 'cf25',
                 'cf26',
                 'cf27',
                 'cf28',
                 'cf29',
                 'cf30',
                 'cf31',
                 'cf32',
                 'cf33',
                 'cf34',
                 'cf35',
                 'cf36',
                 'cf37',
                 'cf38',
                 'cf39',
                 'cf40',
                 'cf41',
                 'cf42',
                 'cf43',
                 'cf44',
                 'cf45',
                 'cf46',
                 'cf47',
                 'cf48',
                 'cg01',
                 'cg02',
                 'cg03',
                 'cg04',
                 'cg05',
                 'cg06',
                 'cg07',
                 'cg08',
                 'cg09',
                 'cg10',
                 'cg11',
                 'cg12',
                 'cg13',
                 'cg14',
                 'cg15',
                 'cg16',
                 'cg17',
                 'cg18',
                 'cg19',
                 'cg20',
                 'cg21',
                 'cg22',
                 'cg23',
                 'cg24',
                 'cg25',
                 'cg26',
                 'cg27',
                 'cg28',
                 'cg29',
                 'cg30',
                 'cg31',
                 'cg32',
                 'cg33',
                 'cg34',
                 'cg35',
                 'cg36',
                 'cg37',
                 'cg38',
                 'cg39',
                 'cg40',
                 'cg41',
                 'cg42',
                 'cg43',
                 'cg44',
                 'cg45',
                 'cg46',
                 'cg47',
                 'cg48',
                 'cg49',
                 'cg50',
                 'cg51',
                 'cg52',
                 'cg53',
                 'cg54',
                 'cg55',
                 'cg56',
                 'cg57',
                 'cg58',
                 'cg59',
                 'cg60',
                 'cg61',
                 'cg62',
                 'cg63',
                 'cg64',
                 'cg65',
                 'cg66',
                 'cg67',
                 'cg68',
                 'cg69',
                 'cg70',
                 'cg71',
                 'cg72',
                 'cg73',
                 'cg74',
                 'cg75',
                 'ch01',
                 'ch02',
                 'ch03',
                 'ch04',
                 'ch05',
                 'ch06',
                 'ch07',
                 'ch08',
                 'ch09',
                 'ch10',
                 'ch11',
                 'ch12',
                 'ch13',
                 'ch14',
                 'ch15',
                 'ch16',
                 'ch17',
                 'ch18',
                 'ch19',
                 'ch20',
                 'ch21',
                 'ch22',
                 'ch23',
                 'ch24',
                 'ch25',
                 'ch26',
                 'ch27',
                 'ch28',
                 'ch29',
                 'ch30',
                 'cj01',
                 'cj02',
                 'cj03',
                 'cj04',
                 'cj05',
                 'cj06',
                 'cj07',
                 'cj08',
                 'cj09',
                 'cj10',
                 'cj11',
                 'cj12',
                 'cj13',
                 'cj14',
                 'cj15',
                 'cj16',
                 'cj17',
                 'cj18',
                 'cj19',
                 'cj20',
                 'cj21',
                 'cj22',
                 'cj23',
                 'cj24',
                 'cj25',
                 'cj26',
                 'cj27',
                 'cj28',
                 'cj29',
                 'cj30',
                 'cj31',
                 'cj32',
                 'cj33',
                 'cj34',
                 'cj35',
                 'cj36',
                 'cj37',
                 'cj38',
                 'cj39',
                 'cj40',
                 'cj41',
                 'cj42',
                 'cj43',
                 'cj44',
                 'cj45',
                 'cj46',
                 'cj47',
                 'cj48',
                 'cj49',
                 'cj50',
                 'cj51',
                 'cj52',
                 'cj53',
                 'cj54',
                 'cj55',
                 'cj56',
                 'cj57',
                 'cj58',
                 'cj59',
                 'cj60',
                 'cj61',
                 'cj62',
                 'cj63',
                 'cj64',
                 'cj65',
                 'cj66',
                 'cj67',
                 'cj68',
                 'cj69',
                 'cj70',
                 'cj71',
                 'cj72',
                 'cj73',
                 'cj74',
                 'cj75',
                 'cj76',
                 'cj77',
                 'cj78',
                 'cj79',
                 'cj80',
                 'ck01',
                 'ck02',
                 'ck03',
                 'ck04',
                 'ck05',
                 'ck06',
                 'ck07',
                 'ck08',
                 'ck09',
                 'ck10',
                 'ck11',
                 'ck12',
                 'ck13',
                 'ck14',
                 'ck15',
                 'ck16',
                 'ck17',
                 'ck18',
                 'ck19',
                 'ck20',
                 'ck21',
                 'ck22',
                 'ck23',
                 'ck24',
                 'ck25',
                 'ck26',
                 'ck27',
                 'ck28',
                 'ck29',
                 'cl01',
                 'cl02',
                 'cl03',
                 'cl04',
                 'cl05',
                 'cl06',
                 'cl07',
                 'cl08',
                 'cl09',
                 'cl10',
                 'cl11',
                 'cl12',
                 'cl13',
                 'cl14',
                 'cl15',
                 'cl16',
                 'cl17',
                 'cl18',
                 'cl19',
                 'cl20',
                 'cl21',
                 'cl22',
                 'cl23',
                 'cl24',
                 'cm01',
                 'cm02',
                 'cm03',
                 'cm04',
                 'cm05',
                 'cm06',
                 'cn01',
                 'cn02',
                 'cn03',
                 'cn04',
                 'cn05',
                 'cn06',
                 'cn07',
                 'cn08',
                 'cn09',
                 'cn10',
                 'cn11',
                 'cn12',
                 'cn13',
                 'cn14',
                 'cn15',
                 'cn16',
                 'cn17',
                 'cn18',
                 'cn19',
                 'cn20',
                 'cn21',
                 'cn22',
                 'cn23',
                 'cn24',
                 'cn25',
                 'cn26',
                 'cn27',
                 'cn28',
                 'cn29',
                 'cp01',
                 'cp02',
                 'cp03',
                 'cp04',
                 'cp05',
                 'cp06',
                 'cp07',
                 'cp08',
                 'cp09',
                 'cp10',
                 'cp11',
                 'cp12',
                 'cp13',
                 'cp14',
                 'cp15',
                 'cp16',
                 'cp17',
                 'cp18',
                 'cp19',
                 'cp20',
                 'cp21',
                 'cp22',
                 'cp23',
                 'cp24',
                 'cp25',
                 'cp26',
                 'cp27',
                 'cp28',
                 'cp29',
                 'cr01',
                 'cr02',
                 'cr03',
                 'cr04',
                 'cr05',
                 'cr06',
                 'cr07',
                 'cr08',
                 'cr09']
```

In [8]:
```
nltk.corpus.gutenberg
```

Out[8]:
```
<PlaintextCorpusReader in 'C:\\Users\\samua\\AppData\\Roaming\\nltk_data\\corpora\\gutenberg'>
```

In [9]:
```
nltk.corpus.gutenberg.fileids()
```

Out[9]:
```
['austen-emma.txt',
 'austen-persuasion.txt',
 'austen-sense.txt',
 'bible-kjv.txt',
 'blake-poems.txt',
 'bryant-stories.txt',
 'burgess-busterbrown.txt',
 'carroll-alice.txt',
 'chesterton-ball.txt',
 'chesterton-brown.txt',
 'chesterton-thursday.txt',
 'edgeworth-parents.txt',
 'melville-moby_dick.txt',
 'milton-paradise.txt',
 'shakespeare-caesar.txt',
 'shakespeare-hamlet.txt',
 'shakespeare-macbeth.txt',
 'whitman-leaves.txt']
```

In [10]:
```
# you can also create your own words
AI = '''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 [11]:
```
AI
```

Out[11]:
```
'Artificial Intelligence refers to the intelligence of machines. This is in contrast to the natural intelligence of\nhumans a
nd animals. With Artificial Intelligence, machines perform functions such as learning, planning, reasoning and\nproblem-solvi
ng. 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\nAI could solve major chall
enges and crisis situations.'
```

In [12]:
```
type(AI)
```

Out[12]:
```
str
```

In [13]:
```
from nltk.tokenize import word_tokenize
```

In [15]:
```
AI_tokens = word_tokenize(AI)
AI_tokens
```

```
Out[15]: ['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 [17]:
```
len(AI_tokens)
```
Out[17]:
```
81
```
In [16]:
```
AI
```
Out[16]:
```
'Artificial Intelligence refers to the intelligence of machines. This is in contrast to the natural intelligence of\nhumans a
nd animals. With Artificial Intelligence, machines perform functions such as learning, planning, reasoning and\nproblem-solvi
ng. 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\nAI could solve major chall
enges and crisis situations.'
```
In [18]:
```
from nltk.tokenize import sent_tokenize
```
In [19]:
```
AI_sent = sent_tokenize(AI)
AI_sent
```
Out[19]:
```
['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, planning, reasoning and\nproblem-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\nAI could solve major challenges and crisis situations.']
```
In [20]:
```
len(AI_sent)
```
Out[20]:
```
6
```
In [21]:
```
AI
```
Out[21]:
```
'Artificial Intelligence refers to the intelligence of machines. This is in contrast to the natural intelligence of\nhumans a
nd animals. With Artificial Intelligence, machines perform functions such as learning, planning, reasoning and\nproblem-solvi
ng. 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\nAI could solve major chall
enges and crisis situations.'
```
In [22]:
```
from nltk.tokenize import blankline_tokenize # Give you how many paragraph
AI_blank = blankline_tokenize(AI)
```

```
AI_blank
#AI_blank
```

Out[22]:
```
['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, planning, reasoning and\nproblem-solv
ing. Most noteworthy, Artificial Intelligence is the simulation of human intelligence by machines.\nIt is probably the fastes
t-growing development in the World of technology and innovation. Furthermore, many experts believe\nAI could solve major chal
lenges and crisis situations.']
```

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

Out[23]:
```
1
```

In [24]:
```
from nltk.tokenize import WhitespaceTokenizer
wt = WhitespaceTokenizer().tokenize(AI)
wt
```

```
Out[24]: ['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 [25]:
```python
print(len(wt))
```

```
70
```

In [26]:
```python
len(AI_tokens)
```

Out[26]:
```
81
```

In [27]:
```python
s = 'Good apple cost $3.88 in hyderbad. Please buy two of them. Thanks.'
```

```
s
```

Out[27]:
```
'Good apple cost $3.88 in hyderbad. Please buy two of them. Thanks.'
```

In [28]:
```
from nltk.tokenize import wordpunct_tokenize
wordpunct_tokenize(s)
```

Out[28]:
```
['Good',
 'apple',
 'cost',
 '$',
 '3',
 '.',
 '88',
 'in',
 'hyderbad',
 '.',
 'Please',
 'buy',
 'two',
 'of',
 'them',
 '.',
 'Thanks',
 '.']
```

In [29]:
```
w_p = wordpunct_tokenize(AI)
w_p
```

```
Out[29]: ['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 [30]:
```
len(w_p)
```

Out[30]:
```
85
```

In [31]:
```
import nltk
```

In [32]:
```
# NEXT WE WILL SEE HOW WE WILL USE UNI-GRAM,BI-GRAM,TRI-GRAM USING NLTK
from nltk.util import bigrams,trigrams,ngrams
```

In [38]:
```
string = 'we are student of prakash senapati from 530pm batch'
quotes_tokens = nltk.word_tokenize(string)
quotes_tokens
```

Out[38]:
```
['we', 'are', 'student', 'of', 'prakash', 'senapati', 'from', '530pm', 'batch']
```

In [39]:
```
string
```

Out[39]:
```
'we are student of prakash senapati from 530pm batch'
```

In [40]:
```
quotes_tokens
```

Out[40]:
```
['we', 'are', 'student', 'of', 'prakash', 'senapati', 'from', '530pm', 'batch']
```

In [37]:
```
len(quotes_tokens)
```

Out[37]:
```
9
```

In [41]:
```
quotes_bigrams = list(nltk.bigrams(quotes_tokens))
quotes_bigrams
```

```
Out[41]: [('we', 'are'),
          ('are', 'student'),
          ('student', 'of'),
          ('of', 'prakash'),
          ('prakash', 'senapati'),
          ('senapati', 'from'),
          ('from', '530pm'),
          ('530pm', 'batch')]
```

```
In [42]: quotes_tokens
```

```
Out[42]: ['we', 'are', 'student', 'of', 'prakash', 'senapati', 'from', '530pm', 'batch']
```

```
In [43]: quotes_trigrams = list(nltk.trigrams(quotes_tokens))
         quotes_trigrams
```

```
Out[43]: [('we', 'are', 'student'),
          ('are', 'student', 'of'),
          ('student', 'of', 'prakash'),
          ('of', 'prakash', 'senapati'),
          ('prakash', 'senapati', 'from'),
          ('senapati', 'from', '530pm'),
          ('from', '530pm', 'batch')]
```

```
In [44]: quotes_ngrams = list(nltk.ngrams(quotes_tokens))
         quotes_ngrams
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[44], line 1
----> 1 quotes_ngrams = list(nltk.ngrams(quotes_tokens))
      2 quotes_ngrams

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

```
In [45]: quotes_ngrams = list(nltk.ngrams(quotes_tokens, 4))
         quotes_ngrams
         # it has given n-gram of length 4
```

```
Out[45]: [('we', 'are', 'student', 'of'),
          ('are', 'student', 'of', 'prakash'),
          ('student', 'of', 'prakash', 'senapati'),
          ('of', 'prakash', 'senapati', 'from'),
          ('prakash', 'senapati', 'from', '530pm'),
          ('senapati', 'from', '530pm', 'batch')]
```

```
In [46]: len(quotes_tokens)
```

```
Out[46]: 9
```

```
In [47]: quotes_ngrams_1 = list(nltk.ngrams(quotes_tokens, 8))
         quotes_ngrams_1
```

```
Out[47]: [('we', 'are', 'student', 'of', 'prakash', 'senapati', 'from', '530pm'),
          ('are', 'student', 'of', 'prakash', 'senapati', 'from', '530pm', 'batch')]
```

```
In [49]: from nltk.stem import PorterStemmer
         pst = PorterStemmer()
```

```
In [50]: pst.stem('affection')
```

```
Out[50]: 'affect'
```

```
In [51]: pst.stem('playing')
```

```
Out[51]: 'play'
```

```
In [52]: pst.stem('maximum')
```

```
Out[52]: 'maximum'
```

```
In [53]: 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 [56]:
```
words_to_stem=['give','giving','given','graved','thinking', 'loving','maximum','samsonkadarikota']
# i am giving these different words to stem, using porter stemmer we get the output

for words in words_to_stem:
    print(words+ ' : ' + pst.stem(words))
```

```
give : give
giving : give
given : given
graved : grave
thinking : think
loving : love
maximum : maximum
samsonkadarikota : samsonkadarikota
```

In [57]:
```
from nltk.stem import LancasterStemmer
lst = LancasterStemmer()

for words in words_to_stem:
    print(words+ ' : ' + lst.stem(words))
```

```
give : giv
giving : giv
given : giv
graved : grav
thinking : think
loving : lov
maximum : maxim
samsonkadarikota : samsonkadarikot
```

In [58]:
```
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
graved : grave
thinking : think
loving : love
maximum : maximum
samsonkadarikota : samsonkadarikota
```

In [67]:
```
stemmer = SnowballStemmer("german") # Choose a language
>>> stemmer.stem("Autobahnen") # Stem a word
```

Out[67]:
```
'autobahn'
```

In [ ]: