necessary libraries

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

```
showing info https://raw.githubusercontent.com/nltk/nltk_data/gh-pages/index.xml
Out[1]: True
```

Loading the sample text

```
In [2]: 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 [3]: AI
```

```
Out[3]: '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 [4]: type(AI)
```

```
Out[4]: str
```

```
In [5]: from nltk.tokenize import word_tokenize
```

Converting Paragraph to Word Tokens

```
In [6]: AI_tokens=word_tokenize(AI)
        AI_tokens
```

```
Out[6]: ['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 [7]:
```
len(AI_tokens)
```

Out[7]:
```
81
```

In [8]:
```
from nltk.tokenize import import sent_tokenize
```

Sentence Tokenization

In [10]:
```
AI_sent=sent_tokenize(AI)
AI_sent
```

Out[10]:
```
['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 [11]:
```
len(AI_sent)
```

Out[11]:
```
6
```

In [12]:
```
AI
```

Out[12]:
```
'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.'
```

Blankline Tokination

In [14]:
```
from nltk.tokenize import blankline_tokenize
AI_blank=blankline_tokenize(AI)
AI_blank
```

Out[14]:
```
['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 [15]:
```
len(AI_blank)
```

Out[15]:
```
1
```

- Tokenization consist of three part like
- 1] Bigrams == Tokens of two consecutive written words.
- 2] Trigrams == Tokens of three consecutive written words.
- 3] Ngrams == Tokens of more then three consecutive written words.

In [16]:
```
from nltk.util import bigrams, trigrams, ngrams
```

In [17]:
```
string = 'The best and most beautiful thing in the world cannot been seen or eve
quotes_tokens=word_tokenize(string)
```

In [19]:
```
quotes_tokens
```

Out[19]:
```
['The',
 'best',
 'and',
 'most',
 'beautiful',
 'thing',
 'in',
 'the',
 'world',
 'can',
 'not',
 'been',
 'seen',
 'or',
 'even',
 'touched',
 ',',
 'they',
 'must',
 'be',
 'felt',
 'with',
 'heart']
```

In [20]:
```
len(quotes_tokens)
```

Out[20]:
```
23
```

In [22]:
```
quotes_bigram=list(nltk.bigrams(quotes_tokens))
```

In [23]:
```
quotes_bigram
```

Out[23]:
```
[('The', 'best'),
 ('best', 'and'),
 ('and', 'most'),
 ('most', 'beautiful'),
 ('beautiful', 'thing'),
 ('thing', 'in'),
 ('in', 'the'),
 ('the', 'world'),
 ('world', 'can'),
 ('can', 'not'),
 ('not', 'been'),
 ('been', 'seen'),
 ('seen', 'or'),
 ('or', 'even'),
 ('even', 'touched'),
 ('touched', ','),
 (',', 'they'),
 ('they', 'must'),
 ('must', 'be'),
 ('be', 'felt'),
 ('felt', 'with'),
 ('with', 'heart')]
```

In [24]:
```
quotes_trigram=list(nltk.trigrams(quotes_tokens))
```

In [25]:
```
quotes_trigram
```

Out[25]:
```
[('The', 'best', 'and'),
 ('best', 'and', 'most'),
 ('and', 'most', 'beautiful'),
 ('most', 'beautiful', 'thing'),
 ('beautiful', 'thing', 'in'),
 ('thing', 'in', 'the'),
 ('in', 'the', 'world'),
 ('the', 'world', 'can'),
 ('world', 'can', 'not'),
 ('can', 'not', 'been'),
 ('not', 'been', 'seen'),
 ('been', 'seen', 'or'),
 ('seen', 'or', 'even'),
 ('or', 'even', 'touched'),
 ('even', 'touched', ','),
 ('touched', ',', 'they'),
 (',', 'they', 'must'),
 ('they', 'must', 'be'),
 ('must', 'be', 'felt'),
 ('be', 'felt', 'with'),
 ('felt', 'with', 'heart')]
```

In [27]:
```
quotes_ngram=list(nltk.ngrams(quotes_tokens))
```

```
---------------------------------------------------------------------------
TypeError                                 Traceback (most recent call last)
Cell In[27], line 1
----> 1 quotes_ngram=list(nltk.ngrams(quotes_tokens))

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

In [28]:
```
quotes_ngram=list(nltk.ngrams(quotes_tokens,4))
```

In [29]:
```
quotes_ngram
```

Out[29]:
```
[('The', 'best', 'and', 'most'),
 ('best', 'and', 'most', 'beautiful'),
 ('and', 'most', 'beautiful', 'thing'),
 ('most', 'beautiful', 'thing', 'in'),
 ('beautiful', 'thing', 'in', 'the'),
 ('thing', 'in', 'the', 'world'),
 ('in', 'the', 'world', 'can'),
 ('the', 'world', 'can', 'not'),
 ('world', 'can', 'not', 'been'),
 ('can', 'not', 'been', 'seen'),
 ('not', 'been', 'seen', 'or'),
 ('been', 'seen', 'or', 'even'),
 ('seen', 'or', 'even', 'touched'),
 ('or', 'even', 'touched', ','),
 ('even', 'touched', ',', 'they'),
 ('touched', ',', 'they', 'must'),
 (',', 'they', 'must', 'be'),
 ('they', 'must', 'be', 'felt'),
 ('must', 'be', 'felt', 'with'),
 ('be', 'felt', 'with', 'heart')]
```

In [30]:
```
len(quotes_tokens)
```

Out[30]:
```
23
```

Stemming

- Normalize the words into its root form.
- there are three types of stemming
- 1] Porterstemmer == It reduces words to their root form
- 2] Lancasterstemmer == It cuts words down to their root form
- 3] Snowballstemmer == It is same like Porterstemmer

In [35]:
```
# we need to make some changes in token is called stemming .stemming give you ro
from nltk.stem import PorterStemmer
pst = PorterStemmer()
```

In [36]:
```
pst.stem('Having')  #it give the root form
```

Out[36]:
```
'have'
```

In [37]:
```
pst.stem('affection')
```

```
Out[37]:  'affect'
```

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

```
Out[39]:  'play'
```

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

```
Out[40]:  'give'
```

```
In [41]: pst.stem('gave')
```

```
Out[41]:  'gave'
```

```
In [43]: word_to_stem =['give','giving','given','gave']
         for words in word_to_stem:
             print(words+ ' : ' +pst.stem(words))
```

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

```
In [45]: words_to_stem=['give','gave','given','giving','thinking','playing','loving','fin
         for words in words_to_stem:
             print(words+ ' : ' +pst.stem(words))
```

```
give : give
gave : gave
given : given
giving : give
thinking : think
playing : play
loving : love
final : final
maximun : maximun
finally : final
```

Lancasterstemmer

```
In [47]: from nltk.stem import LancasterStemmer
         lst =LancasterStemmer()
```

```
In [50]: for words in words_to_stem:
             print(words+ ' : ' +lst.stem(words))
```

```
give : giv
gave : gav
given : giv
giving : giv
thinking : think
playing : play
loving : lov
final : fin
maximun : maximun
finally : fin
```

snowballstemmer

In [52]:
```
from nltk.stem import SnowballStemmer
snt = SnowballStemmer('english')
```

In [53]:
```
for words in words_to_stem:
    print(words+ ' : ' +snt.stem(words))
```

```
give : give
gave : gave
given : given
giving : give
thinking : think
playing : play
loving : love
final : final
maximun : maximun
finally : final
```

Lemmatization

- lemmatize it's gives real words

In [55]:
```
from nltk.stem import WordNetLemmatizer
wnl = WordNetLemmatizer()
```

In [56]:
```
words_to_stem
```

Out[56]:
```
['give',
 'gave',
 'given',
 'giving',
 'thinking',
 'playing',
 'loving',
 'final',
 'maximun',
 'finally']
```

In [57]:
```
for words in words_to_stem:
    print(words+ ' : ' +wnl.lemmatize(words))
```

```
give : give
gave : gave
given : given
giving : giving
thinking : thinking
playing : playing
loving : loving
final : final
maximun : maximun
finally : finally
```

Stopwords

- stopwords is a common word that is usually ignored or removed during text preprocessing because it doesn't carry important meaning.

In [58]:
```
from nltk.corpus import stopwords
```

In [59]:
```
stopwords.words('english')
```

```
Out[59]: ['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 [61]:
```
len(stopwords.words('english'))
```

Out[61]:
```
198
```

In [62]:
```
stopwords.words('spanish')
```

```
Out[62]:  ['de',
           'la',
           'que',
           'el',
           'en',
           'y',
           'a',
           'los',
           'del',
           'se',
           'las',
           'por',
           'un',
           'para',
           'con',
           'no',
           'una',
           'su',
           'al',
           'lo',
           'como',
           'más',
           'pero',
           'sus',
           'le',
           'ya',
           'o',
           'este',
           'sí',
           'porque',
           'esta',
           'entre',
           'cuando',
           'muy',
           'sin',
           'sobre',
           'también',
           'me',
           'hasta',
           'hay',
           'donde',
           'quien',
           'desde',
           'todo',
           'nos',
           'durante',
           'todos',
           'uno',
           'les',
           'ni',
           'contra',
           'otros',
           'ese',
           'eso',
           'ante',
           'ellos',
           'e',
           'esto',
           'mí',
           'antes',
                    'algunos',
                    'qué',
                    'unos',
                    'yo',
                    'otro',
                    'otras',
                    'otra',
                    'él',
                    'tanto',
                    'esa',
                    'estos',
                    'mucho',
                    'quienes',
                    'nada',
                    'muchos',
                    'cual',
                    'poco',
                    'ella',
                    'estar',
                    'estas',
                    'algunas',
                    'algo',
                    'nosotros',
                    'mi',
                    'mis',
                    'tú',
                    'te',
                    'ti',
                    'tu',
                    'tus',
                    'ellas',
                    'nosotras',
                    'vosotros',
                    'vosotras',
                    'os',
                    'mío',
                    'mía',
                    'míos',
                    'mías',
                    'tuyo',
                    'tuya',
                    'tuyos',
                    'tuyas',
                    'suyo',
                    'suya',
                    'suyos',
                    'suyas',
                    'nuestro',
                    'nuestra',
                    'nuestros',
                    'nuestras',
                    'vuestro',
                    'vuestra',
                    'vuestros',
                    'vuestras',
                    'esos',
                    'esas',
                    'estoy',
                    'estás',
                    'está',
                    'estamos',
                    'estáis',
                    'están',
                    'esté',
                    'estés',
                    'estemos',
                    'estéis',
                    'estén',
                    'estaré',
                    'estarás',
                    'estará',
                    'estaremos',
                    'estaréis',
                    'estarán',
                    'estaría',
                    'estarías',
                    'estaríamos',
                    'estaríais',
                    'estarían',
                    'estaba',
                    'estabas',
                    'estábamos',
                    'estabais',
                    'estaban',
                    'estuve',
                    'estuviste',
                    'estuvo',
                    'estuvimos',
                    'estuvisteis',
                    'estuvieron',
                    'estuviera',
                    'estuvieras',
                    'estuviéramos',
                    'estuvierais',
                    'estuvieran',
                    'estuviese',
                    'estuvieses',
                    'estuviésemos',
                    'estuvieseis',
                    'estuviesen',
                    'estando',
                    'estado',
                    'estada',
                    'estados',
                    'estadas',
                    'estad',
                    'he',
                    'has',
                    'ha',
                    'hemos',
                    'habéis',
                    'han',
                    'haya',
                    'hayas',
                    'hayamos',
                    'hayáis',
                    'hayan',
                    'habré',
                    'habrás',
                    'habrá',
                    'habremos',
                    'habréis',
                    'habrán',
                    'habría',
                    'habrías',
                    'habríamos',
                    'habríais',
                    'habrían',
                    'había',
                    'habías',
                    'habíamos',
                    'habíais',
                    'habían',
                    'hube',
                    'hubiste',
                    'hubo',
                    'hubimos',
                    'hubisteis',
                    'hubieron',
                    'hubiera',
                    'hubieras',
                    'hubiéramos',
                    'hubierais',
                    'hubieran',
                    'hubiese',
                    'hubieses',
                    'hubiésemos',
                    'hubieseis',
                    'hubiesen',
                    'habiendo',
                    'habido',
                    'habida',
                    'habidos',
                    'habidas',
                    'soy',
                    'eres',
                    'es',
                    'somos',
                    'sois',
                    'son',
                    'sea',
                    'seas',
                    'seamos',
                    'seáis',
                    'sean',
                    'seré',
                    'serás',
                    'será',
                    'seremos',
                    'seréis',
                    'serán',
                    'sería',
                    'serías',
                    'seríamos',
                    'seríais',
                    'serían',
                    'era',
                    'eras',
                    'éramos',
                    'erais',
                   'eran',
                   'fui',
                   'fuiste',
                   'fue',
                   'fuimos',
                   'fuisteis',
                   'fueron',
                   'fuera',
                   'fueras',
                   'fuéramos',
                   'fuerais',
                   'fueran',
                   'fuese',
                   'fueses',
                   'fuésemos',
                   'fueseis',
                   'fuesen',
                   'sintiendo',
                   'sentido',
                   'sentida',
                   'sentidos',
                   'sentidas',
                   'siente',
                   'sentid',
                   'tengo',
                   'tienes',
                   'tiene',
                   'tenemos',
                   'tenéis',
                   'tienen',
                   'tenga',
                   'tengas',
                   'tengamos',
                   'tengáis',
                   'tengan',
                   'tendré',
                   'tendrás',
                   'tendrá',
                   'tendremos',
                   'tendréis',
                   'tendrán',
                   'tendría',
                   'tendrías',
                   'tendríamos',
                   'tendríais',
                   'tendrían',
                   'tenía',
                   'tenías',
                   'teníamos',
                   'teníais',
                   'tenían',
                   'tuve',
                   'tuviste',
                   'tuvo',
                   'tuvimos',
                   'tuvisteis',
                   'tuvieron',
                   'tuviera',
                   'tuvieras',
                   'tuviéramos',
'tuvierais',
'tuvieran',
'tuviese',
'tuvieses',
'tuviésemos',
'tuvieseis',
'tuviesen',
'teniendo',
'tenido',
'tenida',
'tenidos',
'tenidas',
'tened']
```

In [63]:
```
len(stopwords.words('spanish'))
```

Out[63]:
```
313
```

In [64]:
```
stopwords.words('french')
```

```
Out[64]: ['au',
          'aux',
          'avec',
          'ce',
          'ces',
          'dans',
          'de',
          'des',
          'du',
          'elle',
          'en',
          'et',
          'eux',
          'il',
          'ils',
          'je',
          'la',
          'le',
          'les',
          'leur',
          'lui',
          'ma',
          'mais',
          'me',
          'même',
          'mes',
          'moi',
          'mon',
          'ne',
          'nos',
          'notre',
          'nous',
          'on',
          'ou',
          'par',
          'pas',
          'pour',
          'qu',
          'que',
          'qui',
          'sa',
          'se',
          'ses',
          'son',
          'sur',
          'ta',
          'te',
          'tes',
          'toi',
          'ton',
          'tu',
          'un',
          'une',
          'vos',
          'votre',
          'vous',
          'c',
          'd',
          'j',
          'l',
          'à',
          'm',
          'n',
          's',
          't',
          'y',
          'été',
          'étée',
          'étées',
          'étés',
          'étant',
          'étante',
          'étants',
          'étantes',
          'suis',
          'es',
          'est',
          'sommes',
          'êtes',
          'sont',
          'serai',
          'seras',
          'sera',
          'serons',
          'serez',
          'seront',
          'serais',
          'serait',
          'serions',
          'seriez',
          'seraient',
          'étais',
          'était',
          'étions',
          'étiez',
          'étaient',
          'fus',
          'fut',
          'fûmes',
          'fûtes',
          'furent',
          'sois',
          'soit',
          'soyons',
          'soyez',
          'soient',
          'fusse',
          'fusses',
          'fût',
          'fussions',
          'fussiez',
          'fussent',
          'ayant',
          'ayante',
          'ayantes',
          'ayants',
          'eu',
          'eue',
          'eues',
          'eus',
'ai',
'as',
'avons',
'avez',
'ont',
'aurai',
'auras',
'aura',
'aurons',
'aurez',
'auront',
'aurais',
'aurait',
'aurions',
'auriez',
'auraient',
'avais',
'avait',
'avions',
'aviez',
'avaient',
'eut',
'eûmes',
'eûtes',
'eurent',
'aie',
'aies',
'ait',
'ayons',
'ayez',
'aient',
'eusse',
'eusses',
'eût',
'eussions',
'eussiez',
'eussent']
```

In [66]:
```
len(stopwords.words('french'))
```

Out[66]:
```
157
```

In [67]:
```
stopwords.words('german')
```

```
Out[67]: ['aber',
          'alle',
          'allem',
          'allen',
          'aller',
          'alles',
          'als',
          'also',
          'am',
          'an',
          'ander',
          'andere',
          'anderem',
          'anderen',
          'anderer',
          'anderes',
          'anderm',
          'andern',
          'anderr',
          'anders',
          'auch',
          'auf',
          'aus',
          'bei',
          'bin',
          'bis',
          'bist',
          'da',
          'damit',
          'dann',
          'der',
          'den',
          'des',
          'dem',
          'die',
          'das',
          'dass',
          'daß',
          'derselbe',
          'derselben',
          'denselben',
          'desselben',
          'demselben',
          'dieselbe',
          'dieselben',
          'dasselbe',
          'dazu',
          'dein',
          'deine',
          'deinem',
          'deinen',
          'deiner',
          'deines',
          'denn',
          'derer',
          'dessen',
          'dich',
          'dir',
          'du',
          'dies',
                      'diese',
                      'diesem',
                      'diesen',
                      'dieser',
                      'dieses',
                      'doch',
                      'dort',
                      'durch',
                      'ein',
                      'eine',
                      'einem',
                      'einen',
                      'einer',
                      'eines',
                      'einig',
                      'einige',
                      'einigem',
                      'einigen',
                      'einiger',
                      'einiges',
                      'einmal',
                      'er',
                      'ihn',
                      'ihm',
                      'es',
                      'etwas',
                      'euer',
                      'eure',
                      'eurem',
                      'euren',
                      'eurer',
                      'eures',
                      'für',
                      'gegen',
                      'gewesen',
                      'hab',
                      'habe',
                      'haben',
                      'hat',
                      'hatte',
                      'hatten',
                      'hier',
                      'hin',
                      'hinter',
                      'ich',
                      'mich',
                      'mir',
                      'ihr',
                      'ihre',
                      'ihrem',
                      'ihren',
                      'ihrer',
                      'ihres',
                      'euch',
                      'im',
                      'in',
                      'indem',
                      'ins',
                      'ist',
                      'jede',
                      'jedem',
                      'jeden',
                      'jeder',
                      'jedes',
                      'jene',
                      'jenem',
                      'jenen',
                      'jener',
                      'jenes',
                      'jetzt',
                      'kann',
                      'kein',
                      'keine',
                      'keinem',
                      'keinen',
                      'keiner',
                      'keines',
                      'können',
                      'könnte',
                      'machen',
                      'man',
                      'manche',
                      'manchem',
                      'manchen',
                      'mancher',
                      'manches',
                      'mein',
                      'meine',
                      'meinem',
                      'meinen',
                      'meiner',
                      'meines',
                      'mit',
                      'muss',
                      'musste',
                      'nach',
                      'nicht',
                      'nichts',
                      'noch',
                      'nun',
                      'nur',
                      'ob',
                      'oder',
                      'ohne',
                      'sehr',
                      'sein',
                      'seine',
                      'seinem',
                      'seinen',
                      'seiner',
                      'seines',
                      'selbst',
                      'sich',
                      'sie',
                      'ihnen',
                      'sind',
                      'so',
                      'solche',
                      'solchem',
                      'solchen',
 'solcher',
 'solches',
 'soll',
 'sollte',
 'sondern',
 'sonst',
 'über',
 'um',
 'und',
 'uns',
 'unsere',
 'unserem',
 'unseren',
 'unser',
 'unseres',
 'unter',
 'viel',
 'vom',
 'von',
 'vor',
 'während',
 'war',
 'waren',
 'warst',
 'was',
 'weg',
 'weil',
 'weiter',
 'welche',
 'welchem',
 'welchen',
 'welcher',
 'welches',
 'wenn',
 'werde',
 'werden',
 'wie',
 'wieder',
 'will',
 'wir',
 'wird',
 'wirst',
 'wo',
 'wollen',
 'wollte',
 'würde',
 'würden',
 'zu',
 'zum',
 'zur',
 'zwar',
 'zwischen']
```

```
In [68]: len(stopwords.words('german'))
```

```
Out[68]: 232
```

```
In [70]: stopwords.words('hindi')
```

```
---------------------------------------------------------------------------
OSError                                   Traceback (most recent call last)
Cell In[70], line 1
----> 1 stopwords.words('hindi')

File C:\ProgramData\anaconda3\Lib\site-packages\nltk\corpus\reader\wordlist.py:2
1, in WordListCorpusReader.words(self, fileids, ignore_lines_startswith)
     18 def words(self, fileids=None, ignore_lines_startswith="\n"):
     19     return [
     20         line
---> 21         for line in line_tokenize(self.raw(fileids))
     22         if not line.startswith(ignore_lines_startswith)
     23     ]

File C:\ProgramData\anaconda3\Lib\site-packages\nltk\corpus\reader\api.py:218, in
CorpusReader.raw(self, fileids)
    216 contents = []
    217 for f in fileids:
--> 218     with self.open(f) as fp:
    219         contents.append(fp.read())
    220 return concat(contents)

File C:\ProgramData\anaconda3\Lib\site-packages\nltk\corpus\reader\api.py:231, in
CorpusReader.open(self, file)
    223 """
    224 Return an open stream that can be used to read the given file.
    225 If the file's encoding is not None, then the stream will
   (...)
    228 :param file: The file identifier of the file to read.
    229 """
    230 encoding = self.encoding(file)
--> 231 stream = self._root.join(file).open(encoding)
    232 return stream

File C:\ProgramData\anaconda3\Lib\site-packages\nltk\data.py:333, in FileSystemPa
thPointer.join(self, fileid)
    331 def join(self, fileid):
    332     _path = os.path.join(self._path, fileid)
--> 333     return FileSystemPathPointer(_path)

File C:\ProgramData\anaconda3\Lib\site-packages\nltk\data.py:311, in FileSystemPa
thPointer.__init__(self, _path)
    309 _path = os.path.abspath(_path)
    310 if not os.path.exists(_path):
--> 311     raise OSError("No such file or directory: %r" % _path)
    312 self._path = _path

OSError: No such file or directory: 'C:\\Users\\ritika\\AppData\\Roaming\\nltk_da
ta\\corpora\\stopwords\\hindi'
```

In [71]:
```
AI
```

Out[71]:
```
'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 [72]: import re
         punc=re.compile(r'[-.?!,:,(),|0-9]')
```

```
In [73]: punc
```

```
Out[73]: re.compile(r'[-.?!,:,(),|0-9]', re.UNICODE)
```

```
In [74]: AI
```

```
Out[74]: '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 [75]: AI_tokens
```

```
Out[75]: ['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 [76]:
```
len(AI_tokens)
```

Out[76]:
```
81
```

POS[part of speech]

- It talk always gramatically type of words called verb,adjective,proverb.

In [78]:
```
sent = 'kathy is natural when its come to drawing'
sent_tokens=word_tokenize(sent)
sent_tokens
```

Out[78]:
```
['kathy', 'is', 'natural', 'when', 'its', 'come', 'to', 'drawing']
```

In [81]:
```
for tokens in sent_tokens:
    print(nltk.pos_tag([tokens]))
```

```
[('kathy', 'NN')]
[('is', 'VBZ')]
[('natural', 'JJ')]
[('when', 'WRB')]
[('its', 'PRP$')]
[('come', 'VB')]
[('to', 'TO')]
[('drawing', 'VBG')]
```

In [82]:
```
sent2 = 'john is eating delicious cake'
sent2_tokens=word_tokenize(sent2)
for tokens in sent2_tokens:
    print(nltk.pos_tag([tokens]))
```

```
[('john', 'NN')]
[('is', 'VBZ')]
[('eating', 'VBG')]
[('delicious', 'JJ')]
[('cake', 'NN')]
```

In [ ]:
```
- chunk = chunking means the group of word into chunk
```

In [83]:
```
from nltk import ne_chunk
```

In [84]:
```
NE_sent = 'The US president stay in the WHITEHOUSE'
```

In [86]:
```
NE_tokens=word_tokenize(NE_sent)
NE_tokens
```

Out[86]:
```
['The', 'US', 'president', 'stay', 'in', 'the', 'WHITEHOUSE']
```

In [87]:
```
NE_tag = nltk.pos_tag(NE_tokens)
NE_tag
```

Out[87]:
```
[('The', 'DT'),
 ('US', 'NNP'),
 ('president', 'NN'),
 ('stay', 'NN'),
 ('in', 'IN'),
 ('the', 'DT'),
 ('WHITEHOUSE', 'NNP')]
```

In [93]:
```
new = 'the big cat ate the small mouse who was after fresh cheese'
new_token=nltk.pos_tag(word_tokenize(new))
new_token
```

Out[93]:
```
[('the', 'DT'),
 ('big', 'JJ'),
 ('cat', 'NN'),
 ('ate', 'VBD'),
 ('the', 'DT'),
 ('small', 'JJ'),
 ('mouse', 'NN'),
 ('who', 'WP'),
 ('was', 'VBD'),
 ('after', 'IN'),
 ('fresh', 'JJ'),
 ('cheese', 'NN')]
```

In [94]:
```
from wordcloud import WordCloud
import matplotlib.pyplot as plt
```

In [95]:
```
text=('pyhton java javascript go ruby swift kotlin rust Dart typescript Nodejs r
```

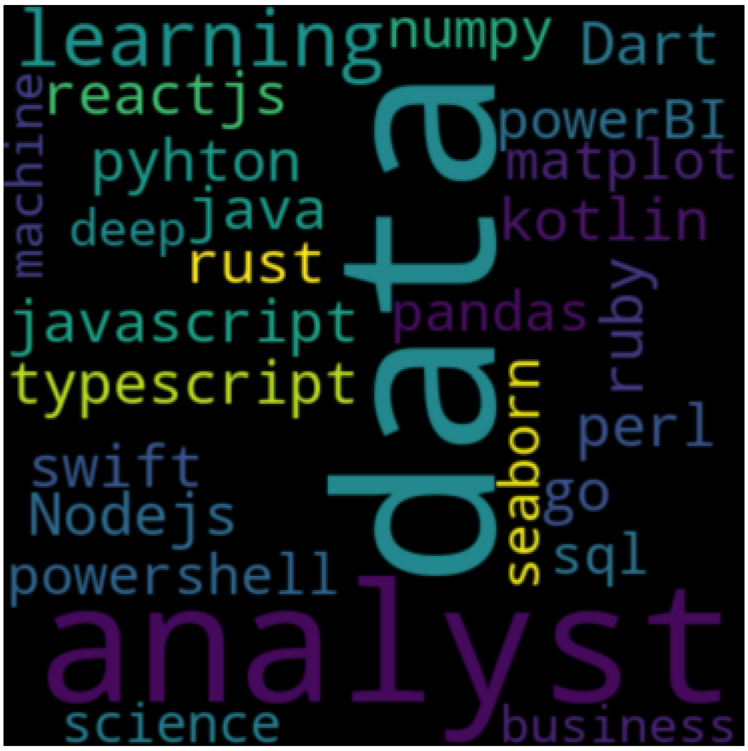
In [96]:
```
text
```

Out[96]:
```
'pyhton java javascript go ruby swift kotlin rust Dart typescript Nodejs reactj
s perl powershell powerBI sql matplot pandas seaborn numpy data science data an
alyst business analyst deep learning machine learning'
```

In [101…
```
#create the object
wordcloud = WordCloud(height=400,width=400,margin=1, background_color='black', m
```

In [104…

```
# Display the generated image:
plt.imshow(wordcloud, interpolation='bicubic')
plt.axis("off")
plt.margins(x=0, y=0)
plt.show()
```


learning
numpy
Dart
reactjs
powerBI
machine
pyhton
matplot
data
deep
java
kotlin
rust
javascript
pandas
ruby
typescript
perl
swift
seaborn
go
Nodejs
sql
powershell
analyst
science
business