In [ ]:
```
#AKSHATA JADHAV
#TE-A 24
#Batch -TA1-TA2
```

In [8]:
```
import pandas as pd
```

In [10]:
```
import nltk
```

In [12]:
```
from nltk.tokenize import sent_tokenize
```

In [14]:
```
text="In literary theory, a text is any object that can be , whether this object is
```

In [17]:
```
sentence=sent_tokenize(text)
```

In [16]:
```
nltk.download('punkt')
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\Akshata\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping tokenizers\punkt.zip.
```

Out[16]:
```
True
```

In [18]:
```
sentence
```

Out[18]:
```
['In literary theory, a text is any object that can be , whether this object is a
work of literature, a street sign, an arrangement of buildings on a city block, or
styles of clothing.',
 'It is a set of signs that is available to be reconstructed by a reader if suffic
ient interpretants are available']
```

In [65]:
```
from nltk.tokenize import word_tokenize
```

In [ ]:
```

```

In [33]:
```
nltk.download('words')
```

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

Out[33]:
```
True
```

In [35]:
```
words = nltk.word_tokenize(text)
```

In [36]:
```
words
```

Out[36]:
```
['In',
 'literary',
 'theory',
 ',',
 'a',
 'text',
 'is',
 'any',
 'object',
 'that',
 'can',
 'be',
 ',',
 'whether',
 'this',
 'object',
 'is',
 'a',
 'work',
 'of',
 'literature',
 ',',
 'a',
 'street',
 'sign',
 ',',
 'an',
 'arrangement',
 'of',
 'buildings',
 'on',
 'a',
 'city',
 'block',
 ',',
 'or',
 'styles',
 'of',
 'clothing',
 '.',
 'It',
 'is',
 'a',
 'set',
 'of',
 'signs',
 'that',
 'is',
 'available',
 'to',
 'be',
 'reconstructed',
 'by',
 'a',
 'reader',
 'if',
 'sufficient',
 'interpretants',
 'are',
 'available']
```

In [37]:

In [64]:
```
pos=pos_tag(words)
```

In [44]:
```
nltk.download('averaged_perceptron_tagger')
```

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

Out[44]:
```
True
```

In [45]:
```
from nltk import pos_tag
```

In [46]:
```
pos=pos_tag(words)
```

In [47]:
```
pos
```

Out[47]:
```
[('In', 'IN'),
 ('literary', 'JJ'),
 ('theory', 'NN'),
 (',', ','),
 ('a', 'DT'),
 ('text', 'NN'),
 ('is', 'VBZ'),
 ('any', 'DT'),
 ('object', 'JJ'),
 ('that', 'WDT'),
 ('can', 'MD'),
 ('be', 'VB'),
 (',', ','),
 ('whether', 'IN'),
 ('this', 'DT'),
 ('object', 'NN'),
 ('is', 'VBZ'),
 ('a', 'DT'),
 ('work', 'NN'),
 ('of', 'IN'),
 ('literature', 'NN'),
 (',', ','),
 ('a', 'DT'),
 ('street', 'NN'),
 ('sign', 'NN'),
 (',', ','),
 ('an', 'DT'),
 ('arrangement', 'NN'),
 ('of', 'IN'),
 ('buildings', 'NNS'),
 ('on', 'IN'),
 ('a', 'DT'),
 ('city', 'NN'),
 ('block', 'NN'),
 (',', ','),
 ('or', 'CC'),
 ('styles', 'NNS'),
 ('of', 'IN'),
 ('clothing', 'NN'),
 ('.', '.'),
 ('It', 'PRP'),
 ('is', 'VBZ'),
 ('a', 'DT'),
 ('set', 'NN'),
 ('of', 'IN'),
 ('signs', 'NNS'),
 ('that', 'WDT'),
 ('is', 'VBZ'),
 ('available', 'JJ'),
 ('to', 'TO'),
 ('be', 'VB'),
 ('reconstructed', 'VBN'),
 ('by', 'IN'),
 ('a', 'DT'),
 ('reader', 'NN'),
 ('if', 'IN'),
 ('sufficient', 'JJ'),
 ('interpretants', 'NNS'),
 ('are', 'VBP'),
 ('available', 'JJ')]
```

In [52]:
```
from nltk.corpus import stopwords
nltk.download('stopwords')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Akshata\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping corpora\stopwords.zip.
```

Out[52]:
```
True
```

In [53]:
```
stop_words=set(stopwords.words("English"))
```

In [54]:
```
stop_words
```

Out[54]:
```
{'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',
 'her',
 'here',
 'hers',
 'herself',
 'him',
```

```
 'himself',
 'his',
 'how',
 'i',
 'if',
 'in',
 'into',
 'is',
 'isn',
 "isn't",
 'it',
 "it's",
 'its',
 'itself',
 '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's",
 'should',
 "should've",
 'shouldn',
 "shouldn't",
 'so',
 'some',
 'such',
 't',
 'than',
 'that',
```

```
"that'll",
'the',
'their',
'theirs',
'them',
'themselves',
'then',
'there',
'these',
'they',
'this',
'those',
'through',
'to',
'too',
'under',
'until',
'up',
've',
'very',
'was',
'wasn',
"wasn't",
'we',
'were',
'weren',
"weren't",
'what',
'when',
'where',
'which',
'while',
'who',
'whom',
'why',
'will',
'with',
'won',
"won't",
'wouldn',
"wouldn't",
'y',
'you',
"you'd",
"you'll",
"you're",
"you've",
'your',
'yours',
'yourself',
'yourselves'}
```

In [55]:
```
filtered_tokens=[word for word in words if word.lower() not in stop_words ]
```

In [56]:
```
filtered_tokens
```

Out[56]:
```
['literary',
 'theory',
 ',',
 'text',
 'object',
 ',',
 'whether',
 'object',
 'work',
 'literature',
 ',',
 'street',
 'sign',
 ',',
 'arrangement',
 'buildings',
 'city',
 'block',
 ',',
 'styles',
 'clothing',
 '.',
 'set',
 'signs',
 'available',
 'reconstructed',
 'reader',
 'sufficient',
 'interpretants',
 'available']
```

In [59]:
```
from nltk.stem import PorterStemmer
```

In [60]:
```
Stemmer=PorterStemmer()
```

In [62]:
```
Stemmer_tokens=[Stemmer.stem(word) for word in filtered_tokens]
```

In [63]:
```
Stemmer_tokens
```

Out[63]:
```
['literari',
 'theori',
 ',',
 'text',
 'object',
 ',',
 'whether',
 'object',
 'work',
 'literatur',
 ',',
 'street',
 'sign',
 ',',
 'arrang',
 'build',
 'citi',
 'block',
 ',',
 'style',
 'cloth',
 '.',
 'set',
 'sign',
 'avail',
 'reconstruct',
 'reader',
 'suffici',
 'interpret',
 'avail']
```

In [80]:
```
from nltk import WordNetLemmatizer
```

In [81]:
```
Lemmatizer=WordNetLemmatizer()
```

In [82]:
```
Lemmatized_tokens = [Lemmatizer.lemmatize(word) for word in filtered_tokens]
```

In [79]:
```
nltk.download('wordnet')
```

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

Out[79]:
```
True
```

In [83]:
```
Lemmatized_tokens
```

Out[83]:
```
['literary',
 'theory',
 ',',
 'text',
 'object',
 ',',
 'whether',
 'object',
 'work',
 'literature',
 ',',
 'street',
 'sign',
 ',',
 'arrangement',
 'building',
 'city',
 'block',
 ',',
 'style',
 'clothing',
 '.',
 'set',
 'sign',
 'available',
 'reconstructed',
 'reader',
 'sufficient',
 'interpretants',
 'available']
```

In [ ]: