In [8]:
```
import nltk
nltk.download('stopwords')
nltk.download('punkt')
nltk.download('wordnet')
nltk.download('averaged_perceptron_tagger')
```

```
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\Apurva\AppData\Roaming\nltk_data...
[nltk_data]   Package stopwords is already up-to-date!
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\Apurva\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\Apurva\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]     C:\Users\Apurva\AppData\Roaming\nltk_data...
[nltk_data]   Package averaged_perceptron_tagger is already up-to-
[nltk_data]       date!
```

Out[8]: True

In [9]:
```
import nltk
```

In [10]:
```
para="Rajgad(literal meaning Ruling Fort) is a hill fort situated in the pune district of maharashtra,India.For
```

In [11]:
```
print(para)
```

```
Rajgad(literal meaning Ruling Fort) is a hill fort situated in the pune district of maharashtra,India.Formerly k
nown as Murumdev,the fort was the capital of Maratha empire under the rule of Chatrapati Shivaji Maharaj for alm
ost 26 years,after which the capital moved to the Raigad Fort.[1]Treasures discovered from an adjacent fort call
ed Torna were used to completely built and fortify the rajgad fort.
```

In [12]:
```
para.split()
```

Out[12]:
```
['Rajgad(literal',
 'meaning',
 'Ruling',
 'Fort)',
 'is',
 'a',
 'hill',
 'fort',
 'situated',
 'in',
 'the',
 'pune',
 'district',
 'of',
 'maharashtra,India.Formerly',
 'known',
 'as',
 'Murumdev,the',
 'fort',
 'was',
 'the',
 'capital',
 'of',
 'Maratha',
 'empire',
 'under',
 'the',
 'rule',
 'of',
 'Chatrapati',
 'Shivaji',
 'Maharaj',
 'for',
 'almost',
 '26',
 'years,after',
 'which',
 'the',
 'capital',
 'moved',
 'to',
 'the',
 'Raigad',
 'Fort.[1]Treasures',
 'discovered',
 'from',
 'an',
 'adjacent',
 'fort',
 'called',
 'Torna',
 'were',
 'used',
 'to',
 'completely',
 'built',
 'and',
 'fortify',
 'the',
 'rajgad',
 'fort.']
```

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

In [14]:
```
import nltk
nltk.download('punkt')
```

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

Out[14]: True

In [15]:
```
import nltk
nltk.download('punkt_tab')
```

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

Out[15]: True

In [16]:
```
sent=sent_tokenize(para)
```

```
sent[1]
```

Out[16]:
```
'[1]Treasures discovered from an adjacent fort called Torna were used to completely built and fortify the rajga
d fort.'
```

In [17]:
```
words=word_tokenize(para)
words
```

Out[17]:
```
['Rajgad',
 '(',
 'literal',
 'meaning',
 'Ruling',
 'Fort',
 ')',
 'is',
 'a',
 'hill',
 'fort',
 'situated',
 'in',
 'the',
 'pune',
 'district',
 'of',
 'maharashtra',
 ',',
 'India.Formerly',
 'known',
 'as',
 'Murumdev',
 ',',
 'the',
 'fort',
 'was',
 'the',
 'capital',
 'of',
 'Maratha',
 'empire',
 'under',
 'the',
 'rule',
 'of',
 'Chatrapati',
 'Shivaji',
 'Maharaj',
 'for',
 'almost',
 '26',
 'years',
 ',',
 'after',
 'which',
 'the',
 'capital',
 'moved',
 'to',
 'the',
 'Raigad',
 'Fort',
 '.',
 '[',
 '1',
 ']',
 'Treasures',
 'discovered',
 'from',
 'an',
 'adjacent',
 'fort',
 'called',
 'Torna',
 'were',
 'used',
 'to',
 'completely',
 'built',
 'and',
 'fortify',
 'the',
 'rajgad',
 'fort',
 '.']
```

In [18]:
```
from nltk.corpus import stopwords
```

In [19]:
```
swords=stopwords.words('english')
swords
```

Out[19]:
```
['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 [20]:

```
x=[word for word in words if word not in swords]
x
```

Out[20]:
```
['Rajgad',
 '(',
 'literal',
 'meaning',
 'Ruling',
 'Fort',
 ')',
 'hill',
 'fort',
 'situated',
 'pune',
 'district',
 'maharashtra',
 ',',
 'India.Formerly',
 'known',
 'Murumdev',
 ',',
 'fort',
 'capital',
 'Maratha',
 'empire',
 'rule',
 'Chatrapati',
 'Shivaji',
 'Maharaj',
 'almost',
 '26',
 'years',
 ',',
 'capital',
 'moved',
 'Raigad',
 'Fort',
 '.',
 '[',
 '1',
 ']',
 'Treasures',
 'discovered',
 'adjacent',
 'fort',
 'called',
 'Torna',
 'used',
 'completely',
 'built',
 'fortify',
 'rajgad',
 'fort',
 '.']
```

In [21]:
```
x=[word for word in words if word.lower() not in swords]
x
```

Out[21]:
```
['Rajgad',
 '(',
 'literal',
 'meaning',
 'Ruling',
 'Fort',
 ')',
 'hill',
 'fort',
 'situated',
 'pune',
 'district',
 'maharashtra',
 ',',
 'India.Formerly',
 'known',
 'Murumdev',
 ',',
 'fort',
 'capital',
 'Maratha',
 'empire',
 'rule',
 'Chatrapati',
 'Shivaji',
 'Maharaj',
 'almost',
 '26',
 'years',
 ',',
 'capital',
 'moved',
 'Raigad',
 'Fort',
 '.',
 '[',
 '1',
 ']',
 'Treasures',
 'discovered',
 'adjacent',
 'fort',
 'called',
 'Torna',
 'used',
 'completely',
 'built',
 'fortify',
 'rajgad',
 'fort',
 '.']
```

In [22]:
```
from nltk.stem import PorterStemmer
ps=PorterStemmer()
ps.stem('working')
```

Out[22]:
```
'work'
```

In [23]:
```
y=[ps.stem(word) for word in x]
y
```

Out[23]:
```
['rajgad',
 '(',
 'liter',
 'mean',
 'rule',
 'fort',
 ')',
 'hill',
 'fort',
 'situat',
 'pune',
 'district',
 'maharashtra',
 ',',
 'india.formerli',
 'known',
 'murumdev',
 ',',
 'fort',
 'capit',
 'maratha',
 'empir',
 'rule',
 'chatrapati',
 'shivaji',
 'maharaj',
 'almost',
 '26',
 'year',
 ',',
 'capit',
 'move',
 'raigad',
 'fort',
 '.',
 '[',
 '1',
 ']',
 'treasur',
 'discov',
 'adjac',
 'fort',
 'call',
 'torna',
 'use',
 'complet',
 'built',
 'fortifi',
 'rajgad',
 'fort',
 '.']
```

In [24]:
```
from nltk.stem import WordNetLemmatizer
wnl=WordNetLemmatizer()
```

In [25]:
```
nltk.download('omw-1.4')
```

```
[nltk_data] Downloading package omw-1.4 to
[nltk_data]     C:\Users\Apurva\AppData\Roaming\nltk_data...
```

Out[25]:
```
True
```

In [30]:
```
wnl.lemmatize('working',pos='v')
```

Out[30]:
```
'work'
```

In [31]:
```
print(ps.stem('went'))         # Output: 'went'  → Stemming doesn't change it.
print(wnl.lemmatize('went', pos='v'))  # Output: 'go'  → Lemmatization gives correct base verb.
```

```
went
go
```

In [32]:
```
z=[wnl.lemmatize(word,pos='v') for word in x]
z
```

Out[32]:
```
['Rajgad',
 '(',
 'literal',
 'mean',
 'Ruling',
 'Fort',
 ')',
 'hill',
 'fort',
 'situate',
 'pune',
 'district',
 'maharashtra',
 ',',
 'India.Formerly',
 'know',
 'Murumdev',
 ',',
 'fort',
 'capital',
 'Maratha',
 'empire',
 'rule',
 'Chatrapati',
 'Shivaji',
 'Maharaj',
 'almost',
 '26',
 'years',
 ',',
 'capital',
 'move',
 'Raigad',
 'Fort',
 '.',
 '[',
 '1',
 ']',
 'Treasures',
 'discover',
 'adjacent',
 'fort',
 'call',
 'Torna',
 'use',
 'completely',
 'build',
 'fortify',
 'rajgad',
 'fort',
 '.']
```

In [33]:
```
import string
```

In [34]:
```
string.punctuation
```

Out[34]:
```
'!"#$%&\'()*+,-./:;<=>?@[\\]^_`{|}~'
```

In [35]:
```
t=[word for word in words if word not in string.punctuation]
t
```

Out[35]:
```
['Rajgad',
 'literal',
 'meaning',
 'Ruling',
 'Fort',
 'is',
 'a',
 'hill',
 'fort',
 'situated',
 'in',
 'the',
 'pune',
 'district',
 'of',
 'maharashtra',
 'India.Formerly',
 'known',
 'as',
 'Murumdev',
 'the',
 'fort',
 'was',
 'the',
 'capital',
 'of',
 'Maratha',
 'empire',
 'under',
 'the',
 'rule',
 'of',
 'Chatrapati',
 'Shivaji',
 'Maharaj',
 'for',
 'almost',
 '26',
 'years',
 'after',
 'which',
 'the',
 'capital',
 'moved',
 'to',
 'the',
 'Raigad',
 'Fort',
 '1',
 'Treasures',
 'discovered',
 'from',
 'an',
 'adjacent',
 'fort',
 'called',
 'Torna',
 'were',
 'used',
 'to',
 'completely',
 'built',
 'and',
 'fortify',
 'the',
 'rajgad',
 'fort']
```

In [37]:
```
from nltk import pos_tag
import nltk
nltk.download('averaged_perceptron_tagger_eng')
```

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

Out[37]:
```
True
```

In [38]:
```
pos_tag(t)
```

Out[38]:
```
[('Rajgad', 'NNP'),
 ('literal', 'JJ'),
 ('meaning', 'NN'),
 ('Ruling', 'NNP'),
 ('Fort', 'NNP'),
 ('is', 'VBZ'),
 ('a', 'DT'),
 ('hill', 'NN'),
 ('fort', 'NN'),
 ('situated', 'VBN'),
 ('in', 'IN'),
 ('the', 'DT'),
 ('pune', 'JJ'),
 ('district', 'NN'),
 ('of', 'IN'),
 ('maharashtra', 'JJ'),
 ('India.Formerly', 'NNP'),
 ('known', 'VBN'),
 ('as', 'IN'),
 ('Murumdev', 'NNP'),
 ('the', 'DT'),
 ('fort', 'NN'),
 ('was', 'VBD'),
 ('the', 'DT'),
 ('capital', 'NN'),
 ('of', 'IN'),
 ('Maratha', 'NNP'),
 ('empire', 'NN'),
 ('under', 'IN'),
 ('the', 'DT'),
 ('rule', 'NN'),
 ('of', 'IN'),
 ('Chatrapati', 'NNP'),
 ('Shivaji', 'NNP'),
 ('Maharaj', 'NNP'),
 ('for', 'IN'),
 ('almost', 'RB'),
 ('26', 'CD'),
 ('years', 'NNS'),
 ('after', 'IN'),
 ('which', 'WDT'),
 ('the', 'DT'),
 ('capital', 'NN'),
 ('moved', 'VBD'),
 ('to', 'TO'),
 ('the', 'DT'),
 ('Raigad', 'NNP'),
 ('Fort', 'NNP'),
 ('1', 'CD'),
 ('Treasures', 'NNS'),
 ('discovered', 'VBN'),
 ('from', 'IN'),
 ('an', 'DT'),
 ('adjacent', 'JJ'),
 ('fort', 'NN'),
 ('called', 'VBN'),
 ('Torna', 'NNP'),
 ('were', 'VBD'),
 ('used', 'VBN'),
 ('to', 'TO'),
 ('completely', 'RB'),
 ('built', 'VBN'),
 ('and', 'CC'),
 ('fortify', 'VB'),
 ('the', 'DT'),
 ('rajgad', 'NN'),
 ('fort', 'NN')]
```

In [39]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
```

In [40]:
```
tfidf=TfidfVectorizer()
v=tfidf.fit_transform(t)
v.shape
```

Out[40]:
```
(67, 50)
```

In [41]:
```
import pandas as pd
pd.DataFrame(v)
```

Out[41]:

	0
0	(0, 35)\t1.0
1	(0, 25)\t1.0
2	(0, 29)\t1.0
3	(0, 37)\t1.0
4	(0, 17)\t1.0
...	...
62	(0, 5)\t1.0
63	(0, 18)\t1.0
64	(0, 40)\t1.0
65	(0, 35)\t1.0
66	(0, 17)\t1.0

67 rows × 1 columns

In [ ]: