In [31]:
```
!pip install textblob
```

```
Requirement already satisfied: textblob in c:\vmwareworkstation10\nikita\lib
\site-packages (0.19.0)
Requirement already satisfied: nltk>=3.9 in c:\vmwareworkstation10\nikita\lib
\site-packages (from textblob) (3.9.1)
Requirement already satisfied: click in c:\vmwareworkstation10\nikita\lib\sit
e-packages (from nltk>=3.9->textblob) (8.0.4)
Requirement already satisfied: joblib in c:\vmwareworkstation10\nikita\lib\si
te-packages (from nltk>=3.9->textblob) (1.2.0)
Requirement already satisfied: regex>=2021.8.3 in c:\vmwareworkstation10\niki
ta\lib\site-packages (from nltk>=3.9->textblob) (2022.7.9)
Requirement already satisfied: tqdm in c:\vmwareworkstation10\nikita\lib\site
-packages (from nltk>=3.9->textblob) (4.65.0)
Requirement already satisfied: colorama in c:\vmwareworkstation10\nikita\lib
\site-packages (from click->nltk>=3.9->textblob) (0.4.6)
```

In [2]:
```
import textblob
from textblob import TextBlob
```

In [3]:
```
text = "Hello everyone! Welcome to my blog post on Medium. We are studying Natu
```

In [4]:
```
import nltk
from nltk.tokenize import word_tokenize
```

In [5]:
```
text = "Hello everyone! Welcome to my blog post on Medium. We are studying Natu
```

In [6]:
```
import nltk
from nltk import sent_tokenize
from nltk import word_tokenize
```

In [7]:
```
import nltk
nltk.download('punkt_tab')
```

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

Out[7]:
```
True
```

In [8]:
```
tokens_sents = nltk.sent_tokenize(text)
print(tokens_sents)
```

```
['Hello everyone!', 'Welcome to my blog post on Medium.', 'We are studying Na
tural Language Processing.']
```

In [9]:
```
tokens_words = nltk.word_tokenize(text)
print(tokens_words)
```

```
['Hello', 'everyone', '!', 'Welcome', 'to', 'my', 'blog', 'post', 'on', 'Medi
um', '.', 'We', 'are', 'studying', 'Natural', 'Language', 'Processing', '.']
```

In [10]:
```
from nltk.stem import PorterStemmer
```

In [11]:
```
ps = PorterStemmer()
word = ("civilization")
ps.stem(word)
```

Out[11]:
```
'civil'
```

In [12]:
```
from nltk.stem.snowball import SnowballStemmer
```

In [13]:
```
stemmer = SnowballStemmer(language = "english")
word = "civilization"
stemmer.stem(word)
```

Out[13]:
```
'civil'
```

In [14]:
```
import nltk
from nltk.stem import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
```

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

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

Out[15]:
```
True
```

In [16]:
```
print(lemmatizer.lemmatize("workers"))
print(lemmatizer.lemmatize("beeches"))
```

```
worker
beech
```

In [17]:
```
text = "Let’s lemmatize a simple sentence. We first tokenize the sentence into
word_list = nltk.word_tokenize(text)
print(word_list)
```

```
['Let', '’', 's', 'lemmatize', 'a', 'simple', 'sentence', '.', 'We', 'first',
'tokenize', 'the', 'sentence', 'into', 'words', 'using', 'nltk.word_tokeniz
e', 'and', 'then', 'we', 'will', 'call', 'lemmatizer.lemmatize', '(', ')']
```

In [18]:
```
lemmatized_output = ' '.join([lemmatizer.lemmatize(w) for w in word_list])
print(lemmatized_output)
```

```
Let ’ s lemmatize a simple sentence . We first tokenize the sentence into wor
d using nltk.word_tokenize and then we will call lemmatizer.lemmatize ( )
```

In [19]:
```
from textblob import TextBlob, Word
```

In [20]:
```
word = 'stripes'
w = Word(word)
w.lemmatize()
```

Out[20]:
```
'stripe'
```

In [21]:
```
text = "The striped bats are hanging on their feet for best"
sent = TextBlob(text)
" ". join([w.lemmatize() for w in sent.words])
```

Out[21]:
```
'The striped bat are hanging on their foot for best'
```

In [22]:
```
import nltk
from nltk import word_tokenize
```

In [23]:
```
import nltk
nltk.download('averaged_perceptron_tagger_eng')
```

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

Out[23]:
```
True
```

In [24]:
```
text = "The striped bats are hanging on their feet for best"
tokens = nltk.word_tokenize(text)
print("Parts of Speech: ",nltk.pos_tag(tokens))
```

```
Parts of Speech:  [('The', 'DT'), ('striped', 'JJ'), ('bats', 'NNS'), ('are',
'VBP'), ('hanging', 'VBG'), ('on', 'IN'), ('their', 'PRP$'), ('feet', 'NNS'),
('for', 'IN'), ('best', 'JJS')]
```

In [25]:
```
import nltk
from nltk import word_tokenize
```

In [26]:
```
text = "The striped bats are hanging on their feet for best"
tokens = nltk.word_tokenize(text)
print("Parts of Speech: ",nltk.pos_tag(tokens))
```

```
Parts of Speech:  [('The', 'DT'), ('striped', 'JJ'), ('bats', 'NNS'), ('are',
'VBP'), ('hanging', 'VBG'), ('on', 'IN'), ('their', 'PRP$'), ('feet', 'NNS'),
('for', 'IN'), ('best', 'JJS')]
```

In [32]:
```
documents = [
"The quick brown fox jumped over the lazy dog's back",
"Now is the time for all good men to come to the aid of their party"
]
```

In [33]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
```

In [34]:
```
documents = [
"The quick brown fox jumped over the lazy dog's back",
"Now is the time for all good men to come to the aid of their party"
]
```

In [35]:
```
vectorizer = TfidfVectorizer(stop_words=["for","is","of","the","to"])
X = vectorizer.fit_transform(documents)
```

In [36]:
```
print(X)
```

```
  (0, 2)	0.3535533905932738
  (0, 5)	0.3535533905932738
  (0, 9)	0.3535533905932738
  (0, 12)	0.3535533905932738
  (0, 8)	0.3535533905932738
  (0, 6)	0.3535533905932738
  (0, 3)	0.3535533905932738
  (0, 14)	0.3535533905932738
  (1, 13)	0.33333333333333337
  (1, 15)	0.33333333333333337
  (1, 0)	0.33333333333333337
  (1, 4)	0.33333333333333337
  (1, 10)	0.33333333333333337
  (1, 7)	0.33333333333333337
  (1, 1)	0.33333333333333337
  (1, 16)	0.33333333333333337
  (1, 11)	0.33333333333333337
```

In [ ]: