siddhu7

April 15, 2025

[20]:
```
!pip install textblob
```

```
Requirement already satisfied: textblob in c:\users\windows
10\anaconda3\lib\site-packages (0.19.0)
Requirement already satisfied: nltk>=3.9 in c:\users\windows
10\anaconda3\lib\site-packages (from textblob) (3.9.1)
Requirement already satisfied: click in c:\users\windows 10\anaconda3\lib\site-
packages (from nltk>=3.9->textblob) (8.1.7)
Requirement already satisfied: joblib in c:\users\windows 10\anaconda3\lib\site-
packages (from nltk>=3.9->textblob) (1.2.0)
Requirement already satisfied: regex>=2021.8.3 in c:\users\windows
10\anaconda3\lib\site-packages (from nltk>=3.9->textblob) (2023.10.3)
Requirement already satisfied: tqdm in c:\users\windows 10\anaconda3\lib\site-
packages (from nltk>=3.9->textblob) (4.65.0)
Requirement already satisfied: colorama in c:\users\windows
10\anaconda3\lib\site-packages (from click->nltk>=3.9->textblob) (0.4.6)
```

[21]:
```
import textblob
from textblob import TextBlob
```

[3]:
```
text = "Hello everyone! Welcome to my blog post on Medium. We are studying␣
 ↪Natural Language Processing."
```

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

[5]:
```
text = "Hello everyone! Welcome to my blog post on Medium. We are studying␣
 ↪Natural Language Processing."
```

[6]:
```
tokens_sents = nltk.sent_tokenize(text)
print(tokens_sents)
```

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

[7]:
```
tokens_words = nltk.word_tokenize(text)
print(tokens_words)
```

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

[8]:
```
from nltk.stem import PorterStemmer
ps = PorterStemmer()
word = ("civilization")
ps.stem(word)
```

[8]:
```
'civil'
```

[9]:
```
from nltk.stem.snowball import SnowballStemmer
stemmer = SnowballStemmer(language = "english")
word = "civilization"
stemmer.stem(word)
```

[9]:
```
'civil'
```

[10]:
```
from nltk.stem import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
nltk.download('wordnet')
```

```
[nltk_data] Downloading package wordnet to C:\Users\WINDOWS
[nltk_data]     10\AppData\Roaming\nltk_data…
[nltk_data]   Package wordnet is already up-to-date!
```

[10]:
```
True
```

[11]:
```
print(lemmatizer.lemmatize("workers"))
print(lemmatizer.lemmatize("beeches"))
```

```
worker
beech
```

[12]:
```
text = "Let's lemmatize a simple sentence. We first tokenize the sentence into␣
  ↪words using nltk.word_tokenize and then we will call lemmatizer.lemmatize()"
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_tokenize',
'and', 'then', 'we', 'will', 'call', 'lemmatizer.lemmatize', '(', ')']
```

[13]:
```
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 word
using nltk.word_tokenize and then we will call lemmatizer.lemmatize ( )
```

[14]:
```
from textblob import TextBlob, Word
word = 'stripes'
w = Word(word)
w.lemmatize()
```

[14]:
```
'stripe'
```

[15]:
```
text = "The striped bats are hanging on their feet for best"
sent = TextBlob(text)
" ". join([w.lemmatize() for w in sent.words])
```

[15]:
```
'The striped bat are hanging on their foot for best'
```

[16]:
```
nltk.download('averaged_perceptron_tagger_eng')
text = "The striped bats are hanging on their feet for best"
tokens = nltk.word_tokenize(text)
print("Parts of Speech: ",nltk.pos_tag(tokens))
```

```
[nltk_data] Downloading package averaged_perceptron_tagger_eng to
[nltk_data]     C:\Users\WINDOWS 10\AppData\Roaming\nltk_data…
[nltk_data]   Unzipping taggers\averaged_perceptron_tagger_eng.zip.
```

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

[17]:
```
from sklearn.feature_extraction.text import TfidfVectorizer
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"
]
vectorizer = TfidfVectorizer(stop_words=["for","is","of","the","to"])
X = vectorizer.fit_transform(documents)
```

[18]:
```
tfidf_matrix = vectorizer.fit_transform(documents)
tfidf_dense = tfidf_matrix.toarray()
print("TF-IDF Matrix (Dense Format):")
print(tfidf_dense)
```

```
TF-IDF Matrix (Dense Format):
[[0.         0.         0.35355339 0.35355339 0.         0.35355339
  0.35355339 0.         0.35355339 0.35355339 0.         0.
  0.35355339 0.         0.35355339 0.         0.        ]
 [0.33333333 0.33333333 0.         0.         0.33333333 0.
  0.         0.33333333 0.         0.         0.33333333 0.33333333
  0.         0.33333333 0.         0.33333333 0.33333333]]
```

[19]:
```
feature_names = vectorizer.get_feature_names_out()
print("\nFeature Names (Terms):")
print(feature_names)
```

```
Feature Names (Terms):
['aid' 'all' 'back' 'brown' 'come' 'dog' 'fox' 'good' 'jumped' 'lazy'
 'men' 'now' 'over' 'party' 'quick' 'their' 'time']
```

[ ]: