Lab Assignment - 2

Textblob

In [1]:
```
import nltk
# nltk.download('punkt')
```

In [2]:
```
nltk.download('averaged_perceptron_tagger')
```

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

Out[2]:
```
True
```

In [3]:
```
from textblob import TextBlob
```

In [4]:
```
text = '''
The titular threat of The Blob has always struck me as the ultimate movie
monster: an insatiably hungry, amoeba-like mass able to penetrate
virtually any safeguard, capable of--as a doomed doctor chillingly
describes it--"assimilating flesh on contact.
Snide comparisons to gelatin be damned, it's a concept with the most
devastating of potential consequences, not unlike the grey goo scenario
proposed by technological theorists fearful of
artificial intelligence run rampant.
'''
```

- pos tagging

In [5]:
```
blob = TextBlob(text)
blob.tags
```

Out[5]:
```
[('The', 'DT'),
 ('titular', 'JJ'),
 ('threat', 'NN'),
 ('of', 'IN'),
 ('The', 'DT'),
 ('Blob', 'NNP'),
 ('has', 'VBZ'),
 ('always', 'RB'),
 ('struck', 'VBN'),
 ('me', 'PRP'),
 ('as', 'IN'),
 ('the', 'DT'),
 ('ultimate', 'JJ'),
 ('movie', 'NN'),
 ('monster', 'NN'),
 ('an', 'DT'),
 ('insatiably', 'RB'),
 ('hungry', 'JJ'),
 ('amoeba-like', 'JJ'),
 ('mass', 'NN'),
 ('able', 'JJ'),
 ('to', 'TO'),
 ('penetrate', 'VB'),
 ('virtually', 'RB'),
 ('any', 'DT'),
 ('safeguard', 'NN'),
 ('capable', 'JJ'),
 ('of', 'IN'),
 ('as', 'IN'),
 ('a', 'DT'),
 ('doomed', 'JJ'),
 ('doctor', 'NN'),
 ('chillingly', 'RB'),
 ('describes', 'VBZ'),
 ('it', 'PRP'),
 ('assimilating', 'VBG'),
 ('flesh', 'NN'),
 ('on', 'IN'),
 ('contact', 'NN'),
 ('Snide', 'JJ'),
 ('comparisons', 'NNS'),
 ('to', 'TO'),
 ('gelatin', 'VB'),
 ('be', 'VB'),
 ('damned', 'VBN'),
 ('it', 'PRP'),
 ("'s", 'VBZ'),
 ('a', 'DT'),
 ('concept', 'NN'),
 ('with', 'IN'),
 ('the', 'DT'),
 ('most', 'RBS'),
 ('devastating', 'JJ'),
 ('of', 'IN'),
 ('potential', 'JJ'),
 ('consequences', 'NNS'),
 ('not', 'RB'),
 ('unlike', 'IN'),
 ('the', 'DT'),
 ('grey', 'NN'),
 ('goo', 'NN'),
 ('scenario', 'NN'),
 ('proposed', 'VBN'),
 ('by', 'IN'),
 ('technological', 'JJ'),
 ('theorists', 'NNS'),
 ('fearful', 'NN'),
 ('of', 'IN'),
 ('artificial', 'JJ'),
 ('intelligence', 'NN'),
```

```
 ('run', 'NN'),
 ('rampant', 'NN')]
```

In [6]:
```
nltk.download('brown')
```

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

Out[6]:
```
True
```

- Noun Phrase Extraction

In [7]:
```
blob.noun_phrases
```

Out[7]:
```
WordList(['titular threat', 'blob', 'ultimate movie monster', 'amoeba-like mass', 'snide',
'potential consequences', 'grey goo scenario', 'technological theorists fearful', 'artific
ial intelligence run rampant'])
```

- sentiment

In [8]:
```
for sentence in blob.sentences:
    print(sentence.sentiment.polarity)
```

```
0.06000000000000001
-0.34166666666666673
```

In [9]:
```
testimonial = TextBlob("Textblob is amazingly simple to use. What great fun!")
testimonial.sentiment
```

Out[9]:
```
Sentiment(polarity=0.39166666666666666, subjectivity=0.4357142857142857)
```

In [10]:
```
testimonial = TextBlob("This computer is good , although it is very costly !")
testimonial.sentiment
```

Out[10]:
```
Sentiment(polarity=0.475, subjectivity=0.45000000000000007)
```

In [11]:
```
testimonial = TextBlob("This computer is very bad , although it is very costly !")
testimonial.sentiment
```

Out[11]:
```
Sentiment(polarity=-0.3299999999999999, subjectivity=0.5833333333333334)
```

- tokanization

In [12]:
```
testimonial.words
```

Out[12]:
```
WordList(['This', 'computer', 'is', 'very', 'bad', 'although', 'it', 'is', 'very', 'costl
y'])
```

- Get Word and Noun Phrase Frequencies

In [13]:
```
testimonial.word_counts['computer']
```

Out[13]:
```
1
```

In [14]:
```
testimonial.words.count('computer', case_sensitive=True)
```

Out[14]:
```
1
```

- n grams

In [15]:
```
testimonial.ngrams(n=3)
```

Out[15]:
```
[WordList(['This', 'computer', 'is']),
 WordList(['computer', 'is', 'very']),
 WordList(['is', 'very', 'bad']),
 WordList(['very', 'bad', 'although']),
 WordList(['bad', 'although', 'it']),
 WordList(['although', 'it', 'is']),
 WordList(['it', 'is', 'very']),
 WordList(['is', 'very', 'costly'])]
```

NLTK

In [16]:
```
# import nltk
```

In [17]:
```
sentence = """At eight o'clock on Thursday morning Arthur didn't feel very good."""
```

- tokenization

In [18]:
```
tokens = nltk.word_tokenize(sentence)
tokens
```

Out[18]:
```
['At',
 'eight',
 "o'clock",
 'on',
 'Thursday',
 'morning',
 'Arthur',
 'did',
 "n't",
 'feel',
 'very',
 'good',
 '.']
```

- pos tagging

In [19]:
```
tagged = nltk.pos_tag(tokens)
tagged
```

Out[19]:
```
[('At', 'IN'),
 ('eight', 'CD'),
 ("o'clock", 'NN'),
 ('on', 'IN'),
 ('Thursday', 'NNP'),
 ('morning', 'NN'),
 ('Arthur', 'NNP'),
 ('did', 'VBD'),
 ("n't", 'RB'),
 ('feel', 'VB'),
 ('very', 'RB'),
 ('good', 'JJ'),
 ('.', '.')]
```

In [20]:
```
# nltk.download('maxent_ne_chunker')
# nltk.download('words')
!pip install svgling
```

```
Requirement already satisfied: svgling in c:\python311\lib\site-packages (0.4.0)
Requirement already satisfied: svgwrite in c:\python311\lib\site-packages (from svgling)
(1.4.3)


[notice] A new release of pip is available: 23.0.1 -> 23.3.2
[notice] To update, run: python.exe -m pip install --upgrade pip
```

In [21]:
```
entities = nltk.chunk.ne_chunk(tagged)
entities
```

Out[21]:
```
Tree('S', [('At', 'IN'), ('eight', 'CD'), ("o'clock", 'NN'), ('on', 'IN'), ('Thursday', 'N
NP'), ('morning', 'NN'), Tree('PERSON', [('Arthur', 'NNP')]), ('did', 'VBD'), ("n't", 'R
B'), ('feel', 'VB'), ('very', 'RB'), ('good', 'JJ'), ('.', '.')])
```

In [22]:
```
from nltk.corpus import treebank
```

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

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

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

In [24]:
```
# import nltk
# from nltk.corpus import treebank

# # Load the Penn Treebank data
# nltk.download('treebank')

# # Get a parsed sentence from the treebank
# t = treebank.parsed_sents('wsj_0001.mrg')[0]

# # Draw the parse tree
# t.draw()
98+31
```

Out[24]:
```
129
```

In [28]:
```
# t = treebank.parsed_sents('wsj_0001.mrg')[0]
# t.draw(max_depth=2)  # Adjust the depth as needed
```

In [26]:
```
t = treebank.parsed_sents('wsj_0001.mrg')[0]
print(t)
```

```
(S
  (NP-SBJ
    (NP (NNP Pierre) (NNP Vinken))
    (, ,)
    (ADJP (NP (CD 61) (NNS years)) (JJ old))
    (, ,))
  (VP
    (MD will)
    (VP
      (VB join)
      (NP (DT the) (NN board))
      (PP-CLR (IN as) (NP (DT a) (JJ nonexecutive) (NN director)))
      (NP-TMP (NNP Nov.) (CD 29))))
  (. .))
```

In [29]:
```
!pip install spacy
```

```
Collecting spacy
  Downloading spacy-3.7.2-cp311-cp311-win_amd64.whl (12.1 MB)
     ---------------------------------------- 0.0/12.1 MB ? eta -:--:--
     ---------------------------------------- 0.0/12.1 MB ? eta -:--:--
     ---------------------------------------- 0.0/12.1 MB 660.6 kB/s eta 0:00:19
     ---------------------------------------- 0.1/12.1 MB 375.8 kB/s eta 0:00:32
     ---------------------------------------- 0.1/12.1 MB 469.7 kB/s eta 0:00:26
     ---------------------------------------- 0.1/12.1 MB 403.5 kB/s eta 0:00:30
      --------------------------------------- 0.2/12.1 MB 787.7 kB/s eta 0:00:16
      --------------------------------------- 0.3/12.1 MB 827.5 kB/s eta 0:00:15
     - -------------------------------------- 0.3/12.1 MB 1.0 MB/s eta 0:00:12
     - -------------------------------------- 0.4/12.1 MB 998.3 kB/s eta 0:00:12
     - -------------------------------------- 0.4/12.1 MB 998.3 kB/s eta 0:00:12
     - -------------------------------------- 0.4/12.1 MB 998.3 kB/s eta 0:00:12
     - -------------------------------------- 0.4/12.1 MB 998.3 kB/s eta 0:00:12
     - -------------------------------------- 0.4/12.1 MB 726.4 kB/s eta 0:00:17
     - -------------------------------------- 0.5/12.1 MB 819.2 kB/s eta 0:00:15
     -- ------------------------------------- 0.6/12.1 MB 922.8 kB/s eta 0:00:13
     -- ------------------------------------- 0.7/12.1 MB 982.7 kB/s eta 0:00:12
```

In [32]:
```
!python -m spacy download en_core_web_sm
```

```
Collecting en-core-web-sm==3.7.1
  Downloading https://github.com/explosion/spacy-models/releases/download/en_core_web_sm-3.7.1/en_core_web_sm-3.7.1-py3-none-any.whl (https://github.com/explosion/spacy-models/releases/download/en_core_web_sm-3.7.1/en_core_web_sm-3.7.1-py3-none-any.whl) (12.8 MB)
     ---------------------------------------- 0.0/12.8 MB ? eta -:--:--
     ---------------------------------------- 0.0/12.8 MB 682.7 kB/s eta 0:00:19
     ---------------------------------------- 0.1/12.8 MB 525.1 kB/s eta 0:00:25
     ---------------------------------------- 0.1/12.8 MB 657.6 kB/s eta 0:00:20
     ---------------------------------------- 0.1/12.8 MB 535.8 kB/s eta 0:00:24
     ---------------------------------------- 0.1/12.8 MB 602.4 kB/s eta 0:00:22
     ---------------------------------------- 0.2/12.8 MB 579.6 kB/s eta 0:00:22
     ---------------------------------------- 0.2/12.8 MB 579.6 kB/s eta 0:00:22
      --------------------------------------- 0.2/12.8 MB 628.1 kB/s eta 0:00:20
      --------------------------------------- 0.3/12.8 MB 682.7 kB/s eta 0:00:19
     - -------------------------------------- 0.3/12.8 MB 720.5 kB/s eta 0:00:18
     - -------------------------------------- 0.4/12.8 MB 839.7 kB/s eta 0:00:15
     - -------------------------------------- 0.5/12.8 MB 829.2 kB/s eta 0:00:15
     - -------------------------------------- 0.5/12.8 MB 814.9 kB/s eta 0:00:16
```

In [33]:
```
import spacy

# Load the English language model
nlp = spacy.load('en_core_web_sm')

# Process a text
doc = nlp("spaCy is a powerful NLP library.")

# Access various annotations
for token in doc:
    print(token.text, token.pos_, token.dep_)
```

```
spaCy INTJ nsubj
is AUX ROOT
a DET det
powerful ADJ amod
NLP PROPN compound
library NOUN attr
. PUNCT punct
```

In [ ]:
```
```