NLTK (Natural Language Toolkit)

[3]:
```python
# Comprehensive library for NLP tasks.

import nltk
nltk.download('punkt')
nltk.download('punkt_tab')
nltk.download('wordnet')
nltk.download('omw-1.4')
from nltk.tokenize import word_tokenize

text = "Natural Language Processing with Python is amazing!"
tokens = word_tokenize(text)
print(tokens)
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\rajpu\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package punkt_tab to
[nltk_data]     C:\Users\rajpu\AppData\Roaming\nltk_data...
[nltk_data]   Unzipping tokenizers\punkt_tab.zip.
['Natural', 'Language', 'Processing', 'with', 'Python', 'is', 'amazing', '!']
[nltk_data] Downloading package wordnet to
[nltk_data]     C:\Users\rajpu\AppData\Roaming\nltk_data...
[nltk_data]   Package wordnet is already up-to-date!
[nltk_data] Downloading package omw-1.4 to
[nltk_data]     C:\Users\rajpu\AppData\Roaming\nltk_data...
[nltk_data]   Package omw-1.4 is already up-to-date!
```

spaCy

[27]:
```
pip install spacy
```

```
Collecting spacyNote: you may need to restart the kernel to use updated packages.

  Downloading spacy-3.8.4-cp312-cp312-win_amd64.whl.metadata (27 kB)
Collecting spacy-legacy<3.1.0,>=3.0.11 (from spacy)
  Downloading spacy_legacy-3.0.12-py2.py3-none-any.whl.metadata (2.8 kB)
Collecting spacy-loggers<2.0.0,>=1.0.0 (from spacy)
  Downloading spacy_loggers-1.0.5-py3-none-any.whl.metadata (23 kB)
Collecting murmurhash<1.1.0,>=0.28.0 (from spacy)
  Downloading murmurhash-1.0.12-cp312-cp312-win_amd64.whl.metadata (2.2 kB)
Collecting cymem<2.1.0,>=2.0.2 (from spacy)
  Downloading cymem-2.0.11-cp312-cp312-win_amd64.whl.metadata (8.8 kB)
Collecting preshed<3.1.0,>=3.0.2 (from spacy)
  Downloading preshed-3.0.9-cp312-cp312-win_amd64.whl.metadata (2.2 kB)
Collecting thinc<8.4.0,>=8.3.4 (from spacy)
  Downloading thinc-8.3.4-cp312-cp312-win_amd64.whl.metadata (15 kB)
Collecting wasabi<1.2.0,>=0.9.1 (from spacy)
  Downloading wasabi-1.1.3-py3-none-any.whl.metadata (28 kB)
Collecting srsly<3.0.0,>=2.4.3 (from spacy)
```

[31]:
```python
import spacy
import spacy.cli

# Ensure the required models are downloaded
spacy.cli.download("en_core_web_sm")

# Load the spaCy model
nlp = spacy.load("en_core_web_sm")

# Process a sample text
doc = nlp("John is learning Natural Language Processing in New York.")

# Print detected entities and their labels
for entity in doc.ents:
    print(entity.text, entity.label_)
```

```
✓ Download and installation successful
You can now load the package via spacy.load('en_core_web_sm')
⚠ Restart to reload dependencies
If you are in a Jupyter or Colab notebook, you may need to restart Python in
order to load all the package's dependencies. You can do this by selecting the
'Restart kernel' or 'Restart runtime' option.
John PERSON
Natural Language Processing ORG
New York GPE
```

TextBlob

[6]:
```
pip install textblob nltk
```

```
Collecting textblob
  Downloading textblob-0.19.0-py3-none-any.whl.metadata (4.4 kB)
Requirement already satisfied: nltk in c:\users\rajpu\anaconda3\lib\site-packages (3.9.1)
Requirement already satisfied: click in c:\users\rajpu\anaconda3\lib\site-packages (from nltk) (8.1.7)
Requirement already satisfied: joblib in c:\users\rajpu\anaconda3\lib\site-packages (from nltk) (1.4.2)
Requirement already satisfied: regex>=2021.8.3 in c:\users\rajpu\anaconda3\lib\site-packages (from nltk) (2024.9.11)
Requirement already satisfied: tqdm in c:\users\rajpu\anaconda3\lib\site-packages (from nltk) (4.66.5)
Requirement already satisfied: colorama in c:\users\rajpu\anaconda3\lib\site-packages (from click->nltk) (0.4.6)
Downloading textblob-0.19.0-py3-none-any.whl (624 kB)
   ---------------------------------------- 0.0/624.3 kB ? eta -:--:--
   ---------------- ----------------------- 262.1/624.3 kB ? eta -:--:--
   --------------------------------- ------ 524.3/624.3 kB 2.1 MB/s eta 0:00:01
   ---------------------------------------- 624.3/624.3 kB 1.1 MB/s eta 0:00:00
Installing collected packages: textblob
Successfully installed textblob-0.19.0
Note: you may need to restart the kernel to use updated packages.
```

[8]:
```python
from textblob import TextBlob

text = "I love programming."
blob = TextBlob(text)
print(blob.sentiment)
```

```
Sentiment(polarity=0.5, subjectivity=0.6)
```

Scikit-learn

[13]:
```
pip install transformers
```

```
Collecting transformers
  Downloading transformers-4.49.0-py3-none-any.whl.metadata (44 kB)
Requirement already satisfied: filelock in c:\users\rajpu\anaconda3\lib\site-packages (from transformers) (3.13.1)
Collecting huggingface-hub<1.0,>=0.26.0 (from transformers)
  Downloading huggingface_hub-0.29.1-py3-none-any.whl.metadata (13 kB)
Requirement already satisfied: numpy>=1.17 in c:\users\rajpu\anaconda3\lib\site-packages (from transformers) (1.26.4)
Requirement already satisfied: packaging>=20.0 in c:\users\rajpu\anaconda3\lib\site-packages (from transformers) (24.1)
Requirement already satisfied: pyyaml>=5.1 in c:\users\rajpu\anaconda3\lib\site-packages (from transformers) (6.0.1)
Requirement already satisfied: regex!=2019.12.17 in c:\users\rajpu\anaconda3\lib\site-packages (from transformers) (2024.9.11)
Requirement already satisfied: requests in c:\users\rajpu\anaconda3\lib\site-packages (from transformers) (2.32.3)
Collecting tokenizers<0.22,>=0.21 (from transformers)
  Downloading tokenizers-0.21.0-cp39-abi3-win_amd64.whl.metadata (6.9 kB)
Collecting safetensors>=0.4.1 (from transformers)
  Downloading safetensors-0.5.2-cp38-abi3-win_amd64.whl.metadata (3.9 kB)
Requirement already satisfied: tqdm>=4.27 in c:\users\rajpu\anaconda3\lib\site-packages (from transformers) (4.66.5)
Requirement already satisfied: fsspec>=2023.5.0 in c:\users\rajpu\anaconda3\lib\site-packages (from huggingface-hub<1.0,>=0.26.0->transformers) (2024.6.1)
Requirement already satisfied: typing-extensions>=3.7.4.3 in c:\users\rajpu\anaconda3\lib\site-packages (from huggingface-hub<1.0,>=0.26.0->transformers)
(4.11.0)
Requirement already satisfied: colorama in c:\users\rajpu\anaconda3\lib\site-packages (from tqdm>=4.27->transformers) (0.4.6)
Requirement already satisfied: charset-normalizer<4,>=2 in c:\users\rajpu\anaconda3\lib\site-packages (from requests->transformers) (3.3.2)
Requirement already satisfied: idna<4,>=2.5 in c:\users\rajpu\anaconda3\lib\site-packages (from requests->transformers) (3.7)
Requirement already satisfied: urllib3<3,>=1.21.1 in c:\users\rajpu\anaconda3\lib\site-packages (from requests->transformers) (2.2.3)
Requirement already satisfied: certifi>=2017.4.17 in c:\users\rajpu\anaconda3\lib\site-packages (from requests->transformers) (2025.1.31)
Downloading transformers-4.49.0-py3-none-any.whl (10.0 MB)
   ---------------------------------------- 0.0/10.0 MB ? eta -:--:--
   ---------------------------------------- 0.0/10.0 MB ? eta -:--:--
   - -------------------------------------- 0.3/10.0 MB ? eta -:--:--
   -- ------------------------------------- 0.5/10.0 MB 932.9 kB/s eta 0:00:11
   -- ------------------------------------- 0.5/10.0 MB 932.9 kB/s eta 0:00:11
   --- ------------------------------------ 0.8/10.0 MB 714.3 kB/s eta 0:00:13
   --- ------------------------------------ 0.8/10.0 MB 714.3 kB/s eta 0:00:13
   ---- ----------------------------------- 1.0/10.0 MB 671.0 kB/s eta 0:00:14
   ---- ----------------------------------- 1.0/10.0 MB 671.0 kB/s eta 0:00:14
   ----- ---------------------------------- 1.3/10.0 MB 671.3 kB/s eta 0:00:13
   ----- ---------------------------------- 1.3/10.0 MB 671.3 kB/s eta 0:00:13
   ------ --------------------------------- 1.6/10.0 MB 671.1 kB/s eta 0:00:13
   ------- -------------------------------- 1.8/10.0 MB 694.4 kB/s eta 0:00:12
   -------- ------------------------------- 2.1/10.0 MB 725.0 kB/s eta 0:00:11
   -------- ------------------------------- 2.1/10.0 MB 725.0 kB/s eta 0:00:11
   --------- ------------------------------ 2.4/10.0 MB 749.8 kB/s eta 0:00:11
   ---------- ----------------------------- 2.6/10.0 MB 766.5 kB/s eta 0:00:10
   ----------- ---------------------------- 2.9/10.0 MB 791.5 kB/s eta 0:00:09
   ------------ --------------------------- 3.1/10.0 MB 809.5 kB/s eta 0:00:09
   ------------- -------------------------- 3.4/10.0 MB 832.0 kB/s eta 0:00:08
   -------------- ------------------------- 3.7/10.0 MB 858.6 kB/s eta 0:00:08
   --------------- ------------------------ 3.9/10.0 MB 883.1 kB/s eta 0:00:07
   ---------------- ----------------------- 4.2/10.0 MB 902.1 kB/s eta 0:00:07
   ----------------- ---------------------- 4.5/10.0 MB 906.9 kB/s eta 0:00:07
   ----------------- ---------------------- 4.5/10.0 MB 906.9 kB/s eta 0:00:07
   ------------------- -------------------- 5.0/10.0 MB 920.7 kB/s eta 0:00:06
   -------------------- ------------------- 5.2/10.0 MB 929.4 kB/s eta 0:00:06
   -------------------- ------------------- 5.2/10.0 MB 929.4 kB/s eta 0:00:06
   --------------------- ------------------ 5.5/10.0 MB 919.3 kB/s eta 0:00:05
   ----------------------- ---------------- 5.8/10.0 MB 917.5 kB/s eta 0:00:05
   ----------------------- ---------------- 5.8/10.0 MB 917.5 kB/s eta 0:00:05
   ------------------------ --------------- 6.0/10.0 MB 922.7 kB/s eta 0:00:05
   -------------------------- ------------- 6.6/10.0 MB 949.7 kB/s eta 0:00:04
   --------------------------- ------------ 6.8/10.0 MB 953.2 kB/s eta 0:00:04
   --------------------------- ------------ 6.8/10.0 MB 953.2 kB/s eta 0:00:04
   ---------------------------- ----------- 7.1/10.0 MB 954.6 kB/s eta 0:00:04
   ----------------------------- ---------- 7.3/10.0 MB 953.6 kB/s eta 0:00:03
   ----------------------------- ---------- 7.3/10.0 MB 953.6 kB/s eta 0:00:03
   ------------------------------ --------- 7.6/10.0 MB 949.1 kB/s eta 0:00:03
   ------------------------------- -------- 7.9/10.0 MB 950.3 kB/s eta 0:00:03
   --------------------------------- ------ 8.4/10.0 MB 983.2 kB/s eta 0:00:02
   ---------------------------------- ----- 8.7/10.0 MB 990.5 kB/s eta 0:00:02
   ----------------------------------- ---- 8.9/10.0 MB 1.0 MB/s eta 0:00:02
   ------------------------------------- -- 9.4/10.0 MB 1.0 MB/s eta 0:00:01
   -------------------------------------- - 9.7/10.0 MB 1.0 MB/s eta 0:00:01
   ---------------------------------------- 10.0/10.0 MB 1.0 MB/s eta 0:00:00
Downloading huggingface_hub-0.29.1-py3-none-any.whl (468 kB)
Downloading safetensors-0.5.2-cp38-abi3-win_amd64.whl (303 kB)
Downloading tokenizers-0.21.0-cp39-abi3-win_amd64.whl (2.4 MB)
   ---------------------------------------- 0.0/2.4 MB ? eta -:--:--
   ---- ----------------------------------- 0.3/2.4 MB ? eta -:--:--
   -------- ------------------------------- 0.5/2.4 MB 1.3 MB/s eta 0:00:02
   ----------------- ---------------------- 1.0/2.4 MB 1.7 MB/s eta 0:00:01
   --------------------- ------------------ 1.3/2.4 MB 1.8 MB/s eta 0:00:01
   ------------------------------ --------- 1.8/2.4 MB 1.9 MB/s eta 0:00:01
   ----------------------------------- ---- 2.1/2.4 MB 1.9 MB/s eta 0:00:01
   ---------------------------------------- 2.4/2.4 MB 1.8 MB/s eta 0:00:00
Installing collected packages: safetensors, huggingface-hub, tokenizers, transformers
Successfully installed huggingface-hub-0.29.1 safetensors-0.5.2 tokenizers-0.21.0 transformers-4.49.0
Note: you may need to restart the kernel to use updated packages.
```

[10]:
```python
from transformers import pipeline

summarizer = pipeline("summarization")
text = """Natural Language Processing (NLP) is a fascinating field of artificial intelligence
that focuses on the interaction between computers and humans through natural language.
It enables machines to understand, interpret, and generate human language effectively."""

print(summarizer(text, max_length=30, min_length=10))
```

```
No model was supplied, defaulted to sshleifer/distilbart-cnn-12-6 and revision a4f8f3e (https://huggingface.co/sshleifer/distilbart-cnn-12-6).
Using a pipeline without specifying a model name and revision in production is not recommended.
Device set to use cpu
[{'summary_text': ' Natural Language Processing (NLP) is a fascinating field of artificial intelligence that focuses on the interaction between computers a
nd humans through natural language .'}]
```

Topic Modeling with Gensim (LDA)

[1]:
```python
import genism
from gensim import corpora
from nltk.tokenize import word_tokenize
import string

#Sample documents
documents = ["I love programming in Python. Python is great for data analysis.",
             "I enjoy learning machine learning techniques.",
             "Data science is a mix of statistics, programming, and machine learning.",
             "Machine learning is part of the broader field of artificial intelligence.",
             "Statistics and data science are closely related fields.",
            ]

# Preprocessing: Tokenization and removing stopwords/punctuation
stopwords = set(['is', 'a', 'the', 'for', 'and', 'of', 'in', 'to'])

def preprocess(text):
#Tokenize and remove punctuation and stopwords
    tokens = word_tokenize(text.lower()) # lowercase and tokenize
    tokens = [t for t in tokens if t not in stopwords and t not in string.punctuation]
    return tokens
#Preprocess all documents
    processed_docs = [preprocess(doc) for doc in documents]

#Create a dictionary from the processed documents
dictionary = corpora.Dictionary(processed_docs)

#Create a document-term matrix (DTM)
corpus = [dictionary.doc2bow(doc) for doc in processed_docs]

# Build the LDA model
Ida_model = gensim.models.LdaMulticore(corpus, num_topics=2, id2word=dictionary, passes=10)

# Print the topics discovered by the LDA model
topics = Ida_model.print_topics(num_words=4)

for topic in topics:
    print(topic)
```

```
---------------------------------------------------------------------------
ModuleNotFoundError                       Traceback (most recent call last)
Cell In[1], line 1
----> 1 import genism
      2 from gensim import corpora
      3 from nltk.tokenize import word_tokenize

ModuleNotFoundError: No module named 'genism'
```

Pre-processing Pipeline in Python

[11]:
```python
import string
import nltk
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.stem import PorterStemmer

# Download necessary NLTK data
nltk.download('punkt')
nltk.download('stopwords')

# Sample text
text = "I love programming in Python, especially for data science!"

#Tokenization
tokens = word_tokenize(text)

# Lowercasing
tokens = [token.lower() for token in tokens]

# Removing punctuation
tokens = [token for token in tokens if token not in string.punctuation]

# Removing stopwords
stop_words = set(stopwords.words('english'))
tokens = [token for token in tokens if token not in stop_words]

# Stemming
stemmer = PorterStemmer()
tokens = [stemmer.stem(token) for token in tokens]

# Resulting pre-processed text
print(tokens)
```

```
[nltk_data] Downloading package punkt to
[nltk_data]     C:\Users\rajpu\AppData\Roaming\nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package stopwords to
[nltk_data]     C:\Users\rajpu\AppData\Roaming\nltk_data...
['love', 'program', 'python', 'especi', 'data', 'scienc']
[nltk_data]   Unzipping corpora\stopwords.zip.
```

Stemming in Python using NLTK

[28]:
```python
import nltk
from nltk.stem import PorterStemmer, LancasterStemmer, SnowballStemmer

# Sample words
words = ["running", "runner", "happily", "better", "fishing", "jumps"]

# Initialize different stemmers
porter_stemmer = PorterStemmer()
lancaster_stemmer = LancasterStemmer()
snowball_stemmer = SnowballStemmer("english")

# Apply stemming using different stemmers
print("Porter Stemmer:")
for word in words:
    print(f" {word} -> {porter_stemmer.stem(word)}")

print("\nLancaster Stemmer:")
for word in words:
    print(f" {word} -> {lancaster_stemmer.stem(word)}")

print("\nSnowball Stemmer:")
for word in words:
    print(f" {word} -> {snowball_stemmer.stem(word)}")
```

```
Porter Stemmer:
 running -> run
 runner -> runner
 happily -> happili
 better -> better
 fishing -> fish
 jumps -> jump

Lancaster Stemmer:
 running -> run
 runner -> run
 happily -> happy
 better -> bet
 fishing -> fish
 jumps -> jump

Snowball Stemmer:
 running -> run
 runner -> runner
 happily -> happili
 better -> better
 fishing -> fish
 jumps -> jump
```

Lemmatization in Python using NLTK

[33]:
```python
import nltk
from nltk.stem import WordNetLemmatizer
from nltk.tokenize import word_tokenize

# Sample text
text = "The Lt. M. J. Kundaliya Arts & Commerce Mahila College Running the Computer Science Department."

#Tokenize the text
tokens = word_tokenize(text)

# Initialize the Lemmatizer
lemmatizer = WordNetLemmatizer()

#Lemmatize the tokens
lemmatized_tokens = [lemmatizer.lemmatize(token, pos='v') for token in tokens]

# Print the lemmatized tokens
print(lemmatized_tokens)
```

```
['The', 'Lt.', 'M.', 'J.', 'Kundaliya', 'Arts', '&', 'Commerce', 'Mahila', 'College', 'Running', 'the', 'Computer', 'Science', 'Department', '.']
```

Chunking

[1]:
```
pip install nltk
```

```
Requirement already satisfied: nltk in c:\users\rajpu\anaconda3\lib\site-packages (3.9.1)
Requirement already satisfied: click in c:\users\rajpu\anaconda3\lib\site-packages (from nltk) (8.1.7)
Requirement already satisfied: joblib in c:\users\rajpu\anaconda3\lib\site-packages (from nltk) (1.4.2)
Requirement already satisfied: regex>=2021.8.3 in c:\users\rajpu\anaconda3\lib\site-packages (from nltk) (2024.9.11)
Requirement already satisfied: tqdm in c:\users\rajpu\anaconda3\lib\site-packages (from nltk) (4.66.5)
Requirement already satisfied: colorama in c:\users\rajpu\anaconda3\lib\site-packages (from click->nltk) (0.4.6)
Note: you may need to restart the kernel to use updated packages.
```

[3]:
```python
import nltk
nltk.download('averaged_perceptron_tagger')
```

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

[3]: True

[11]:
```python
import nltk
from nltk.tokenize import word_tokenize
from nltk import pos_tag
from nltk.chunk import RegexpParser

# Download necessary NLTK data
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')

#Sample sentence
sentence = "The quick brown fox jumps over the lazy dog"

#Tokenize the sentence
tokens = word_tokenize(sentence)

#Get the part-of-speech tags for each token
pos_tags = pos_tag(tokens)

# Define the chunking pattern (rules for noun and verb phrases)
chunk_grammar = """
NP: {<DT>?<JJ>*<NN>} # Noun Phrase
VP: {<VB.*>} # Verb Phrase
"""
#Create the chunk parser with the defined grammar
chunk_parser = RegexpParser(chunk_grammar)

#Parse the POS tagged tokens to identify chunks
chunked_tree = chunk_parser.parse(pos_tags)

# Display the chunked tree
chunked_tree.pretty_print()
```

```
                                    S
    ________________________________|_____________________________
   |              NP                NP        VP              NP
   |       _______|________         |         |        _______|______
over/IN The/DT quick/JJ brown/NN fox/NN jumps/VBZ the/DT lazy/JJ dog/NN

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

[ ]: