```
# Install necessary libraries: nltk, spacy, and matplotlib.
# Download the English language model for spaCy.
!pip install nltk spacy matplotlib
!python -m spacy download en_core_web_sm
```

```
Requirement already satisfied: nltk in /usr/local/lib/python3.12/dist-packages (3.9.1)
Requirement already satisfied: spacy in /usr/local/lib/python3.12/dist-packages (3.8.11)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.12/dist-packages (3.10.0)
Requirement already satisfied: click in /usr/local/lib/python3.12/dist-packages (from nltk) (8.3.1)
Requirement already satisfied: joblib in /usr/local/lib/python3.12/dist-packages (from nltk) (1.5.3)
Requirement already satisfied: regex>=2021.8.3 in /usr/local/lib/python3.12/dist-packages (from nltk) (2025.11.3)
Requirement already satisfied: tqdm in /usr/local/lib/python3.12/dist-packages (from nltk) (4.67.1)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in /usr/local/lib/python3.12/dist-packages (from spacy) (3.0.12)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (1.0.5)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (1.0.15)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.0.13)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.12/dist-packages (from spacy) (3.0.12)
Requirement already satisfied: thinc<8.4.0,>=8.3.4 in /usr/local/lib/python3.12/dist-packages (from spacy) (8.3.10)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.12/dist-packages (from spacy) (1.1.3)
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.5.2)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.0.10)
Requirement already satisfied: weasel<0.5.0,>=0.4.2 in /usr/local/lib/python3.12/dist-packages (from spacy) (0.4.3)
Requirement already satisfied: typer-slim<1.0.0,>=0.3.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (0.21.1)
Requirement already satisfied: numpy>=1.19.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.0.2)
Requirement already satisfied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.32.4)
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.1
Requirement already satisfied: jinja2 in /usr/local/lib/python3.12/dist-packages (from spacy) (3.1.6)
Requirement already satisfied: setuptools in /usr/local/lib/python3.12/dist-packages (from spacy) (75.2.0)
Requirement already satisfied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (25.0)
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (1.3.3)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (0.12.1)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (4.61.1)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (1.4.9)
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (11.3.0)
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (3.3.1)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (2.9.0.post0)
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!=1.8.1,<3.
Requirement already satisfied: pydantic-core==2.41.4 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!=1.8.1,<3.0
Requirement already satisfied: typing-extensions>=4.14.1 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!=1.8.1,
Requirement already satisfied: typing-inspection>=0.4.2 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!=1.8.1,<
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.7->matplotlib) (1.17
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0->spacy) (3.
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0->spac
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0->spac
Requirement already satisfied: blis<1.4.0,>=1.3.0 in /usr/local/lib/python3.12/dist-packages (from thinc<8.4.0,>=8.3.4->spacy) (
Requirement already satisfied: confection<1.0.0,>=0.0.1 in /usr/local/lib/python3.12/dist-packages (from thinc<8.4.0,>=8.3.4->sp
Requirement already satisfied: cloudpathlib<1.0.0,>=0.7.0 in /usr/local/lib/python3.12/dist-packages (from weasel<0.5.0,>=0.4.2-
Requirement already satisfied: smart-open<8.0.0,>=5.2.1 in /usr/local/lib/python3.12/dist-packages (from weasel<0.5.0,>=0.4.2->s
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-packages (from jinja2->spacy) (3.0.3)
Requirement already satisfied: wrapt in /usr/local/lib/python3.12/dist-packages (from smart-open<8.0.0,>=5.2.1->weasel<0.5.0,>=0
Collecting en-core-web-sm==3.8.0
  Downloading https://github.com/explosion/spacy-models/releases/download/en_core_web_sm-3.8.0/en_core_web_sm-3.8.0-py3-none-any
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 12.8/12.8 MB 88.4 MB/s eta 0:00:00
✓ 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.
```

```
# Define the essay text as a multi-line string.
essay = """
Artificial intelligence is transforming modern education.
It improves learning processes and supports students in understanding complex concepts.
Researchers develop algorithms to analyze large datasets and produce accurate results.
"""
```

```
# Print the defined essay to display its content.
print(essay)
```

```

Artificial intelligence is transforming modern education.
It improves learning processes and supports students in understanding complex concepts.
Researchers develop algorithms to analyze large datasets and produce accurate results.

```

```
# Import the NLTK library for natural language processing.
# Download the 'punkt' tokenizer models for word tokenization.
import nltk
nltk.download('punkt')
nltk.download('punkt_tab')

# Tokenize the essay into individual words and punctuation marks.
tokens = nltk.word_tokenize(essay)
print(tokens)
```

```
['Artificial', 'intelligence', 'is', 'transforming', 'modern', 'education', '.', 'It', 'improves', 'learning', 'processes', 'and
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Package punkt is already up-to-date!
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Package punkt_tab is already up-to-date!
```

```
# Download the 'averaged_perceptron_tagger_eng' model for Part-of-Speech tagging.
nltk.download('averaged_perceptron_tagger_eng')

# Perform Part-of-Speech tagging on the NLTK tokens.
nltk_pos = nltk.pos_tag(tokens)
print(nltk_pos)
```

```
[('Artificial', 'JJ'), ('intelligence', 'NN'), ('is', 'VBZ'), ('transforming', 'VBG'), ('modern', 'JJ'), ('education', 'NN'), ('
[nltk_data] Downloading package averaged_perceptron_tagger_eng to
[nltk_data]     /root/nltk_data...
[nltk_data]   Package averaged_perceptron_tagger_eng is already up-to-
[nltk_data]       date!
```

```
# Import the spaCy library.
# Load the English language model for spaCy.
import spacy
nlp = spacy.load('en_core_web_sm')
# Process the essay text with the spaCy model to create a Doc object.
doc = nlp(essay)

# Extract all nouns from the processed document.
nouns = [token.text for token in doc if token.pos_ == "NOUN"]
# Extract all verbs from the processed document.
verbs = [token.text for token in doc if token.pos_ == "VERB"]

# Print the extracted nouns and verbs.
print("Nouns:", nouns)
print("Verbs:", verbs)
```

```
Nouns: ['intelligence', 'education', 'processes', 'students', 'concepts', 'Researchers', 'algorithms', 'datasets', 'results']
Verbs: ['transforming', 'improves', 'learning', 'supports', 'understanding', 'develop', 'analyze', 'produce']
```

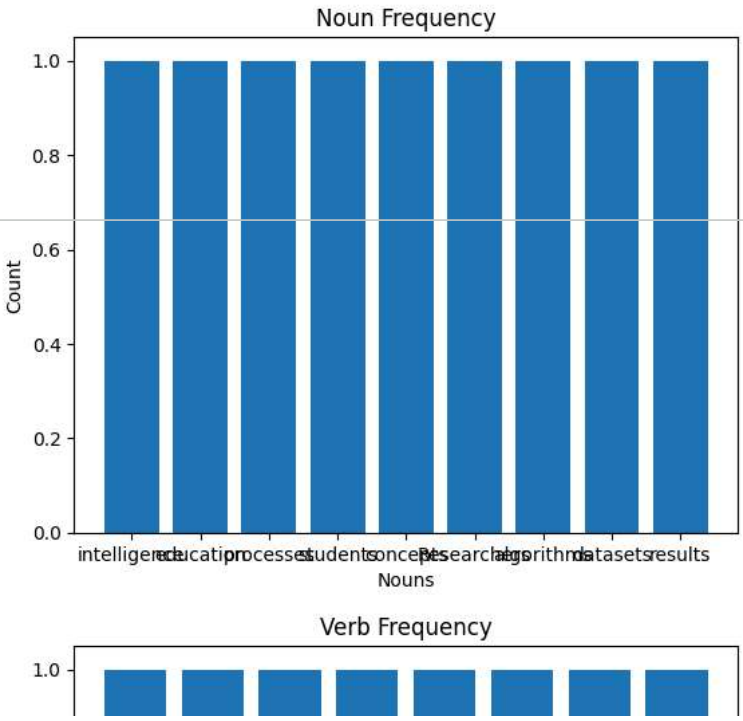
```
# Import the Counter class for counting frequencies.
from collections import Counter

# Calculate the frequency of each noun.
noun_freq = Counter(nouns)
# Calculate the frequency of each verb.
verb_freq = Counter(verbs)

# Print the noun and verb frequencies.
print("Noun Frequency:", noun_freq)
print("Verb Frequency:", verb_freq)
```

```
Noun Frequency: Counter({'intelligence': 1, 'education': 1, 'processes': 1, 'students': 1, 'concepts': 1, 'Researchers': 1, 'alg
Verb Frequency: Counter({'transforming': 1, 'improves': 1, 'learning': 1, 'supports': 1, 'understanding': 1, 'develop': 1, 'anal
```

```
# Import the pandas library for data manipulation.
import pandas as pd

# Create a DataFrame from noun frequencies.
noun_df = pd.DataFrame(noun_freq.items(), columns=["Noun", "Frequency"])
# Create a DataFrame from verb frequencies.
verb_df = pd.DataFrame(verb_freq.items(), columns=["Verb", "Frequency"])

# Display both DataFrames.
noun_df, verb_df
```

```
(           Noun  Frequency
 0  intelligence          1
 1     education          1
 2     processes          1
 3      students          1
 4      concepts          1
 5   Researchers          1
 6    algorithms          1
 7      datasets          1
 8       results          1,
             Verb  Frequency
 0   transforming          1
 1       improves          1
 2       learning          1
 3       supports          1
 4  understanding          1
 5        develop          1
 6        analyze          1
 7        produce          1)
```

```
# Import the matplotlib library for plotting.
import matplotlib.pyplot as plt

# Create a bar chart for noun frequencies.
plt.figure()
plt.bar(noun_df["Noun"], noun_df["Frequency"])
plt.title("Noun Frequency")
plt.xlabel("Nouns")
plt.ylabel("Count")
plt.show()

# Create a bar chart for verb frequencies.
plt.figure()
plt.bar(verb_df["Verb"], verb_df["Frequency"])
plt.title("Verb Frequency")
plt.xlabel("Verbs")
plt.ylabel("Count")
plt.show()
```


Noun Frequency
Count
1.0
0.8
0.6
0.4
0.2
0.0
intelligence
education
processes
students
concepts
Researchers
algorithms
datasets
results
Nouns

Verb Frequency
1.0