

```
!pip install nltk spacy matplotlib seaborn pandas
!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: seaborn in /usr/local/lib/python3.12/dist-packages (0.13.2)
Requirement already satisfied: pandas in /usr/local/lib/python3.12/dist-packages (2.2.2)
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)
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)
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.12/dist-packages (from spacy)
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.12/dist-packages (from spacy)
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.0.7)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.12/dist-packages (from spacy) (3.0.9)
Requirement already satisfied: thinc<8.4.0,>=8.3.4 in /usr/local/lib/python3.12/dist-packages (from spacy) (8.3.11)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.12/dist-packages (from spacy) (1.0.1)
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.4.8)
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) (0.4.2)
Requirement already satisfied: typer-slim<1.0.0,>=0.3.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (0.10.0)
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) (2.32.0)
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) (2.10.6)
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) (1.3.0)
Requirement already satisfied: cyclor>=0.10 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (0.12.0)
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (4.55.0)
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (3.0.7)
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) (3.2.0)
Requirement already satisfied: python-dateutil>=2.7 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (2.9.0)
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.2)
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.3)
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.12/dist-packages (from pydantic) (0.7.0)
Requirement already satisfied: pydantic-core==2.41.4 in /usr/local/lib/python3.12/dist-packages (from pydantic) (2.41.4)
```

Requirement already satisfied: typing-extensions>=4.14.1 in /usr/local/lib/python3.12/dist-packages (from pyda
Requirement already satisfied: typing-inspection>=0.4.2 in /usr/local/lib/python3.12/dist-packages (from pydar
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.7-
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requ
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests<3.
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests<3.
Requirement already satisfied: blis<1.4.0,>=1.3.0 in /usr/local/lib/python3.12/dist-packages (from thinc<8.4.6
Requirement already satisfied: confection<1.0.0,>=0.0.1 in /usr/local/lib/python3.12/dist-packages (from thinc
Requirement already satisfied: cloudpathlib<1.0.0,>=0.7.0 in /usr/local/lib/python3.12/dist-packages (from wea
Requirement already satisfied: smart-open<8.0.0,>=5.2.1 in /usr/local/lib/python3.12/dist-packages (from wease
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-packages (from jinja2->spacy)
Requirement already satisfied: wrapt in /usr/local/lib/python3.12/dist-packages (from smart-open<8.0.0,>=5.2.1
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.tar.gz 12.8/12.8 MB 91.5 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

```
import nltk
import spacy
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from nltk.tokenize import word_tokenize
from collections import Counter
```

```
nltk.download('punkt')
nltk.download('averaged_perceptron_tagger')
```

```
[nltk_data] Downloading package punkt to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt.zip.
[nltk_data] Downloading package averaged_perceptron_tagger to
[nltk_data]   /root/nltk_data...
[nltk_data]   Unzipping taggers/averaged_perceptron_tagger.zip.
True
```

```
essay_text = """
Artificial Intelligence has transformed modern computing by enabling machines
to learn from data and make intelligent decisions. Researchers argue that AI
improves efficiency, enhances accuracy, and supports innovation across industries.
However, ethical concerns arise when algorithms influence human behavior without
transparency. Therefore, responsible development and regulation of AI systems
remain essential for sustainable technological progress.
"""
```

```
print(essay_text)
```

```
Artificial Intelligence has transformed modern computing by enabling machines
to learn from data and make intelligent decisions. Researchers argue that AI
improves efficiency, enhances accuracy, and supports innovation across industries.
However, ethical concerns arise when algorithms influence human behavior without
transparency. Therefore, responsible development and regulation of AI systems
remain essential for sustainable technological progress.
```

```
nltk.download('punkt_tab')
tokens = word_tokenize(essay_text)
print(tokens)
```

```
[nltk_data] Downloading package punkt_tab to /root/nltk_data...
[nltk_data]   Unzipping tokenizers/punkt_tab.zip.
['Artificial', 'Intelligence', 'has', 'transformed', 'modern', 'computing', 'by', 'enabling', 'machines', 'to',
```

```
nltk.download('averaged_perceptron_tagger_eng')
nltk_pos_tags = nltk.pos_tag(tokens)
nltk_pos_tags
```

```
[nltk_data] Downloading package averaged_perceptron_tagger_eng to
[nltk_data]   /root/nltk_data...
[nltk_data]   Unzipping taggers/averaged_perceptron_tagger_eng.zip.
[('Artificial', 'JJ'),
 ('Intelligence', 'NNP'),
 ('has', 'VBZ'),
```

```
('transformed', 'VBN'),
('modern', 'JJ'),
('computing', 'NN'),
('by', 'IN'),
('enabling', 'VBG'),
('machines', 'NNS'),
('to', 'TO'),
('learn', 'VB'),
('from', 'IN'),
('data', 'NNS'),
('and', 'CC'),
('make', 'VB'),
('intelligent', 'JJ'),
('decisions', 'NNS'),
('.', '.'),
('Researchers', 'NNP'),
('argue', 'VBP'),
('that', 'IN'),
('AI', 'NNP'),
('improves', 'VBZ'),
('efficiency', 'NN'),
(',', ','),
('enhances', 'VBZ'),
('accuracy', 'NN'),
(',', ','),
('and', 'CC'),
('supports', 'VBZ'),
('innovation', 'NN'),
('across', 'IN'),
('industries', 'NNS'),
('.', '.'),
('However', 'RB'),
(',', ','),
('ethical', 'JJ'),
('concerns', 'NNS'),
('arise', 'VB'),
('when', 'WRB'),
('algorithms', 'JJ'),
('influence', 'NN'),
('human', 'JJ'),
('behavior', 'NN'),
('without', 'IN'),
```

```
('transparency', 'NN'),  
('.', '.'),  
('Therefore', 'RB'),  
(',', ','),  
('responsible', 'JJ'),  
('development', 'NN'),  
('and', 'CC'),  
('regulation', 'NN'),  
('of', 'IN'),
```

```
nlp = spacy.load("en_core_web_sm")  
doc = nlp(essay_text)
```

```
spacy_pos_tags = [(token.text, token.pos_) for token in doc]  
spacy_pos_tags
```

```
[('\n', 'SPACE'),  
('Artificial', 'PROPN'),  
('Intelligence', 'PROPN'),  
('has', 'AUX'),  
('transformed', 'VERB'),  
('modern', 'ADJ'),  
('computing', 'NOUN'),  
('by', 'ADP'),  
('enabling', 'VERB'),  
('machines', 'NOUN'),  
('\n', 'SPACE'),  
('to', 'PART'),  
('learn', 'VERB'),  
('from', 'ADP'),  
('data', 'NOUN'),  
('and', 'CCONJ'),  
('make', 'VERB'),  
('intelligent', 'ADJ'),  
('decisions', 'NOUN'),  
('.', 'PUNCT'),  
('Researchers', 'NOUN'),  
('argue', 'VERB'),  
('that', 'SCONJ'),  
('AI', 'PROPN'),  
('\n', 'SPACE'),
```

```
('improves', 'VERB'),
('efficiency', 'NOUN'),
(',', 'PUNCT'),
('enhances', 'VERB'),
('accuracy', 'NOUN'),
(',', 'PUNCT'),
('and', 'CCONJ'),
('supports', 'VERB'),
('innovation', 'NOUN'),
('across', 'ADP'),
('industries', 'NOUN'),
('.', 'PUNCT'),
('\n', 'SPACE'),
('However', 'ADV'),
(',', 'PUNCT'),
('ethical', 'ADJ'),
('concerns', 'NOUN'),
('arise', 'VERB'),
('when', 'SCONJ'),
('algorithms', 'NOUN'),
('influence', 'VERB'),
('human', 'ADJ'),
('behavior', 'NOUN'),
('without', 'ADP'),
('\n', 'SPACE'),
('transparency', 'NOUN'),
('.', 'PUNCT'),
('Therefore', 'ADV'),
(',', 'PUNCT'),
('responsible', 'ADJ'),
('development', 'NOUN'),
('and', 'CCONJ'),
```

Start coding or [generate](#) with AI.

```
df_pos_comparison = pd.DataFrame({
    'Token': [token for token, pos in nltk_pos_tags],
    'NLTK_POS': [pos for token, pos in nltk_pos_tags]
})
```



```

spacy_tokens = [token.text for token in doc if not token.is_space]
spacy_pos = [token.pos_ for token in doc if not token.is_space]

df_spacy = pd.DataFrame({
    'Token': spacy_tokens,
    'SpaCy_POS': spacy_pos
})

df_comparison = pd.merge(df_pos_comparison, df_spacy, on='Token', how='outer')
display(df_comparison)

```

	Token	NLTK_POS	SpaCy_POS	
0	,	,	PUNCT	
1	,	,	PUNCT	
2	,	,	PUNCT	
3	,	,	PUNCT	
4	,	,	PUNCT	
...	
90	to	TO	PART	
91	transformed	VBN	VERB	
92	transparency	NN	NOUN	
93	when	WRB	SCONJ	
94	without	IN	ADP	

95 rows × 3 columns

Next steps:

[Generate code with df_comparison](#)

[New interactive sheet](#)

```

nouns = [token.text.lower() for token in doc if token.pos_ == "NOUN"]
verbs = [token.text.lower() for token in doc if token.pos_ == "VERB"]

```

```
print("Extracted Nouns:", nouns)
print("Extracted Verbs:", verbs)
```

```
Extracted Nouns: ['computing', 'machines', 'data', 'decisions', 'researchers', 'efficiency', 'accuracy', 'innov
Extracted Verbs: ['transformed', 'enabling', 'learn', 'make', 'argue', 'improves', 'enhances', 'supports', 'ari
```

```
noun_freq = Counter(nouns)
verb_freq = Counter(verbs)
```



```
noun_freq, verb_freq
```

```
(Counter({'computing': 1,
          'machines': 1,
          'data': 1,
          'decisions': 1,
          'researchers': 1,
          'efficiency': 1,
          'accuracy': 1,
          'innovation': 1,
          'industries': 1,
          'concerns': 1,
          'algorithms': 1,
          'behavior': 1,
          'transparency': 1,
          'development': 1,
          'regulation': 1,
          'systems': 1,
          'progress': 1}),
 Counter({'transformed': 1,
          'enabling': 1,
          'learn': 1,
          'make': 1,
          'argue': 1,
          'improves': 1,
          'enhances': 1,
          'supports': 1,
          'arise': 1,
          'influence': 1,
          'remain': 1}))
```



```
noun_df = pd.DataFrame(noun_freq.items(), columns=["Noun", "Frequency"])
verb_df = pd.DataFrame(verb_freq.items(), columns=["Verb", "Frequency"])
```

noun_df


	Noun	Frequency	
0	computing	1	
1	machines	1	
2	data	1	
3	decisions	1	
4	researchers	1	
5	efficiency	1	
6	accuracy	1	
7	innovation	1	
8	industries	1	
9	concerns	1	
10	algorithms	1	
11	behavior	1	
12	transparency	1	
13	development	1	
14	regulation	1	
15	systems	1	
16	progress	1	

Next steps:

[Generate code with noun_df](#)

[New interactive sheet](#)

verb_df

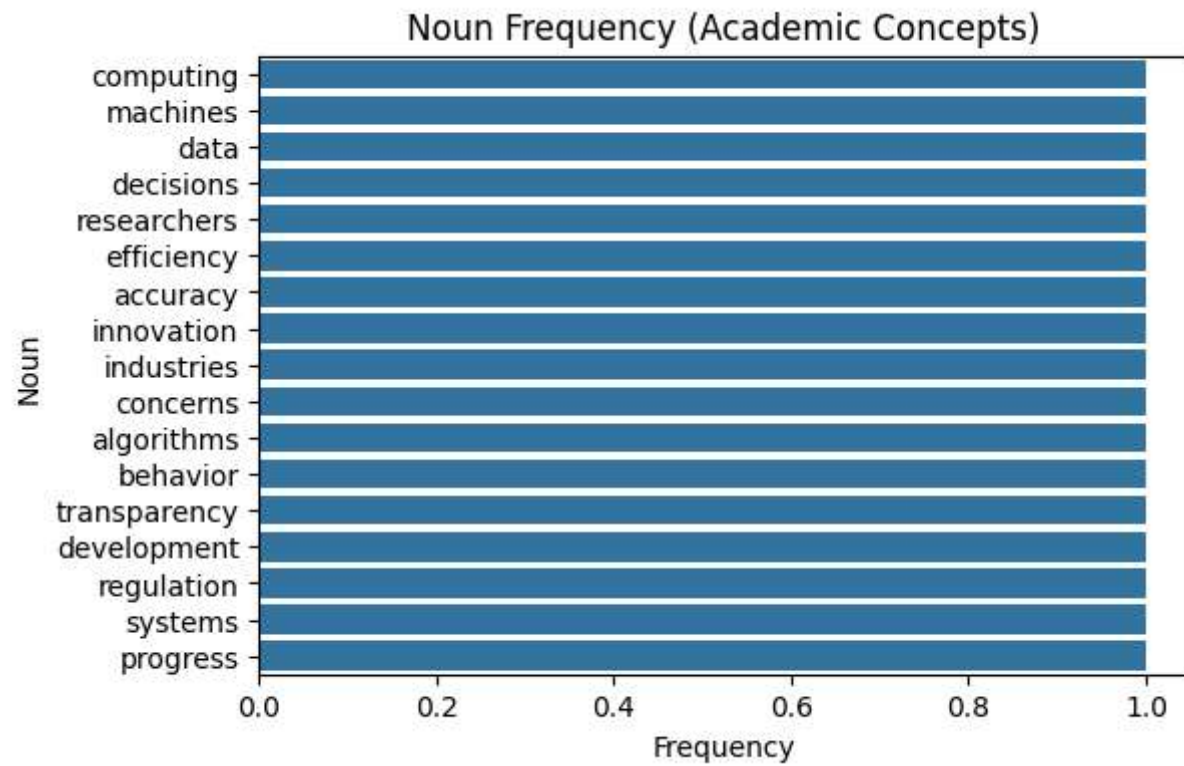
	Verb	Frequency	
0	transformed	1	
1	enabling	1	
2	learn	1	
3	make	1	
4	argue	1	
5	improves	1	
6	enhances	1	
7	supports	1	
8	arise	1	
9	influence	1	
10	remain	1	

Next steps:

[Generate code with verb_df](#)

[New interactive sheet](#)

```
plt.figure(figsize=(6,4))
sns.barplot(x="Frequency", y="Noun", data=noun_df)
plt.title("Noun Frequency (Academic Concepts)")
plt.show()
```



```
plt.figure(figsize=(6,4))
sns.barplot(x="Frequency", y="Verb", data=verb_df)
plt.title("Verb Frequency (Arguments)")
plt.show()
```

Verb Frequency (Arguments)

