```
!pip install spacy pandas matplotlib
!python -m spacy download en_core_web_sm
```

```
Requirement already satisfied: spacy in /usr/local/lib/python3.12/dist-packages (3.8.11)
Requirement already satisfied: pandas in /usr/local/lib/python3.12/dist-packages (2.2.2)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.12/dist-packages (3.10.0)
Requirement already satisfied: spacy-legacy<3.1.0,>=3.0.11 in /usr/local/lib/python3.12/dist-packages (from
Requirement already satisfied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.12/dist-packages (from
Requirement already satisfied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.12/dist-packages (from s
Requirement already satisfied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.12/dist-packages (from spacy)
Requirement already satisfied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.12/dist-packages (from spacy
Requirement already satisfied: thinc<8.4.0,>=8.3.4 in /usr/local/lib/python3.12/dist-packages (from spacy)
Requirement already satisfied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.12/dist-packages (from spacy)
Requirement already satisfied: srsly<3.0.0,>=2.4.3 in /usr/local/lib/python3.12/dist-packages (from spacy)
Requirement already satisfied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.12/dist-packages (from spa
Requirement already satisfied: weasel<0.5.0,>=0.4.2 in /usr/local/lib/python3.12/dist-packages (from spacy)
Requirement already satisfied: typer-slim<1.0.0,>=0.3.0 in /usr/local/lib/python3.12/dist-packages (from sp
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python3.12/dist-packages (from spacy)
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 spa
Requirement already satisfied: pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 in /usr/local/lib/python3.12/dist-packa
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.
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from pand
Requirement already satisfied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.
Requirement already satisfied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas) (202
Requirement already satisfied: contourpy>=1.0.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (0
Requirement already satisfied: fonttools>=4.22.0 in /usr/local/lib/python3.12/dist-packages (from matplotli
Requirement already satisfied: kiwisolver>=1.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotli
Requirement already satisfied: pillow>=8 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (11.3
Requirement already satisfied: pyparsing>=2.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib
Requirement already satisfied: annotated-types>=0.6.0 in /usr/local/lib/python3.12/dist-packages (from pyda
Requirement already satisfied: pydantic-core==2.41.4 in /usr/local/lib/python3.12/dist-packages (from pydan
Requirement already satisfied: typing-extensions>=4.14.1 in /usr/local/lib/python3.12/dist-packages (from p
Requirement already satisfied: typing-inspection>=0.4.2 in /usr/local/lib/python3.12/dist-packages (from py
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from re
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
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests
Requirement already satisfied: blis<1.4.0,>=1.3.0 in /usr/local/lib/python3.12/dist-packages (from thinc<8.
Requirement already satisfied: confection<1.0.0,>=0.0.1 in /usr/local/lib/python3.12/dist-packages (from th
Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.12/dist-packages (from typer-slim<1.0
Requirement already satisfied: cloudpathlib<1.0.0,>=0.7.0 in /usr/local/lib/python3.12/dist-packages (from
Requirement already satisfied: smart-open<8.0.0,>=5.2.1 in /usr/local/lib/python3.12/dist-packages (from we
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-packages (from jinja2->spa
Requirement already satisfied: wrapt in /usr/local/lib/python3.12/dist-packages (from smart-open<8.0.0,>=5.
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_
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 12.8/12.8 MB 111.7 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.
```

```
import spacy
import pandas as pd
import matplotlib.pyplot as plt
from collections import Counter
from spacy.matcher import Matcher
```

```
nlp = spacy.load("en_core_web_sm")
df = pd.read_csv("/content/arxiv_data.csv")
print(df.head())
```

```
                                              titles  \
0  Survey on Semantic Stereo Matching / Semantic ...
1  FUTURE-AI: Guiding Principles and Consensus Re...
2  Enforcing Mutual Consistency of Hard Regions f...
```

```
3  Parameter Decoupling Strategy for Semi-supervi...
4  Background-Foreground Segmentation for Interio...

                                          summaries  \
0  Stereo matching is one of the widely used tech...
1  The recent advancements in artificial intellig...
2  In this paper, we proposed a novel mutual cons...
3  Consistency training has proven to be an advan...
4  To ensure safety in automated driving, the cor...

                        terms
0           ['cs.CV', 'cs.LG']
1  ['cs.CV', 'cs.AI', 'cs.LG']
2           ['cs.CV', 'cs.AI']
3                   ['cs.CV']
4           ['cs.CV', 'cs.LG']
```

```
data = pd.read_csv("arxiv_data.csv")
abstracts = data['summaries'].dropna().head(50)
```

```
docs = [nlp(text) for text in abstracts]
```

```
sample_doc = docs[0]
tokens = [token.text for token in sample_doc]

print(tokens[:30])
```

```
['Stereo', 'matching', 'is', 'one', 'of', 'the', 'widely', 'used', 'techniques', 'for', 'inferring', 'depth
```

```
noun_phrases = []

for doc in docs:
    for chunk in doc.noun_chunks:
        noun_phrases.append(chunk.text.lower())

np_freq = Counter(noun_phrases)
top_noun_phrases = np_freq.most_common(10)

print("Top Noun Phrases:")
for np, count in top_noun_phrases:
    print(np, ":", count)
```

```
Top Noun Phrases:
we : 143
which : 43
that : 34
it : 28
the-art : 23
image segmentation : 14
this paper : 14
this work : 14
semantic segmentation : 11
medical image segmentation : 11
```

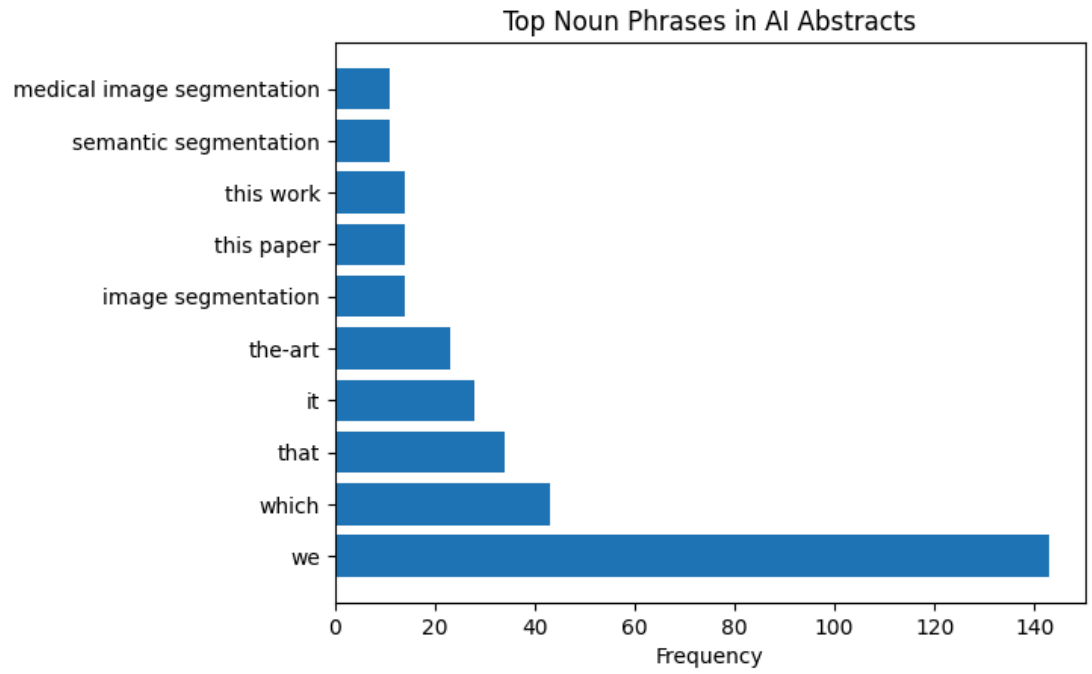
```
entities = []

for doc in docs:
    for ent in doc.ents:
        if ent.label_ in ["ORG", "DATE", "PRODUCT", "GPE"]:
            entities.append(ent.text)

entity_freq = Counter(entities)
print(entity_freq.most_common(10))
```

```
[('3D', 9), ('AI', 6), ('U-Net', 6), ('CNN', 5), ('FSS', 5), ('CT', 4), ('MIA', 4), ('EHT', 3), ('linear',
```

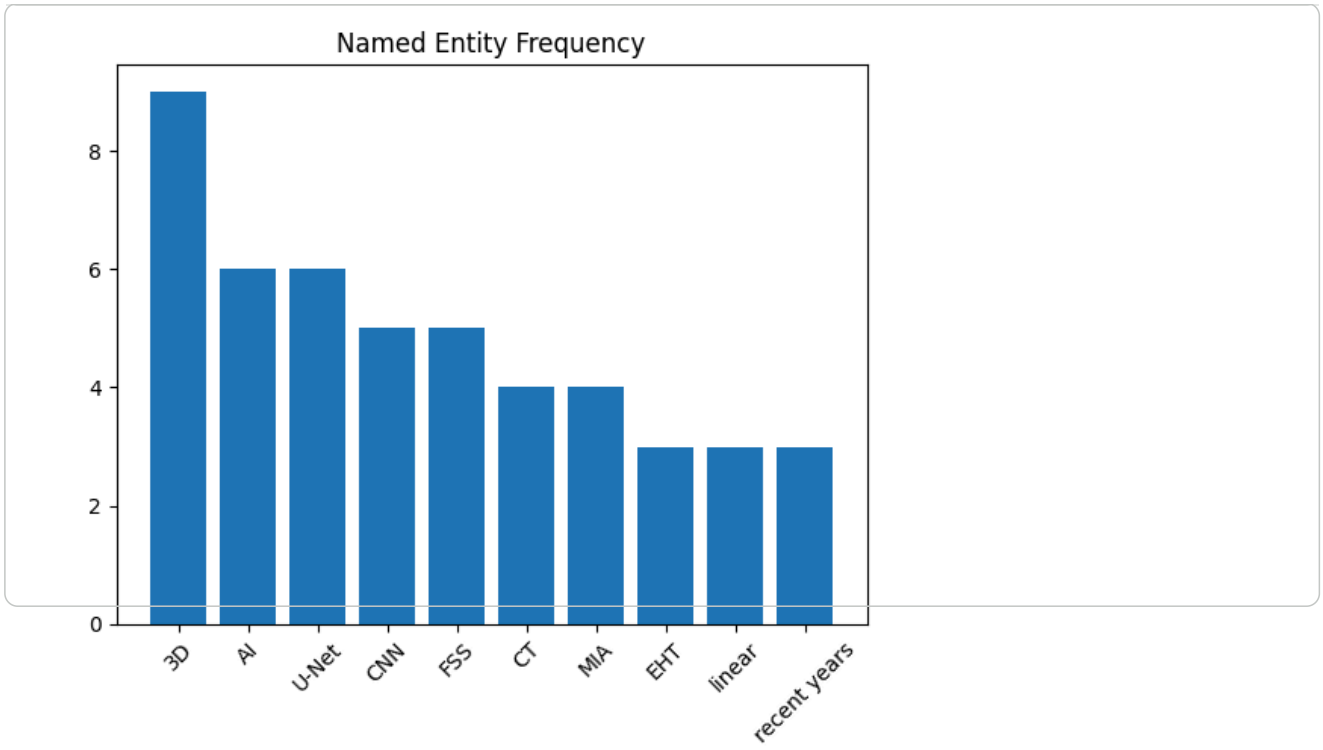
```
matcher = Matcher(nlp.vocab)

# Pattern: Adjective + Noun (e.g., "deep learning")
pattern1 = [{"POS": "ADJ"}, {"POS": "NOUN"}]

# Pattern: Noun + Noun (e.g., "neural network")
pattern2 = [{"POS": "NOUN"}, {"POS": "NOUN"}]

matcher.add("TECH_TERMS", [pattern1, pattern2])

matches = []
for doc in docs:
    for match_id, start, end in matcher(doc):
        matches.append(doc[start:end].text)

print("Sample Technical Terms:")
print(matches[:10])
```

```
Sample Technical Terms:
['stereo images', 'autonomous driving', 'robotic navigation', '3D reconstruction', 'other fields', 'reflect
```

```
phrases, counts = zip(*top_noun_phrases)

plt.figure()
plt.barh(phrases, counts)
plt.xlabel("Frequency")
plt.title("Top Noun Phrases in AI Abstracts")
plt.show()
```




```
entity_items = entity_freq.most_common(10)
entities, ent_counts = zip(*entity_items)

plt.figure()
plt.bar(entities, ent_counts)
plt.xticks(rotation=45)
plt.title("Named Entity Frequency")
plt.show()
```


Named Entity Frequency
0
2
4
6
8
3D
AI
U-Net
CNN
FSS
CT
MIA
EHT
linear
recent years