```
!pip install spacy matplotlib pandas seaborn
!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: matplotlib in /usr/local/lib/python3.12/dist-packages (3.10.0)
Requirement already satisfied: pandas in /usr/local/lib/python3.12/dist-packages (2.2.2)
Requirement already satisfied: seaborn in /usr/local/lib/python3.12/dist-packages (0.13.2)
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: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (4.67.1)
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)
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.post
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!=1.8,!=1.8.1
Requirement already satisfied: pydantic-core==2.41.4 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!=1.8.1,
Requirement already satisfied: typing-extensions>=4.14.1 in /usr/local/lib/python3.12/dist-packages (from pydantic!=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
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.7->matplotlib) (
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.
Requirement already satisfied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0->spacy)
Requirement already satisfied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0->
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0->
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->spac
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
Requirement already satisfied: click>=8.0.0 in /usr/local/lib/python3.12/dist-packages (from typer-slim<1.0.0,>=0.3.0->spacy
Requirement already satisfied: cloudpathlib<1.0.0,>=0.7.0 in /usr/local/lib/python3.12/dist-packages (from weasel<0.5.0,>=0.
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.
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-packages (from jinja2->spacy) (3.0.3)
Requirement already satisfied: wrap in /usr/local/lib/python3.12/dist-packages (from smart-open<8.0.0,>=5.2.1->weasel<0.5.0
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
  ──────────────────────────────────── 12.8/12.8 MB 75.9 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 pandas as pd
import spacy
from collections import Counter
import matplotlib.pyplot as plt
import seaborn as sns

from spacy.matcher import Matcher
```

```
df = pd.read_csv("/content/arxiv_data.csv", engine='python', on_bad_lines='skip')

# Display dataset structure
df.head()
```

	titles	summaries	terms	⊞
0	Survey on Semantic Stereo Matching / Semantic ...	Stereo matching is one of the widely used tech...	['cs.CV', 'cs.LG']	
1	FUTURE-AI: Guiding Principles and Consensus Re...	The recent advancements in artificial intellig...	['cs.CV', 'cs.AI', 'cs.LG']	
2	Enforcing Mutual Consistency of Hard Regions f...	In this paper, we proposed a novel mutual cons...	['cs.CV', 'cs.AI']	
3	Parameter Decoupling Strategy for Semi-supervi...	Consistency training has proven to be an advan...	['cs.CV']	
4	Background-Foreground Segmentation for Interio...	To ensure safety in automated driving, the cor...	['cs.CV', 'cs.LG']	

Next steps: (Generate code with `df`)   (New interactive sheet)

```python
# Drop missing abstracts
df = df.dropna(subset=['summaries'])

# Use only first 200 abstracts for speed
texts = df['summaries'].head(200).tolist()

print("Number of abstracts used:", len(texts))
```

```
Number of abstracts used: 200
```

```python
nlp = spacy.load("en_core_web_sm")
```

```python
doc = nlp(texts[0])

tokens = [token.text for token in doc if not token.is_punct]

tokens[:20]
```

```
['Stereo',
 'matching',
 'is',
 'one',
 'of',
 'the',
 'widely',
 'used',
 'techniques',
 'for',
 'inferring',
 'depth',
 'from',
 '\n',
 'stereo',
 'images',
 'owing',
 'to',
 'its',
 'robustness']
```

```python
noun_list = []

for text in texts:
    doc = nlp(text)
    for token in doc:
        if token.pos_ == "NOUN":
            noun_list.append(token.lemma_.lower())

print("Total number of nouns:", len(noun_list))
```

```
Total number of nouns: 11622
```

```python
noun_freq = Counter(noun_list)

top_nouns = noun_freq.most_common(10)
top_nouns
```
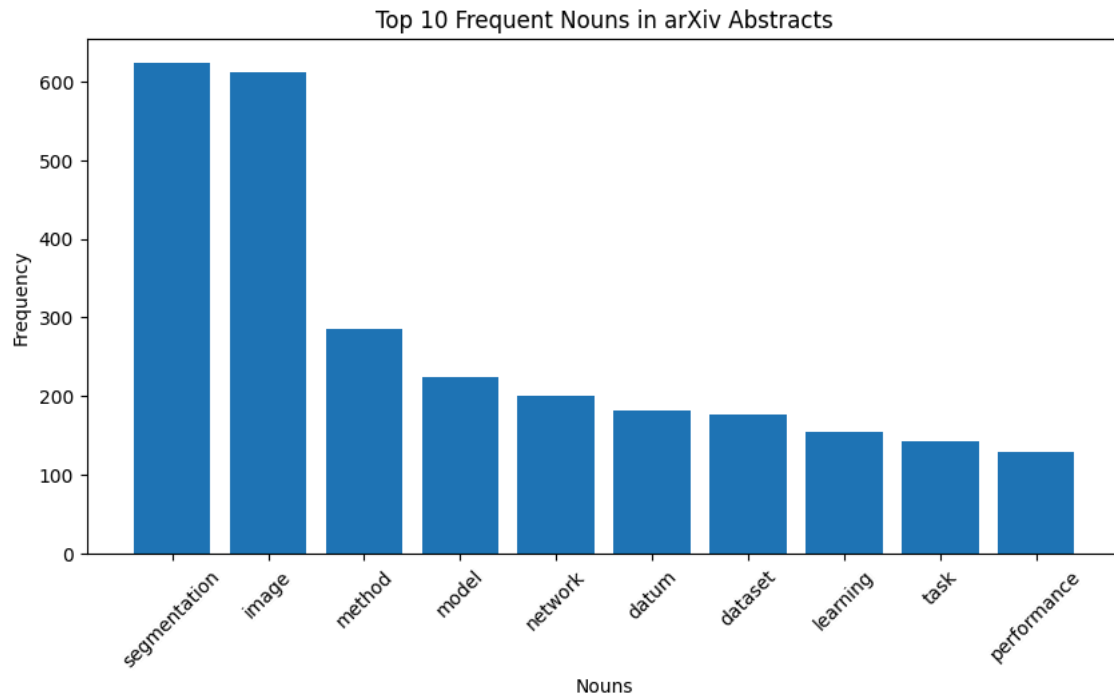
```
[('segmentation', 623),
 ('image', 612),
 ('method', 285),
 ('model', 224),
 ('network', 201),
 ('datum', 181),
 ('dataset', 177),
 ('learning', 155),
```

```
    ('task', 142),
    ('performance', 129)]
```

```python
nouns, counts = zip(*top_nouns)

plt.figure(figsize=(10,5))
plt.bar(nouns, counts)
plt.title("Top 10 Frequent Nouns in arXiv Abstracts")
plt.xlabel("Nouns")
plt.ylabel("Frequency")
plt.xticks(rotation=45)
plt.show()
```


Top 10 Frequent Nouns in arXiv Abstracts

```python
noun_phrases = []

for text in texts:
    doc = nlp(text)
    for chunk in doc.noun_chunks:
        noun_phrases.append(chunk.text.lower())

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

```
[('we', 540),
 ('which', 172),
 ('that', 144),
 ('it', 120),
 ('this paper', 74),
 ('the-art', 72),
 ('our method', 50),
 ('image segmentation', 47),
 ('this work', 47),
 ('medical image segmentation', 37)]
```

```python
entity_list = []

for text in texts:
    doc = nlp(text)
    for ent in doc.ents:
        if ent.label_ in ["ORG", "DATE", "PRODUCT", "GPE"]:
            entity_list.append((ent.text, ent.label_))
```

```python
entity_texts = [ent[0] for ent in entity_list]
entity_freq = Counter(entity_texts)

top_entities = entity_freq.most_common(10)
top_entities
```

```
[('3D', 23),
 ('U-Net', 21),
 ('CNN', 18),
 ('UDA', 14),
 ('CT', 13),
 ('Transformer', 12),
 ('recent years', 9),
 ('Cityscapes', 8),
 ('Dice', 8),
 ('AI', 7)]
```

```python
entities, freq = zip(*top_entities)

plt.figure(figsize=(10,5))
sns.barplot(x=list(freq), y=list(entities))
plt.title("Top Named Entities in arXiv Abstracts")
plt.xlabel("Frequency")
plt.ylabel("Entity")
plt.show()
```


Top Named Entities in arXiv Abstracts
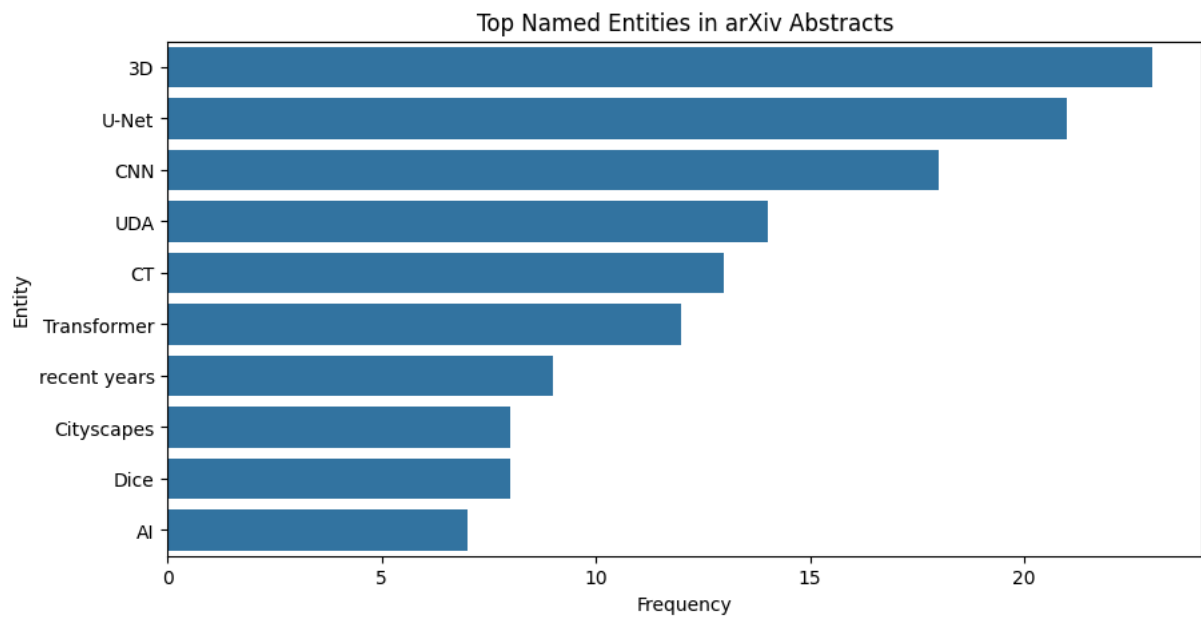
```python
matcher = Matcher(nlp.vocab)

pattern = [
    {"POS": "ADJ"},
    {"POS": "NOUN"}
]

matcher.add("TECH_TERM", [pattern])
```

```python
matches_found = []

for text in texts:
    doc = nlp(text)
    matches = matcher(doc)
    for match_id, start, end in matches:
        matches_found.append(doc[start:end].text.lower())

Counter(matches_found).most_common(10)
```

```
[('medical image', 98),
 ('semantic segmentation', 49),
 ('deep learning', 39),
 ('medical images', 25),
 ('experimental results', 21),
 ('contextual information', 17),
 ('neural networks', 16),
 ('extensive experiments', 16),
 ('neural network', 16),
 ('contrastive learning', 15)]
```