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

```
ied: spacy in /usr/local/lib/python3.12/dist-packages (3.8.11)
ied: pandas in /usr/local/lib/python3.12/dist-packages (2.2.2)
ied: matplotlib in /usr/local/lib/python3.12/dist-packages (3.10.0)
ied: spacy-legacy<3.1.0,>=3.0.11 in /usr/local/lib/python3.12/dist-packages (from spacy) (3.0.12)
ied: spacy-loggers<2.0.0,>=1.0.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (1.0.5)
ied: murmurhash<1.1.0,>=0.28.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (1.0.15)
ied: cymem<2.1.0,>=2.0.2 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.0.13)
ied: preshed<3.1.0,>=3.0.2 in /usr/local/lib/python3.12/dist-packages (from spacy) (3.0.12)
ied: thinc<8.4.0,>=8.3.4 in /usr/local/lib/python3.12/dist-packages (from spacy) (8.3.10)
ied: wasabi<1.2.0,>=0.9.1 in /usr/local/lib/python3.12/dist-packages (from spacy) (1.1.3)
ied: srsly<3.0.0,>=2.4.3 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.5.2)
ied: catalogue<2.1.0,>=2.0.6 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.0.10)
ied: weasel<0.5.0,>=0.4.2 in /usr/local/lib/python3.12/dist-packages (from spacy) (0.4.3)
ied: typer-slim<1.0.0,>=0.3.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (0.21.1)
ied: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (4.67.1)
ied: numpy>=1.19.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.0.2)
ied: requests<3.0.0,>=2.13.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.32.4)
ied: pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4 in /usr/local/lib/python3.12/dist-packages (from spacy) (2.12.3)
ied: jinja2 in /usr/local/lib/python3.12/dist-packages (from spacy) (3.1.6)
ied: setuptools in /usr/local/lib/python3.12/dist-packages (from spacy) (75.2.0)
ied: packaging>=20.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (25.0)
ied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from pandas) (2.9.0.post0)
ied: pytz>=2020.1 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.2)
ied: tzdata>=2022.7 in /usr/local/lib/python3.12/dist-packages (from pandas) (2025.3)
ied: contourpy>=1.0.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (1.3.3)
ied: cycler>=0.10 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (0.12.1)
ied: fonttools>=4.22.0 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (4.61.1)
ied: kiwisolver>=1.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (1.4.9)
ied: pillow>=8 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (11.3.0)
ied: pyparsing>=2.3.1 in /usr/local/lib/python3.12/dist-packages (from matplotlib) (3.3.1)
ied: annotated-types>=0.6.0 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4->spacy
ied: pydantic-core==2.41.4 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4->spacy)
ied: typing-extensions>=4.14.1 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4->spa
ied: typing-inspection>=0.4.2 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!=1.8.1,<3.0.0,>=1.7.4->spa
ied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.8.2->pandas) (1.17.0)
```

```
ied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0->spacy) (3.4.4)
ied: idna<4,>=2.5 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0->spacy) (3.11)
ied: urllib3<3,>=1.21.1 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0->spacy) (2.5.0)
ied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0->spacy) (2026.1.4)
ied: blis<1.4.0,>=1.3.0 in /usr/local/lib/python3.12/dist-packages (from thinc<8.4.0,>=8.3.4->spacy) (1.3.3)
ied: confection<1.0.0,>=0.0.1 in /usr/local/lib/python3.12/dist-packages (from thinc<8.4.0,>=8.3.4->spacy) (0.1.5)
ied: click>=8.0.0 in /usr/local/lib/python3.12/dist-packages (from typer-slim<1.0.0,>=0.3.0->spacy) (8.3.1)
ied: cloudpathlib<1.0.0,>=0.7.0 in /usr/local/lib/python3.12/dist-packages (from weasel<0.5.0,>=0.4.2->spacy) (0.23.0)
ied: smart-open<8.0.0,>=5.2.1 in /usr/local/lib/python3.12/dist-packages (from weasel<0.5.0,>=0.4.2->spacy) (7.5.0)
ied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-packages (from jinja2->spacy) (3.0.3)
ied: wrapt in /usr/local/lib/python3.12/dist-packages (from smart-open<8.0.0,>=5.2.1->weasel<0.5.0,>=0.4.2->spacy) (2.0.1
=3.8.0
ub.com/explosion/spacy-models/releases/download/en_core_web_sm-3.8.0/en_core_web_sm-3.8.0-py3-none-any.whl (12.8 MB)
━━━━━━━━━━━━━━━━━━━━ 12.8/12.8 MB 52.5 MB/s eta 0:00:00
on successful
ge via spacy.load('en_core_web_sm')
dencies
 Colab notebook, you may need to restart Python in
age's dependencies. You can do this by selecting the
rt runtime' option.
```

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

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

```
import pandas as pd
df = pd.read_csv("arxiv_data.csv", engine='python', on_bad_lines='skip')
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']

```
# Find abstract column
abstract_col = None
for col in df.columns:
    if "abstract" in col.lower() or "summaries" in col.lower():
        abstract_col = col
        break

if abstract_col is None:
    print("Error: No suitable column found for abstracts. Available columns:")
    print(df.columns.tolist())
else:
    abstracts = df[abstract_col].dropna().astype(str).tolist()
    print("Total abstracts:", len(abstracts))
```

```
Total abstracts: 11171
```

```
doc = nlp(abstracts[0])
tokens = [token.text for token in doc]
print(tokens)
```

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

```
matcher = Matcher(nlp.vocab)

tech_pattern = [
    {"POS": "ADJ", "OP": "+"},
    {"POS": "NOUN", "OP": "+"}
]

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

```
import spacy
import pandas as pd
from spacy.matcher import Matcher

nlp = spacy.load("en_core_web_sm")

# Define matcher and tech_pattern (brought into this cell for self-containment)
matcher = Matcher(nlp.vocab)
tech_pattern = [
    {"POS": "ADJ", "OP": "+"},
    {"POS": "NOUN", "OP": "+"}
]
matcher.add("TECH_TERM", [tech_pattern])

# Define abstracts (brought into this cell, assuming 'summaries' column and df exists)
abstract_col = 'summaries' # Based on previous successful identification in IipNuj8KNMYu
abstracts = df[abstract_col].dropna().astype(str).tolist()

noun_phrases = []
entities = []
tech_terms = []

for doc in nlp.pipe(abstracts[:300], batch_size=20):
```

```
    # Noun phrases
    for chunk in doc.noun_chunks:
        noun_phrases.append(chunk.text.lower())

    # Named entities
    for ent in doc.ents:
        entities.append(ent.label_)

    # Matcher patterns
    matches = matcher(doc)
    for match_id, start, end in matches:
        span = doc[start:end]
        tech_terms.append(span.text.lower())
```

```
np_counter = Counter(noun_phrases)
top_np = np_counter.most_common(15)

print("\nTop Noun Phrases:\n")
for np, freq in top_np:
    print(np, ":", freq)
```

```
Top Noun Phrases:

we : 795
which : 257
that : 190
it : 176
the-art : 115
this paper : 112
image segmentation : 79
our method : 71
this work : 59
- : 55
segmentation : 42
medical image segmentation : 41
semantic segmentation : 40
```

```
this : 40
training : 38
```

```
ent_counter = Counter(entities)
top_entities = ent_counter.most_common(10)

print("\nEntity Frequency:\n")
for ent, freq in top_entities:
    print(ent, ":", freq)
```

```
Entity Frequency:

ORG : 787
CARDINAL : 461
PERSON : 101
ORDINAL : 92
PERCENT : 79
GPE : 67
NORP : 48
DATE : 47
PRODUCT : 24
MONEY : 13
```

```
tech_counter = Counter(tech_terms)
top_tech = tech_counter.most_common(15)

print("\nTop Technical Term Patterns:\n")
for term, freq in top_tech:
    print(term, ":", freq)
```

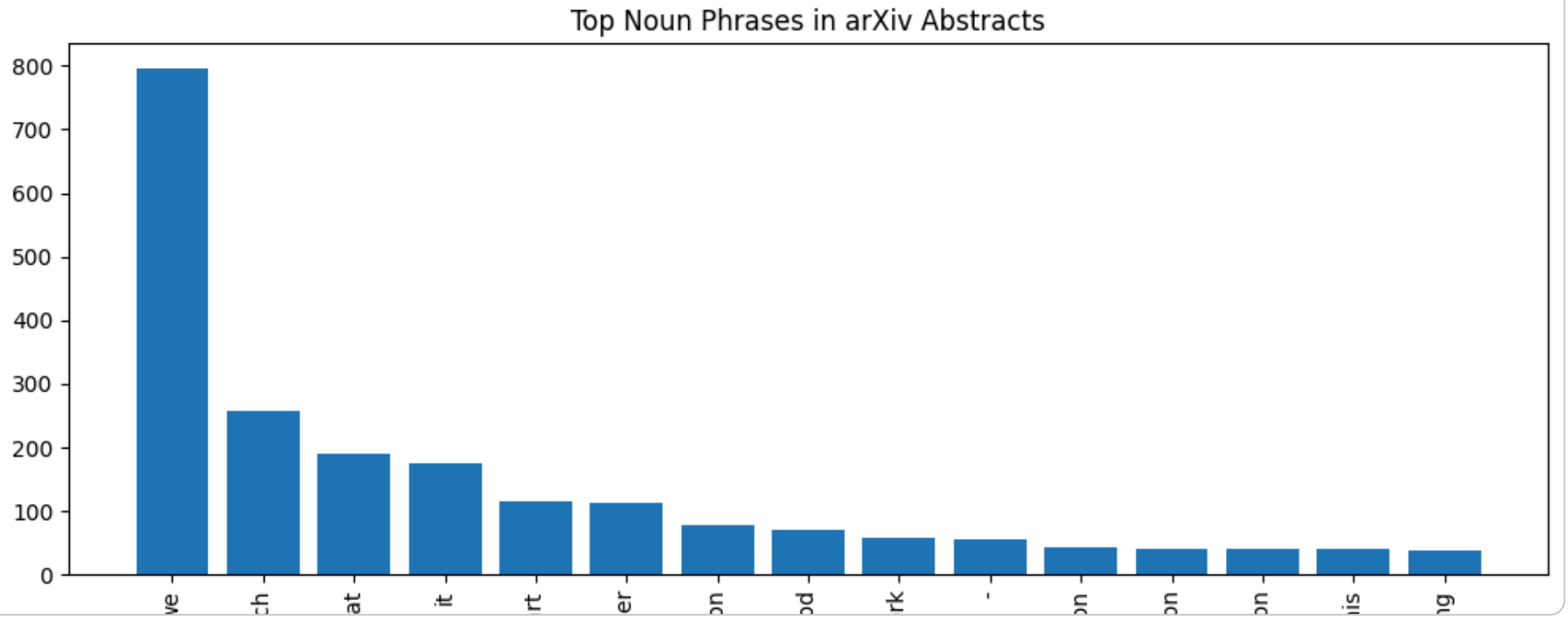
```
Top Technical Term Patterns:

medical image : 133
medical image segmentation : 88
```

```
semantic segmentation : 70
deep learning : 69
medical images : 33
experimental results : 32
extensive experiments : 30
neural networks : 26
neural network : 23
semantic image : 22
medical imaging : 20
unlabeled data : 20
contrastive learning : 19
supervised learning : 19
active learning : 18
```

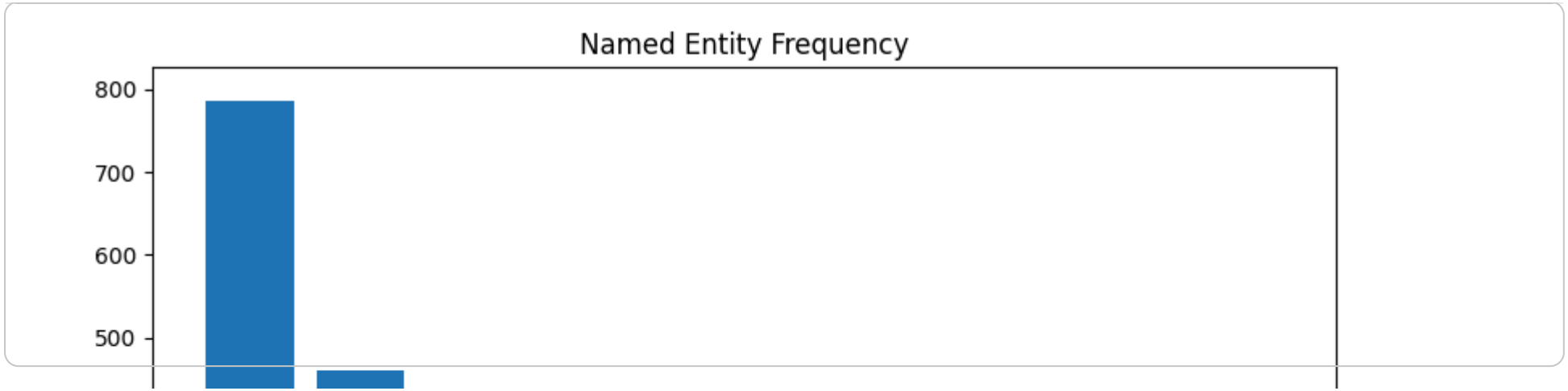
```
phrases, counts = zip(*top_np)

plt.figure(figsize=(10,6))
plt.bar(phrases, counts)
plt.xticks(rotation=90)
plt.title("Top Noun Phrases in arXiv Abstracts")
plt.tight_layout()
plt.show()
```




```
labels, counts = zip(*top_entities)

plt.figure(figsize=(8,5))
plt.bar(labels, counts)
plt.title("Named Entity Frequency")
plt.tight_layout()
plt.show()
```


Named Entity Frequency
800
700
600
500