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

!pip install spacy nltk matplotlib pandas
!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: nltk in /usr/local/lib/python3.12/dist-
packages (3.9.1)
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: 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) (2.12.3)
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: 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: 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: 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,<3.0.0,>=1.7.4->spacy) (0.7.0)
Requirement already satisfied: pydantic-core==2.41.4 in
/usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!
=1.8.1,<3.0.0,>=1.7.4->spacy) (2.41.4)
Requirement already satisfied: 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->spacy) (4.15.0)
Requirement already satisfied: 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->spacy) (0.4.2)
Requirement already satisfied: six>=1.5 in
/usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.7-
>matplotlib) (1.17.0)
Requirement already satisfied: charset_normalizer<4,>=2 in
/usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0-
>spacy) (3.4.4)
Requirement already satisfied: idna<4,>=2.5 in
/usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0-
>spacy) (3.11)
```

```
Requirement already satisfied: 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)
Requirement already satisfied: certifi>=2017.4.17 in
/usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0-
>spacy) (2026.1.4)
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) (1.3.3)
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-
>spacy) (0.1.5)
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-
>spacy) (0.23.0)
Requirement already satisfied: smart-open<8.0.0,>=5.2.1 in
/usr/local/lib/python3.12/dist-packages (from weasel<0.5.0,>=0.4.2-
>spacy) (7.5.0)
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,>=0.4.2->spacy) (2.0.1)
Collecting en-core-web-sm==3.8.0
  Downloading
https://github.com/explosion/spacy-models/releases/download/en_core_we
b_sm-3.8.0/en_core_web_sm-3.8.0-py3-none-any.whl (12.8 MB)
━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 12.8/12.8 MB 99.3 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.
```

```
df = pd.read_csv("/content/arxiv_data.csv")
df
```


```
{"summary":"{\n  \"name\": \"df\",\n  \"rows\": 51774,\n  \"fields\":
[\n    {\n      \"column\": \"titles\",\n      \"properties\": {\n
\"dtype\": \"string\",\n        \"num_unique_values\": 38972,\n
\"samples\": [\n          \"Sum-Product-Transform Networks: Exploiting
Symmetries using Invertible Transformations\",\n          \"A Primal-
Dual Subgradient Approachfor Fair Meta Learning\",\n
\"Adversarial Multi-Source Transfer Learning in Healthcare:
Application to Glucose Prediction for Diabetic People\"\n        ],\n
\"semantic_type\": \"\",\n        \"description\": \"\"\n      }\
```

n   },\n   {\n   \"column\": \"summaries\",\n \"properties\": {\n   \"dtype\": \"string\",\n \"num_unique_values\": 38979,\n   \"samples\": [\n \"Continual learning (CL) is a setting in which an agent has to learn from an\\nincoming stream of data during its entire lifetime. Although major advances\\nhave been made in the field, one recurring problem which remains unsolved is\\nthat of Catastrophic Forgetting (CF). While the issue has been extensively\\nstudied empirically, little attention has been paid from a theoretical angle.\\nIn this paper, we show that the impact of CF increases as two tasks\\nincreasingly align. We introduce a measure of task similarity called the NTK\\noverlap matrix which is at the core of CF. We analyze common projected gradient\\nalgorithms and demonstrate how they mitigate forgetting. Then, we propose a\\nvariant of Orthogonal Gradient Descent (OGD) which leverages structure of the\\ndata through Principal Component Analysis (PCA). Experiments support our\\ntheoretical findings and show how our method can help reduce CF on classical CL\\ndatasets.\",\n   \"Few-shot learning is a challenging task since only few instances are given\\nfor recognizing an unseen class. One way to alleviate this problem is to\\nacquire a strong inductive bias via meta-learning on similar tasks. In this\\npaper, we show that such inductive bias can be learned from a flat collection\\nof unlabeled images, and instantiated as transferable representations among\\nseen and unseen classes. Specifically, we propose a novel part-based\\nself-supervised representation learning scheme to learn transferable\\nrepresentations by maximizing the similarity of an image to its discriminative\\npart. To mitigate the overfitting in few-shot classification caused by data\\nscarcity, we further propose a part augmentation strategy by retrieving extra\\nimages from a base dataset. We conduct systematic studies on miniImageNet and\\ntieredImageNet benchmarks. Remarkably, our method yields impressive results,\\noutperforming the previous best unsupervised methods by 7.74% and 9.24% under\\n5-way 1-shot and 5-way 5-shot settings, which are comparable with\\nstate-of-the-art supervised methods.\",\n   \"Surgical instrument segmentation is extremely important for computer-assisted\\nsurgery. Different from common object segmentation, it is more challenging due\\nto the large illumination and scale variation caused by the special surgical\\nscenes. In this paper, we propose a novel bilinear attention network with\\nadaptive receptive field to solve these two challenges. For the illumination\\nvariation, the bilinear attention module can capture second-order statistics to\\nencode global contexts and semantic dependencies between local pixels. With\\nthem, semantic features in challenging areas can be inferred from their\\nneighbors and the distinction of various semantics can be boosted. For the\\nscale variation, our adaptive receptive field module aggregates multi-scale\\nfeatures and automatically fuses them with different weights. Specifically, it\\nencodes the semantic relationship between channels to emphasize feature maps\\nwith appropriate scales, changing the

receptive field of subsequent\\nconvolutions. The proposed network achieves the best performance 97.47% mean\\nIOU on Cata7 and comes first place on EndoVis 2017 by 10.10% IOU overtaking\\nsecond-ranking method.\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   },\n   {\n   \"column\": \"terms\",\n   \"properties\": {\n   \"dtype\": \"category\",\n   \"num_unique_values\": 3157,\n   \"samples\": [\n   \"['cs.LG', 'cs.CE', 'q-fin.ST', 'stat.ML']\",\n   \"['cs.LG', 'physics.comp-ph', 'physics.flu-dyn']\",\n   \"['cs.LG', 'cs.CV', 'math.AT']\"\n   ],\n   \"semantic_type\": \"\",\n   \"description\": \"\"\n   }\n   }\n]\n}","type":"dataframe","variable_name":"df"}

```
abstracts = df['summaries'].dropna().head(100)

nlp = spacy.load("en_core_web_sm")

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

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)

top_noun_phrases
```

```
[('we', 265),
 ('which', 74),
 ('that', 73),
 ('it', 72),
 ('the-art', 42),
 ('this paper', 34),
 ('medical image segmentation', 25),
 ('our method', 25),
 ('this work', 24),
 ('image segmentation', 22)]
```

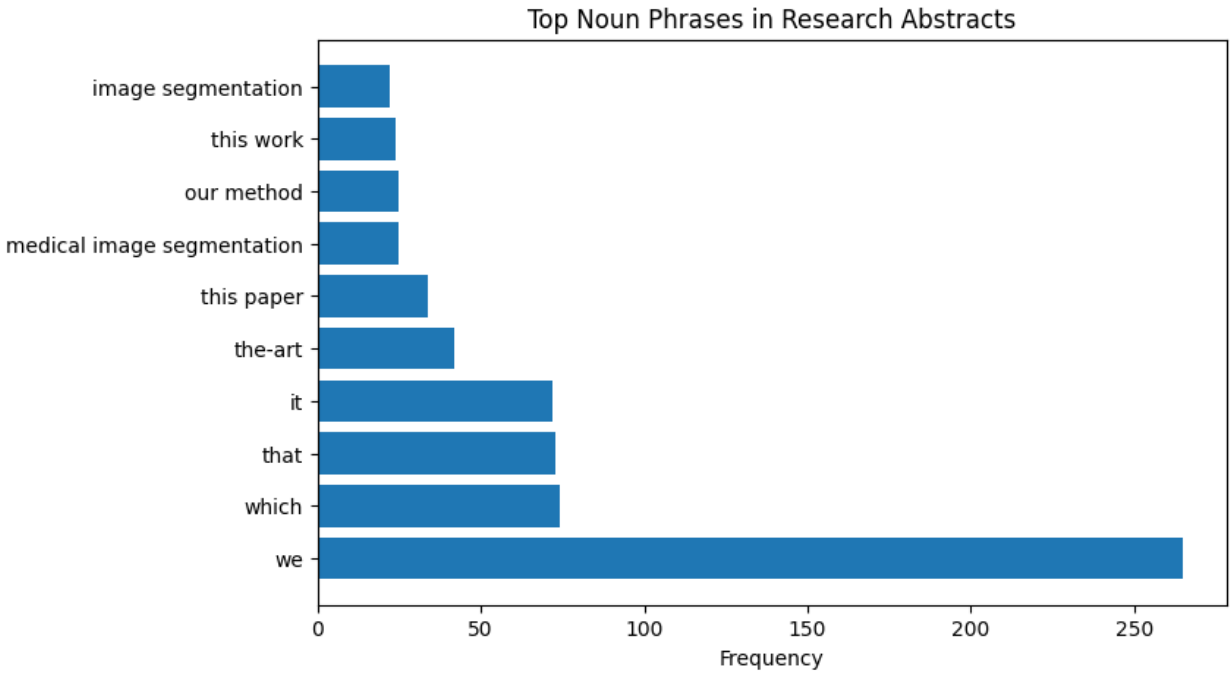
```
entities = []

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

entity_freq = Counter(entities)
entity_freq
```

```
Counter({'DATE': 13,
         'GPE': 21,
         'CARDINAL': 132,
         'NORP': 15,
         'ORG': 247,
         'ORDINAL': 37,
         'WORK_OF_ART': 2,
         'PERSON': 31,
         'PERCENT': 19,
         'PRODUCT': 6,
         'MONEY': 4,
         'TIME': 2,
         'LOC': 1,
         'LAW': 1,
         'EVENT': 1,
         'FAC': 3})
```

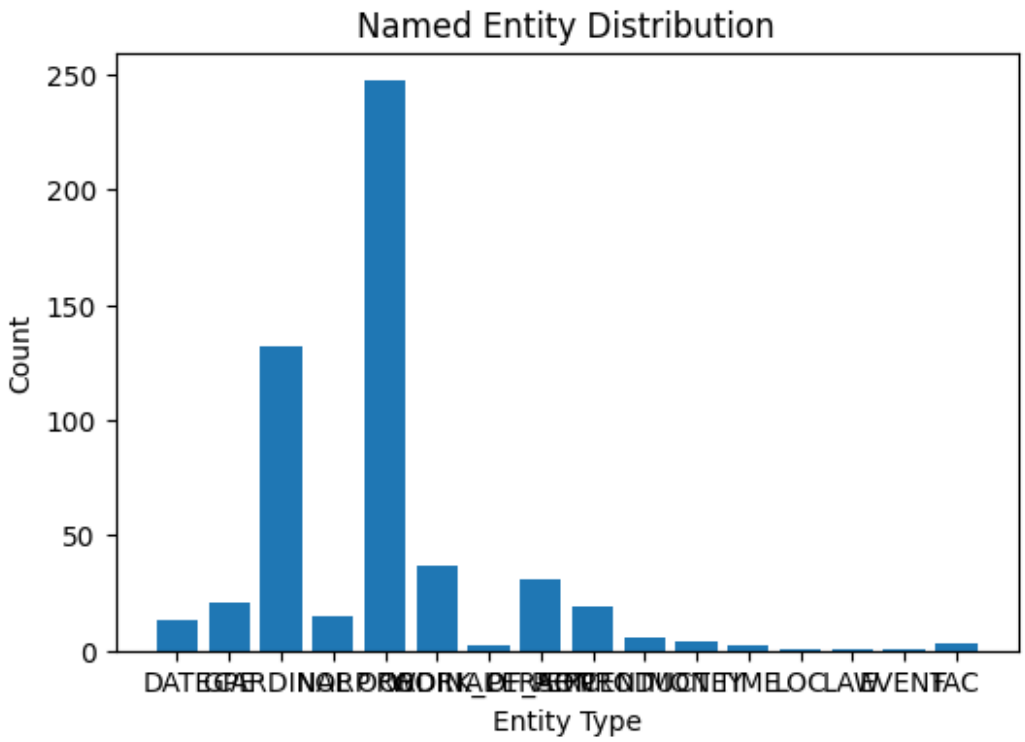
```
matcher = Matcher(nlp.vocab)

pattern1 = [{"POS": "ADJ"}, {"POS": "NOUN"}]
pattern2 = [{"POS": "NOUN"}, {"POS": "NOUN"}]

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

tech_terms = []

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

Counter(tech_terms).most_common(10)
```

```
[('image segmentation', 111),
 ('medical image', 60),
 ('semantic segmentation', 24),
 ('segmentation tasks', 17),
 ('deep learning', 15),
 ('training data', 15),
 ('neural networks', 12),
 ('contextual information', 12),
 ('unlabeled data', 11),
 ('semantic image', 11)]
```

```
labels, values = zip(*top_noun_phrases)

plt.figure(figsize=(8,5))
plt.barh(labels, values)
plt.title("Top Noun Phrases in Research Abstracts")
plt.xlabel("Frequency")
plt.show()
```




```
labels, values = zip(*entity_freq.items())

plt.figure(figsize=(6,4))
plt.bar(labels, values)
plt.title("Named Entity Distribution")
plt.xlabel("Entity Type")
plt.ylabel("Count")
plt.show()
```


Named Entity Distribution
Count
250
200
150
100
50
0
Entity Type