```
!python -m spacy download en_core_web_sm
```

```
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-any
     ━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━━ 12.8/12.8 MB 71.3 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.
```

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

```
pip install spacy transformers torch pandas
```

```
Requirement already satisfied: spacy in /usr/local/lib/python3.12/dist-packages (3.8.11)
Requirement already satisfied: transformers in /usr/local/lib/python3.12/dist-packages (5.0.0)
Requirement already satisfied: torch in /usr/local/lib/python3.12/dist-packages (2.10.0+cpu)
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) (0.24.0)
Requirement already satisfied: tqdm<5.0.0,>=4.38.0 in /usr/local/lib/python3.12/dist-packages (from spacy) (4.67.3)
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
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) (26.0)
Requirement already satisfied: filelock in /usr/local/lib/python3.12/dist-packages (from transformers) (3.24.2)
Requirement already satisfied: huggingface-hub<2.0,>=1.3.0 in /usr/local/lib/python3.12/dist-packages (from transformers) (1.4
Requirement already satisfied: pyyaml>=5.1 in /usr/local/lib/python3.12/dist-packages (from transformers) (6.0.3)
Requirement already satisfied: regex!=2019.12.17 in /usr/local/lib/python3.12/dist-packages (from transformers) (2025.11.3)
Requirement already satisfied: tokenizers<=0.23.0,>=0.22.0 in /usr/local/lib/python3.12/dist-packages (from transformers) (0.2
Requirement already satisfied: safetensors>=0.4.3 in /usr/local/lib/python3.12/dist-packages (from transformers) (0.7.0)
Requirement already satisfied: typing-extensions>=4.10.0 in /usr/local/lib/python3.12/dist-packages (from torch) (4.15.0)
Requirement already satisfied: sympy>=1.13.3 in /usr/local/lib/python3.12/dist-packages (from torch) (1.14.0)
Requirement already satisfied: networkx>=2.5.1 in /usr/local/lib/python3.12/dist-packages (from torch) (3.6.1)
Requirement already satisfied: fsspec>=0.8.5 in /usr/local/lib/python3.12/dist-packages (from torch) (2025.3.0)
Requirement already satisfied: python-dateutil>=2.8.2 in /usr/local/lib/python3.12/dist-packages (from pandas) (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: hf-xet<2.0.0,>=1.2.0 in /usr/local/lib/python3.12/dist-packages (from huggingface-hub<2.0,>=1.3
Requirement already satisfied: httpx<1,>=0.23.0 in /usr/local/lib/python3.12/dist-packages (from huggingface-hub<2.0,>=1.3.0->
Requirement already satisfied: shellingham in /usr/local/lib/python3.12/dist-packages (from huggingface-hub<2.0,>=1.3.0->trans
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,<3
Requirement already satisfied: typing-inspection>=0.4.2 in /usr/local/lib/python3.12/dist-packages (from pydantic!=1.8,!=1.8.1
Requirement already satisfied: six>=1.5 in /usr/local/lib/python3.12/dist-packages (from python-dateutil>=2.8.2->pandas) (1.17
Requirement already satisfied: charset_normalizer<4,>=2 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13
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->spa
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0->spa
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.12/dist-packages (from sympy>=1.13.3->torch) (1.3.
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)
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: typer>=0.24.0 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,>=0.4.
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-
Requirement already satisfied: MarkupSafe>=2.0 in /usr/local/lib/python3.12/dist-packages (from jinja2->spacy) (3.0.3)
Requirement already satisfied: anyio in /usr/local/lib/python3.12/dist-packages (from httpx<1,>=0.23.0->huggingface-hub<2.0,>=
Requirement already satisfied: httpcore==1.* in /usr/local/lib/python3.12/dist-packages (from httpx<1,>=0.23.0->huggingface-hu
Requirement already satisfied: h11>=0.16 in /usr/local/lib/python3.12/dist-packages (from httpcore==1.*->httpx<1,>=0.23.0->hug
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,>=
```

```
import spacy
from spacy import displacy
from transformers import pipeline, AutoTokenizer, AutoModelForTokenClassification
```

```
import torch
import pandas
```

```
print("Visualizing Entities with displacy:")
print("-" * 40)

for i, sentence in enumerate(sentences):
    doc = nlp(sentence)
    print(f"\nSentence {i+1}:")
    displacy.render(doc, style="ent", jupyter=True)
    print("-" * 40)
```

```
Visualizing Entities with displacy:
----------------------------------------

Sentence 1:
 Apple Inc. ORG  was founded by  Steve Jobs PERSON , Steve Wozniak PERSON , and  Ronald Wayne PERSON .
----------------------------------------

Sentence 2:
 Google ORG 's headquarters are in  Mountain View GPE , California GPE .
----------------------------------------

Sentence 3:
 The Eiffel Tower LOC  is in  Paris GPE , France GPE , and was designed by  Gustave Eiffel PERSON .
----------------------------------------

Sentence 4:
 Barack Obama PERSON  was the  44th ORDINAL  President of  the United States GPE .
----------------------------------------
```

```
entity_data = []

for sentence_text in sentences:
    doc = nlp(sentence_text)
    if doc.ents:
        for ent in doc.ents:
            entity_data.append({"Sentence": sentence_text, "Entity": ent.text, "Label": ent.label_})
    else:
        # If no entities are found, you might want to record the sentence anyway
        # or just skip it. For this task, we'll only record sentences with entities.
        pass

entity_df = pandas.DataFrame(entity_data)
display(entity_df)
```

	Sentence	Entity	Label
0	Apple Inc. was founded by Steve Jobs, Steve Wo...	Apple Inc.	ORG
1	Apple Inc. was founded by Steve Jobs, Steve Wo...	Steve Jobs	PERSON
2	Apple Inc. was founded by Steve Jobs, Steve Wo...	Steve Wozniak	PERSON
3	Apple Inc. was founded by Steve Jobs, Steve Wo...	Ronald Wayne	PERSON
4	Google's headquarters are in Mountain View, Ca...	Google	ORG
5	Google's headquarters are in Mountain View, Ca...	Mountain View	GPE
6	Google's headquarters are in Mountain View, Ca...	California	GPE
7	The Eiffel Tower is in Paris, France, and was ...	The Eiffel Tower	LOC
8	The Eiffel Tower is in Paris, France, and was ...	Paris	GPE
9	The Eiffel Tower is in Paris, France, and was ...	France	GPE
10	The Eiffel Tower is in Paris, France, and was ...	Gustave Eiffel	PERSON
11	Barack Obama was the 44th President of the Uni...	Barack Obama	PERSON
12	Barack Obama was the 44th President of the Uni...	44th	ORDINAL
13	Barack Obama was the 44th President of the Uni...	the United States	GPE

Feature	spaCy	Hugging Face Transformers
Model Type	Rule-based, statistical, CNNs	Transformer-based (BERT, RoBERTa, etc.)
Speed	Generally faster and more lightweight	Can be slower due to larger model sizes
Accuracy (qualitative)	Good for general-purpose NER (e.g., `en_core_web_sm` models)	Often state-of-the-art accuracy due to complex architectures
Context Handling	Local context with some global awareness	Excellent, leveraging deep contextual embeddings
Confidence Score	No direct confidence scores for entities (can be derived using custom methods)	Provides direct confidence scores for each token/entity

```
def clean_transformer_output(ner_results):
    cleaned_entities = []
    current_entity = {"word": "", "entity": "", "score": 0.0, "count": 0}

    for item in ner_results:
        word = item['word']
        entity_label = item['entity']
        score = item['score']

        # Check if it's a beginning token (B-) or continuation of a previous entity (I-)
        if entity_label.startswith('B-') or not current_entity["word"] or not entity_label.startswith('I-'):
            if current_entity["word"]:
                # Append the previous entity if it exists
                cleaned_entities.append({
                    "Entity": current_entity["word"].replace('##', ''),
                    "Label": current_entity["entity"].replace('B-', '').replace('I-', ''),
                    "Confidence_Score": current_entity["score"] / current_entity["count"]
                })
            # Start a new entity
            current_entity = {
                "word": word,
                "entity": entity_label,
                "score": score,
                "count": 1
            }
        else: # Continuation of an entity (I-)
            current_entity["word"] += word.replace('##', '')
            current_entity["score"] += score
            current_entity["count"] += 1

    # Append the last entity after the loop finishes
    if current_entity["word"]:
        cleaned_entities.append({
            "Entity": current_entity["word"].replace('##', ''),
            "Label": current_entity["entity"].replace('B-', '').replace('I-', ''),
            "Confidence_Score": current_entity["score"] / current_entity["count"]
        })

    return cleaned_entities

all_cleaned_entity_data = []

for sentence_text in sentences:
    ner_results = classifier(sentence_text)
    cleaned = clean_transformer_output(ner_results)
    for entity_info in cleaned:
        all_cleaned_entity_data.append({
            "Sentence": sentence_text,
            "Entity": entity_info['Entity'],
            "Label": entity_info['Label'],
            "Confidence_Score": entity_info['Confidence_Score']
        })

cleaned_transformer_df = pandas.DataFrame(all_cleaned_entity_data)
display(cleaned_transformer_df)
```

	Sentence	Entity	Label	Confidence_Score
0	Apple Inc. was founded by Steve Jobs, Steve Wo...	AppleInc	ORG	0.999405
1	Apple Inc. was founded by Steve Jobs, Steve Wo...	SteveJobs	PER	0.983681
2	Apple Inc. was founded by Steve Jobs, Steve Wo...	SteveWozniak	PER	0.971392
3	Apple Inc. was founded by Steve Jobs, Steve Wo...	RonaldWayne	PER	0.999695
4	Google's headquarters are in Mountain View, Ca...	Google	ORG	0.998801
5	Google's headquarters are in Mountain View, Ca...	MountainView	LOC	0.997765
6	Google's headquarters are in Mountain View, Ca...	California	LOC	0.999266
7	The Eiffel Tower is in Paris, France, and was ...	Eiff	LOC	0.821108
8	The Eiffel Tower is in Paris, France, and was ...	elTower	LOC	0.895773
9	The Eiffel Tower is in Paris, France, and was ...	Paris	LOC	0.999587
10	The Eiffel Tower is in Paris, France, and was ...	France	LOC	0.999647
11	The Eiffel Tower is in Paris, France, and was ...	Gustav	PER	0.998398
12	The Eiffel Tower is in Paris, France, and was ...	eEiffel	PER	0.948596
13	Barack Obama was the 44th President of the Uni...	BarackObama	PER	0.999533
14	Barack Obama was the 44th President of the Uni...	UnitedStates	LOC	0.999509

```
print("Applying Transformer NER to sentences:")
print("-" * 40)

for i, sentence in enumerate(sentences):
    print(f"\nSentence {i+1}: {sentence}")
    ner_results = classifier(sentence)

    if ner_results:
        print("Entities detected (Word | Label | Confidence):")
        for entity in ner_results:
            # Some entities might be broken into sub-words by the tokenizer.
            # We'll display them as they are returned.
            print(f"  - '{entity['word']}' | '{entity['entity']}' | {entity['score']:.4f}")
    else:
        print("  No entities found.")
    print("-" * 40)
```

```
Applying Transformer NER to sentences:
----------------------------------------

Sentence 1: Apple Inc. was founded by Steve Jobs, Steve Wozniak, and Ronald Wayne.
Entities detected (Word | Label | Confidence):
  - 'Apple' | 'B-ORG' | 0.9995
  - 'Inc' | 'I-ORG' | 0.9993
  - 'Steve' | 'B-PER' | 0.9997
  - 'Job' | 'I-PER' | 0.9995
  - '##s' | 'I-PER' | 0.9518
  - 'Steve' | 'B-PER' | 0.9998
  - 'W' | 'I-PER' | 0.9997
  - '##oz' | 'I-PER' | 0.9992
  - '##nia' | 'I-PER' | 0.9993
  - '##k' | 'I-PER' | 0.8590
  - 'Ronald' | 'B-PER' | 0.9997
  - 'Wayne' | 'I-PER' | 0.9997
----------------------------------------

Sentence 2: Google's headquarters are in Mountain View, California.
Entities detected (Word | Label | Confidence):
  - 'Google' | 'B-ORG' | 0.9988
  - 'Mountain' | 'B-LOC' | 0.9967
  - 'View' | 'I-LOC' | 0.9988
  - 'California' | 'B-LOC' | 0.9993
----------------------------------------

Sentence 3: The Eiffel Tower is in Paris, France, and was designed by Gustave Eiffel.
Entities detected (Word | Label | Confidence):
  - 'E' | 'B-LOC' | 0.9925
  - '##iff' | 'I-LOC' | 0.6498
  - '##el' | 'B-LOC' | 0.7989
  - 'Tower' | 'I-LOC' | 0.9927
  - 'Paris' | 'B-LOC' | 0.9996
  - 'France' | 'B-LOC' | 0.9996
  - 'Gustav' | 'B-PER' | 0.9984
  - '##e' | 'B-PER' | 0.9302
  - 'E' | 'I-PER' | 0.9986
  - '##iff' | 'I-PER' | 0.8858
  - '##el' | 'I-PER' | 0.9797
----------------------------------------

Sentence 4: Barack Obama was the 44th President of the United States.
Entities detected (Word | Label | Confidence):
  - 'Barack' | 'B-PER' | 0.9995
  - 'Obama' | 'I-PER' | 0.9995
  - 'United' | 'B-LOC' | 0.9997
  - 'States' | 'I-LOC' | 0.9993
----------------------------------------
```

```
model_name = "dslim/bert-base-NER"

# 1. Load tokenizer
tokenizer = AutoTokenizer.from_pretrained(model_name)

# 2. Load model
model = AutoModelForTokenClassification.from_pretrained(model_name)

# 3. Create NER pipeline
classifier = pipeline("ner", model=model, tokenizer=tokenizer)
```

```
/usr/local/lib/python3.12/dist-packages/huggingface_hub/utils/_auth.py:94: UserWarning:
The secret `HF_TOKEN` does not exist in your Colab secrets.
To authenticate with the Hugging Face Hub, create a token in your settings tab (https://huggingface.co/settings/tokens), set it
You will be able to reuse this secret in all of your notebooks.
Please note that authentication is recommended but still optional to access public models or datasets.
  warnings.warn(
config.json:  100%                                        829/829  [00:00<00:00,  85.5kB/s]

tokenizer_config.json:  100%                                        59.0/59.0  [00:00<00:00,  8.01kB/s]

Warning: You are sending unauthenticated requests to the HF Hub. Please set a HF_TOKEN to enable higher rate limits and faster d
WARNING:huggingface_hub.utils._http:Warning: You are sending unauthenticated requests to the HF Hub. Please set a HF_TOKEN to en
vocab.txt:       213k/?  [00:00<00:00,  15.0MB/s]

added_tokens.json:  100%                                        2.00/2.00  [00:00<00:00,  216B/s]

special_tokens_map.json:  100%                                        112/112  [00:00<00:00,  13.4kB/s]

model.safetensors:  100%                                        433M/433M  [00:03<00:00,  206MB/s]

Loading weights:  100%                                        199/199  [00:00<00:00,  706.90it/s,  Materializing  param=classifier.weight]
BertForTokenClassification LOAD REPORT from: dslim/bert-base-NER
Key                         | Status     |  |
-------------------------+------------+--+-
bert.pooler.dense.bias    | UNEXPECTED |  |
bert.pooler.dense.weight | UNEXPECTED |  |

Notes:
- UNEXPECTED    :can be ignored when loading from different task/architecture; not ok if you expect identical arch.
```

```
from collections import Counter

all_entity_labels = []
for sentence in sentences:
    doc = nlp(sentence)
    for ent in doc.ents:
        all_entity_labels.append(ent.label_)

label_counts = Counter(all_entity_labels)

print("Entity Label Occurrences:")
print("-" * 30)
for label, count in label_counts.items():
    print(f"  - {label}: {count}")
print("-" * 30)
```

```
Entity Label Occurrences:
------------------------------
  - ORG: 2
  - PERSON: 5
  - GPE: 5
  - LOC: 1
  - ORDINAL: 1
------------------------------
```

```
sentences = [
    "Apple Inc. was founded by Steve Jobs, Steve Wozniak, and Ronald Wayne.",
    "Google's headquarters are in Mountain View, California.",
    "The Eiffel Tower is in Paris, France, and was designed by Gustave Eiffel.",
    "Barack Obama was the 44th President of the United States."
]

print("Applying spaCy NER to sentences:")
print("-" * 40)

for i, sentence in enumerate(sentences):
    doc = nlp(sentence)
    print(f"\nSentence {i+1}: {sentence}")
    print("Entities detected:")
    if doc.ents:
        for ent in doc.ents:
            print(f"  - Text: '{ent.text}', Label: '{ent.label_}'")
    else:
        print("  No entities found.")
    print("-" * 40)
```

```
Applying spaCy NER to sentences:
----------------------------------------

Sentence 1: Apple Inc. was founded by Steve Jobs, Steve Wozniak, and Ronald Wayne.
Entities detected:
  - Text: 'Apple Inc.', Label: 'ORG'
  - Text: 'Steve Jobs', Label: 'PERSON'
  - Text: 'Steve Wozniak', Label: 'PERSON'
  - Text: 'Ronald Wayne', Label: 'PERSON'
----------------------------------------

Sentence 2: Google's headquarters are in Mountain View, California.
Entities detected:
  - Text: 'Google', Label: 'ORG'
  - Text: 'Mountain View', Label: 'GPE'
  - Text: 'California', Label: 'GPE'
----------------------------------------

Sentence 3: The Eiffel Tower is in Paris, France, and was designed by Gustave Eiffel.
Entities detected:
  - Text: 'The Eiffel Tower', Label: 'LOC'
  - Text: 'Paris', Label: 'GPE'
  - Text: 'France', Label: 'GPE'
  - Text: 'Gustave Eiffel', Label: 'PERSON'
----------------------------------------

Sentence 4: Barack Obama was the 44th President of the United States.
Entities detected:
  - Text: 'Barack Obama', Label: 'PERSON'
  - Text: '44th', Label: 'ORDINAL'
  - Text: 'the United States', Label: 'GPE'
----------------------------------------
```