

```
import warnings
warnings.filterwarnings('ignore') # Suppress warnings related to unauthenticated Hugging Face Hub requests.
!pip install spacy transformers torch pandas # Install necessary libraries: spaCy for NLP, transformers for Hugging Face models,
!python -m spacy download en_core_web_sm # Download the small English language model for spaCy, which includes pre-trained NER c

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->trans
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,<3
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->sp
Requirement already satisfied: certifi>=2017.4.17 in /usr/local/lib/python3.12/dist-packages (from requests<3.0.0,>=2.13.0->sp
Requirement already satisfied: mpmath<1.4,>=1.1.0 in /usr/local/lib/python3.12/dist-packages (from sympy>=1.13.3->torch) (1.3.6
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,>=1
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,>
Requirement already satisfied: click>=8.2.1 in /usr/local/lib/python3.12/dist-packages (from typer>=0.24.0->typer-slim<1.0.0,>
Requirement already satisfied: rich>=12.3.0 in /usr/local/lib/python3.12/dist-packages (from typer>=0.24.0->typer-slim<1.0.0,>
Requirement already satisfied: annotated-doc>=0.0.2 in /usr/local/lib/python3.12/dist-packages (from typer>=0.24.0->typer-slim
Requirement already satisfied: markdown-it-py>=2.2.0 in /usr/local/lib/python3.12/dist-packages (from rich>=12.3.0->typer>=0.2
Requirement already satisfied: pygments<3.0.0,>=2.13.0 in /usr/local/lib/python3.12/dist-packages (from rich>=12.3.0->typer>=0
Requirement already satisfied: mdurl~0.1 in /usr/local/lib/python3.12/dist-packages (from markdown-it-py>=2.2.0->rich>=12.3.0
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-any.whl
12.8/12.8 MB 94.3 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.

```
import spacy # Import the spaCy library for Natural Language Processing (NLP).
from spacy import displacy # Import displacy from spaCy for visualizing named entities.
from transformers import pipeline, AutoTokenizer, AutoModelForTokenClassification # Import components from the transformers lib
import torch # Import PyTorch, a deep learning framework.
import pandas as pd # Import pandas for data manipulation and creating DataFrames.
```

```
nlp = spacy.load("en_core_web_sm") # Load the English language model 'en_core_web_sm' into spaCy for processing text.
```

```
sentences = [ # Define a list of sentences to be used for Named Entity Recognition (NER).
```

```
"Apple launched the new iPhone 15 in California.",
"Elon Musk announced new Tesla updates in Texas.",
"The Indian cricket team won the World Cup in Australia.",
"Prime Minister Narendra Modi visited the United States.",
```

```
"Google is investing $5 billion in artificial intelligence research."
]
```

```
print("---- spaCy NER Results ----\n") # Print a header for the spaCy NER results.

spacy_results = [] # Initialize an empty list to store spaCy's NER results.

for sentence in sentences: # Iterate through each sentence in the predefined list.
    doc = nlp(sentence) # Process the sentence with the loaded spaCy model to create a Doc object.
    print(f"Sentence: {sentence}") # Print the current sentence.

    for ent in doc.ents: # Iterate through each named entity detected in the Doc object.
        print(f"Entity: {ent.text}, Label: {ent.label_}") # Print the entity text and its assigned label.
        spacy_results.append([sentence, ent.text, ent.label_]) # Append the sentence, entity, and label to the results list.

    print() # Print an empty line for better readability between sentences.
```

```
---- spaCy NER Results ----

Sentence: Apple launched the new iPhone 15 in California.
Entity: Apple, Label: ORG
Entity: 15, Label: CARDINAL
Entity: California, Label: GPE

Sentence: Elon Musk announced new Tesla updates in Texas.
Entity: Elon Musk, Label: PERSON
Entity: Tesla, Label: ORG
Entity: Texas, Label: GPE

Sentence: The Indian cricket team won the World Cup in Australia.
Entity: Indian, Label: NORP
Entity: the World Cup, Label: EVENT
Entity: Australia, Label: GPE

Sentence: Prime Minister Narendra Modi visited the United States.
Entity: Narendra Modi, Label: PERSON
Entity: the United States, Label: GPE

Sentence: Google is investing $5 billion in artificial intelligence research.
Entity: Google, Label: ORG
Entity: $5 billion, Label: MONEY
```

```
df_spacy = pd.DataFrame(spacy_results, columns=["Sentence", "Entity", "Label"]) # Create a pandas DataFrame from the spaCy NER
df_spacy # Display the DataFrame containing spaCy's NER output.
```

	Sentence	Entity	Label
0	Apple launched the new iPhone 15 in California.	Apple	ORG
1	Apple launched the new iPhone 15 in California.	15	CARDINAL
2	Apple launched the new iPhone 15 in California.	California	GPE
3	Elon Musk announced new Tesla updates in Texas.	Elon Musk	PERSON
4	Elon Musk announced new Tesla updates in Texas.	Tesla	ORG
5	Elon Musk announced new Tesla updates in Texas.	Texas	GPE
6	The Indian cricket team won the World Cup in A...	Indian	NORP
7	The Indian cricket team won the World Cup in A...	the World Cup	EVENT
8	The Indian cricket team won the World Cup in A...	Australia	GPE
9	Prime Minister Narendra Modi visited the Unite...	Narendra Modi	PERSON
10	Prime Minister Narendra Modi visited the Unite...	the United States	GPE
11	Google is investing \$5 billion in artificial i...	Google	ORG
12	Google is investing \$5 billion in artificial i...	\$5 billion	MONEY

```
displacy.render(nlp(sentences[0]), style="ent", jupyter=True) # Render the first sentence with its named entities highlighted u
```

```
Apple ORG launched the new iPhone 15 CARDINAL in California GPE .
```

```
model_name = "dslim/bert-base-NER" # Specify the name of the pre-trained BERT-based NER model from Hugging Face.
```

```
tokenizer = AutoTokenizer.from_pretrained(model_name) # Load the tokenizer associated with the specified pre-trained model.
model = AutoModelForTokenClassification.from_pretrained(model_name) # Load the pre-trained model for token classification (NER)

ner_pipeline = pipeline("ner", model=model, tokenizer=tokenizer) # Create an NER pipeline using the loaded model and tokenizer.
```

config.json: 100%829/829 [00:00<00:00, 11.5kB/s]

tokenizer\_config.json: 100%59.0/59.0 [00:00<00:00, 885B/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, 2.62MB/s]

added\_tokens.json: 100%2.00/2.00 [00:00<00:00, 34.4B/s]

special\_tokens\_map.json: 100%112/112 [00:00<00:00, 2.10kB/s]

model.safetensors: 100%433M/433M [00:04<00:00, 179MB/s]

Loading weights: 100%199/199 [00:00<00:00, 288.88it/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.

```
print("---- Hugging Face NER Results ----\n") # Print a header for the Hugging Face NER results.

hf_results = [] # Initialize an empty list to store Hugging Face's NER results.

for sentence in sentences: # Iterate through each sentence.
    print(f"Sentence: {sentence}") # Print the current sentence.
    entities = ner_pipeline(sentence) # Process the sentence with the Hugging Face NER pipeline to extract entities.

    for ent in entities: # Iterate through each entity detected by the pipeline.
        # Print the word, its predicted label, and the confidence score, rounded to 4 decimal places.
        print(f"Word: {ent['word']}, Label: {ent['entity']}, Score: {round(ent['score'],4)}")
        hf_results.append([sentence, ent['word'], ent['entity'], ent['score']]) # Append the sentence, word, label, and score t

    print() # Print an empty line for better readability.
```

---- Hugging Face NER Results ----

Sentence: Apple launched the new iPhone 15 in California.  
Word: Apple, Label: B-ORG, Score: 0.9975000023841858  
Word: iPhone, Label: B-MISC, Score: 0.9983999729156494  
Word: 15, Label: I-MISC, Score: 0.998199999332428  
Word: California, Label: B-LOC, Score: 0.9997000098228455

Sentence: Elon Musk announced new Tesla updates in Texas.  
Word: El, Label: B-ORG, Score: 0.9986000061035156  
Word: ##on, Label: I-ORG, Score: 0.9944999814033508  
Word: Mu, Label: I-ORG, Score: 0.9968000054359436  
Word: ##sk, Label: I-ORG, Score: 0.9861999750137329  
Word: Te, Label: B-MISC, Score: 0.595300018787384  
Word: ##sla, Label: I-ORG, Score: 0.6690000295639038  
Word: Texas, Label: B-LOC, Score: 0.9995999932289124

Sentence: The Indian cricket team won the World Cup in Australia.  
Word: Indian, Label: B-MISC, Score: 0.9995999932289124  
Word: World, Label: B-MISC, Score: 0.9936000108718872  
Word: Cup, Label: I-MISC, Score: 0.9987999796867371  
Word: Australia, Label: B-LOC, Score: 0.9998000264167786

Sentence: Prime Minister Narendra Modi visited the United States.  
Word: Na, Label: B-PER, Score: 0.9997000098228455  
Word: ##ren, Label: I-PER, Score: 0.6021000146865845  
Word: ##dra, Label: I-PER, Score: 0.9045000076293945  
Word: Mo, Label: I-PER, Score: 0.9995999932289124  
Word: ##di, Label: I-PER, Score: 0.9732999801635742  
Word: United, Label: B-LOC, Score: 0.9997000098228455  
Word: States, Label: I-LOC, Score: 0.9994000196456909

Sentence: Google is investing \$5 billion in artificial intelligence research.  
Word: Google, Label: B-ORG, Score: 0.9984999895095825

```
clean_results = [] # Initialize an empty list to store cleaned Hugging Face NER results.

for sentence in sentences: # Iterate through each sentence.
    entities = ner_pipeline(sentence) # Process the sentence with the Hugging Face NER pipeline.

    merged_word = "" # Initialize an empty string for merging subword tokens (not used in this specific implementation but kept
    current_label = "" # Initialize an empty string for the current label (not used in this specific implementation but kept fr
    current_score = 0 # Initialize a score (not used in this specific implementation but kept from original).

    for ent in entities: # Iterate through each entity detected.
        word = ent["word"].replace("##", "") # Get the word and remove '##' prefix from subword tokens for cleaner display.
        label = ent["entity"] # Get the entity label.
        score = ent["score"] # Get the confidence score for the entity.

        clean_results.append([sentence, word, label, score]) # Append the sentence, cleaned word, label, and score to the resul

df_hf = pd.DataFrame(clean_results, # Create a pandas DataFrame from the cleaned Hugging Face NER results.
                      columns=["Sentence", "Entity", "Label", "Confidence Score"]) # Define column names for the DataFrame.

df_hf # Display the DataFrame containing the cleaned Hugging Face NER output.
```

	Sentence	Entity	Label	Confidence Score
0	Apple launched the new iPhone 15 in California.	Apple	B-ORG	0.997462
1	Apple launched the new iPhone 15 in California.	iPhone	B-MISC	0.998376
2	Apple launched the new iPhone 15 in California.	15	I-MISC	0.998245
3	Apple launched the new iPhone 15 in California.	California	B-LOC	0.999744
4	Elon Musk announced new Tesla updates in Texas.	El	B-ORG	0.998633
5	Elon Musk announced new Tesla updates in Texas.	on	I-ORG	0.994543
6	Elon Musk announced new Tesla updates in Texas.	Mu	I-ORG	0.996818
7	Elon Musk announced new Tesla updates in Texas.	sk	I-ORG	0.986198
8	Elon Musk announced new Tesla updates in Texas.	Te	B-MISC	0.595255
9	Elon Musk announced new Tesla updates in Texas.	sla	I-ORG	0.668992
10	Elon Musk announced new Tesla updates in Texas.	Texas	B-LOC	0.999635
11	The Indian cricket team won the World Cup in A...	Indian	B-MISC	0.999630
12	The Indian cricket team won the World Cup in A...	World	B-MISC	0.993581
13	The Indian cricket team won the World Cup in A...	Cup	I-MISC	0.998754
14	The Indian cricket team won the World Cup in A...	Australia	B-LOC	0.999780
15	Prime Minister Narendra Modi visited the Unite...	Na	B-PER	0.999656
16	Prime Minister Narendra Modi visited the Unite...	ren	I-PER	0.602087
17	Prime Minister Narendra Modi visited the Unite...	dra	I-PER	0.904483
18	Prime Minister Narendra Modi visited the Unite...	Mo	I-PER	0.999624
19	Prime Minister Narendra Modi visited the Unite...	di	I-PER	0.973345
20	Prime Minister Narendra Modi visited the Unite...	United	B-LOC	0.999736
21	Prime Minister Narendra Modi visited the Unite...	States	I-LOC	0.999445
22	Google is investing \$5 billion in artificial i...	Google	B-ORG	0.998542

