****

**Superior University Lahore**

***Lab Task # 9***

**Name:** Muhammad Abdullah Kashif

**Roll No**: SU92-BSAIM-F23-070

**Instructor**: Sir Rasikh Ali

# Course: Programming for Artificial Intelligence (Lab)

**NLP Task: Information Extraction (Named Entity Recognition using Spacy)**

**Step-by-Step Code Explanation**

**1. Import Required Libraries**

import pandas as pd

import spacy

from spacy import displacy

* pandas: Used to create a structured table (DataFrame) of extracted entities.
* spacy: The core NLP library used for processing and analyzing text.
* displacy: spaCy’s visualizer module to render named entities visually in HTML.

**2. Load spaCy Language Model**

nlp = spacy.load("en\_core\_web\_sm")

* Loads the **English small model** that contains vocabulary, syntax, and trained pipelines for tasks like part-of-speech tagging and named entity recognition.
* Make sure you have it installed with:

bash

CopyEdit

python -m spacy download en\_core\_web\_sm

**3. Input Text Content**

content = """Trinamool Congress leader Mahua Moitra has moved the Supreme Court ..."""

* This is a sample paragraph mentioning political entities, persons, and organizations.
* The model will extract named entities from this text.

**4. Process the Text with spaCy**

doc = nlp(content)

* The nlp() function processes the text and returns a Doc object that contains linguistic annotations and entities.

**5. Print Named Entities**

print("\nNamed Entities:")

for ent in doc.ents:

print(f"{ent.text:<30} Start: {ent.start\_char:<3} End: {ent.end\_char:<3} Label: {ent.label\_}")

* Iterates through each named entity in the document.
* Prints:
  + The entity text (e.g., "Mahua Moitra")
  + Its start and end character positions in the original text
  + Its label (e.g., PERSON, ORG, GPE)

**6. Create DataFrame of Entities**

entities = [(ent.text, ent.label\_, ent.lemma\_) for ent in doc.ents]

df = pd.DataFrame(entities, columns=['Text', 'Type', 'Lemma'])

print("\nEntity DataFrame:")

print(df)

* Extracts text, type (label), and lemma (base form).
* Stores them in a DataFrame for a clean tabular view.

**7. Render NER Visualization to HTML**

html = displacy.render(doc, style="ent", page=True)

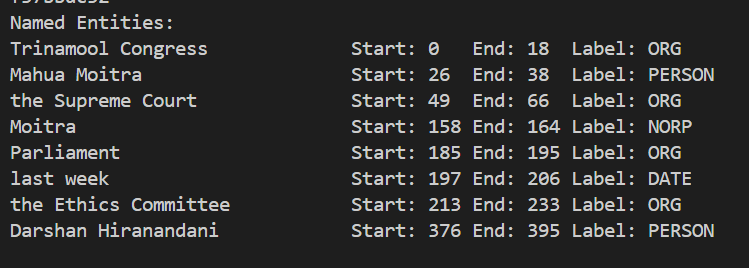
with open("ner\_visualization.html", "w", encoding="utf-8") as f:

f.write(html)

print("\n✔ Entity visualization saved to ner\_visualization.html")

* Uses displacy to create a visual representation of entities.
* style="ent" highlights entities in the text.
* page=True returns complete HTML code.
* Saves the visualization to a local HTML file: ner\_visualization.html.

**8. Output**

****

****