|  |
| --- |
| Experiment No.7 |
| Implement Named Entity Recognizer for the given Text input |
| Date of Performance: |
| Date of Submission: |

CSDL7013: Natural Language Processing Lab



Aim: Implement Named Entity Recognizer for the given Text input Objective: Understand the importance of NER in NLP and Implement NER.

Theory:

The named entity recognition (NER) is one of the most data preprocessing task. It involves the identification of key information in the text and classification into a set of predefined categories. An entity is basically the thing that is consistently talked about or refer to in the text.

NER is the form of NLP.

At its core, NLP is just a two-step process, below are the two steps that are involved:

* Detecting the entities from the text
* Classifying them into different categories

Some of the categories that are the most important architecture in NER such that:

* Person
* Organization
* Place/ location

Other common tasks include classifying of the following:

* date/time.
* expression
* Numeral measurement (money, percent, weight, etc)
* E-mail address

Ambiguity in NE

For a person, the category definition is intuitively quite clear, but for computers, there is some ambiguity in classification. Let’s look at some ambiguous example:

England (Organisation) won the 2019 world cup vs The 2019 world cup happened in England(Location).

Washington(Location) is the capital of the US vs The first president of the US was Washington(Person).

CSDL7013: Natural Language Processing Lab

import spacy

Load the spaCy language model

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

text = "Apple Inc. is a company based in Cupertino, California. John works for Google in Mountain View." Process the text using spaCy

import spacy

# Load the spaCy language model nlp = spacy.load("en\_core\_web\_sm")

# Sample text input text = "Apple Inc. is a company based in Cupertino, California. John works for Google in Mountain View."

# Process the text using spaCy doc = nlp(text)

# Initialize variables to store named entities named\_entities = []

# Define a function to extract named entities def extract\_named\_entities(doc):

entities = [] current\_entity = None

for token in doc:

if token.ent\_type\_:

if current\_entity and token.ent\_type\_ == current\_entity[1]:

current\_entity = (current\_entity[0] + " " + token.text, token.ent\_type\_)

else:

if current\_entity:

entities.append(current\_entity)

current\_entity = (token.text, token.ent\_type\_) else:

if current\_entity:

entities.append(current\_entity)

current\_entity = None

if current\_entity: entities.append(current\_entity) return entities

# Extract named entities

named\_entities = extract\_named\_entities(doc)

# Print the named entities for entity, label in named\_entities:

print(f"Entity: {entity}, Label: {label}")

** Entity: Apple Inc., Label: ORG

Entity: Cupertino, Label: GPE

Entity: California, Label: GPE

Entity: John, Label: PERSON

Entity: Google, Label: ORG

Entity: Mountain View, Label: GPE

**Conclusion:**

A Named Entity Recognizer (NER) is a critical NLP tool for identifying entities in text, such as names of people, places, organizations, dates, and more. Its application in the provided text would require the text content itself to assess and identify specific named entities. However, without the actual text input, it's impossible to perform NER or draw conclusions about the recognized entities.