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

Exp. No.: 7

Title: Implement Named Entity Recognizer for the given text input.

Theory:

Named Entity Recognition (NER) is a subtask of Information Extraction that involves identifying and classifying key entities in a text into predefined categories such as names, organizations, locations, dates, and more. NER plays a crucial role in various Natural Language Processing (NLP) applications, including information retrieval, question answering, and text summarization.

- NER is the process of locating and classifying named entities in text into predefined categories.
- It focuses on extracting relevant information from unstructured text, converting it into a structured format.

2. Common Entity Categories

NER typically recognizes entities in the following categories:

- **Person**: Names of individuals (e.g., "Albert Einstein")
- **Organization**: Names of companies, agencies, or institutions (e.g., "OpenAI", "United Nations")
- **Location**: Geographical locations, such as cities, countries, and landmarks (e.g., "Paris", "Mount Everest")
- **Date/Time**: Dates and times (e.g., "January 1, 2024", "10:00 AM")
- **Miscellaneous**: Other entities like monetary values, percentages, etc. (e.g., "$500", "75%")

3. Importance of NER

- **Information Retrieval**: Helps in extracting relevant information from large volumes of text.

- **Knowledge Graph Construction**: Facilitates the creation of structured knowledge bases.
- **Sentiment Analysis**: Enhances sentiment analysis by identifying key entities associated with positive or negative sentiments.
- **Chatbots and Virtual Assistants**: Improves the understanding of user queries by recognizing important entities.

4. Challenges in NER

- **Ambiguity**: Words can have multiple meanings or interpretations depending on context (e.g., "Apple" can refer to a fruit or a technology company).
- **Variability**: Entities may be expressed in various forms (e.g., "New York", "NY", "the Big Apple").
- **Context Sensitivity**: The classification of an entity can change based on surrounding text.

5. Approaches to NER

1. **Rule-Based Approaches**:
    - Use predefined lists of keywords and patterns to identify entities.
    - Example: Regular expressions to match patterns for dates or names.
2. **Machine Learning Approaches**:
    - Train models using annotated datasets where entities are labeled.
    - Algorithms include Conditional Random Fields (CRFs), Support Vector Machines (SVMs), and Naive Bayes.
3. **Deep Learning Approaches**:
    - Utilize neural networks, particularly Recurrent Neural Networks (RNNs), Long Short-Term Memory networks (LSTMs), and Transformers.
    - These models capture complex patterns in the data and have shown significant improvements in NER performance.

6. Tools and Libraries for NER

- **spaCy**: An efficient NLP library that provides a built-in NER model for various languages.
- **NLTK**: Offers basic NER functionalities and interfaces with other NER models.
- **Stanford NER**: A Java-based NER toolkit with models trained on various datasets.
- **Hugging Face Transformers**: Provides pre-trained models for NER tasks using state-of-the-art transformers.

Code:

```
# Install spaCy
!pip install spacy

# Download the spaCy model for English
!python -m spacy download en_core_web_sm
import spacy

# Load the pre-trained spaCy model
nlp = spacy.load('en_core_web_sm')
```

```
# Sample text for NER
text = "Apple Inc. is looking at buying U.K. startup for $1 billion. The meeting is scheduled for September 12,
2024."
```

```
# Process the text with spaCy
doc = nlp(text)
```

```
# Extract entities
for ent in doc.ents:
    print(f"{ent.text}: {ent.label_}")
```

```
Apple Inc.: ORG
U.K.: GPE
$1 billion: MONEY
September 12, 2024: DATE
```

```
# Install Hugging Face transformers
!pip install transformers
from transformers import pipeline

# Initialize the NER pipeline
ner_pipeline = pipeline("ner", model="dbmdz/bert-large-cased-finetuned-conll03-english")
```

```
# Sample text for NER
text = "Apple Inc. is looking at buying U.K. startup for $1 billion. The meeting is scheduled for September 12,
2024."
```

```
# Perform NER
results = ner_pipeline(text)
```

```
# Display the results
for entity in results:
    print(f"{entity['word']}: {entity['entity']}")
```

```
/usr/local/lib/python3.10/dist-packages/transformers/tokenization_utils_base.py:1601: FutureWarning: `clean_up_tokenization_spaces` was not set. It will be set to `True` by default. This
  warnings.warn(
Apple: I-ORG
Inc: I-ORG
U: I-LOC
K: I-LOC
```

Conclusion: