Natural Language Processing using Python Programming

Notebook 05.1: Introduction to Named Entity Recognition (NER)

Python 3.8+ NLTK Latest SpaCy Latest License MIT

Part of the comprehensive learning series: Natural Language Processing using Python Programming

Learning Objectives:

- Master Named Entity Recognition (NER) concepts and real-world applications
- Understand common entity types: PERSON, ORG, LOC, DATE, GPE and their significance
- Implement entity extraction using NLTK's chunking-based approach
- Learn the IOB/BILOU tagging scheme for sequence labeling
- Build foundation for advanced entity recognition with SpaCy

---

- **Named Entity Recognition (NER)** is the task of identifying and classifying named entities (such as people, organizations, locations, and dates) in text.

- It's a critical step for converting unstructured text into structured data, often serving as the foundation for search engines, question-answering systems, and knowledge graph construction.

1. What are Named Entities and Why are They Important?

- Named Entities are real-world objects that can be denoted with a name.

- Identifying them answers the 'who, what, where, and when' questions in a text.

Common Entity Types

- While various tagsets exist, the following are standard:

Entity Type	Description	Example
PERSON	People, including fictional characters.	*Rohit Sharma, Will Smith*
ORG	Companies, agencies, institutions.	*Apple, World Health Organization*

Entity Type	Description	Example
LOC	Non-geopolitical locations (geographical features).	*Mount Everest, Amazon River*
DATE	Absolute or relative dates/periods.	*2025, next week, March*
GPE	Geopolitical entity (countries, cities, states).	*India, London, California*

Importance (Real-World Application)

- In a legal context, NER can quickly extract all parties, dates, and locations from a contract, allowing for automated indexing and search.

2. Basic NER with NLTK (Entity Chunking)

- NLTK's approach to NER is based on a process called **Chunking** (or shallow parsing), which combines grammatically related tokens into larger chunks.
- NLTK uses a pre-trained classifier that relies on POS tags (Chapter 4.1) to identify and label named entities.

Note on NLTK NER:

- NLTK's NER is useful for understanding the conceptual process but is generally less accurate and robust than SpaCy's for production use.
- It requires tokens to be POS-tagged first.

In [1]:

```
# NLTK Named Entity Recognition (NER) Example
# This code demonstrates how to perform Named Entity Recognition using NLTK.
# It includes tokenization, POS tagging, and applying NER chunking.
import nltk
from nltk.tokenize import word_tokenize

# Ensure necessary NLTK resources are downloaded
nltk.download('maxent_ne_chunker_tab', quiet=True)
nltk.download('words', quiet=True)
nltk.download('averaged_perceptron_tagger', quiet=True)

sample_sentence = "Tim Cook, the CEO of Apple, visited London on Tuesday."

# 1. Tokenize the text
tokens = word_tokenize(sample_sentence)

# 2. POS Tag the tokens
tagged_tokens = nltk.pos_tag(tokens)

# 3. Apply NER (ne_chunk) to the tagged tokens
ner_tree = nltk.ne_chunk(tagged_tokens)

print("NLTK NER Chunking Result (Tree Format):")
print(ner_tree)
```

```
NLTK NER Chunking Result (Tree Format):
(S
  (PERSON Tim/NNP)
  (GPE Cook/NNP)
  ,/,
  the/DT
  (ORGANIZATION CEO/NNP)
  of/IN
  (GPE Apple/NNP)
  ,/,
  visited/VBD
  (GPE London/NNP)
  on/IN
  Tuesday/NNP
  ./.)
```

- NLTK's built-in NER (based on maxent_ne_chunker_tab) is not state-of-the-art — it can make tagging errors:
  - 'CEO' isn't an organization, but might be labeled as such.
  - 'Apple' being tagged as GPE (instead of ORGANIZATION) can happen.

Extracting Entities Programmatically

- The output is a tree structure where chunks are labeled with entity types.
- We can iterate through the tree to extract the entities in a simple list format.

In [2]:

```python
# Function to extract entities from the NER tree
def extract_entities_from_tree(tree):
    entities = []
    for chunk in tree:
        # Check if the chunk is a Named Entity (i.e., has a label)
        if hasattr(chunk, 'label') and chunk.label():
            entity_type = chunk.label()
            # Join the words/tokens within the chunk
            entity_text = ' '.join([c[0] for c in chunk])
            entities.append((entity_text, entity_type))
    return entities

extracted_entities = extract_entities_from_tree(ner_tree)

print("Extracted Entities (NLTK):")
for entity, entity_type in extracted_entities:
    print(f" - {entity:<15} : {entity_type}")
```

```
Extracted Entities (NLTK):
 - Tim             : PERSON
 - Cook            : GPE
 - CEO             : ORGANIZATION
 - Apple           : GPE
 - London          : GPE
```

3. The IOB/BILOU Tagging Scheme (Conceptual)

- Behind the scenes, NER models often use a sequence tagging scheme to label multi-word entities.

- The most common is the **BIO** (Begin, Inside, Outside) or **BILOU** scheme.

Tag	Meaning	Example
B-ORG	**B**eginning of an Organization entity	*B-ORG (Google)*
I-ORG	**I**nside an Organization entity	*I-ORG (Labs)* (in 'Google Labs')
O	**O**utside an entity (not an entity)	*O (is), O (the)*

Example Sequence: `[B-PER John] [I-PER Smith] O is [B-ORG CEO] O of [B-ORG IBM] O.`

- Understanding this is key if you ever need to train or evaluate a custom NER model.

4. Summary and Next Steps

- We introduced NER, its core entity types, and demonstrated how NLTK uses chunking (relying on POS tags) to extract entities.

- While NLTK is great for theory, the industry standard for production NER is **SpaCy** due to its speed and high-accuracy models.

Key Takeaways

- **NER Fundamentals:** We successfully mastered Named Entity Recognition concepts, understanding how to identify and classify real-world entities in text.

- **Entity Types Mastery:** We learned the standard entity categories (PERSON, ORG, LOC, DATE) and their critical importance for structured data extraction.

- **NLTK Implementation:** We implemented entity extraction using NLTK's chunking approach, understanding how POS tags enable entity identification.

- **Sequence Tagging Understanding:** We explored the IOB/BILOU tagging scheme, building theoretical foundation for advanced NER models.

Next Notebook Preview

- With NER fundamentals mastered, we're ready to explore **production-grade entity recognition**.

- The next notebook will dive into **NER with SpaCy**, leveraging state-of-the-art models for high-accuracy entity extraction and classification.

About This Project

This notebook is part of the **Natural Language Processing using Python Programming for Beginners** repository - a comprehensive, beginner-friendly guide for mastering NLP using Python, NLTK, and SpaCy.

Repository: `NLP`

Author

Prakash Ukhalkar

GitHub prakash-ukhalkar

Built with ❤ for the Python community