UNIVERSIDADE CATÓLICA PORTUGUESA

BRAGA

Behavior Analysis Technologies

Session 14

Named Entity Recognition

Applied Data Science

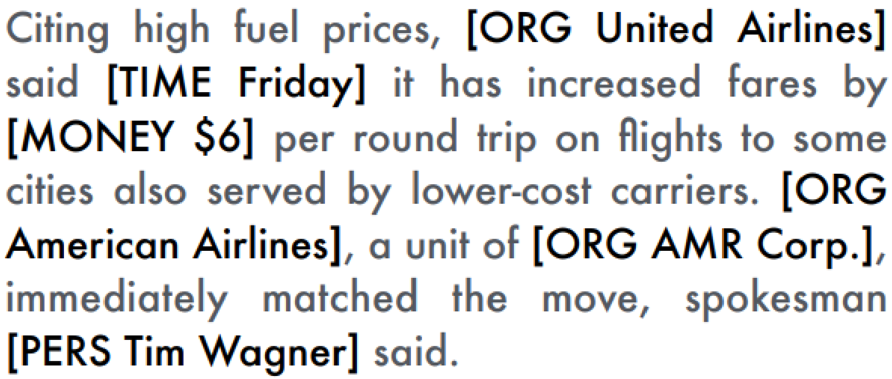
2024/2025

Named Entity Recognition (NER)

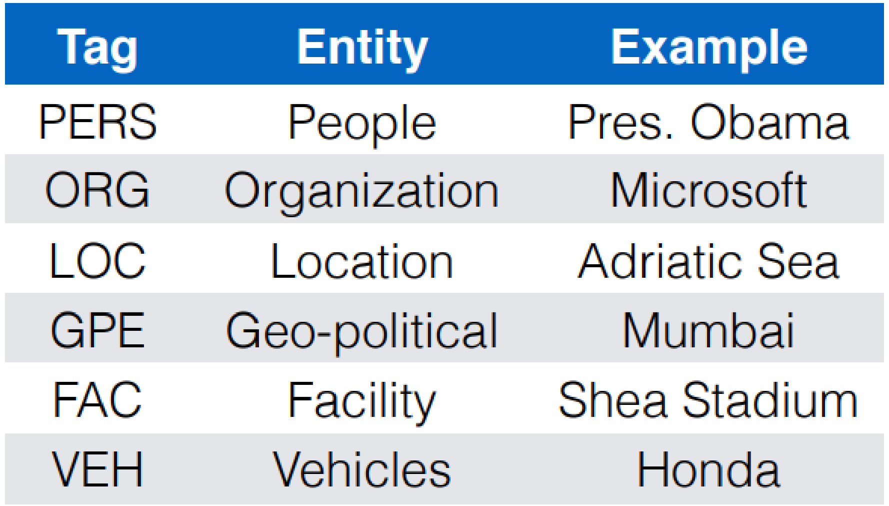
- NER is a subtask of information extraction that **identifies named entities** in text and classifies them into **predefined categories**.

- **Examples of entities**:
  - Person names (e.g., *Albert Einstein*)
  - Locations (e.g., *Paris*)
  - Organizations (e.g., *United Nations*)
  - Dates, monetary values, etc.

NER Example and Tags



Citing high fuel prices, [ORG United Airlines] said [TIME Friday] it has increased fares by [MONEY $6] per round trip on flights to some cities also served by lower-cost carriers. [ORG American Airlines], a unit of [ORG AMR Corp.], immediately matched the move, spokesman [PERS Tim Wagner] said.



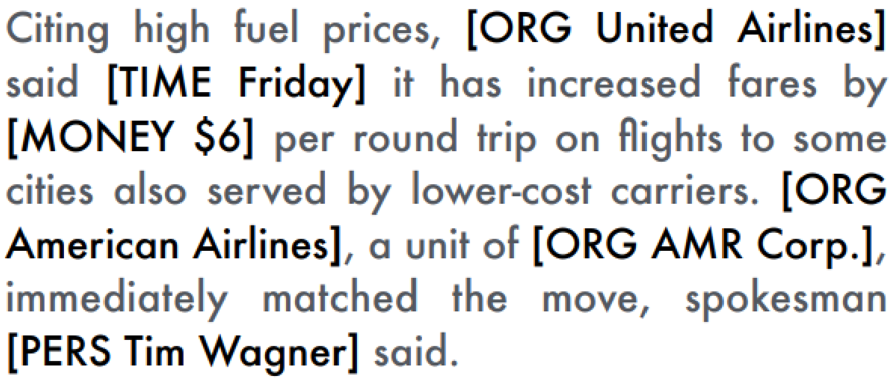
Tag	Entity	Example
PERS	People	Pres. Obama
ORG	Organization	Microsoft
LOC	Location	Adriatic Sea
GPE	Geo-political	Mumbai
FAC	Facility	Shea Stadium
VEH	Vehicles	Honda

Named Entity Recognition (NER)

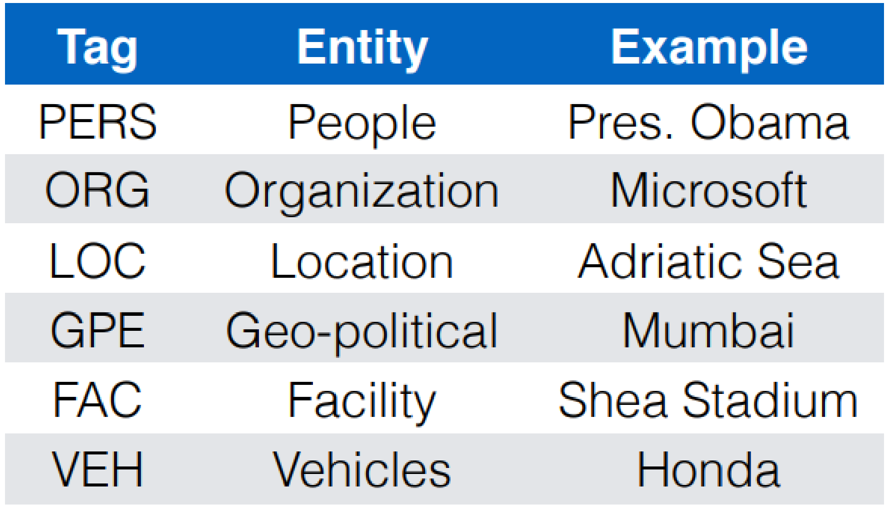
- Clauses are typically assigned an **entity type** from a **predefined list**, with each type having distinct **contextual indicators** that help identify entities of that category.

- For example, dates and times often appear in **recognizable formats**, while names of people are frequently introduced with **specific cues** in the surrounding text (e.g., "**Dr.** John Smith" or "**Ms.** Jane Doe").

NER Example and Tags



Citing high fuel prices, [ORG United Airlines] said [TIME Friday] it has increased fares by [MONEY $6] per round trip on flights to some cities also served by lower-cost carriers. [ORG American Airlines], a unit of [ORG AMR Corp.], immediately matched the move, spokesman [PERS Tim Wagner] said.



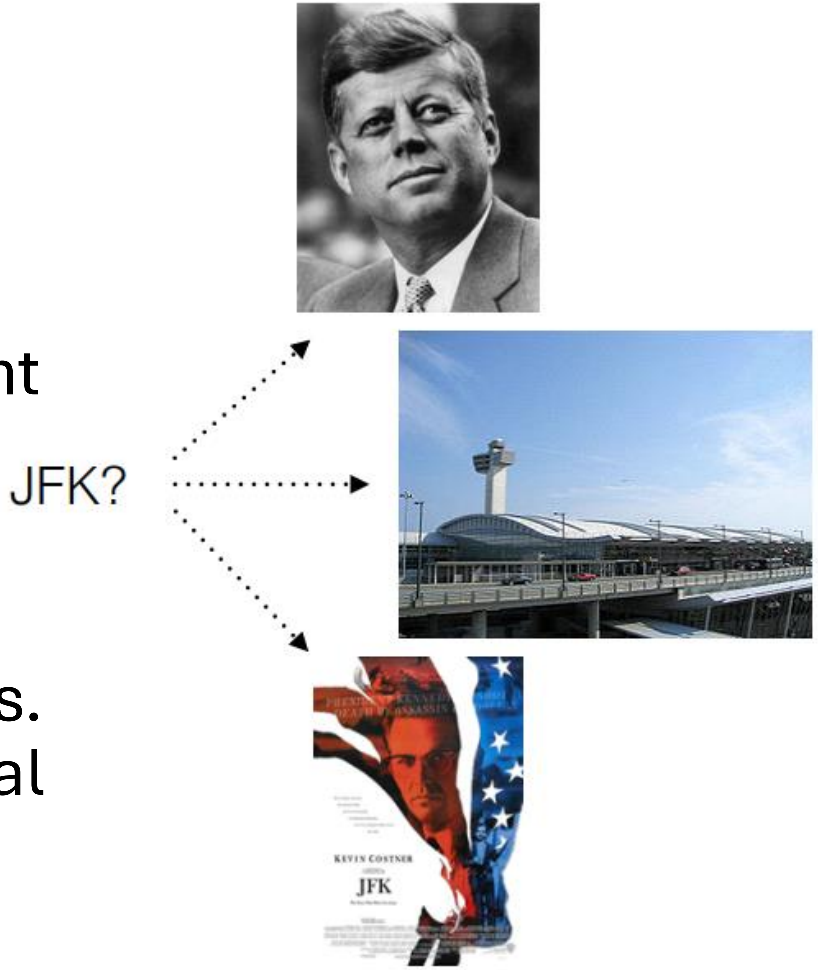
Tag	Entity	Example
PERS	People	Pres. Obama
ORG	Organization	Microsoft
LOC	Location	Adriatic Sea
GPE	Geo-political	Mumbai
FAC	Facility	Shea Stadium
VEH	Vehicles	Honda

Why is NER Important?

- Applications:
  - **Information Extraction:** Extract key information from documents.
  - **Question Answering:** Identify answers within large texts.
  - **Sentiment Analysis:** Analyze opinions tied to specific entities (e.g., brands).
  - **Search Engines:** Improve search relevance by identifying named entities.

Ambiguity in NER

- NER systems often have to deal with several important types of ambiguity:
  - **Reference resolution:** the same name can refer to different entities of the same type. For instance, JFK can refer to a former US president or his son.
  - **Cross-type Confusion:** the identical entity mentions can refer to entities of different types. For instance, JFK also names an airport, several schools, bridges, etc.



JFK?

How does NER work?

- Key steps:

1. **Tokenization:** Splits text into words or phrases.

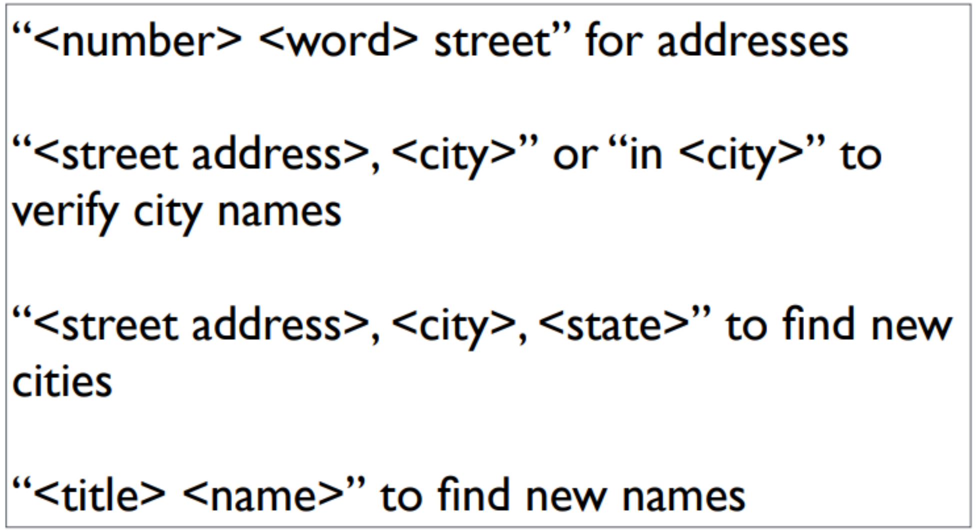
2. **Feature Extraction:** Uses features like shape, POS tags, and surrounding words.

3. **Classification:** Models predict entity types based on extracted features.

Rule-Based NER

- **Rule-based systems** for NER are effective for certain entity classes.
- Many of them use **lexicons**, which lists names, organizations, locations, etc.
- Rules can also be crafted using **regular expressions** or other pattern matching tools. The rules may be **built by hand**, or with **machine learning**.
- **Fast** and **interpretable** but **limited by language rules**.

Entity Patterns



"<number> <word> street" for addresses

"<street address>, <city>" or "in <city>" to verify city names

"<street address>, <city>, <state>" to find new cities

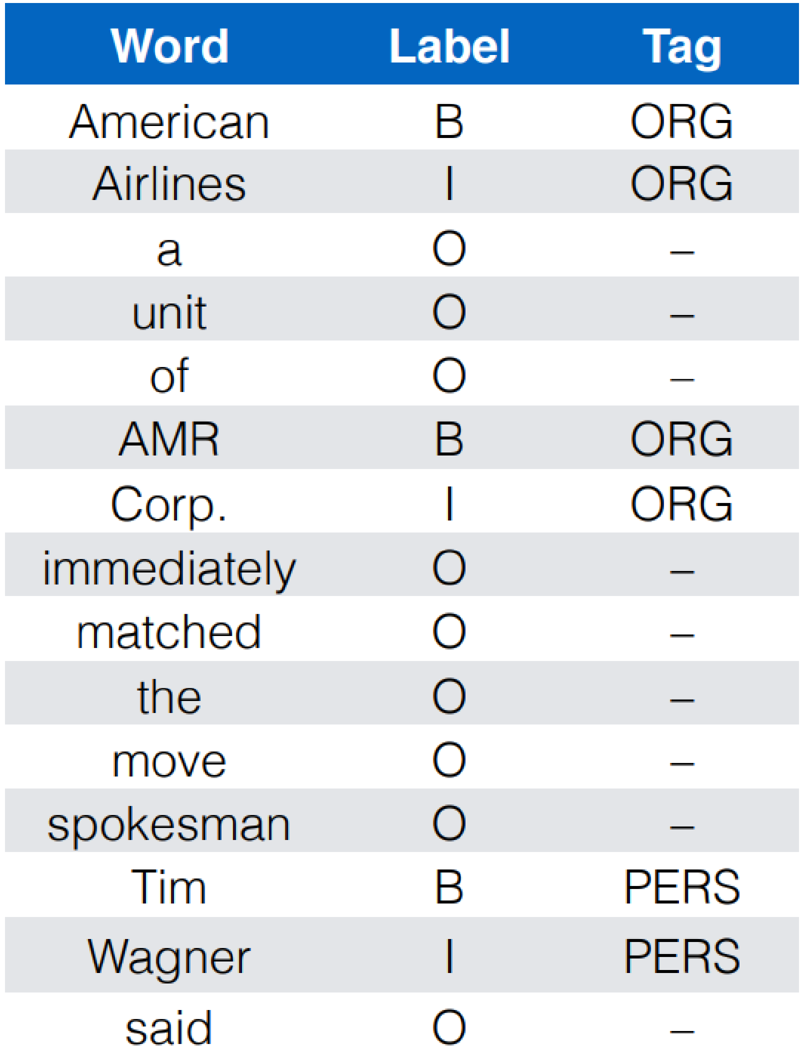
"<title> <name>" to find new names

NER with Sequence Tagging

- Sequence tagging is a common ML approach to NER.

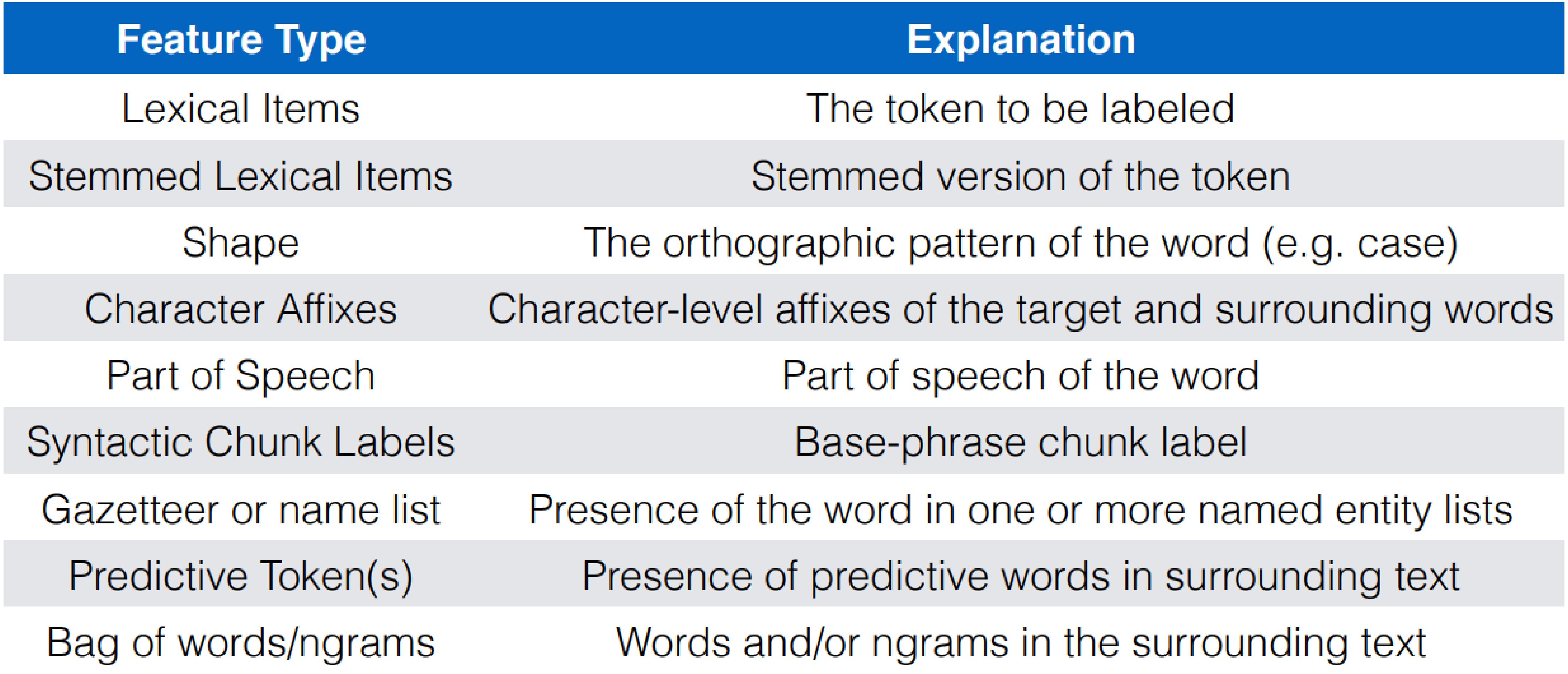
- Tokens are labeled as one of:
  - **B**: Beginning of an entity
  - **I**: Inside an entity
  - **O**: Outside an entity

- **Machine Learning** models are trained on a variety of **text features** to accomplish this.



Word	Label	Tag
American	B	ORG
Airlines	I	ORG
a	O	–
unit	O	–
of	O	–
AMR	B	ORG
Corp.	I	ORG
immediately	O	–
matched	O	–
the	O	–
move	O	–
spokesman	O	–
Tim	B	PERS
Wagner	I	PERS
said	O	–

Features for Sequence Tagging



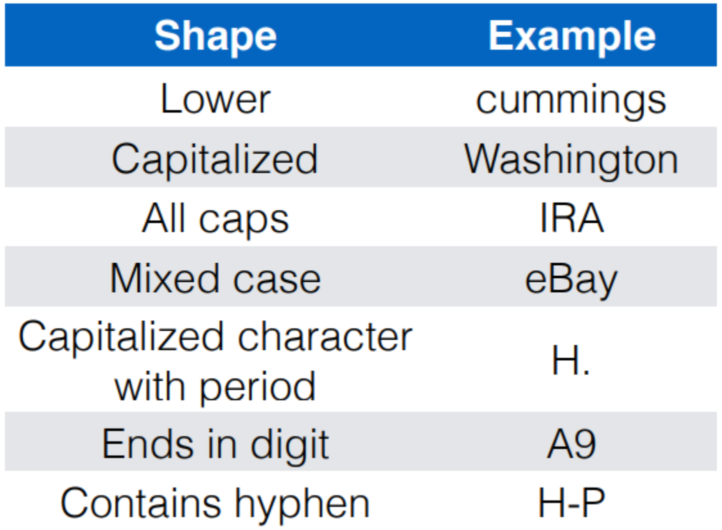
Feature Type	Explanation
Lexical Items	The token to be labeled
Stemmed Lexical Items	Stemmed version of the token
Shape	The orthographic pattern of the word (e.g. case)
Character Affixes	Character-level affixes of the target and surrounding words
Part of Speech	Part of speech of the word
Syntactic Chunk Labels	Base-phrase chunk label
Gazetteer or name list	Presence of the word in one or more named entity lists
Predictive Token(s)	Presence of predictive words in surrounding text
Bag of words/ngrams	Words and/or ngrams in the surrounding text

Word Shape

- In English, the shape feature is one of the most predictive of entity names.

- It is particularly useful for identifying businesses and products like Yahoo!, eBay, or iMac.

- Shape is also a strong predictor of certain technical terms, such as gene names.



Shape	Example
Lower	cummings
Capitalized	Washington
All caps	IRA
Mixed case	eBay
Capitalized character with period	H.
Ends in digit	A9
Contains hyphen	H-P

Context-Based NER

- **Rule-based NER** systems will inevitably **miss some entities**.
- All **lexicons are incomplete**, because new names are continually invented.
- Pattern matching doesn’t work for every entity type, and is at odds with the creativity put into writing.
- **Statistical NER** techniques instead identify entities using the terms in and around them.

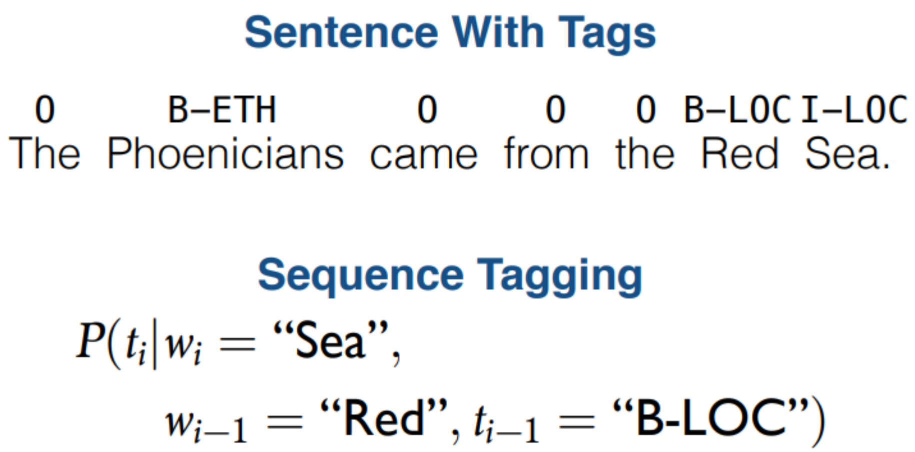
Machine Learning Approaches

- Traditional machine learning approaches for NER rely on **hand-crafted features** and supervised algorithms to classify tokens as specific entities.

- Some common algorithms used are **Conditional Random Fields** (CRFs) and **Hidden Markov Models** (HMMs).

Hidden Markov Models

- **Hidden Markov Model (HMM):** A machine learning framework for labeling sequences.
- **Sequential Labeling:** Each item in a sequence is tagged based on the assumption that its label depends on a limited number of preceding items.
- **Dependency on Prior Decisions:** Decisions made for previous items influence the labeling of the next item in the sequence.



Sentence With Tags

O	B-ETH	O	O	O	B-LOC	I-LOC
The	Phoenicians	came	from	the	Red	Sea.

Sequence Tagging

$$P(t_i \mid w_i = \text{“Sea”}, w_{i-1} = \text{“Red”}, t_{i-1} = \text{“B-LOC”})$$

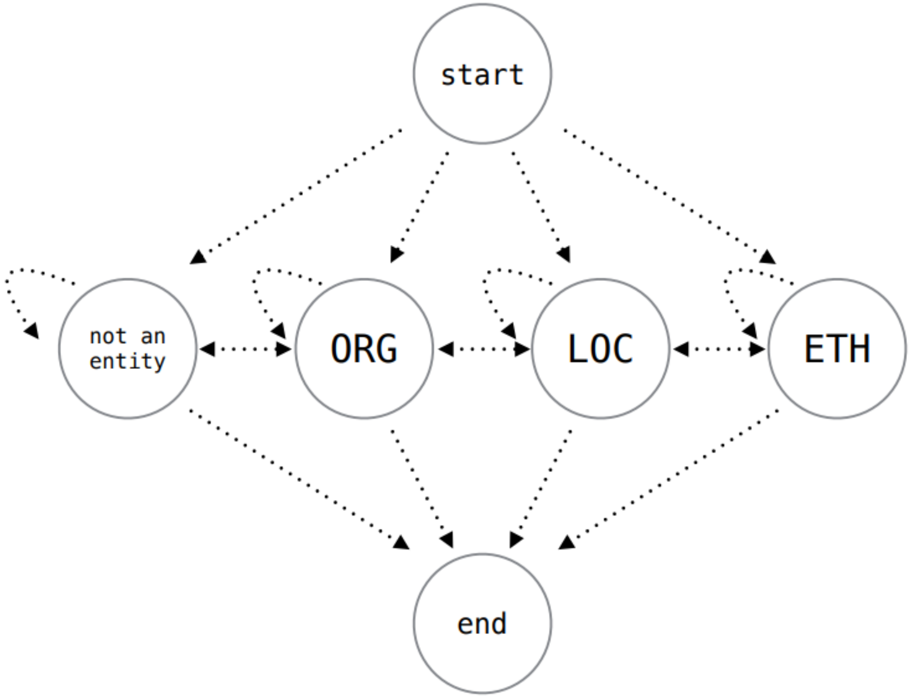
Hidden Markov Models

- A HMM describes a process as a series of states, each with some **probability distribution over the vocabulary**.

- When we want to assign a tag to some word w_i in a sentence, we only consider:
  - The properties of $\mathbf{w_i}$
  - The properties and tags assigned to $\mathbf{w_{i-1}}$ **through** $\mathbf{w_{i-k}}$ for some small constant k

- We assume that for words before w_{i-k} or after w_i have no information about the tag for w_i, mainly because this simplifies computation.

State diagram for NER tagging



* Any entity type state can transition to any other. Some arrows omitted for clarity.

Forward-Backward Algorithm

- HMMs are commonly trained using a dynamic programming technique called the **Forward-Backward algorithm**. This algorithm has three steps:

1. **Forward step:** Move through the sequence in increasing order calculating $P(t_i|w_1, \ldots, w_i)$.
2. **Backward step:** Move backward through the sequence, calculating $P(t_i|w_{i+1}, \ldots, w_n)$.
3. **Smoothing step:** Smooth together the two probabilities to calculate $P(t_i|w_1, \ldots, w_n)$

Deep Learning Approaches

- Deep learning has significantly advanced NER performance, especially with models that can **automatically learn complex features and contextual relationships from data**.

- Some popular deep learning models for NER include **Recurrent Neural Networks (RNNs)** and **Transformer-based models** like BERT.

Challenges in NER

- **Ambiguity:** Words may belong to multiple entity types (e.g., "Apple" as fruit or company).

- **Complex Phrases:** Multi-word entities are harder to detect

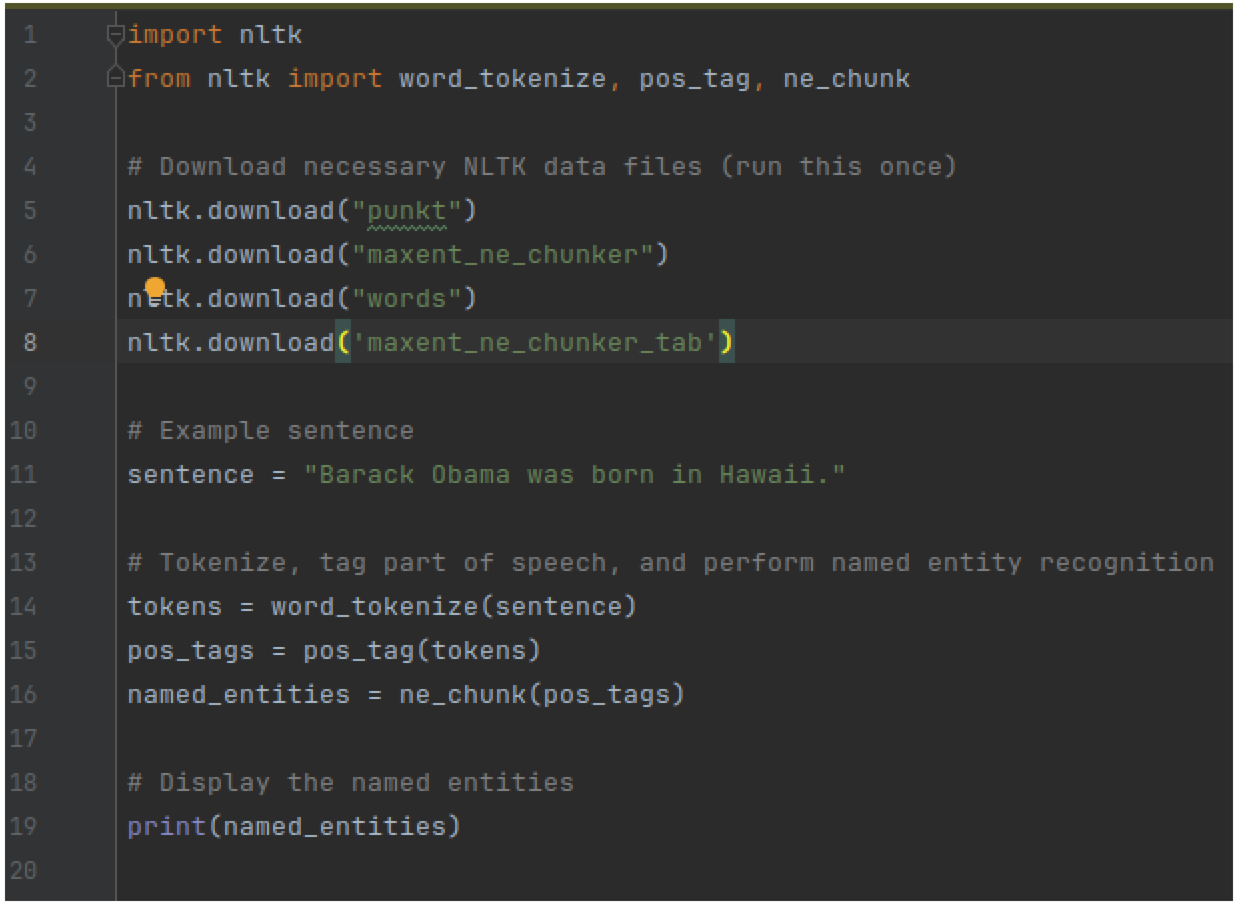
- **Domain Adaptation:** Different fields require specific entity types and vocabularies.

- **Low-resource Languages:** Less data for effective training in some languages.

NER in Practice

- Libraries and Tools:
  - **SpaCy:** Fast and easy-to-use NLP library.
  - **NLTK:** Classic NLP library, though not optimized for NER.
  - **Hugging Face Transformers:** Pre-trained models for advanced NER tasks.

NER with nltk



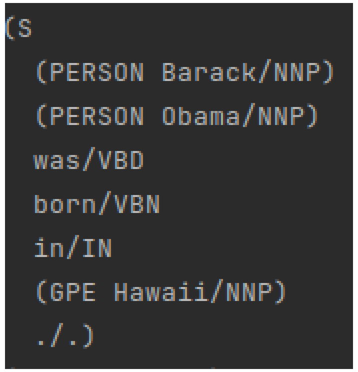
```python
import nltk
from nltk import word_tokenize, pos_tag, ne_chunk

# Download necessary NLTK data files (run this once)
nltk.download("punkt")
nltk.download("maxent_ne_chunker")
nltk.download("words")
nltk.download('maxent_ne_chunker_tab')

# Example sentence
sentence = "Barack Obama was born in Hawaii."

# Tokenize, tag part of speech, and perform named entity recognition
tokens = word_tokenize(sentence)
pos_tags = pos_tag(tokens)
named_entities = ne_chunk(pos_tags)

# Display the named entities
print(named_entities)
```



```
(S
  (PERSON Barack/NNP)
  (PERSON Obama/NNP)
  was/VBD
  born/VBN
  in/IN
  (GPE Hawaii/NNP)
  ./.)
```