Applied Text Analytics & Natural Language Processing

with Dr. Mahdi Roozbahani
& Wafa Louhichi

Sequence Labelling
Named Entity Recognition (NER)

Learning Objectives

In this lesson, you will learn a sequence labeling method named Named Entity Recognition (NER)

- NER Chunker
- Transformer Models

Named Entity Recognition (NER)

- Named Entity (NE): (Abstract or physical) real-world objects that can be given a proper name.
  - Examples: Georgia Tech, France, United Airline
  - It is often a multi-word phrase (which makes NER difficult.)
- Named Entity Recognition (NER): finding the part of a text containing a NE and attributing a tag to it
- Most common tags:
  - **PER** (Person): “Marie Curie”
  - **LOC** (Location): “Atlanta”
  - **ORG** (Organization): “Georgia Tech”
  - **GPE** (Geo-Political Entity): "Boulder, Colorado"
  - Also includes non-entities: **dates**, **times**, **prices**
- Application:
  - Sentiment analysis
  - Question answering: when a question asks about an entity in the sentence
  - Information summarization: we can use the named entities in a text to extract important information

NE Type	Examples
ORGANIZATION	*Georgia-Pacific Corp., WHO*
PERSON	*Eddy Bonte, President Obama*
LOCATION	*Murray River, Mount Everest*
DATE	*June, 2008-06-29*
TIME	*two fifty a m, 1:30 p.m.*
MONEY	*175 million Canadian Dollars, GBP 10.40*
PERCENT	*twenty pct, 18.75 %*
FACILITY	*Washington Monument, Stonehenge*
GPE	*South East Asia, Midlothian*

https://huggingface.co/course/chapter1/1
https://en.wikipedia.org/wiki/Named_entity
https://web.stanford.edu/~jurafsky/slp3/slides/8_POSNER_intro_May_6_2021.pdf

Named Entity Recognition (NER)

- **Some of the algorithms for NER:**
  - Sequence models: RNN and Transformer
  - Large language models: BERT (fine-tuned)
  - NE Chunker classifier: Uses MaxEnt classifier (Maximum Entropy classifier)

  (NOTE: POS tag outputs as input for the NER task)
- **NER challenges:**
  - Segmentation: Unlike POS tagging, where each word is assigned a tag, NER requires finding entities which are often multi-word phrases.
  - Text ambiguity: The meaning of a word/phrase can vary in different contexts.
    - E.g., "Franklin" can be a name of a person or the name of a city in North Carolina. So, should it be tagged as a person (PER) or a location (LOC)?
- **IOB:**
  - Using IOB (or BIO) tagging, we will have one tag (person, location, etc.) per NE. We also specify the beginning of the NE (B), between or inside the NE (I), and outside any NE (O).

https://huggingface.co/course/chapter1/1
https://en.wikipedia.org/wiki/Named_entity
https://web.stanford.edu/~jurafsky/slp3/slides/8_POSNER_intro_May_6_2021.pdf
https://mattshomepage.com/articles/2016/May/23/nltk_nec/

Named Entity Recognition (NER)

- **Some of the algorithms for NER:**
  - Sequence models: RNN and Transformer
  - Large language models: BERT (fine-tuned)
  - NE Chunker classifier: Uses MaxEnt classifier (Maximum Entropy classifier)

  (NOTE: POS tag outputs as input for the NER task)
- **NER challenges:**
  - Segmentation: Unlike POS tagging, where each word is assigned a tag, NER requires finding entities which are often multi-word phrases.
  - Text ambiguity: The meaning of a word/phrase can vary in different contexts.
    - E.g., "Franklin" can be a name of a person or the name of a city in North Carolina. So, should it be tagged as a person (PER) or a location (LOC)?
- **IOB:**
  - Using IOB (or BIO) tagging, we will have one tag (person, location, etc.) per NE. We also specify the beginning of the NE (B), between or inside the NE (I), and outside any NE (O). i.e. Mahdi Roozbahani visited Florida last month. The IOB: **Mahdi**-B-PER **Roozbahani**-I-PER **visited**-O **Florida**-B-LOC **last**-O **month**-O

https://huggingface.co/course/chapter1/1
https://en.wikipedia.org/wiki/Named_entity
https://web.stanford.edu/~jurafsky/slp3/slides/8_POSNER_intro_May_6_2021.pdf
https://mattshomepage.com/articles/2016/May/23/nltk_nec/

Named Entity Recognition (NER) - NLTK

```
import nltk //a Python library for NLP tasks
from nltk.tokenize import word_tokenize
from nltk.tag import pos_tag

text = 'Professor Roozbahani traveled from Atlanta to Paris via United Airlines to visit the Louvre
Museum' // text to be processed

raw_words = word_tokenize(text) //divides strings into lists of substrings
tags = pos_tag(raw_words) // attaches a POS tag to each word
ne = nltk.ne_chunk(tags,binary=True) // NLTK classifier trained to recognize named entities

from nltk.chunk import tree2conlltags
iob = tree2conlltags(ne)
print(iob)
```

- binary: if True, named entities are tagged as NE or not being a named entity (O). Otherwise, it will show the category labels, e.g., PERSON

https://analyticsindiamag.com/guide-to-named-entity-recognition-with-spacy-and-nltk/

Named Entity Recognition (NER) - NLTK

binary=True

```
>> [('Professor', 'NNP', 'B-NE'),
 ('Roozbahani', 'NNP', 'I-NE'),
 ('traveled', 'VBD', 'O'), VBD: verb, past tense
 ('from', 'IN', 'O'),
 ('Atlanta', 'NNP', 'B-NE'), NNP: singular noun
 ('to', 'TO', 'O'), TO: preposition or infinitive
marker
 ('Paris', 'NNP', 'B-NE'),
 ('via', 'IN', 'O'), IN: preposition or conjunction,
subordinating
 ('United', 'NNP', 'B-NE'),
 ('Airline', 'NNP', 'I-NE'),
 ('to', 'TO', 'O'),
 ('visit', 'VB', 'O'), VB: verb, base form
 ('the', 'DT', 'O'), DT: determiner
 ('Louvre', 'NNP', 'B-NE'),
 ('Museum', 'NNP', 'I-NE')]
```

binary=False

```
>> [('Professor', 'NNP', 'O'),
 ('Roozbahani', 'NNP', 'B-PERSON'),
 ('traveled', 'VBD', 'O'),
 ('from', 'IN', 'O'),
 ('Atlanta', 'NNP', 'B-GPE'),
 ('to', 'TO', 'O'),
 ('Paris', 'NNP', 'B-GPE'),
 ('via', 'IN', 'O'),
 ('United', 'NNP', 'B-GPE'),
 ('Airline', 'NNP', 'I-GPE'),
 ('to', 'TO', 'O'),
 ('visit', 'VB', 'O'),
 ('the', 'DT', 'O'),
 ('Louvre', 'NNP', 'B-ORGANIZATION'),
 ('Museum', 'NNP', 'I-ORGANIZATION')]
```

- B-NE: token that begins a NE
- I-NE: tokens inside a NE
- O: tokens outside of any NE

https://web.stanford.edu/~jurafsky/slp3/slides/8_POSNER_intro_May_6_2021.pdf

Named Entity Recognition (NER) – Transformer Models

Fine-tuning a transformer architecture with an attention mechanism

Using Keras, we can create a fine-tuned transformer and test it to get better performance for our example. We can choose the datasets to train on according to the testing sample

Steps for building and training a transformer model:

1. Preparing the dataset by creating tags for the NER task

```
iob_labels = ["B", "I"]

ner_labels = ["PER", "ORG", "LOC", "MISC"]

Example of iob tagging: [('Alex', 'B-PER'), ('is','O), ('going', 'O'), ('to', 'O'), ('Los','B-LOC'), ('Angeles','I-LOC')]
```

2. Creating the transformer architecture for NER and training on the training dataset - creating the Transformer block, Token and Position Embeddings
3. Tokenizing the text and converting it to IDs for the model
4. Compile and fit the model on training data
5. Testing the results on the example text

https://keras.io/examples/nlp/ner_transformers/
https://www.clips.uantwerpen.be/conll2003/ner/

Named Entity Recognition (NER) – Transformer Models

Fine-tuning a transformer architecture with an attention mechanism

Text Example:

```
'Professor Roozbahani traveled from Atlanta to Paris via United Airlines to visit the Louvre Museum’
```

Output:

```
>> ['O', 'O', 'O', 'O', 'B-ORG', 'O', 'B-ORG', 'O', 'B-LOC', 'I-ORG', 'O', 'O', 'O', 'O', 'O']
```

Named Entity Recognition (NER) – BERT Model

Using pre-trained BERT model from Hugging Face

```
!pip install transformers //Library from Hugging Face with all the models
from transformers import AutoTokenizer, TFBertForTokenClassification //importing tokenizer
and BERT models for the NER task
from transformers import pipeline

text = 'Professor Roozbahani traveled from Atlanta to Paris via United Airlines to
visit the Louvre Museum' // text to be processed

model_name = ”dslim/bert-base-NER” // a model fine-tuned for NER task
#This model only exists in PyTorch, hence we use from_pt flag to import in tensorflow
model = TFBertForTokenClassification.from_pretrained(model_name, from_pt = True)
tokenizer = AutoTokenizer.from_pretrained(model_name)
classifier = pipeline(‘ner’, model = model, tokenizer = tokenizer)

classifier(text) //running the model on the example text
```

https://huggingface.co/dslim/bert-base-NER

Named Entity Recognition (NER) – BERT Model

Using pre-trained BERT model from Hugging Face

```
>> [{'entity': 'B-PER', 'score': 0.99917233, 'index': 2, 'word': 'R', 'start': 10, 'end': 11}, {'entity':
'I-PER', 'score': 0.81094116, 'index': 3, 'word': '##oo', 'start': 11, 'end': 13}, {'entity': 'I-PER',
'score': 0.81378126, 'index': 4, 'word': '##z', 'start': 13, 'end': 14}, {'entity': 'I-PER', 'score':
0.98243654, 'index': 5, 'word': '##bah', 'start': 14, 'end': 17}, {'entity': 'I-PER', 'score':
0.96564984, 'index': 6, 'word': '##ani', 'start': 17, 'end': 20}, {'entity': 'B-LOC', 'score':
0.99956256, 'index': 9, 'word': 'Atlanta', 'start': 35, 'end': 42}, {'entity': 'B-LOC', 'score':
0.9995933, 'index': 11, 'word': 'Paris', 'start': 46, 'end': 51}, {'entity': 'B-ORG', 'score': 0.9988399,
'index': 13, 'word': 'United', 'start': 56, 'end': 62}, {'entity': 'I-ORG', 'score': 0.9974515, 'index':
14, 'word': 'Airlines', 'start': 63, 'end': 71}, {'entity': 'B-LOC', 'score': 0.71856314, 'index': 18,
'word': 'Lou', 'start': 85, 'end': 88}, {'entity': 'B-LOC', 'score': 0.95321137, 'index': 19, 'word':
'##vre', 'start': 88, 'end': 91}, {'entity': 'I-LOC', 'score': 0.8275989, 'index': 20, 'word': 'Museum',
'start': 92, 'end': 98}]
```

Summary

- We learned about NER using
- NLTK (NER Chunker)
- Encoder-decoder-based transfer model
- BERT (Encoder-based transformer model)