

```
In [1]: import sys
        sys.path.append('../')
        from cnt.model import load_ner_model
```

# Named Entity Recognition

## Define the path and name of the model

```
In [2]: model_directory = "../cnt/trained_model/ner/"
        model_name = "english_cno"
```

## Load the model

```
In [3]: model = load_ner_model(model_directory, model_name)
```

## Define an input sentence

```
In [4]: sentence = "Bare-headed bust of Antoninus Pius, right, wearing cuirass and paludamentum."
```

## There are three different outputs.

- use `predict\_single\_sentence` to receive the position of the predictions
- use `predict_single_sentence_clear` to receive the string representation instead of the position
- use `predict_single_sentence_clear` with `as_doc=True` to receive a spacy object that can be visualised using `displacy`

---

### Option 1

```
In [5]: prediction = model.predict_single_sentence(sentence)
```

```
In [6]: prediction
```

```
Out[6]: [(12, 16, 'OBJECT'),
         (20, 34, 'PERSON'),
         (51, 58, 'OBJECT'),
         (63, 75, 'OBJECT')]
```

---

### Option 2

```
In [7]: prediction = model.predict_single_sentence_clear(sentence, as_doc=False)
```

```
In [8]: prediction
```

```
Out[8]: [('bust', 'OBJECT'),
         ('Antoninus Pius', 'PERSON'),
         ('cuirass', 'OBJECT'),
         ('paludamentum', 'OBJECT')]
```

---

### Option 3

```
In [9]: prediction = model.predict_single_sentence_clear(sentence, as_doc=True)
```

```
In [10]: from spacy import displacy
colors = {'PERSON': 'mediumpurple', 'OBJECT': 'greenyellow', 'ANIMAL': 'orange', 'PLANT': 'red'}
options = {'ent': ['PERSON', 'OBJECT', 'ANIMAL', 'PLANT'], 'colors': colors}
displacy.render(prediction,
                 style='ent', jupyter=True, options=options)
```

Bare-headed bust **OBJECT** of Antoninus Pius **PERSON** , right, wearing cuirass **OBJECT** and paludamentum **OBJECT** .

---

## Relation Extraction

```
In [11]: from cnt.model import load_pipeline, predict_re_single_sentence
```

### Define the path and name of the model

```
In [12]: re_model_directory = "../cnt/trained_model/re/"
re_model_name = "english_cno"
```

### Load the model

```
In [13]: model = load_pipeline(re_model_directory, re_model_name)
```

### Define an input sentence

```
In [14]: sentence = "Bare-headed bust of Antoninus Pius, right, wearing cuirass and paludamentum."
```

### Use the predict\_re\_single\_sentence function for predicting on a single sentence

```
In [15]: y_pred = predict_re_single_sentence(model, sentence)
```

```
In [16]: y_pred
```

```
Out[16]: [('Antoninus Pius', 'PERSON', 'wearing', 'cuirass', 'OBJECT'),
          ('Antoninus Pius', 'PERSON', 'wearing', 'paludamentum', 'OBJECT')]
```