# Named Entity Recognition

## Objective

The aim is to study Named Entity Recognition ready for the next assignment. We will use spaCy for this.

## What is Named Entity Recognition?

In Assignment 1 you created a .json containing 20 queries. Each had a type:

{

"number":1,

"original\_query":"Who played keyboards in the band Dire Straits",

"keyword\_query":"dire straits keyboard",

"kibana\_query":

{

"query":

{

"multi\_match":

{

"query":"dire straits keyboard",

"fields":[],

"type":"best\_fields"

}

}

},

**"answer\_type":"person"**,

...

This was the expected type of the answer. The allowed types were:

'person', 'animal', 'organisation', 'event', 'place', 'date', 'reason', 'other'.

These are types of ‘Named Entities’. The term was invented at the *Message Understanding Conferences* (called MUC, similar to TREC) in the 1990s. These were organised by Ralph Grishman and Beth Sundheim, and funded by ARPA. See https://apps.dtic.mil/sti/pdfs/ADA633406.pdf.

A Named Entity (NE) is a piece of information which we might be interested in, which falls into a particular class.

When we specified the NE type in the .json, we were indicating the expected type of the answer. When creating the Gold Standard, you checked many documents to see if the required NE was present.

Following the MUC conferences, considerable interest was created in NEs. As a result, there has beein considerable work on trying to recognise NEs automatically. We call this Named Entity Recognition (NER).

In this lab, we are going to study NER and apply it to your Assignment 1 data. In Assignment 2, we will add NER to your existing work and use it to create a Question Answering system which will return and exact answer to a question, not just a document containing the answer.

To do this, we will carry out NER using a Python package called spaCy.

## Using spaCy for Named Entity Recognition (NER)

**Note: You must do this at the command line using exactly the commands shown. We assume you have a directory called c:\ir as previously instructed.**

First, check whether the spaCy package is installed:

c:\ir> python

>>> import spacy

>>>

If this succeeds and gives no error, then you have spaCy installed on your machine. SpaCy is in the CSEE labs. (If it does not work, see section **Installing spaCy** below.)

Now, leave Python using quit() and run the following at the command line:

c:\ir> python -m spacy download en\_core\_web\_sm

This will download the language data for English. You should see output like this:

Collecting en-core-web-sm==3.7.1

... a lot more output...

If this succeeds, everything should be ready. You can try this in Python:

import spacy

nlp = spacy.load("en\_core\_web\_sm")

doc = nlp( "Richard is a person." )

for ent in doc.ents:

print(ent.text, ent.start\_char, ent.end\_char, ent.label\_)

You should get output like this:

Richard 0 7 PERSON

That means that in the input ‘Richard is a person’ there was a Named Entity of type PERSON. It started at offset 0 and ended at offset 7.

## Applying NER to your Assignment Data

First, have a look at this very informative page: https://spacy.io/usage/processing-pipelines.

The NE types spaCy recognises are:

PERSON - People, including fictional.

NORP - Nationalities or religious or political groups.

FAC - Buildings, airports, highways, bridges, etc.

ORG - Companies, agencies, institutions, etc.

GPE - Countries, cities, states.

LOC - Non-GPE locations, mountain ranges, bodies of water.

PRODUCT - Objects, vehicles, foods, etc. (Not services.)

EVENT - Named hurricanes, battles, wars, sports events, etc.

WORK\_OF\_ART - Titles of books, songs, etc.

LAW - Named documents made into laws.

LANGUAGE - Any named language.

DATE - Absolute or relative dates or periods.

TIME - Times smaller than a day.

PERCENT - Percentage, including "%".

MONEY - Monetary values, including unit.

QUANTITY - Measurements, as of weight or distance.

ORDINAL - "first", "second", etc.

CARDINAL - Numerals that do not fall under another type.

Now, do these four tasks:

### Task 1: Examples of NEs

Choose any FOUR of the NE types in the above list EXCEPT PERSON (which we already showed). For each of the four, try to compose a sentence which contains the NE, and show that spaCY finds it, using the above code.

We will show the method, using another person. We compose a sentence: ‘Will Smith is a fictional character.’. Now enter into the above code and see the results:

doc = nlp( " Will Smith is a fictional character." )

for ent in doc.ents:

print(ent.text, ent.start\_char, ent.end\_char, ent.label\_)

Will Smith 1 11 PERSON

This produces the results as shown. Now, copy and paste all those four lines (doc = nlp...PERSON) into your .docx answers file (ready for upload to Moodle).

Do this for four NE types not including PERSON. You will find that the NER in spaCy usually works well.

## Task 2: NER in your Queries

Now, do exactly the same thing or your 20 queries, using the original\_query form. Paste the output just as above, plus a one line note about whether it is correct. Example:

doc = nlp( "Who played keyboards in the band Dire Straits" )

for ent in doc.ents:

print(ent.text, ent.start\_char, ent.end\_char, ent.label\_)

Dire Straits 33 45 LOC

**Note:** Incorrect. Dire Straits is an ORG not a LOC.

Do this for all your 20 queries and paste the results in your .docx answers file. As you see, the NER was incorrect, so our Note stated this. If the NER is correct you can simply write ‘Note: Correct’ on the last line.

## Task 3: NER in your Matching Documents

Now, choose supporting sentences from FOUR of your queries (preferably queries of 4 different query types, or as many as you have (two was the minimum requirement). Do exactly the same thing for those. Give the original\_query and the query\_type, then the code, and finally the Note. Here is an example.

"original\_query":"Who played keyboards in the band Dire Straits"

"answer\_type":"person"

doc = nlp( "At the time of their initial breakup in September 1988, the band featured Mark Knopfler, Illsley, rhythm guitarist and backing vocalist Jack Sonni, drummer Terry Williams and keyboardists Alan Clark and Guy Fletcher." )

for ent in doc.ents:

print(ent.text, ent.start\_char, ent.end\_char, ent.label\_)

September 1988 40 54 DATE

Mark Knopfler 74 87 PERSON

Illsley 89 96 PERSON

Jack Sonni 136 146 PERSON

Terry Williams 156 170 PERSON

Alan Clark 188 198 PERSON

Guy Fletcher 203 215 PERSON

**Note: All correct.**

Do that for each of four sentences for four different queries and add the complete text (from original\_query right up to **All correct**) to your answers file.

If NER is not correct for your sentences, do not worry. Just note at the end which NEs are incorrect. The above example happened to work perfectly, this is not always the case.

That is everything. Hopefully it gives you an idea of what NER is and how accurate (or in accurate) it is on your actual queries and answer sentences.

## Installing spaCy

In case your PC has not got spaCy installed (or on your own PC at home) you can do the following. Note: The first step is to check that pip itself is up-to-date. Most problems occurring when using pip are because it is not up-to-date, so always check this first. Here are the steps, all from the command line:

python.exe -m pip install --upgrade pip

pip install -U spacy

python -m spacy download en\_core\_web\_sm

You should now be able to use spaCy as described above.

## Questions for Upload

Put the results for Tasks 1-3 into the .docx answer file and upload to Moodle by the deadline.