# COM6115: Lab Class 7

## NER (GATE vs spaCy)

This lab aims to perform some standard NLP processing using GATE and spaCy on a corpus from Snopes' (a fact checking website) Twitter account, using a list of countries and a list of languages to create gazetteers.

## Data Description

The dataset is a corpus of text from the Snopes Twitter account . The provided file is comma-separated. For each sample, we are interested in the **claim** column only. In the following table you can find several example sentences.

| SentenceID | Text |
|---|---|
| 21 | A video shows Dutch politician Tunahan Kuzu putting a grilled cheese in his pocket before an interview. |
| 27 | British actor Rowan Atkinson, best known as 'Mr. Bean,' has died. |
| 102 | Mike Pence once said that smoking doesn't kill people. See Example(s) |

Two other lists are also available:

- Comma-separated list of countries and their two digit code: `countries.csv`
- Comma-separated list of languages and a position (irrelevant for this lab): `languages.csv`

## Gazetteers with GATE

**GATE** is an open source software toolkit with multiple capabilities to solve text processing problems. The traditional version is a Java application, however, we can now use a lot of functionalities in Python as well (https://gatenlp.github.io/python-gatenlp/).

In addition, multiple tools developed by the GATE team in Sheffield are also available via APIs in the **GATE Cloud** (let me know if you are interested in this, I may be able to get some additional quota for you).

### Roadmap

Follow the instructions in this notebook in Google Colab: https://colab.research.google.com/drive/1pfsiAq6gVHd9AHvjdN95zNz97M5Itpon?usp=sharing[1].

You will need to **save a copy** of the Colab Notebook in your own Google Drive before starting. Various approaches for creating the gazetteers are shown in this document.

Please run them and familiarise yourself with the approaches. You can also note that there is a way to use some of the functionality of **NLTK** as well.

[1]This notebook is a copy of the documentation available at https://nbviewer.org/urls/gatenlp.github.io/python-gatenlp/gazetteers.ipynb

After getting familiar with the code and how to create a gazetteer in GATE, you will be able to create two gazetteers:

- Country: use the list in `countries.csv`. As features you should use the two digit code available in the document and a Wikipedia URL.

  The URL can be built by adding the country name to the end of the pattern `https://en.wikipedia.org/wiki/`. For example, for the country Brazil you will have `https://en.wikipedia.org/wiki/Brazil`. For country names with more than one word (e.g. "Bosnia and Herzegovina"), you should replace the spaces by an underscore (_) (e.g. `https://en.wikipedia.org/wiki/Bosnia_and_Herzegovina`).

- Languages: use the list in `languages.csv`. As features, you should use the Wikipedia URL.

  Again, this is not available, but you can create it starting with the pattern `https://en.wikipedia.org/wiki/` + Language name + `_language`. If a language name has more than one word, you should use the same strategy as above.

We want to answer the following question: What are the entities and how do they co-occur?

To annotate the data you can use any of the approaches shown in the notebook (just choose the one you think will work best).

1. Get some statistics about the corpus
   - Create a graphic to show the distribution of the number of entities per claim.
   - What are the 20 most popular countries in that corpus? (popular means that appear more frequently)
3. Test the model
   - Find claims where the entities in the gazetteers occur and visualise them.
2. Fool the model
   - Generate some sentences that will be well/badly processed by your gazetteers.
   - What are the flaws of the model?

## Notes and comments

- Consider using the **Pandas** library to load the data. https://pandas.pydata.org/.
- Use Google Colab (it has dependencies already installed): https://colab.research.google.com/drive/1pfsiAq6gVHd9AHvjdN95zNz97M5Itpon?usp=sharing
- Go to Files (icon on the left hand side) → Upload the files `snopes.csv`, `countries.csv`, `languages.csv`.
- Alternatively, to run a notebook, you need to install jupyter https://jupyter.org:
  - with conda: `conda install -c conda-forge notebook`

- with pip: `pip install notebook`
- Then you will need to follow the instructions here to create an environment for the GATE python toolkit: https://gatenlp.github.io/python-gatenlp/installation.html
- then run the command `jupyter notebook` in your terminal (you may need to change the kernel to use the `gatenlp` environment).

# NER with spaCy

**spaCy** (https://spacy.io/api) is an NLP API that processes documents into a pipeline that can tokenize, lemmatize, tag and extract entities (among others) from them.

In this lab we are interested in the NER task. The model integrated into spaCy uses neural networks[2].

The goal is to create an NER system by setting up a correct pipeline, study the strength and weaknesses of the model and extract some useful information about the data itself and the detected named entities. See below for details.

## Roadmap

Use the python notebook `NERLab.ipynb`.

What are the entities and how do they co-occur?

1. Implement some preprocessing steps using Pandas and spaCy:
   - This is made very easy by spaCy. Look at the language processing pipeline: https://spacy.io/usage/processing-pipelines
   - We are interested in tokenization and entity recognition (POS tagger and dependency parser can be deactivated).
   - Remove the duplicate lines (use the **drop_duplicates** function from Pandas).
   - Print the first 5 lines along with the detected entities.
   - What is the impact of lower- or upper-casing the text on the detected entities?

6. Get some statistics about the corpus
   - Create a graphic to show the distribution of the number of entities per claim.
   - What are the 20 most popular entities in that corpus? (popular means that appear more frequently)

3. Fool the model
   - Generate some sentences that will be well/badly processed by the spaCy NER model.

[2]While outside the scope of this lab, those who are interested can look at https://spacy.io/universe/project/video-spacys-ner-model

- What are the flaws of the model?

3. Display the co-occurring entities:
   - Use the provided code to display the graph of cooccurrences.
   - What do you observe?
   - How can you improve the graph?

You should obtain something similar to this:

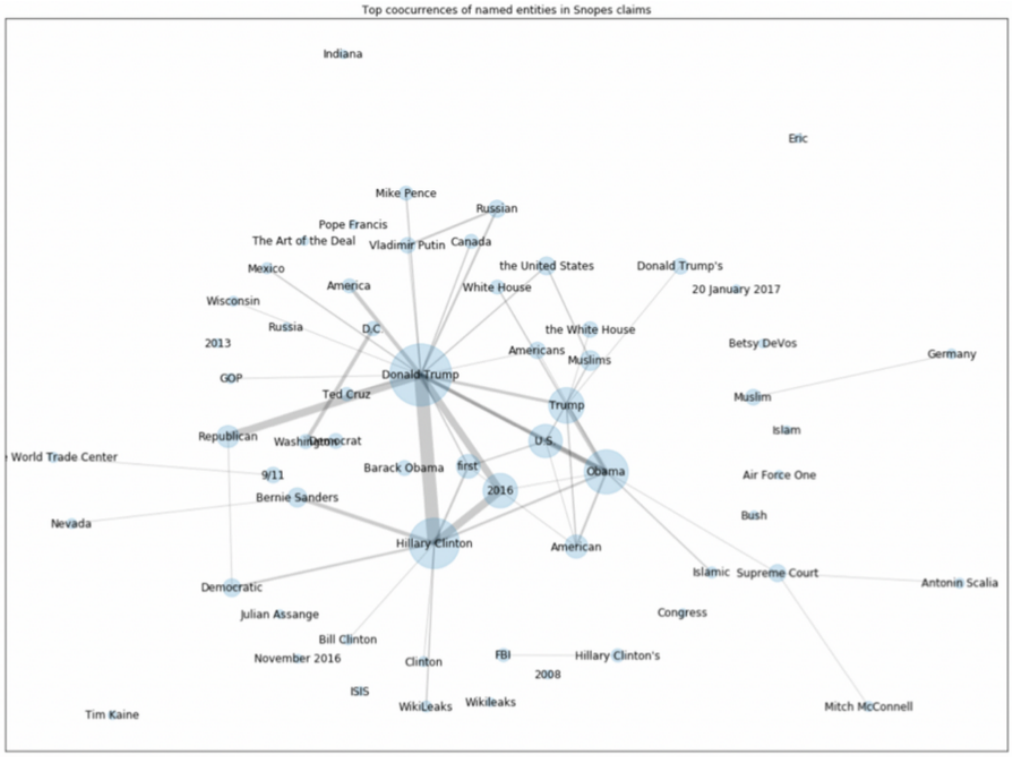


Figure 1: Named entity co-occurrences in the Snopes Tweets corpus

## Going Further

1. Generate a word cloud, see https://pypi.org/project/wordcloud/
2. Use the NLTK NER as presented here: https://pythonprogramming.net/named-entity-recognition-stanford-ner-tagger/

## Notes and comments

- Consider using the **Pandas** library to load the data. https://pandas.pydata.org/.
- To run a notebook, you need to install jupyter https://jupyter.org:

  - with conda: `conda install -c conda-forge notebook`
  - with pip: `pip install notebook`
  - then run the command `jupyter notebook` in your terminal.

- Alternatively, use Google colab: https://colab.research.google.com/drive/110iooLem3cuYrxwxhGIte3yaSjtS63Ck?usp=sharing
- Go to Files (icon on the left hand side) → Upload the file `snopes.csv`