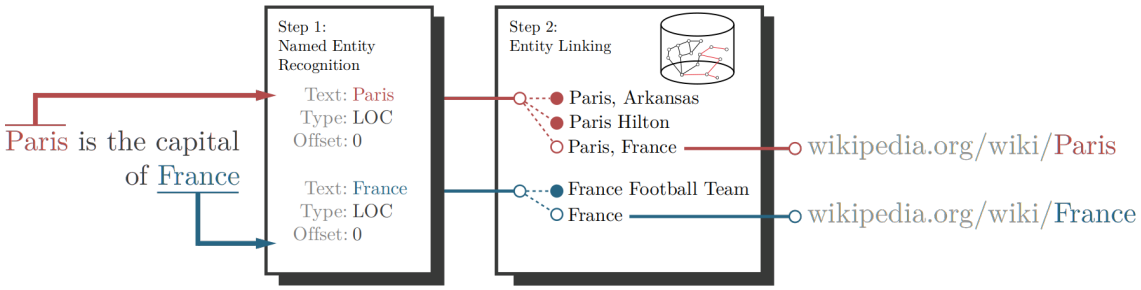
{EPITECH}

TRAVEL ORDER RESOLVER

BOOTSTRAP

TRAVEL ORDER RESOLVER

Let's explore an important component of NLP: the **N**amed **E**ntity **R**ecognition.



Specifically, NER is a subcategory of token classification. The goal is to assign each word to a category such as a person's name, a location, a brand name, and so on.

Annotation Format and Grammar

In order to train and practice entity detection, annotated textual data is required.
The annotation format may vary depending on the source. However, there are certain conventions such as the IOB (Inside-Outside-Beginning) annotation.

First NER models

The goal of this Bootstrap is to help you understand how Named Entity Recognition works.

In a classical model, you have to predict a label for each unit in your dataset (for example, for each article in a newspaper, you have to predict the topic: war, economy, environment, etc.).
The assumption usually made is that the entities on which you make predictions are independent. However, this assumption is broken in the case of text analysis because the prediction of one word depends on the surrounding words, and there is even interdependence among all the words in the sentence.

Different methods exist for making predictions, such as RNNs, LSTMs, Transformers (BERT and CamemBert). It's up to you to explore and understand these different methods.

We don't expect you to code a token classification model from scratch.

For instance, you can compare the NER functionality of spaCy with some more recent models like CamemBERT.

You must have a thorough understanding of the models you use and the underlying methods: Feed Forward Neural Networks, RNNs, LSTMs, Transformers, attention, self-attention, Bert, etc…

You may have already used all these methods without realizing it, especially with a tool called ChatGPT…

Evaluation of Results

Evaluation of the results is an essential aspect of your process.
Beyond a quality measure of classification, as in traditional supervised learning, you need to consider that certain entities can span multiple words.

This blog provides a quick description of different evaluation methods as well as the implementation of a library.

Consider displaying different metrics for different named entity categories to evaluate their strengths and weaknesses.

Warm-up

You're provided with a corpus zip-file, coming from a Kaggle challenge. It is a NER dataset annotated in a very standard way, where each word is associated with its corresponding label.

The file `ner_dataset.csv` contains different sentences; each line corresponds to a token with its NER tag. Note that all elements of the sentence are preserved, including commas, periods, etc. These are all very important for identifying entities (e.g., "I saw Mr. Poubelle yesterday": In this case, even without the capitalization of "Poubelle," you can guess that it refers to a named entity and not an everyday object).

Start by applying a classical NER model like spaCy to get a first idea of the results.

Then, calculate some more **advanced metrics**.

Training

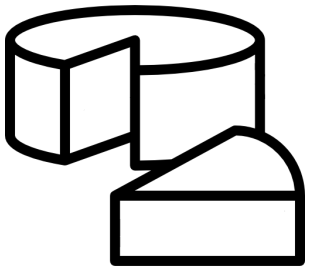
You now have a `bottins.csv` dataset with annotated entries from 20th-century directories.

The specificity of this dataset is that the annotation is directly done in the text. To test the NER, you will need to start by splitting the annotation to have the original text on one side and the annotated tokens on the other. This phase requires a bit more work than the previous one before you can get the initial results.

Once the cleaning is done, try running spaCy as well as a NER model based on Transformers (e.g., BERT), and evaluate the results.

Is one model better than the other?

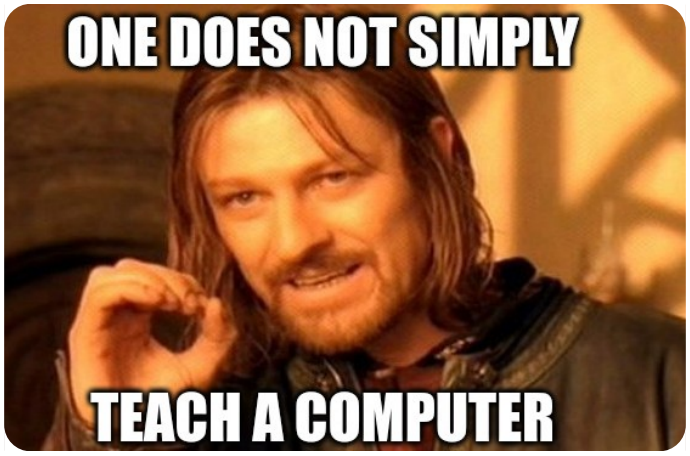
Pay attention, the text here is in French, so you need to choose a model that is suitable for the French language: either a specific French model or a multilingual one.



Project utils

As you have seen, evaluating the results is crucial for assessing a model on texts containing thousands of lines/sentences.

Evaluation requires annotation, and annotation requires… work!!



This exercise directly prepares you for the project: you will start an initial annotation phase.

Set a goal for each group to write 100 properly annotated sentences directly related to the project's objective.

For instance:

- ✓ Je veux aller depuis <*Dep*>Paris<*EndDep*> vers <*Arr*>Monaco<*EndArr*>.
- ✓ Je veux aller à <*Arr*>Monaco<*EndArr*> et je me trouve à <*Dep*>Paris<*EndDep*>.
- ✓ Avec Albert, on voudrait faire <*Arr*>Paris<*EndArr*>-<*Dep*>Monaco<*EndDep*>.
- ✓ Depuis <*Dep*>Paris<*EndDep*>, je veux aller à <*Arr*>Albert<*EndArr*> pour boire un Monaco.
- ✓ D'<*Dep*>Albert<*EndDep*>, je veux aller à <*Arr*>Monaco<*EndArr*> pour aller voir Paris.
- ✓ …

Annotation can be done quickly if you use appropriate tools such as this one.
If you are resourceful, you can even use ChatGPT (with the right prompts) to help you with annotation.

To realize the deepness and the complexity of NLP, you can dig the linguistics of Noam CHOMSKY or give an ear to a famous french sketch by Raymond DEVOS.

v2