# TD 02 — Extraction d'entités nommées

## 1. Objectifs du TD

L'extraction d'entités nommées, ou *Named Entity Recognition* (NER), est une technique du traitement automatique des langues (TAL) qui consiste à identifier automatiquement certaines expressions dans un texte et à leur attribuer une catégorie.

Elle permet notamment de repérer :

- les personnes ;
- les lieux géographiques ;
- les organisations ;
- les dates et les expressions temporelles ;
- d'autres entités selon les catégories prévues par le modèle utilisé.

En analyse littéraire, cette technique peut aider à repérer les personnages, à cartographier les lieux d'un récit et à étudier les références historiques présentes dans un roman.

À la fin de ce TD, l'étudiant sera capable de :

- utiliser spaCy pour extraire des entités nommées ;
- identifier les limites d'un modèle de reconnaissance automatique ;
- utiliser un modèle Transformer spécialisé dans la reconnaissance d'entités nommées en français ;
- comparer les résultats obtenus avec deux modèles différents ;
- interpréter les catégories et les scores de confiance ;
- évaluer les résultats en tenant compte du contexte littéraire.

## Exercice 1 — Démonstration : identifier les personnes, les lieux et les dates

### 1.1. Problématique

Considérons le texte suivant :

> Albert Camus est né à Mondovi en 1913. Il a vécu en Algérie avant de s'installer en France. En 1942, il publie L'Étranger, un roman dont l'histoire se déroule notamment à Alger.

Notre objectif est d'identifier automatiquement les personnes, les lieux et les dates mentionnés dans ce texte.

Nous allons utiliser deux modèles successivement pour observer leurs résultats et leurs limites.

## 1.2. Première approche : utiliser spaCy

Le modèle français de spaCy doit être installé dans l'environnement Python utilisé par Thonny.

**Programme Python**

```
import spacy


# Charger le modèle français
nlp = spacy.load("fr_core_news_sm")


# Texte à analyser
texte = """
Albert Camus est né à Mondovi en 1913.
Il a vécu en Algérie avant de s'installer en France.
En 1942, il publie L'Étranger, un roman
dont l'histoire se déroule notamment à Alger.
"""


# Analyser le texte
doc = nlp(texte)


# Afficher les entités reconnues
print("ENTITÉS NOMMÉES AVEC SPACY")
print("--------------------------")


for entite in doc.ents:
    print(entite.text, "->", entite.label_)
```

**Explication du programme**

- `import spacy` importe la bibliothèque spaCy.

- `spacy.load("fr_core_news_sm")` charge un modèle linguistique français.
- `nlp(texte)` analyse le texte et produit un document annoté.
- `doc.ents` contient les entités reconnues par le modèle.
- `entite.text` affiche l'expression extraite.
- `entite.label_` affiche la catégorie attribuée à cette expression.

**Résultats obtenus**

Lors de notre première exécution, le programme a produit les résultats suivants :

```
ENTITÉS NOMMÉES
----------------
Albert Camus -> MISC
Mondovi -> LOC
Algérie -> LOC
France -> LOC
L'Étranger -> MISC
Alger -> LOC
```

Les résultats montrent que le modèle a reconnu plusieurs entités, mais qu'il n'a pas identifié les deux années, 1913 et 1942.

Il a également classé Albert Camus dans la catégorie `MISC`, alors qu'il s'agit d'une personne.

**Interprétation**

Ces résultats illustrent plusieurs limites possibles de la reconnaissance automatique :

1. Une entité peut être détectée, mais recevoir une catégorie incorrecte.
2. Certaines entités présentes dans le texte peuvent ne pas être reconnues.
3. Un titre d'œuvre littéraire peut être classé dans une catégorie générale plutôt que dans une catégorie spécifiquement consacrée aux œuvres.
4. Les résultats dépendent du modèle utilisé et des données sur lesquelles il a été entraîné.

**À retenir :** la reconnaissance automatique ne garantit ni l'exhaustivité ni l'exactitude des résultats.

## 1.3. Deuxième approche : utiliser un modèle Transformer spécialisé

Pour améliorer la reconnaissance, nous allons utiliser un modèle de la bibliothèque Hugging Face Transformers :

`Jean-Baptiste/camembert-ner-with-dates`

Ce modèle repose sur CamemBERT, une architecture de type Transformer adaptée au français. Il a été entraîné pour reconnaître des entités nommées, notamment des personnes, des lieux, des organisations et des dates.

Contrairement à notre première approche, nous allons utiliser un modèle spécialisé dans cette tâche.

**Installation des bibliothèques**

Dans le logiciel Thonny, rechercher et installer les deux plugins (add-ons) : transformers et torch

**Programme Python**

```
from transformers import pipeline

# Charger le modèle spécialisé en reconnaissance d'entités
modele = "Jean-Baptiste/camembert-ner-with-dates"

reconnaitre_entites = pipeline(
    "token-classification",
    model=modele,
    aggregation_strategy="simple"
)

# Texte à analyser
texte = """
Albert Camus est né à Mondovi en 1913.
Il a vécu en Algérie avant de s'installer en France.
En 1942, il publie L'Étranger, un roman
dont l'histoire se déroule notamment à Alger.
"""

# Extraire automatiquement les entités
entites = reconnaitre_entites(texte)

# Afficher les résultats
print("ENTITÉS NOMMÉES AVEC CAMEMBERT")
```

```
print("----------------------------")


for entite in entites:
    print(
        entite["word"],
        "->",
        entite["entity_group"],
        f"(confiance : {entite['score']:.2f})"
    )
```

**Explication du programme**

- `pipeline()` crée une interface permettant d'utiliser le modèle.
- `"token-classification"` indique que le modèle attribue des catégories aux tokens du texte.
- `model=modele` précise le modèle préentraîné à utiliser.
- `aggregation_strategy="simple"` regroupe les prédictions sur les tokens pour obtenir des entités complètes.
- `reconnaitre_entites(texte)` analyse le texte et renvoie les entités détectées.
- `entite["word"]` contient l'expression reconnue.
- `entite["entity_group"]` indique la catégorie attribuée.
- `entite["score"]` fournit le score de confiance associé à la prédiction.

**Remarque :** lors du premier lancement, le modèle doit être téléchargé. Son utilisation nécessite donc une connexion Internet pour ce téléchargement initial. Une fois les fichiers mis en cache, le modèle peut généralement être réutilisé sans les télécharger de nouveau.

## 1.4. Comprendre les résultats et les scores de confiance

Le programme peut produire des résultats de la forme suivante :

```
Albert Camus -> PER (confiance : 1.00)
Mondovi -> LOC (confiance : 0.99)
en 1913 -> DATE (confiance : 1.00)
Algérie -> LOC (confiance : 0.96)
France -> LOC (confiance : 0.99)
```

```
En 1942 -> DATE (confiance : 1.00)
L'Étranger -> MISC (confiance : 0.98)
Alger -> LOC (confiance : 0.98)
```

Ces valeurs illustrent le type de résultat obtenu ; les scores exacts peuvent varier selon le modèle et sa version.

Le score de confiance est un indicateur calculé par le modèle. Il renseigne sur la force de sa prédiction, mais ne garantit pas que la catégorie attribuée est correcte.

Par exemple :

```
L'Étranger -> MISC (confiance : 0.98)
```

Le modèle peut être très confiant dans cette classification sans fournir la catégorie la plus précise pour une analyse littéraire.

Dans le jeu de catégories de ce modèle, `MISC` peut inclure des entités diverses, notamment des titres d'œuvres. Il ne faut donc pas supposer qu'un modèle de REN distingue nécessairement toutes les catégories utiles à l'analyse littéraire.

## 1.5. Comparaison des deux approches

| Critère | spaCy `fr_core_news_sm` | CamemBERT spécialisé |
|---|---|---|
| Type de modèle | Modèle linguistique statistique généraliste | Modèle neuronal de type Transformer spécialisé en REN |
| Reconnaissance des personnes | Possible, avec des erreurs éventuelles | Possible, avec des erreurs éventuelles |
| Reconnaissance des dates | Possible, mais certaines dates peuvent être oubliées | La reconnaissance des dates fait partie de ses capacités ciblées |
| Score de confiance par entité | Non affiché directement par l'interface utilisée | Fourni pour chaque prédiction |
| Installation et utilisation | Relativement simples | Nécessite plusieurs bibliothèques et le téléchargement du modèle |
| Exactitude garantie | Non | Non |

Les deux modèles peuvent produire des erreurs. Le recours à un modèle spécialisé permet de tester une autre approche, mais il ne dispense pas d'évaluer les résultats.

**Point important :** une entité nommée n'est pas nécessairement un personnage. Un lieu peut jouer un rôle essentiel dans l'univers narratif, tandis qu'un personnage peut être désigné par un prénom, un nom, un titre ou un pronom que le système ne saura pas nécessairement relier à une personne précise.

# Exercice 2 — À vous de jouer : cartographier les lieux d'un récit

## 2.1. Problématique

Vous devez étudier les références géographiques présentes dans un extrait littéraire.

Votre objectif est d'extraire automatiquement les entités nommées, puis d'examiner les personnes, les lieux et les dates identifiés par le modèle.

Vous utiliserez le modèle Transformer présenté dans l'exercice 1.

## 2.2. Travail demandé

1. Choisissez un extrait littéraire.
2. Remplacez le contenu de la variable `texte` par cet extrait.
3. Exécutez le programme de reconnaissance avec CamemBERT.
4. Affichez les entités reconnues et leurs catégories.
5. Comptez le nombre d'entités de chaque catégorie.
6. Affichez uniquement les entités classées comme personnes ou lieux.
7. Examinez au moins cinq entités et vérifiez si leur classification est correcte dans le contexte du texte.
8. Repérez les entités oubliées et les éventuelles erreurs de classification.

## 2.3. Aide : compter les catégories

Pour compter les catégories, ajoutez le code suivant après la reconnaissance des entités :

```
from collections import Counter
categories = Counter(
    entite["entity_group"] for entite in entites
)
print("\nNOMBRE D'ENTITÉS PAR CATÉGORIE")
print("------------------------------")
for categorie, nombre in categories.items():
    print(categorie, ":", nombre)
```

Pour afficher uniquement les personnes et les lieux :

```
print("\nPERSONNES ET LIEUX")
```

```
print("------------------")

for entite in entites:
    if entite["entity_group"] in ["PER", "LOC"]:
        print(entite["word"], "->", entite["entity_group"])
```

### 2.4. Aide possible avec un LLM

Vous pouvez demander à un LLM :

> Modifie ce programme Python utilisant Hugging Face Transformers pour afficher uniquement les entités classées comme personnes et lieux, puis compter les entités de chaque catégorie. Conserve la structure générale du programme et explique les modifications. N'ajoute pas de règles manuelles pour identifier les noms propres.

# Bilan du TD

L'extraction d'entités nommées permet d'identifier automatiquement des personnes, des lieux, des dates et d'autres informations présentes dans un texte.

L'utilisation de spaCy constitue une première approche pour découvrir cette technique. Cependant, un modèle généraliste peut oublier certaines entités ou leur attribuer une catégorie incorrecte.

L'utilisation d'un modèle Transformer spécialisé permet d'explorer une autre approche de la reconnaissance automatique. Elle offre également un score de confiance pour chaque prédiction, ce qui peut faciliter l'examen des résultats.

Toutefois, **un modèle plus spécialisé n'est pas nécessairement correct dans tous les contextes**. Un score élevé ne garantit pas l'exactitude d'une classification, et les catégories proposées peuvent ne pas correspondre précisément aux besoins de l'analyse littéraire.

Enfin, l'extraction automatique d'entités nommées ne suffit pas, à elle seule, à identifier tous les personnages d'un roman ni à déterminer l'importance narrative des lieux. Les résultats doivent être interprétés en tenant compte du contexte et, lorsque cela est nécessaire, vérifiés par une analyse humaine.