A series of thin, black, overlapping geometric lines, primarily triangles and quadrilaterals, are scattered across the top half of the slide, creating a complex, abstract pattern.

# RECONNAISSANCE D'ENTITÉS NOMMÉES : PRÉSENTATION-ATELIER

Alexia Schneider : Master TDL (unistra) & stagiaire Bidnum BNU

[email](#)

[GitHub](#) | [GitLab](#)

Lundi 8 juillet 2024 – Le Lab

# PLAN

1. Introduction : qu'est-ce que la reconnaissance d'Entités Nommées
2. Exemple d'une chaîne de traitement pour de la fouille de texte avec NER
3. Atelier : présentation et utilisation des programmes développés (regex, NER, entraînement de modèle, alignement avec idRef)



# INTRODUCTION

# LES ENTITÉS (NOMMÉES) (EN)

## Définition

- Terme employé en Traitement Automatique des Langues (TAL)
- Extraction/reconnaissance de certains mots :
  - Les noms propres (personnes, lieux, organisations)
  - Dates
  - Monnaies et symboles (ex: €, euro)
  - Chiffres et nombres

# OBJECTIFS DE LA RECONNAISSANCE D'ENTITÉS NOMMÉES

Des corpus « augmentés » pour l'exploration textuelle :

- Extraction de données précises (date, lieux) pour des questions de recherches transversales : exemple Analyse de réseaux (quelles personnes sont citées dans quels textes)

Interopérabilité :

- Les données dont le contenu est structuré peuvent être indexées et échangées de manière + efficace et probante.

Automatisation de la NER :

- Volume de données + important

# TÂCHES DE TAL POUR LA FOUILLE DE TEXTE

- Tâches de bas niveau :
  - Tokenisation (séquençage du texte en « mot »)
  - Lemmatisation (retrouver la forme canonique d'un mot/token)
- Tâches complexes :
  - Classification de tokens
    - Analyse morpho-syntaxique (« part of speech tagging ») : NOM, DET, PREP, ADV etc.
    - **Reconnaissance d'EN**
    - Analyse en dépendance (« dependency parsing »)
  - Génération de texte

# QUELLES SONT LES DIFFICULTÉS POUR LA RECONNAISSANCE D'ENTITÉS ?

François RITTER, *Catalogue des Incunables et Livres du XVI<sup>e</sup> siècle de la Bibliothèque Municipale de Strasbourg*, Strasbourg-Zurich, P.-H. Heitz, 1948, gr. in-8°, 923 pp.

Cet ouvrage fait pendant au Répertoire bibliographique des Incunables et des Livres imprimés en Alsace au XVI<sup>e</sup> siècle de la Bibliothèque nationale et universitaire, dont 3 volumes ont déjà paru chez l'éditeur Heitz, et dont le dernier, prêt depuis 1936, est impatiemment attendu. Avant de nous le donner, le D<sup>r</sup> Fr. Ritter a tenu à acquitter une dette envers la Bibliothèque municipale de Strasbourg, et il a rassemblé ici plus de 2.000 titres d'ouvrages couvrant la même période. Il lui restera pour achever son œuvre à nous donner le catalogue des livres des XVe et XVI<sup>e</sup> siècle du Grand Séminaire, et surtout l'Histoire de l'Imprimerie du XVe et XVI<sup>e</sup> siècle en Alsace, qui seule permettra un jugement d'ensemble, notamment sur la contribution apportée à la Réforme en Alsace par la diffusion du livre.

Le D<sup>r</sup> Ritter ne s'y est pas essayé dans l'introduction de l'ouvrage que nous présentons ici. Il se contente d'y décrire les vicissitudes de la Bibliothèque municipale de cette ville, qui, brûlée en 1870, fut entièrement refaite, par les soins surtout du bibliothécaire Rod. Reuss (1870-1895).

# DIFFÉRENCES ENTRE UN HUMAIN ET UNE MACHINE

## Un humain :

- **Comprend le texte :**
  - Lexique : compréhension des mots ressemblants (Strazburg = Strasbourg)
  - Syntaxe & morpho-syntaxe : importance de la ponctuation, de la nature des mots & de l'ordre des mots (Pierre, charpentier de Paris. =/= Pierre Charpentier, de Paris.)
  - Contexte : différence entre un nom d'auteur et un nom d'éditeur dans une notice bibliographique, différence entre une erreur d'OCRisation et un nouveau mot, nuance de sens.



# DIFFÉRENCES ENTRE UN HUMAIN ET UNE MACHINE

## Une machine :

- Ne peut pas interroger le texte comme nous le faisons, **elle ne « comprend » pas un texte.**
- **Mais, on peut recomposer la logique de la langue et d'une tâche précise :**  
**Apprentissage** avec des systèmes à base de règle ou des systèmes inductifs à partir de grands volumes de données (modélisation de la langue )

# EXEMPLE DU RITTER : UN RÉPERTOIRE BIBLIOGRAPHIQUE

REPERTOIRE BIBLIOGRAPHIQUE DES LIVRES 1  
IMPRIMES EN ALSACE  
AUX XV<sup>e</sup> ET XVI<sup>e</sup> SIÈCLES

---

PREMIERE PARTIE

## CATALOGUE DES INCUNABLES DE LA BIBLIOTHEQUE NATIONALE ET UNIVERSITAIRE DE STRASBOURG

PAR

FR. RITTER

---

# EXEMPLE DE PAGES DU RITTER

Notices

Renvoi

Adresse bibliographique

Nom d'auteur

Imprimeur

Lieu d'impression

Référentiels et  
provenance

Commentaire/  
retranscription  
de la première  
de couverture

CARACCIOLUS Robertus: Sermones per Adventum. [Strasbourg, M. Schott, vers 1485], in-fol.

G W 6050, Hain 44/1, Pellechet: Colmar 440 Proctor 403, Voulliéme: Berlin 2243. 2243, 2.

K 1074. Prov.: Biblioth. Ch. Schmidt, Strasbourg, 10/IV. 1895.

124

CARACCIOLUS Robertus: Sermones quadragesimales de poenitentia. Strasbourg, [Martin Schott], 3 sept. 1485, in-fol.

G W 6078, Hain 4436, Madsen 1035, Pellechet 3253, Pellechet: Colmar 436, Proctor 395, Voulliéme: Berlin 2231, 5; Walter: Sélestat 130.

K 1063. Prov.: C. Geyling héritiers, Vienne, 18/I. 1886, 8 M. Notes mss.

125

CARLETTI Angelo. Voir: *Angelus de Clavasio*.

CASSIODORUS M. Aurelius: Historia ecclesiastica tripartita. [Strasbourg, J. Prüss, vers 1500], in-fol.

G W 6167, Hain 45/2, Madsen 1062, Pellechet 3348, Pellechet: Colmar 453, Proctor 583, Voulliéme: Berlin 2369, 5.

K 1093. Prov.: K. Trübner, Strasbourg, 10/VI. 1887, 6 M.

126

STURM Johannes

Strasbourg, J. Rihel, 1566

Ioannis Sturmij || Partitionum Dialecticarum || Libri IIII.  
|| Emendati & aucti.

Marque typ. de Josias Rihel (H. & B. Planche XXXI, N° 8 ;  
Silvestre, N° 821).

Cum Gratia et Privi || legio Caesareo ad annos octo.

Anno. M. D. LXVI. (Verso blanc.)

A la fin : Argentorati || per Iosiam Rihelium.

In-8°, car. rom., 8 ff. non ch., 292 ff. ch., titre courant, réclames, notes  
marg., init. ornées D, B, M.

Réimpression de l'édition 1560.

R. 102.068. Prov.: Cromer, cure de Saverne, 29/VII, 1879.

Au milieu du titre : « Parochia Tabernensis, en bas : P. Wellerius »  
Biblioth. BM Strasbourg, n° 2000

2214

STURM Johannes

Strasbourg, S. Emmel, 1567

Epistolae || Duae duorum a- || micorum, Bartholomaei  
La- || tomi, & Ioannis Sturmij, de dissidio peri || culoque  
Germaniae, & per quos stet, quo minus || concordiae ratio  
inter partes || ineatur.

Item Alia Qvaedam Stvr- || mij, de emendatione Ecclesiae,  
& Religionis || controuersijs.

Marque typ. de Sam. Emmel. (H. & B. Planche XXXV,  
N° 1 ; Silvestre, N° 822).

Argentorati M. D. LXVII. || Samuel Emmel exprimebat.  
(Verso blanc.)

In-8°, car. rom., 80 ff. ch., titre courant, réclames, init. ornées.  
fol. 2<sup>a</sup>-10<sup>b</sup> : Bartholomeus || Latomus Ioanni || Sturmio... — Bononiae,  
II. die Feb. Anno || M. D. XL.

Volume 1

Volume 5

# LES NOTICES BIBLIOGRAPHIQUES ET L'AUTOMATISATION DE L'EXTRACTION

## **Le Ritter a des avantages :**

- Des données structurées avec un ordre attendu
- Des données limitées (zone géographique et période restreinte)

# LES NOTICES BIBLIOGRAPHIQUES ET L'AUTOMATISATION DE L'EXTRACTION

## **Le Ritter a des avantages :**

- Des données structurées avec un ordre attendu
- Des données limitées (zone géographique et période restreinte)

## **Mais aussi des spécificités contraignantes:**

- Noms de personne et noms de référentiel identiques (Schmidt, Hain)
- Notices en français ou en allemand + latin
- Structure variable d'un tome à un autre
- Variations linguistiques (Strasbourg, Strassburg, Strazburg)



EXEMPLE DE  
CHAÎNE DE  
TRAITEMENT  
POUR LA  
FOUILLE DE  
TEXTE



# GRANDES ÉTAPES POSSIBLES D'UNE CHAÎNE DE TRAITEMENT

**1. Numérisation**

**2. OCR (Optical Character Recognition)**

**3. Annotation**

**Annotation manuelle**

**Annotation automatique**

**4. Alignement des données annotées avec un référentiel**

# 1. NUMÉRISATION

- Passage d'une page physique à une image

## 2. OCR

(HTR : handwritten text recognition)

- Transcription du texte d'une image vers un document « .txt » brut.
- Évaluation de la qualité de la retranscription :
  - Character Error Rate/Word Error Rate



## 3. ANNOTATION

- Ajout de notes à un texte brut
- Valorisation du contenu sémantique d'un texte.
- Formats standardisés comme XML-TEI pour interopérabilité.

### 3.A ANNOTATION MANUELLE

- Réflexion préalable sur les besoins : quelles étiquettes, quelles données, quels buts ?
- Etape nécessaire pour l'évaluation : établir une « vérité de terrain »

**Exemple de travail d'annotation manuelle :**

<https://tecoholic.github.io/ner-annotator/>

NER Annotator

File

Annotations

Tags

Text Separator

An Empty Line

Annotation Precision

Word Level

Tagging Progress (0/1)

Keyboard Shortcuts ?

✓

PER

2

LOC

3

DATE

NEW TAG

EDIT TAGS

PARATUS ( Idem ) . — [

Strasbourg

LOC

✕

,

Martin Flach

PER

✕

s. d.

DATE

✕

) , in-4°. Hain 12402 , Polain

3515 , Madsen 3040 , Voulliéme : Berlin 2516. 554 PARATUS ( Idem ) . — (

Strasbourg

LOC

✕

,

Martin Flach

s. d.

DATE

✕

]

in-4°. Hain 12403 , Polain 3516 , Proctor 722. 555 556 > PARATUS ( Idem ) . —

Strasbourg

LOC

✕

,

Martin Flach

PER

✕

s. d.

DATE

✕

) , in-fol. Hain 12404 ; Proctor 730. 556 PARATUS ( Idem ) . —

Strasbourg

LOC

✕

,

Martin Flach

PER

✕

1491

DATE

✕

,

in-fol. Copinger II , 4600. 557 PARATUS ( Idem ) .

RESET

BACK

SKIP

SAVE

<https://tecoholic.github.io/ner-annotator/>

18

```
<TEI xmlns="http://www.tei-c.org/ns/1.0">
```

```
<text>
```

```
<body>
```

```
<div>
```

```
<p><div type="notice" id="1162">
```

```
<lb/>manuscript 107.
```

```
<lb/>1163
```

```
</div><div type="notice" id="1164"><bibl>
```

```
<lb/><persName>BUCER, Martin</persName> <title>(Job) : Commentaria in librum <persName>Job</persName>.</title> –
```

```
<lb/><placeName>Strasbourg</placeName> <date>1528</date>, in-fol.
```

```
</bibl>
```

```
<lb/>Mentz, n° 22; Stupperich, n° 23.
```

```
<lb/>1164
```

## Annotation en XML d'une notice bibliographique



## 3.B ANNOTATION AUTOMATIQUE

### **Systèmes à base de règles**

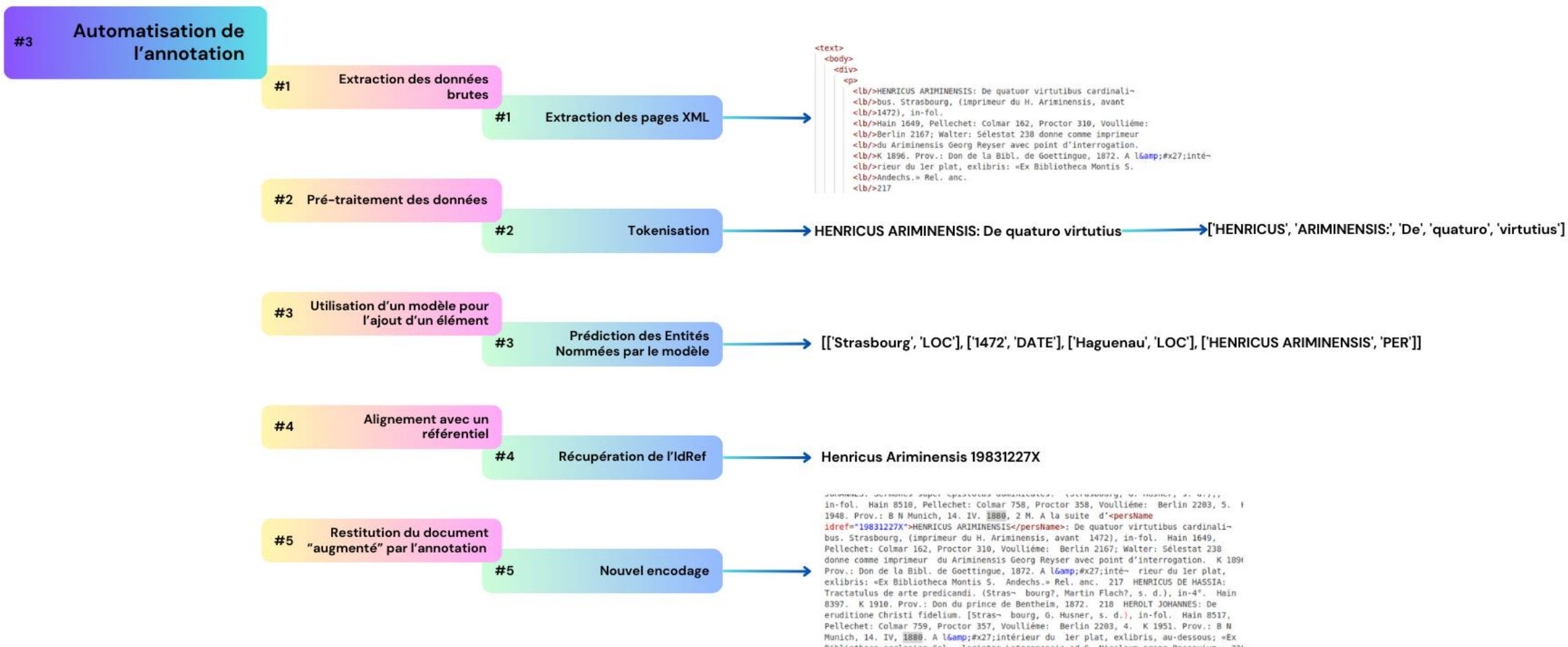
**Exemple d'un système à base de règle pour l'extraction de données brutes : les expressions régulières.**

### **Systèmes prédictifs à partir d'une modélisation de la langue :**

**Les « modèles de langue » et la représentation des données**

**Applications : modèles spaCy et Transformers disponibles via Hugging Face.**

**Les modèles génératifs**



# SYSTÈME À BASE DE RÈGLES

Exploitation des ressources connues et vérifiées pour établir des règles :

## Les tables alphabétiques de noms d'auteur

### TABLE ALPHABÉTIQUE DES NOMS PROPRES

(sauf les noms d'imprimeurs ou d'auteurs qui figurent les premiers dans la liste des imprimeurs, les autres dans le corps de l'ouvrage comme vedettes).

Alberti Joh. 167.	Aurifaber Aegidius 440, 441.
Albertus Episcopus 201.	Bader 109.
Alburg 380.	Badius Ascensius 114.
Alexander Magnus 281-283.	Bâle 4, 25.
Altendorffer Henr. 204.	Bamberg 169, 241, 258.
Altomünster (Bavière) 210.	Baer (antiquaire) 2, 7, 91, 123,
Amberg (Bavière) 35.	136, 143, 150, 151, 254, 260,
Aemilius Montfordensis 354.	277, 299, 300, 335, 356, 415,
Andechs 61, 158, 217, 297, 382.	417, 420, 426, 436, 442, 491,
Anvers 354.	Baum Joh. Wilh. 172, 263, 421,
Arensberg 434.	460.
Aristoteles 445.	Beck 145, 295.
Arnoldus de Villa Nova 408.	Belial 250, 251.
Artur Henri 413.	Belling J. C. 16.
Auerbach Joh. 464.	Bensheimer (Libr. Strasbourg)
Augsbourg 6, 19, 49, 51, 54, 67,	344, 361.
81, 85, 98, 118, 121, 139, 142,	Bentheim Prince de 28, 89, 152,
180, 192, 193, 197, 198, 206,	214, 218, 223, 232, 301, 310,
211, 251, 252, 256, 266, 270,	319, 324, 334, 359, 366, 375,
271, 278, 292, 297, 298, 304,	406, 418.
305, 308, 313, 318, 320, 322,	Berlin 1, 2, 44a, 94, 127, 132,
327, 328, 346, 352, 353, 358,	

# SYSTÈME À BASE DE RÈGLES

Les **Expressions Régulières** (regex) : ensemble de caractères spéciaux pour la recherche de motifs.

Exemples :

- Trouver toutes les dates du 20<sup>e</sup> siècle càd motif de 4 chiffres en 19XX.

`19[1-9][1-9]`

- Trouver les variantes de « Strasbourg » : Strassburg, Strazburg

`Stra[sz]s?bo?urg`

Site pour tester ses regex : <https://regex101.com/>

# LES REGEX EN PYTHON

```
1  import re
2
3  exemple = "Caracciolus Robertus: Sermones quadragesimales de \
4  | poenitentia. Strasbourg, [Martin Schott], 3 sept. 1485,"
5
6
7  # division de l'exemple en token ou mot :
8  for token in exemple.split():
9      # extraction des dates : chiffres romains | XIV-XVIe siècle | "Anno" :
10     if re.fullmatch(r'(M?\.[LXI]+\.?|1[456]\d\d\.[?]?|Anno\W?)', token) :
11         label = 'DATE'
12         # affichage
13         print(f'\nEntitée Nommée détectée : "{token}" -> {label}.\n')
14
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

```
• proj-py3.11test@ubu:~/bnu_ner$ /home/test/.cache/pypoetry/virtualenvs/proj-EifN0aK
Entitée Nommée détectée : "1485," -> DATE.
```

exemple complet sur le Ritter : [https://github.com/lab-bnu/ritter\\_ner/blob/main/regex\\_ner.py](https://github.com/lab-bnu/ritter_ner/blob/main/regex_ner.py)

NB : bien vérifier les paramètres : [https://github.com/lab-bnu/ritter\\_ner/blob/main/params.py](https://github.com/lab-bnu/ritter_ner/blob/main/params.py)



# EVALUATION : MESURES D'ÉVALUATION

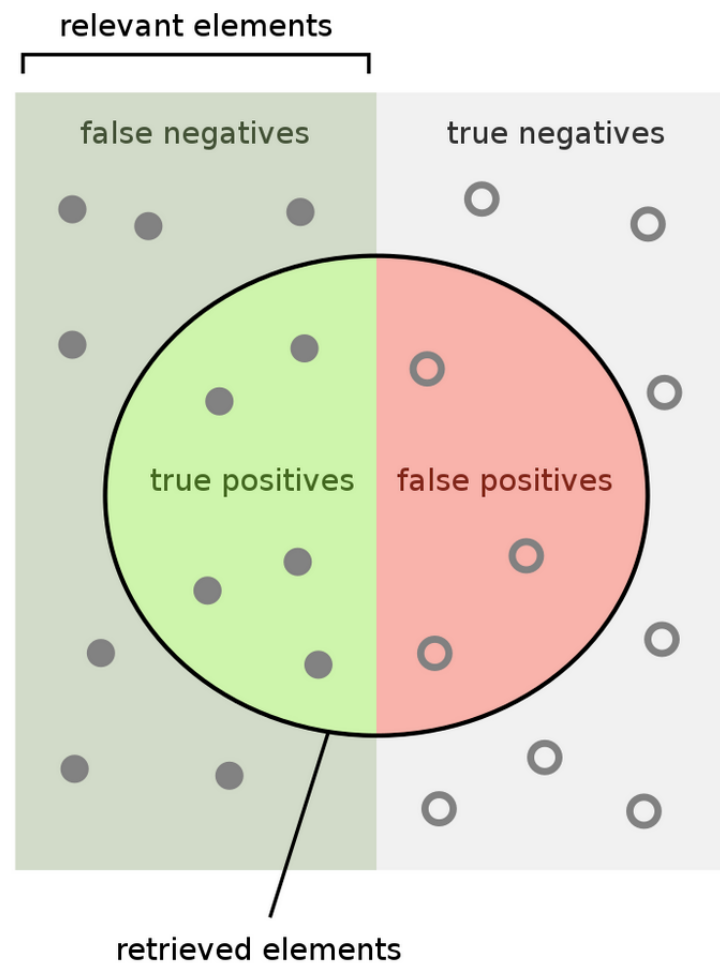
Précision (*precision*) : vrais PER prédits / ensemble des fois où le modèle a prédit PER

Rappel (*recall*) : vrais PER prédits / ensemble des fois où le modèle devait prédire PER

Mesure F1 (*F1-score*) : moyenne de la précision et du rappel

Valeurs de 0 à 1 avec 0 score le + bas et 1 le + élevé.

NB : il est toujours nécessaire d'avoir une vérité de terrain (annotation manuelle) pour évaluer les performances d'un modèle



[https://en.wikipedia.org/wiki/Precision\\_and\\_recall#/media/File:Precisionrecall.svg](https://en.wikipedia.org/wiki/Precision_and_recall#/media/File:Precisionrecall.svg)

How many retrieved items are relevant?

$$\text{Precision} = \frac{\text{true positives}}{\text{true positives} + \text{false positives}}$$

How many relevant items are retrieved?

$$\text{Recall} = \frac{\text{true positives}}{\text{true positives} + \text{false negatives}}$$

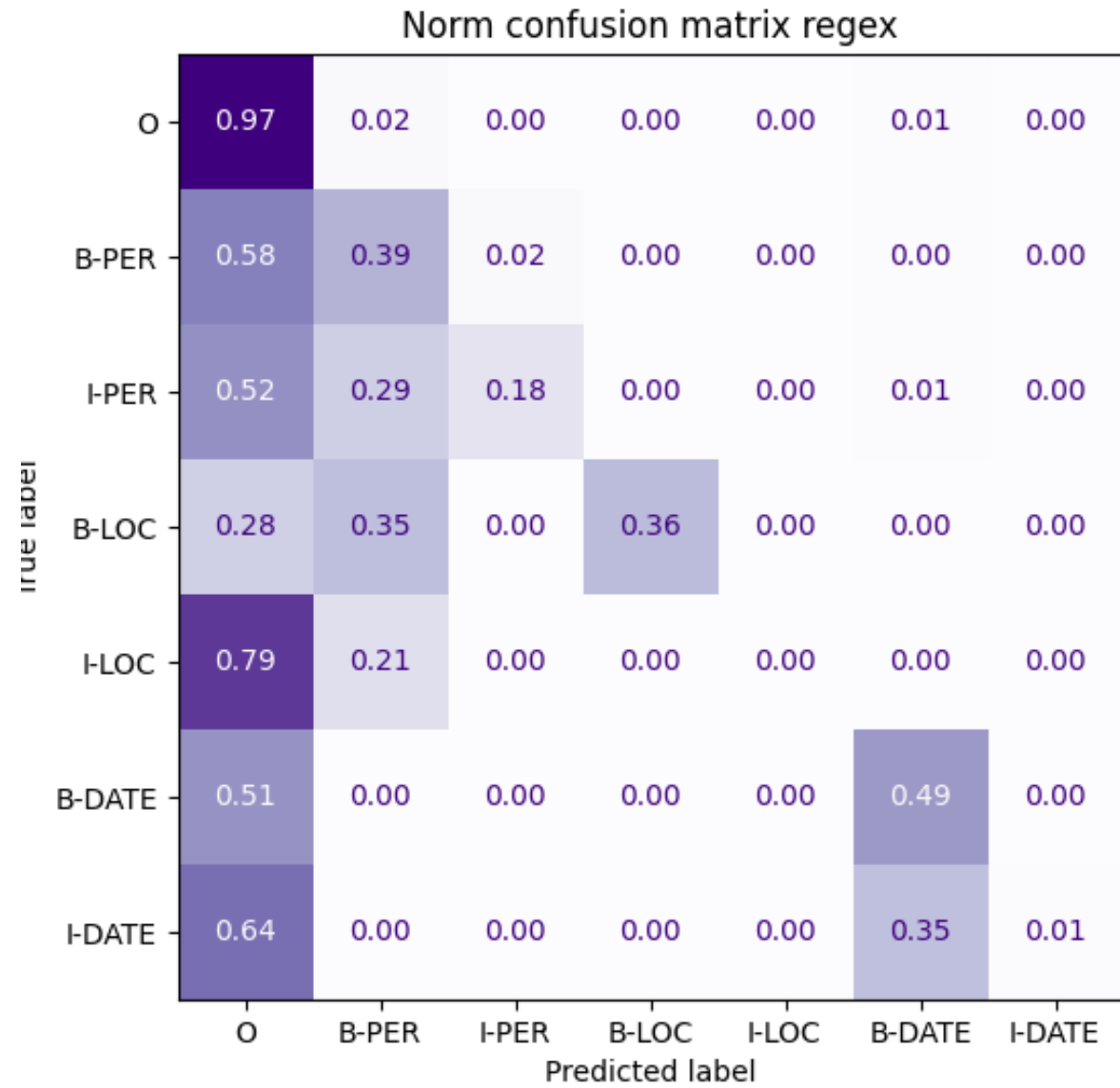
# EVALUATION DU PROGRAMME REGEX

Précision (*precision*) : vrais PER / ensemble des fois où le modèle a prédit PER

Rappel (*recall*) : vrais PER / ensemble des fois où le modèle devait prédire PER

Mesure F1 (*F1-score*) : moyenne de la précision et du rappel

		precision	recall	f1-score	support
	0	0.92	0.97	0.94	19330
	B-PER	0.23	0.39	0.29	762
	I-PER	0.75	0.18	0.29	1072
	B-LOC	0.99	0.36	0.53	515
	I-LOC	0.00	0.00	0.00	78
	B-DATE	0.42	0.49	0.45	366
	I-DATE	0.50	0.01	0.01	278
	accuracy			0.87	22401
	macro avg	0.55	0.34	0.36	22401
	weighted avg	0.88	0.87	0.86	22401

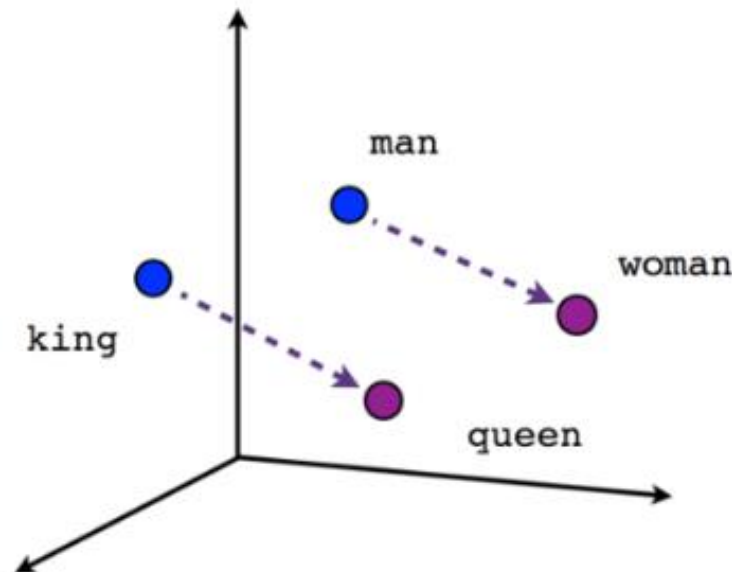


# SYSTÈMES PRÉDICTIFS À PARTIR DE LA MODÉLISATION D'UNE LANGUE

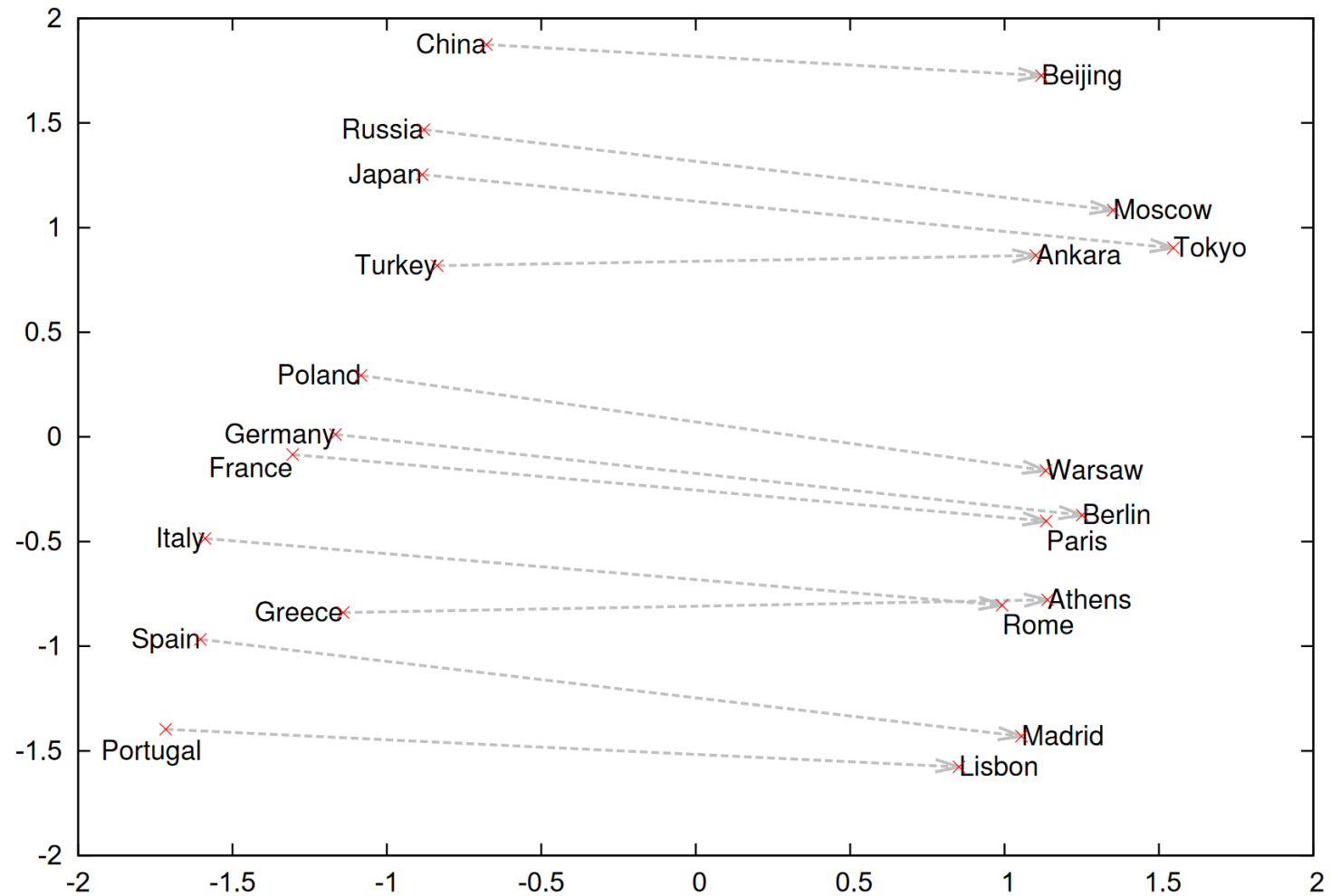
**Modèles de langue** : représentation vectorielle de la langue à partir de l'encodage d'un volume (ou très très grand volume) de données textuelles.

Transformers (BERT) : modélisation selon un encodage qui prend en compte le contexte (ordre des mots, cooccurrences des mots dans une phrase/document).

Vecteurs :



Country and Capital Vectors Projected by PCA



Mikolov et al.,  
2013. Distributed  
representations of  
words and phrases  
and their  
compositionality,  
[NIPS'13:  
Proceedings of the  
26th International  
Conference on  
Neural Information  
Processing Systems  
- Volume 2](#), p. 3111  
– 3113.

# MODÈLES DE SPACY

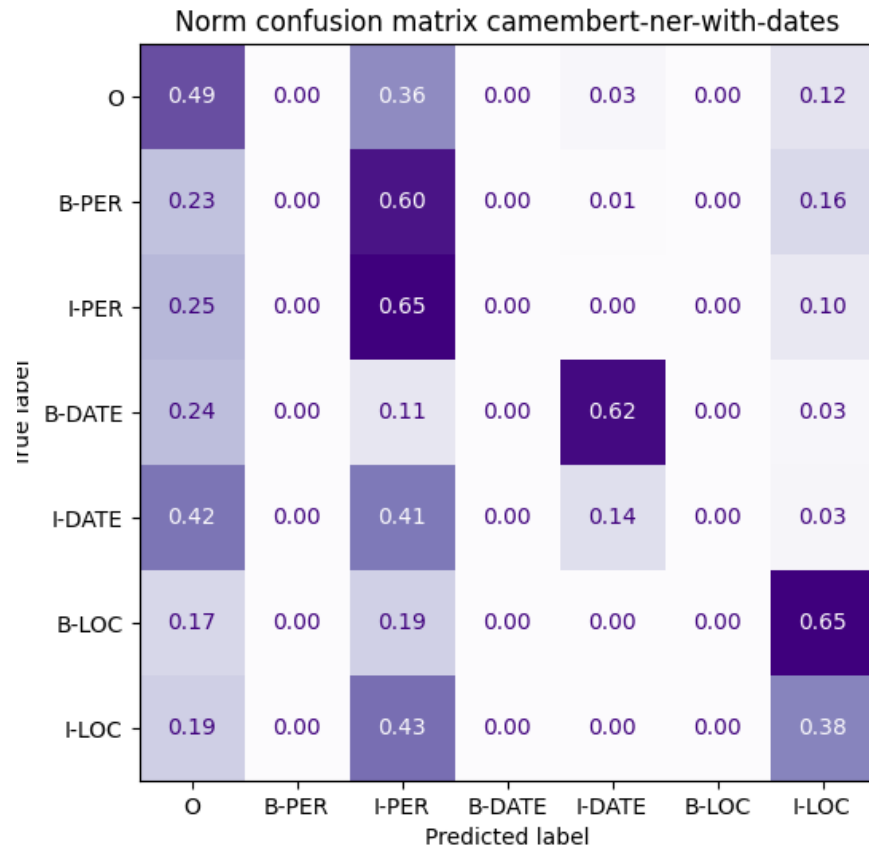
- Bibliothèque open-source pour le TAL : <https://spacy.io/>
- Visualisation de la tâche de reconnaissance d'EN avec les modèles spaCy : <https://demos.explosion.ai/displacy-ent>
- On peut aussi entraîner des modèles avec spaCy. (voir partie sur les entraînements)
- Le programme [https://github.com/lab-bnu/ritter\\_ner/blob/main/programmes\\_standalone/spacy\\_ner.py](https://github.com/lab-bnu/ritter_ner/blob/main/programmes_standalone/spacy_ner.py) permet d'utiliser un modèle spaCy de son choix pour le NER. Le modèle par défaut est fr\_core\_news\_sm.

# MODÈLES DISPONIBLES VIA LA PLATEFORME HUGGING FACE

- Plateforme <https://huggingface.co> sur laquelle on peut mettre à disposition un modèle : les grandes entreprises de l'IA (Google, Meta, etc.) comme les universités ou les particuliers.
- Utilisation simplifiée des modèles grâce à la bibliothèque Python transformers
- On peut aussi entraîner ces modèles. (voir partie sur les entraînements)
- Le programme [https://github.com/lab-bnu/ritter\\_ner/blob/main/programmes\\_standalone/testing\\_hf\\_ner.py](https://github.com/lab-bnu/ritter_ner/blob/main/programmes_standalone/testing_hf_ner.py) permet d'utiliser un modèle disponible sur HF de son choix pour le NER. Le modèle par défaut est : <https://huggingface.co/Jean-Baptiste/camembert-ner-with-dates>



# EVALUATION DE CAMEMBERT-NER-WITH-DATES



				precision	recall	f1-score	support
			0	0.90	0.49	0.63	7997
		B-PER		0.00	0.00	0.00	513
		I-PER		0.10	0.65	0.17	555
		B-DATE		0.00	0.00	0.00	184
		I-DATE		0.04	0.14	0.07	115
		B-LOC		0.00	0.00	0.00	358
		I-LOC		0.01	0.38	0.02	42
	accuracy					0.44	9764
	macro avg			0.15	0.24	0.13	9764
	weighted avg			0.75	0.44	0.53	9764

# ENTRAINEMENT DE MODÈLE : AFFINAGE SUR NOS DONNÉES

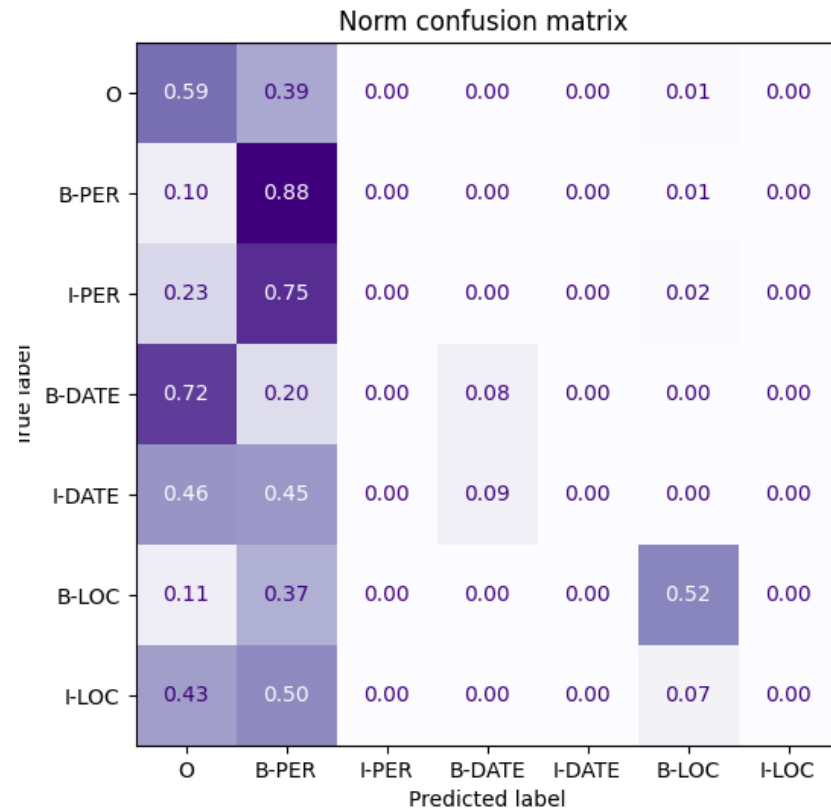
---

- Gros potentiel des modèles de langue mais limite majeure: difficulté à généraliser sur des données qui ne « ressemblent » pas aux données de pré-entraînement.
- On peut poursuivre l'entraînement d'un modèle : pour une tâche demandée, et pour des données précises.

**Tutoriel pour affinage de modèle avec la bibliothèque spaCy :**

[https://github.com/lab-bnu/ritter\\_ner/blob/main/documentation/5.Entraînement\\_modèle\\_avec\\_spacy.md](https://github.com/lab-bnu/ritter_ner/blob/main/documentation/5.Entraînement_modèle_avec_spacy.md)

# EVALUATION DE CAMEMBERT AFFINÉ SUR NOS DONNÉES



	precision	recall	f1-score	support
0	0.92	0.59	0.72	7997
B-PER	0.11	0.88	0.19	513
I-PER	0.04	0.00	0.00	555
B-DATE	0.33	0.08	0.13	184
I-DATE	0.00	0.00	0.00	115
B-LOC	0.63	0.52	0.57	358
I-LOC	0.00	0.00	0.00	42
accuracy			0.55	9764
macro avg	0.29	0.30	0.23	9764
weighted avg	0.79	0.55	0.62	9764

Importance de l'entraînement : ces résultats correspondent à un entraînement de **10 époques**

# EVALUATION DE CAMEMBERT AFFINÉ SUR NOS DONNÉES

Entité	Précision	Rappel	F1-score
B-PER	0.8835	0.7521	0.8125
I-PER	0.8731	0.7091	0.7826
B-LOC	0.87	0.9355	0.9015
I-LOC	0.8333	0.4545	0.5882
B-DATE	0.7917	0.95	0.8636
I-DATE	0.8421052632	0.8727272727	0.8571
Moyenne pondérée	0.8580	0.802	0.8291

Importance de l'entraînement : ces résultats correspondent à un entraînement de **40 époques**

# TÉLÉCHARGEMENT ET UTILISATION DU MODÈLE AFFINÉ POUR LA NER SUR LES DONNÉES DU RITTER :

Le modèle est disponible : [https://huggingface.co/alexiaschn/fr\\_camembert\\_ritter](https://huggingface.co/alexiaschn/fr_camembert_ritter)

Pour le télécharger et l'utiliser en ligne de commande (en plus des modules spaCy)

```
pip install
```

```
https://huggingface.co/alexiaschn/fr\_camembert\_ritter/resolve/main/fr\_camembert\_ritter-any-py3-none-any.whl
```

Puis on peut l'utiliser grâce à la pipeline de spaCy :

```
import spacy
```

```
nlp = spacy.load("fr_camembert_ritter")
```

```
exemples = ["Exemple de phrase avec des EN comme Strasbourg et Henri Eppendorf.",  
"Deuxième phrase d'exemple sans EN."]
```

```
print(*[(ent.text, ent.label_) for doc in nlp.pipe(exemples, disable=["tok2vec",  
"tagger", "parser", "attribute_ruler", "lemmatizer"]) for ent in doc.ents], sep="\n")
```

# LES MODÈLES GÉNÉRATIFS

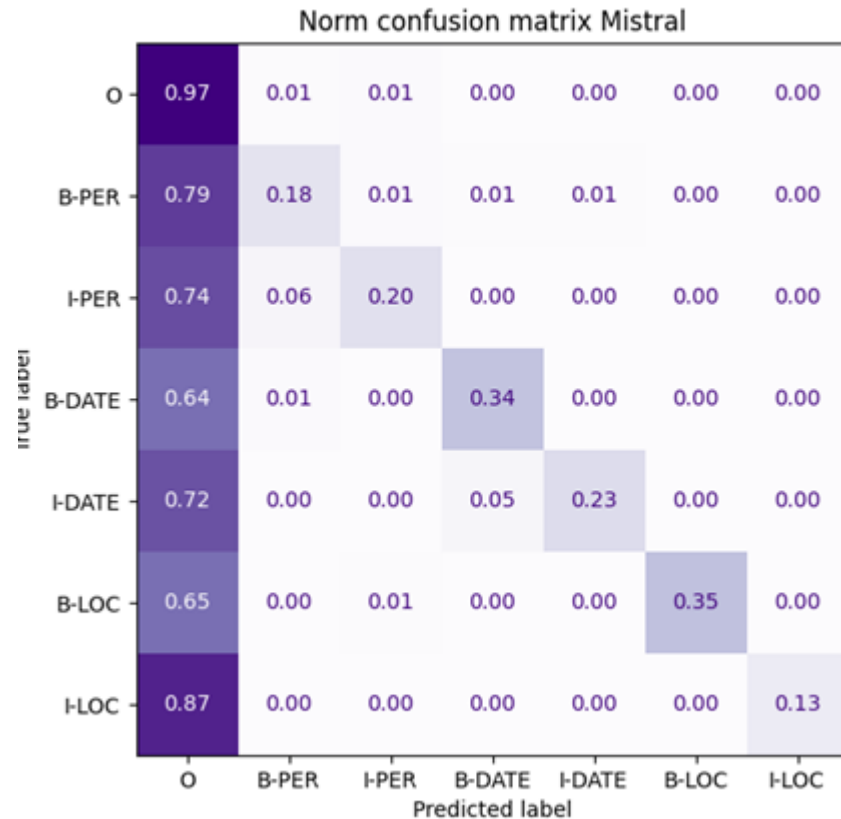
- Des Large Language Models (LLMs) (entraînés sur de très grand volume de données) : capacité de généralisation accrue. Reçoit des instructions en langue naturelle (prompt = instruction).
- Concurrence croissante car capacité à gérer des tâches diverses sans affinage.
- Grande fenêtre contextuelle (plusieurs milliers de tokens).
- Disponibilité de ces modèles via des interfaces web simples d'utilisation :
  - <https://chat.mistral.ai/chat> - MistralAI: concurrent français d'OpenAI.
  - Réponse en langue naturelle : post-traitement nécessaire.

# LES MODÈLES GÉNÉRATIFS : UTILISATION DE CES MODÈLES

---

- **Division du texte en entrée en « batches » :**
  - Programme : [https://github.com/lab-bnu/ritter\\_ner/blob/main/programmes\\_standalone/create\\_batches.py](https://github.com/lab-bnu/ritter_ner/blob/main/programmes_standalone/create_batches.py)
- **Utilisation possible via l'API pour automatiser le traitement :**
  - Nécessite une clé API payante pour la plupart des modèles
  - Meilleur contrôle des paramètres du modèle :
    - Température
    - Nombre de tokens générés en sortie
    - Type de tâche demandées : tchat ou complétion de texte.

# EVALUATION DE MISTRAL LARGE



	precision	recall	f1-score	support
O	0.86	0.97	0.92	2799
B-PER	0.52	0.18	0.27	158
I-PER	0.47	0.20	0.29	181
B-DATE	0.67	0.34	0.45	76
I-DATE	0.64	0.23	0.33	40
B-LOC	0.96	0.35	0.51	124
I-LOC	0.40	0.13	0.20	15
accuracy			0.85	3393
macro avg	0.65	0.34	0.42	3393
weighted avg	0.82	0.85	0.82	3393





## 4. ALIGNEMENT AVEC UN RÉFÉRENTIEL

**Les référentiels**

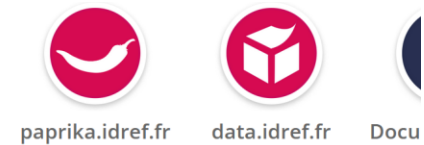
**Différentes stratégies pour la récupération d'un URI**

# ALIGNEMENT AVEC UN RÉFÉRENTIEL

- Référentiel : base de données dans laquelle sont référencés des noms de personnes, de lieux etc. à partir d'un identifiant unique (URI).
- IdRef : référentiel français pour les noms de personne.
- Programme : [https://github.com/lab-bnu/ritter\\_ner/blob/main/programmes\\_standalone/alignement.py](https://github.com/lab-bnu/ritter_ner/blob/main/programmes_standalone/alignement.py) permet de requêter automatiquement l'URI d'un nom entré en ligne de commande. Option `-sparql`

# RÉCUPÉRATION DE L'URI

- Requêtage de IdRef :



Nouvelle recherche	Modifier la recherche
<p>l'autorité</p> <p>de personne</p> <p>de collectivité</p> <p>rès</p> <p>commun</p> <p>e ou genre Rameau</p> <p>géographique</p> <p>lle</p>	<p>Termes de recherche</p> <div><input data-bbox="1332 915 1900 991" type="text" value="martin bucer"/>1 résultat</div> <p>Bucer, Martin (1491-1551)</p> <p><input type="checkbox"/> Troncature automatique</p>

<https://www.idref.fr/>

# BUCER, MARTIN (1491-1551)



Contrôle Paprika



data.idref



Élément EAD



Export biblio



Export XML



Export JSON




Export RDF




Améliorer la notice



Signaler une erreur

Identifiant pérenne de la notice : <https://www.idref.fr/02749957X> 

URI de l'entité : <http://www.idref.fr/02749957X/id> 

Notice de type : Personne

## Point d'accès autorisé

Bucer, Martin (1491-1551)

Pseudonyme

<https://www.idref.fr/>

# RÉCUPÉRATION DE L'URI : SOLR

- Requêtage de IdRef via le moteur de recherche interne Solr : construction de la requête dans l'URL

URL de base : <https://www.idref.fr/Sru/Solr/>

# RÉCUPÉRATION DE L'URI : SOLR

Requête à partir d'index (critères de recherche):

<https://documentation.abes.fr/aideidrefdeveloppeur/index.html#index>

Utiliser l'API Solr pour interroger IdRef > |

Recherche "contient les mots" Index de type <i>text</i> : <i>_t</i>	Recherche exacte Index de type <i>string</i> : <i>_s</i>	Contenu de l'index
<b>persname_t</b>	persname_s	Nom et prénom de personne et leurs variantes respectives
<b>nom_t</b>	nom_s	Nom et variantes de nom de personne
<b>prenom_t</b>	prenom_s	Prénom et variantes de prénom de personne
<b>bestnom_t</b>	bestnom_s	Nom privilégié de la personne (200\$a)
<b>bestprenom_t</b>	bestprenom_s	Prénom privilégié de la personne (200\$b)
<b>corpname_t</b>	corpname_s	Nom de collectivité
<b>conference_t</b>	conference_s	Congrès
<b>datenaissance_dt</b>		Année de naissance
<b>datemort_dt</b>		Année de décès

# RÉCUPÉRATION DE L'URI : SOLR

- Requêtage de IdRef via le moteur de recherche interne Solr : construction de la requête dans l'URL

URL de base : <https://www.idref.fr/Sru/Solr/>

- Index retenus pour notre requête : persname\_t et anneenaissance\_dt:

Ex URL avec requête pour Bucer :

[https://www.idref.fr/Sru/Solr?q=persname\\_t:\(Martin%20AND%20Flach\)%20AND%20anneenaissance\\_dt:\[1400-01-01T23:59:59.999Z TO 1650-01-01T23:59:59.999Z\]](https://www.idref.fr/Sru/Solr?q=persname_t:(Martin%20AND%20Flach)%20AND%20anneenaissance_dt:[1400-01-01T23:59:59.999Z TO 1650-01-01T23:59:59.999Z])

- Par défaut la réponse est donnée en format HTML, mais on peut changer ça avec l'option wt suivi du format JSON par exemple

[https://www.idref.fr/Sru/Solr?q=persname\\_t:\(Martin%20AND%20Flach\)%20AND%20anneenaissance\\_dt:\[1400-01-01T23:59:59.999Z TO 1650-01-01T23:59:59.999Z\]&wt=json](https://www.idref.fr/Sru/Solr?q=persname_t:(Martin%20AND%20Flach)%20AND%20anneenaissance_dt:[1400-01-01T23:59:59.999Z TO 1650-01-01T23:59:59.999Z]&wt=json)

# RÉCUPÉRATION DE L'URI : SPARQL (VIRTUOSO)

- Requêtage de la base IdRef via l'endpoint Virtuoso en SPARQL :

```
PREFIX bio: <http://purl.org/vocab/bio/0.1/>
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
select ?nom ?naissance ?id
where {{
    ?id foaf:name ?nom;
        bio:event [a bio:Birth ; bio:date ?naissance];
        ?nom a foaf:name.

        ?nom bif:contains "nom".

    FILTER (xsd:integer(?naissance)<1650)

}}
```

LIMIT 10

[On peut essayer via : https://data.idref.fr/yasgui.html](https://data.idref.fr/yasgui.html)





ATELIER :  
PRÉSENTATION  
DES  
PROGRAMMES DE  
LAB-  
BNU/RITTER\_NER



# PROGRAMMES PRÉSENTÉS

**Installation : création de l'environnement virtuel Python et téléchargement des modules requis.**

**Chaîne de traitement complète sur de nouvelles données**

**Création jeux de données**

**Test avec spaCy**

**Test avec un modèle dispo sur HF**

**Test avec le modèle entraîné sur nos données (dispo sur HF)**

**Test extraction de l'URI IdRef**

# CRÉATION D'UN VENV POUR LE PROJET

- Cloner le repo : [https://github.com/lab-bnu/ritter\\_ner](https://github.com/lab-bnu/ritter_ner) (ou le télécharger + dézipper)

- Se déplacer en ligne de commande dans le dossier :

```
cd ./ritter_ner
```

- Venv avec Conda : `conda create -f requirements.txt -n ner`

- Venv avec poetry :

```
poetry shell #choisir ^3.10 ou ^3.11 pour la version de Python
```

```
poetry init
```

```
pip install -r requirements.txt
```

Si pip n'est pas installé : <https://pip.pypa.io/en/stable/installation/>

# AJOUT DES ENTITÉS NOMMÉES À DES DOCUMENTS XML NON ANNOTÉS

- programme : `xml2encoded_xml_ner.py`
- Paramètres sur `params.py` `param_xml2encoded_xml_ner`
  - `Outdir` : dossier de sortie où se trouveront les docs .xml encodés
  - `Csvdocname` : nom du document .csv qui contiendra les données traitées
  - `Xml_dir` : dossier contenant les pages .xml d'origine (sans aucune annotation)
  - `Model_spacy` : nom du modèle spaCy à utiliser pour la reconnaissance d'EN

# FORMAT DES DONNÉES POUR LE TRAITEMENT

- Passage d'un document xml annoté à un fichier csv avec `creation_jeu_donnees.py`
  - Paramètres généraux:
    - `Tags_to_extract` : liste des balises à extraire ‘
    - `Class_names` : nom des classes (alignées avec les balises) en correspondance avec `tags_to_extract`
    - Systèmes OIB (Outside, Inside, Beginning) avec B-LABEL pour le 1<sup>e</sup> token et I-LABEL pour les tokens suivants portant le même token.
      - Exemple : « Alexia Schneider » : Alexia B-PER | Schneider I-PER
    - Dossier et document en sortie (sans extension)
  - Paramètres spécifiques à la création du jeu de données :
    - `By_element`: les pages XML peuvent être divisées en élément, comme c'est le cas avec ‘div’, sinon prendra la page entière avec « body »
    - `Tokenizer` : peut être « split », « spacy » ou un modèle dispo sur HF
    - `Xml_dir` : dossier où se trouvent les pages XML
    - `Train_test_split` : True/False : détermine si le jeu complet doit être divisé en jeux d'entraînement et de test.

# FORMAT DES DONNÉES POUR LE TRAITEMENT

originaltext	text	tags	tag_ids	ents	prov
PARATUS (Idem). â€” [Strasbourg, Martin Flach s. d.), [', 'PARATUS', ',', '(', 'Idem', ')', ', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'C[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 1, 2, 0, 5, [(20, 30, 'B-LOC'), (32, 38, 'B-PER'), (39, 44, 'Catalogue des incuna					
PARATUS (Idem). â€” (Strasbourg, Martin Flach s. d.) i [', 'PARATUS', ',', '(', 'Idem', ')', ', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'C[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 1, 2, 0, 5, [(20, 30, 'B-LOC'), (32, 38, 'B-PER'), (39, 44, 'Catalogue des incuna					
556> PARATUS (Idem). â€” Strasbourg, Martin Flach s [556', '>', ',', 'PARATUS', ',', '(', 'Id', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'C[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 1, 2, 0, [(24, 34, 'B-LOC'), (36, 42, 'B-PER'), (43, 48, 'Catalogue des incuna					
PARATUS (Idem). â€” Strasbourg, Martin Flach 1491, i [', 'PARATUS', ',', '(', 'Idem', ')', ', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'C[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 1, 2, 0, 5, 0, [(19, 29, 'B-LOC'), (31, 37, 'B-PER'), (38, 43, 'Catalogue des incuna					
PARATUS (Idem). â€” Haguenau, Henri Gran 1500, in-z [', 'PARATUS', ',', '(', 'Idem', ')', ', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'C[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 1, 2, 0, 5, 0, [(19, 27, 'B-LOC'), (29, 34, 'B-PER'), (35, 39, 'Catalogue des incuna					
PAULSDORF, Martin. Voir : Agenda Wratislaviensis, nÂ [', 'PAULSDORF', ',', 'Martin', ',', 'O', 'B-PER', 'I-PER', 'I-PER', '[0, 1, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0] [(1, 10, 'B-PER'), (10, 11, 'I-PER'), (12, 18, 'I-P Catalogue des incuna					
PELBARTUS de Themeswar : Expositio libri Psalmorum [', ',', 'PELBARTUS', 'de', 'Theme', 'O', 'O', 'B-PER', 'I-PER', 'I-PER', '[0, 0, 1, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, [(2, 11, 'B-PER'), (12, 14, 'I-PER'), (15, 24, 'I-P Catalogue des incuna					
PELBARTUS de Themeswar : Stellarium coronae Beata [', 'PELBARTUS', 'de', 'Themesw', 'O', 'B-PER', 'I-PER', 'I-PER', '[0, 1, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, [(1, 10, 'B-PER'), (11, 13, 'I-PER'), (14, 23, 'I-P Catalogue des incuna					
PELBARTUS de Themeswar: Stellarium coronae B. V. M [', 'PELBARTUS', 'de', 'Themesw', 'O', 'B-PER', 'I-PER', 'I-PER', '[0, 1, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, [(1, 10, 'B-PER'), (11, 13, 'I-PER'), (14, 23, 'I-P Catalogue des incuna					
PELBARTUS DE THEMESWAR : Sermones Pomerii. â€” [', 'PELBARTUS', 'DE', 'THEMES', 'O', 'B-PER', 'I-PER', 'I-PER', '[0, 1, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 1, [(1, 10, 'B-PER'), (11, 13, 'I-PER'), (14, 23, 'I-P Catalogue des incuna					
SCHWENCKFELD, Caspar : Bekantnuss vom H. Sacra [', 'SCHWENCKFELD', ',', 'Casp', 'O', 'B-PER', 'I-PER', 'I-PER', '[0, 1, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, [(1, 13, 'B-PER'), (13, 14, 'I-PER'), (15, 21, 'I-P 20211206-002466.xm					
SCHWENCKFELD, Caspar : Summarium. â€” Haguenæ [', 'SCHWENCKFELD', ',', 'Casp', 'O', 'B-PER', 'I-PER', 'I-PER', '[0, 1, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 3, 0, 0, 1, 2, [(1, 13, 'B-PER'), (13, 14, 'I-PER'), (15, 21, 'I-P 20211206-002466.xm					
SCHWENCKFELD, Caspar : Von erbawung des gewiss [', 'SCHWENCKFELD', ',', 'Caspa', 'O', 'B-PER', 'I-PER', 'I-PER', '[0, 1, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, [(1, 13, 'B-PER'), (13, 14, 'I-PER'), (15, 21, 'I-P 20211206-002466.xm					
SCHWENCKFELD, C. Voir aussi: SPECKER, Melch., nÂ [', 'SCHWENCKFELD', ',', 'C.', ', 'O', 'B-PER', 'I-PER', 'I-PER', '[0, 1, 2, 2, 0, 0, 0, 0, 1, 2, 2, 2, 0, 0, 0, 0, [(1, 13, 'B-PER'), (13, 14, 'I-PER'), (15, 17, 'I-P 20211206-002466.xm					
SCRIPTUM: collocutorum Augustanae Confessionis, i [', ',', 'SCRIPTUM', ':', 'collocut', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'O', 'C[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, [(146, 156, 'B-LOC'), (159, 165, 'B-PER'), (16, 20211206-002466.xm					
SEBASTIANUS, Claudius Metensis : Bellum musicale i [', 'SEBASTIANUS', ',', 'Claudius', 'O', 'B-PER', 'I-PER', 'I-PER', '[0, 1, 2, 2, 2, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, [(1, 12, 'B-PER'), (12, 13, 'I-PER'), (14, 22, 'I-P 20211206-002466.xm					

# TEST DES PROGRAMMES DE NER

Spacy :

```
python programmes_standalone/spacy_ner.py data/demo/demo_notices.txt [--  
model "fr_core_news_sm" --outpath output_doc.txt]
```

HF :

```
python programmes_standalone/testing_hf_ner.py data/demo/demo_notices.txt -  
-outpath output_doc.txt]
```

Notre modèle :

```
python programmes_standalone/spacy_ner.py data/demo/demo_notices.txt --  
model "fr_camembert_ritter" [--outpath output_doc.txt]
```

# EVALUATION :

Modèles dispo sur HF mais hors spaCy (le modèle camembert\_ritter ne fonctionnera pas) :

`python hf_ner.py`

Avec paramètres spécifiques dans `params.py` :

- `Model` : nom du modèle, par défaut « Jean-Baptiste/camembert-ner-with-dates »
- `Outdir/outdoc` : dossier et nom du document de sortie sans extension
- `Tokenized_with_model` : `True` si le jeu de donnée comprend une colonne « text » avec les tokens produit par le modèle
- `Ents_annotated` : `True` si le jeu de donnée en entrée (`param_general`) comprend une colonne « tags » avec les étiquettes alignées avec les tokens
- `Eval` : `True` si l'évaluation doit être faite (incompatible avec `ents_annotated False`)



# TEST DES REGEX SUR NOS DONNÉES

Programme regex :

Python regex\_ner.py [--writeconll --confmatrixnorm]

Avec paramètres sur params.py:

Le document d'entrée est la sortie de param\_general (datadir/datadoc.csv)

Doc\_table\_alpha : document de référence contenant une liste des noms de personnes à chercher.

Outdir : dossier de sortie

Antidictionnaire : mots à ne pas sélectionner pour la NER.

# TÉLÉCHARGEMENT ET UTILISATION DU MODÈLE AFFINÉ POUR LA NER SUR LES DONNÉES DU RITTER :

Le modèle est disponible : [https://huggingface.co/alexiaschn/fr\\_camembert\\_ritter](https://huggingface.co/alexiaschn/fr_camembert_ritter)

Pour le télécharger et l'utiliser en ligne de commande (en plus des modules spaCy)

```
pip install
```

```
https://huggingface.co/alexiaschn/fr\_camembert\_ritter/resolve/main/fr\_camembert\_ritter-any-py3-none-any.whl
```

Puis on peut l'utiliser grâce à la pipeline de spaCy :

```
import spacy
```

```
nlp = spacy.load("fr_camembert_ritter")
```

```
exemples = ["Exemple de phrase avec des EN comme Strasbourg et Henri Eppendorf.",  
"Deuxième phrase d'exemple sans EN."]
```

```
print(*[(ent.text, ent.label_) for doc in nlp.pipe(exemples, disable=["tok2vec",  
"tagger", "parser", "attribute_ruler", "lemmatizer"]) for ent in doc.ents], sep="\n")
```

# DIVISION DES DONNÉES BRUTES EN BATCHES POUR LES LLMS GÉNÉRATIFS

- Nécessite une clé API (MistralAI ou OpenAI) valide.

Soit :

```
python programmes_standalone/create_batches.py doc_template.txt  
dossier/contenant/textes out/dir
```

Soit avec paramètres sur params.py:

```
python generative_models_ner.py
```

Paramètres :

Model : pour l'instant seul mistral est géré (mistral-large)

template : instruction qui sera préfixée à chacune des portions du texte

Txtdir : dossier contenant au moins un document .txt qui sera passé en prompt au modèle

Outdir : dossier de sortie (les batches seront format .txt)

Max\_length : nombre de tokens pour chaque batch (3000 recommandé)

# UTILISATION DE L'API POUR LES MODÈLES

Nécessite une clé API (MistralAI ou OpenAI) valide

`python generative_models_ner.py`

Paramètres :

`Model` : pour l'instant seul mistral est géré (mistral-large)

`template` : instruction qui sera préfixée à chacune des portions du texte

`Txtdir` : dossier contenant au moins un document .txt qui sera passé en prompt au modèle

`Outdir` : dossier de sortie (les batches seront format .txt)

`Max_length` : nombre de tokens pour chaque batch (3000 recommandé) Nécessite une clé API (MistralAI ou OpenAI) valide

# ALIGNEMENT ET ENCODAGE FINAL EN XML AVEC LES EN ET LEUR IDREF

On peut tester le requêtage des iDref dans une version légère du programme principal :

```
Python programmes_standalone/alignement.py "Schneider" [--sparql]
```

Ou on peut se servir du programme principal avec params.py pour automatiser la recherche d'IdRef et encoder le texte en XML avec les Entités Nommées et idRef :

```
Python alignement_encodage_xml.py [--writetsv --writexml]
```

Paramètres :

Ents\_docs\_path : document csv en entrée avec colonnes données en paramètres pour signaler quelles sont les colonnes pour :

Ents\_id\_colname : colonne où se trouve les étiquettes des EN (identifiant numérique)

tokens\_colname : colonne du texte tokenisé

Originaltext\_colname : colonne du texte sans tokenisation (peut-être identique à tokens\_colname)

Outdir : dossier de sortie des documents XML généré (le texte est rassemblé par provenance et les « div » sont rassemblés si les paramètres de création du jeu de donnée le mentionne)



MERCI POUR VOTRE  
ATTENTION

Alexia Schneider [email](#)

[GitHub](#) | [GitLab](#)

Lundi 8 juillet 2024 – Le Lab