UNIVERSITÉ DE PARIS CITÉ CAMPUS DE SAINT GERMAIN DES PRÉS
FACULTÉ DES SCIENCES FONDAMENTALES ET BIOMÉDICALES

En vue de l'obtention du diplôme de Master 1 en apprentissage machine pour la science des données

Thème du mémoire :

Exploration de documents biomédicaux par reconnaissance d'entités nommées

Réalisé par

Hayet LOUNI

Ikram CHERFAOUI

Melissa SFIHI

Youssef HASSAN

Encadré par **Mme. Séverine Affeldt**

Promotion 2022/2023

Table des matières

Introduction

La reconnaissance d'entités nommées (NER), également appelée extraction d'entités nommées, est une technique d'extraction d'informations utilisée dans le domaine du traitement automatique du langage naturel (NLP). Cette technique consiste à identifier et extraire des entités nommées telles que des noms de personnes, des lieux, d'organisations, des dates, des montants, etc. à partir des textes non structurés.
Les entités nommées sont des éléments clés de la plupart des textes. Par exemple, lors de la lecture d'un article de journal, les noms de personnes, de lieux et d'organisations peuvent fournir des informations importantes sur le sujet traité. De même, dans le cadre d'une analyse financière, les montants et les dates peuvent être cruciaux pour évaluer les performances d'une entreprise.
Les applications de la reconnaissance d'entités nommées (NER) sont très diverses et couvrent de nombreux domaines. Parmi lesquels nous pouvons citer :
-**La recherche d'informations** : La NER peut aider à trouver des informations spécifiques dans des textes volumineux en identifiant les entités pertinentes.
-**La traduction automatique** : En reconnaissant les entités nommées, la NER peut aider les systèmes de traduction automatique à mieux comprendre les textes source et à produire des traductions plus précises.
-**L'analyse de sentiment** : En identifiant les noms de personnes, d'organisations ou de produits mentionnés dans des commentaires ou des critiques en ligne, la NER peut aider à déterminer l'opinion générale des consommateurs.
Dans ce mémoire nous allons nous intéresser au **domaine biomédical**. En effet, la NER est essentielle pour extraire des informations à partir de textes biomédicaux non structurés tels que des articles scientifiques, des rapports de laboratoire, des dossiers médicaux et des bases de données médicales. En identifiant et en extrayant des entités nommées telles que des gènes, des protéines, des maladies, des traitements et des symptômes, la NER peut aider les chercheurs et les professionnels de la santé à accélérer la découverte de nouveaux traitements et thérapies pour des maladies spécifiques.
La reconnaissance d'entités nommées dans le domaine biomédical présente également des défis particuliers. Les textes biomédicaux sont souvent très techniques et contiennent des

termes complexes et spécialisés. De plus, les entités biomédicales peuvent avoir des noms très similaires ou des variantes de noms qui nécessitent une analyse précise pour éviter les erreurs de classification.
Cependant, la reconnaissance d'entités nommées est devenue un outil indispensable pour les chercheurs en biomédecine, les professionnels de la santé et les scientifiques pour extraire des informations importantes à partir de textes non structurés. Grâce à cette technique, il est possible d'accélérer la découverte de nouveaux traitements, d'améliorer les soins de santé et de mieux comprendre les maladies et les processus biologiques.

Objectifs :
La reconnaissance d'entités nommées dans les textes biomédicaux ne constitue pas un sujet récent. Ce sujet a été abordé depuis de nombreuses années et cela s'explique en grande partie par la complexité du langage utilisé dans les textes médicaux, qui contiennent de nombreux termes techniques et des acronymes spécifiques au domaine de la médecine.

Dès les années 1990, des travaux de recherche ont été menés pour développer des approches de reconnaissance d'entités nommées dans les textes biomédicaux. Ces approches étaient basées sur des règles et des dictionnaires, et ont permis d'identifier avec succès des entités nommées telles que les noms de maladies, de médicaments et d'organismes.

Avec l'avènement des techniques d'apprentissage automatique et l'augmentation exponentielle du volume de données médicales disponibles, de nouvelles approches ont été développées. Ces approches utilisent des algorithmes d'apprentissage automatique tel que les réseaux de neurones et les SVM, qui ont permis d'améliorer significativement les performances de la reconnaissance d'entités nommées. On citera également l'approche basée un modèle discriminatif de champs aléatoire conditionnel (CRFs, Conditionnal Random Fields) et un réseau profond de type LSTM (Long short Term memory) qui était l'une des premières approches de NER ayant atteint de très bons résultats.

Malgré les progrès accomplis, la reconnaissance d'entités nommées dans les textes biomédicaux reste un défi important. En effet, le langage médical évolue constamment et de nouveaux matière et acronymes sont créés régulièrement. De plus, les textes biomédicaux peuvent être très hétérogènes en termes de style et de format, ce qui peut compliquer la tâche de la reconnaissance d'entités nommées. Les modèles existants, que nous évoquerons un peu plus loin dans ce rapport, sont donc très spécifiques. On retrouvera des modèles dédiés à la détection de gêne, d'autres modèles centrés sur la détection des maladies ...etc.

L'objectif de ce projet sera donc de proposer un modèle plus générique, qui serait capable de détecter l'ensemble des entités relatives au domaine biomédical, qu'il s'agisse d'entités biologiques, de procédés biomédicaux ou de relations de cause à effet.

En conclusion, bien que la reconnaissance d'entités nommées dans les textes biomédicaux ne soit pas un sujet récent, elle reste un défi important pour la communauté scientifique.

Les progrès dans les techniques d'apprentissage automatique ont permis d'améliorer significativement les performances de la reconnaissance d'entités nommées, mais il reste encore beaucoup à faire pour développer des approches robustes et efficaces pour la reconnaissance d'entités nommées dans les textes biomédicaux.

Plan de travail :

Ce rapport synthétisera les travaux réalisés dans le cadre de notre projet de fin d'études en vue de la réalisation des objectifs initialement définis. Nous commencerons par exposer deux bibliothèques Python, à savoir **SpaCy** et **Flair**, en faisant une analyse comparative de leurs fonctionnalités respectives. Nous présenterons ensuite une liste exhaustive des modèles de reconnaissance d'entités nommées disponibles dans le domaine biomédical qui repose sur ces deux technologies.

Nous décrirons par la suite la méthodologie que nous avons adoptée, de la collecte des données à la mise en place du modèle de reconnaissance. Nous exposerons les résultats obtenus suite à l'entraînement du modèle et les difficultés rencontrées notamment dans la détection de certaines entités. Nous expliquerons enfin comment nous avons surmonté ces obstacles et nous présenterons les résultats de notre travail à travers une application Streamlit.

Chapitre 1

Étude de l'existant

1 Introduction

SpaCy et Flair sont deux bibliothèques de traitement du langage naturel (NLP) largement utilisées pour la détection des entités nommées (NER) dans le domaine biomédical. Le NER est une tâche clé dans le domaine biomédical qui consiste à extraire des informations importantes tels que les noms de médicaments, les maladies, les symptômes, les protéines et les gènes à partir des textes biomédicaux.

2 SpaCy

SpaCy est une bibliothèque de traitement du langage naturel (NLP) gratuite et open source, développée par Matt Honnibal de Explosion AI. Écrite en Cython, elle est conçue pour une utilisation en production avec une API simple et concise. C'est une bibliothèque de bas niveau mais intuitive et performante.
SpaCy permet de créer des applications pour traiter et comprendre de grandes quantités de texte, y compris pour extraire des informations, comprendre le langage naturel, ou prétraiter des textes pour le Deep Learning.[Rédac, 2022]

SpaCy dispose de plusieurs modèles pré-entraînés pour la reconnaissance d'entités nommées (NER) dans différentes langues, y compris l'anglais, le français, l'allemand, l'espagnol et bien d'autres. Ces modèles ont été entraînés sur des grands ensembles de données annotées et sont capables d'identifier des entités nommées telles que les noms de personnes, les organisations, les lieux, les produits, les événements, les dates, les montants d'argent, etc.
SpaCy propose également des modèles pré-entraînés spécifiquement conçus pour le traitement de textes biomédicaux et pour la reconnaissance d'entités nommées dans ce domaine. Ces modèles peuvent être utilisés tels quels ou affinés sur des données spécifiques à un domaine pour améliorer leur précision.[Spacy, 2023]
Il existe une extension de SpaCy appelée scispaCy, qui propose des modèles pré-entraînés spécialisés pour le domaine biomédical.

3 SciSpaCy

SciSpaCy [AI2, 2022] est un package Python contenant des modèles spaCy pour le traitement de textes biomédicaux, scientifiques ou cliniques.
SciSpaCy propose des modèles pré-entraînés spécialisés pour le domaine biomédical, tels que les noms de maladies, les noms de médicaments, les gènes, les protéines, etc. De plus, scispaCy offre également des fonctionnalités supplémentaires telles que la lemmatisation spécialisée pour les termes biomédicaux et l'extraction de relations pour identifier les rela-

tions entre les entités biomédicales. Le tableau ci dessous résume les modèles pré-entraîner qui sont contenu dans ce package [Mooney and Kaggle, cess]

Modèle	F1	Entity type
en_ner_craft_md	76.75	GGP, SO, TAXON, CHEBI, GO, CL
en_ner_jnlpba_md	72.28	DNA, CELL_TYPE, CELL_LINE, RNA, PROTEIN
en_ner_bc5cdr_md	84.53	DISEASE, CHEMICAL
en_ner_bionlp13cg_md	76.57	AMINO_ACID,ANATOMICAL_SYSTEM, CANCER, CELL, CELLULAR_COMPONENT, DEVELOPING_ANATOMICAL_STRUCTURE, GENE_OR_GENE_PRODUCT, IMMATERIAL_ANATOMICAL_ENTITY, MULTI_TISSUE_STRUCTURE, ORGAN, ORGANISM, ORGANISM_SUBDIVISION, ORGANISM_SUBSTANCE, PATHOLOGICAL_FORMATION, SIMPLE_CHEMICAL, TISSUE

4 Flair

Flair [Akbik and contributors, cess] est une bibliothèque open source pour le traitement du langage naturel (NLP). Elle permet de résoudre différentes tâches de NLP telles que la reconnaissance d'entités nommées (NER), l'étiquetage grammatical (POS tagging), la classification de texte, la traduction de langue et bien plus encore. Flair est conçue pour être facile à utiliser et à configurer, avec une API simple et une documentation détaillée. Elle est basée sur des modèles neuronaux pré-entraînés, qui peuvent être finement ajustés pour des tâches spécifiques en utilisant des techniques d'apprentissage en transfert. La bibliothèque offre également un pipeline de traitement complet qui peut être personnalisé en fonction des besoins de l'utilisateur. Flair est compatible avec plusieurs langues, notamment l'anglais, l'allemand, le français, l'espagnol et l'italien. Elle dispose également d'un module de traitement pour les langues à écrire de droite à gauche, comme l'arabe et l'hébreu. En somme, Flair est une bibliothèque NLP complète et flexible, offrant des solutions pour une variété de tâches de traitement de texte.

5 Modèles Flair prè-entrainés existants

Voici quelques exemples de modèles pré-entraînés pour la reconnaissance d'entités dans les domaines de la santé et de la biologie disponibles dans la bibliothèque Flair :

- Drug-NER : un modèle pré-entraîné spécifiquement pour la reconnaissance d'entités nommées liées aux médicaments, tels que les noms de médicaments, les doses, les voies d'administration.
- Symptom-NER : un modèle pré-entraîné pour la reconnaissance des entités liées aux symptômes dans les textes médicaux. Il se concentre spécifiquement sur l'identification des symptômes rapportés par les patients.
- GSC-News-Data : un modèle pré-entraîné pour la reconnaissance d'entités nommées dans les articles de presse scientifique, y compris les maladies, les traitements, les découvertes, les institutions.
- Med-NER : un modèle pré-entraîné pour la reconnaissance d'entités nommées dans les textes médicaux, en se focalisant sur les maladies, les symptômes, les traitements.

6 Critique de l'existant

Les modèles pré-entraînés pour la reconnaissance d'entités nommées dans les textes biomédicaux sont des outils très utiles pour identifier des entités biomédicales telles que des maladies, des symptômes, des gènes, des protéines, etc. Ils sont spécialement conçus pour identifier les maladies dans les publications biomédicales ou pour des besoins très spécifiques. De plus, chaque modèle reconnaît en moyenne une dizaine de types d'entités ce qui est très peu pour ce qu'il en est.

Cependant, il est important de noter que l'utilisation de modèles pré-entraînés pour la reconnaissance d'entités biomédicales peut présenter certaines limites en matière de couverture et de précision des entités détectées. De plus, ces modèles peuvent ne pas être adaptés à des domaines spécifiques et nécessiter une adaptation ou un entraînement supplémentaire pour être efficacement utilisés dans des contextes médicaux précis.

Dans notre cas, notre recherche concerne la reconnaissance de plusieurs types d'entités, à savoir les maladies, les symptômes, les facteurs environnementaux et les gènes. Trouver un modèle pré-entraîné qui inclut toutes ces entités est une tâche complexe car il existe de nombreux modèles pré-entraînés spécifiques à des domaines particuliers, mais il n'en existe pas de généralistes correspondant à nos besoins.

Nous avons donc opté pour la création de notre propre modèle en l'entraînant sur un corpus spécialisé appelé NERO, que nous détaillerons dans la section suivante [Akbik et al., 2021].

Chapitre 2

Méthodologie

1 Introduction

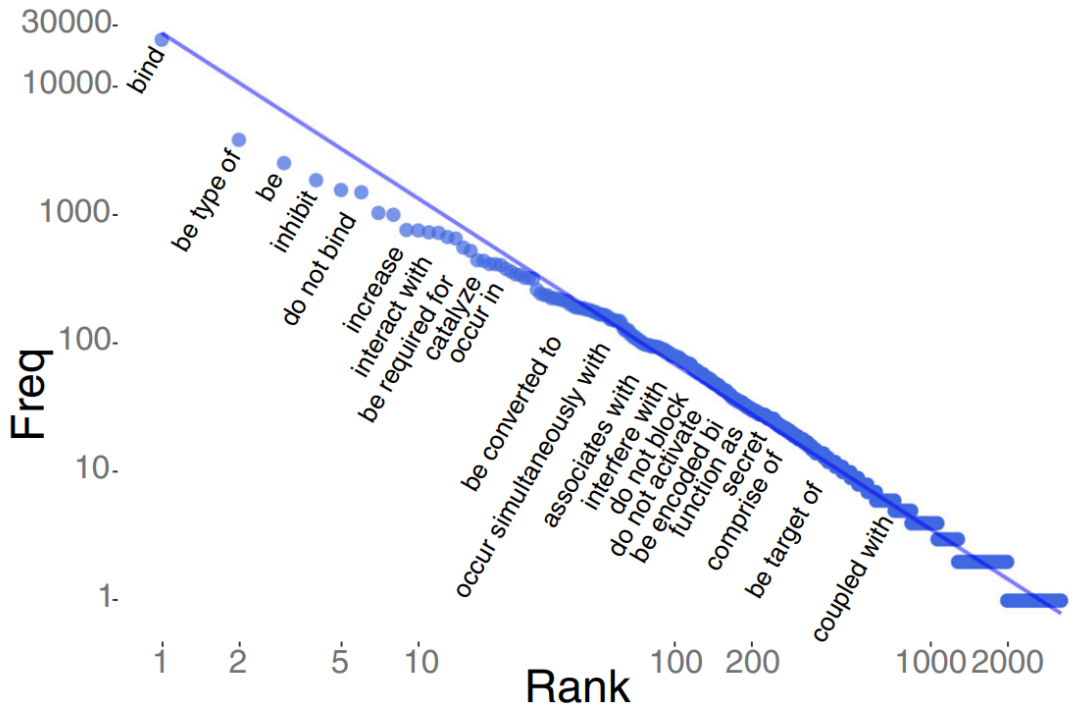
Le chapitre de la méthodologie vise à présenter les approches et les étapes clés utilisées dans notre étude. Dans cette introduction, nous aborderons les aspects essentiels de notre méthodologie, en mettant l'accent sur la présentation du corpus NERO et de ses spécificités. Le corpus NERO est un élément central de notre recherche, Comprendre la structure et les caractéristiques de ce corpus est crucial pour la suite de notre étude.

2 Présentation du corpus NERO et de ses spécificités

Dans cette partie, nous allons exposer le corpus qui aura servi à entraîner notre modèle de reconnaissance d'entités nommées dans le contexte biomédical. Nous allons, dans un premier temps, découvrir l'ontologie NERO [BioPortal, cessa], pour ensuite explorer le corpus qui lui est associé.

2.1 Découvrir NERO

Une ontologie est un ensemble de catégories qui définissent les types d'entités nommées dans les textes et les relations entre elles. L'utilisation d'une ontologie facilite la reconnaissance et l'extraction d'informations à partir des textes en structurant les termes et les concepts de manière systématique. NERO en est une, elle a spécifiquement été développée pour décrire les entités textuelles dans les textes biomédicaux. Elle répertorie les termes pertinents dans le domaine biomédical tel que des protéines, des gènes, des maladies, des médicaments, des organes, des tissus, des symptômes et des signes cliniques et peut aider à identifier de nouvelles associations significatives entre différentes entités biomédicales. Par exemple, en utilisant NERO pour identifier les relations entre les protéines et les maladies, les chercheurs peuvent identifier de nouveaux biomarqueurs pour des maladies spécifiques ou de nouvelles cibles thérapeutiques. En effet, en utilisant des techniques d'embedding, il est possible de découvrir que les protéines A et B sont souvent mentionnées ensemble, ce qui suggère une certaine relation entre elles qui pourrait s'exprimer à travers les termes suivants :

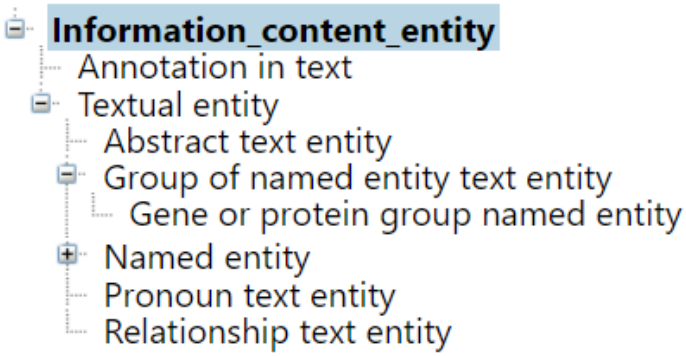


Les ontologies existantes étant très spécifiques, NERO se distingue par :

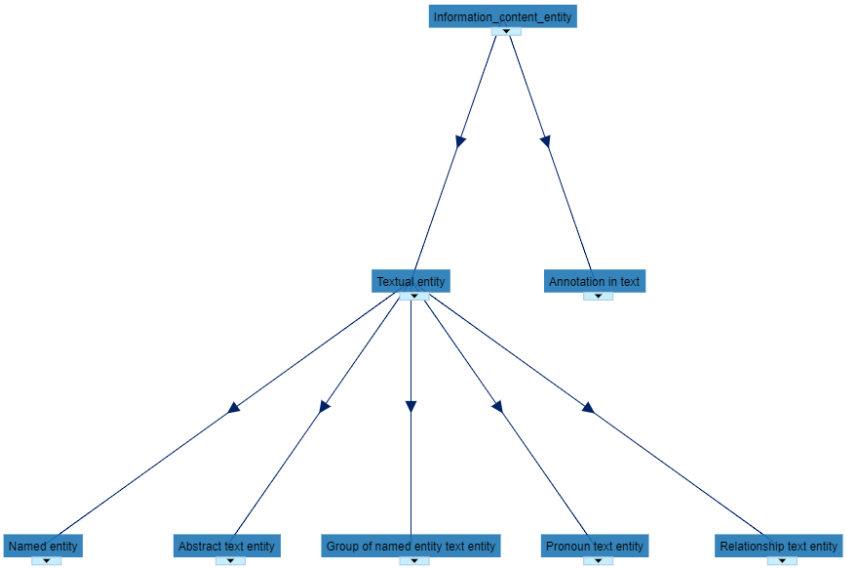
1. **Sa taille :** 189965 phrases annotées qui encapsulent 26488 entités nommées, 13 relations répetoriées et 98 labels répértoriés)
2. **La qualité de son ensemble de données :** des lignes directrices ont été mises à disposition d'experts humains afin d'annoter les noms des entités dans les textes.
3. **Sa couverture étendue des entités :** certaines entités nommées peuvent être ambiguës, c'est-à-dire qu'elles peuvent correspondre à plusieurs types d'entités différentes. Par exemple, "hémoglobine 2 mutante" peut faire référence à la fois à une protéine et à un gène. Pour exprimer cette ambiguïté, NERO définit des concepts ambigus tels que "GeneOrProtein", qui englobe les termes "Gene" et "Protein". Cela permet de représenter les entités textuelles et de préserver l'incertitude liée au texte.
4. **Sa flexibilité et extensibilité :** des pictogrammes (symboles graphiques) ont été ajoutés pour toutes les entités nommées.
5. **L'extraction d'associations significatives :** des modèles d'embeddings qui démontrent les promesses des associations biomédicales intégrées dans ce corpus.

Ces caractéristiques permettent une reconnaissance plus précise des entités nommées et une extraction plus complète d'informations pertinentes dans les textes biomédicaux.

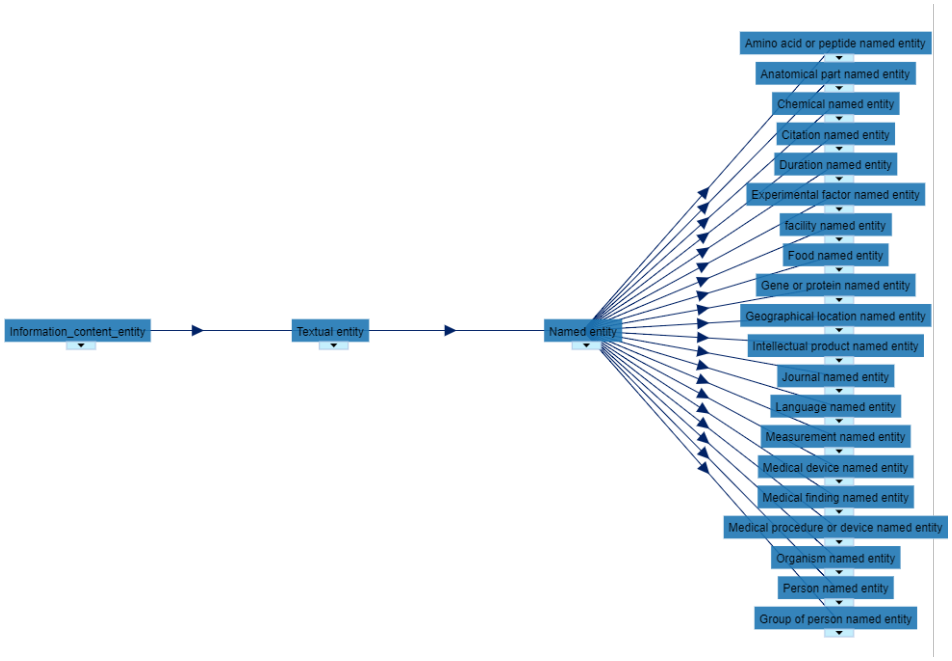
De plus, NERO est conçu comme une extension de l'ontologie IAO [Name, cessa], qui est l'une des ontologies OBO [Name, cessc] les plus utilisées pour représenter l'information. La classe InformationContentEntity est utilisée comme classe de niveau supérieur dans NERO. Elle a deux sous-classes : TextualEntity définie par IAO et la classe AnnotationInText pour capturer des informations sur les annotations [BioPortal, cessb].



La branche TextualEntity dans IAO inclut diverses parties de texte (par exemple, titre de document, résumé et tableau), mais elle n'inclut pas les entités qui sont importantes pour les processus de fouille de texte, telles que les entités nommées. NERO permet la représentation des entités nommées et définit formellement des classes telles que :



NamedEntity : une entité nommée est un objet spécifique dans le texte qui a été nommé, tel qu'un gène, une protéine, une maladie, un médicament, un organe, un tissu, un symptôme ou un signe clinique.



NamedEntityGroup : un groupe d'entités nommées est une collection d'entités nommées qui sont liées les unes aux autres, telles que des gènes impliqués dans une voie de signalisation, des protéines qui interagissent dans une interaction protéine-protéine, ou des symptômes associés à une maladie.

RelationshipAssertion : une assertion de relation est une déclaration qui décrit la relation entre deux ou plusieurs entités nommées, telles qu'une interaction protéine-protéine, une association maladie-gène ou un effet secondaire d'un médicament.

EntityMention : une mention d'entité est une occurrence spécifique d'une entité nommée dans le texte, telle que la mention d'un gène dans une phrase ou la mention d'un symptôme dans un rapport de cas.

2.2 Explorer NERO :

Les concepteurs et créateurs de l'ontologie NERO, ont mis à disposition, un corpus accessible et open-source en ligne, qu'il est possible d'utiliser après installation [Name, cessb] :

```
pip install NERO-nlp
```

et importation :

```
from NERO.corpus import corpus
```

Le corpus de NERO contient **362740 textes annotés**. Il possède 31 colonnes dont **entity1**, **entity2** qui sont les entités détectées dans la phrase, et **semantic class entity**, **semantic class entity** qui sont les annotations attribuées à ces entités.

Suite à plusieurs traitements, nous récupérons **189965 textes annotés** sur 362740 initiaux Nous créons ensuite, les colonnes **sentenceText**, **entity** et **semantic class entity** et nous maintenons la colonne **procset**.

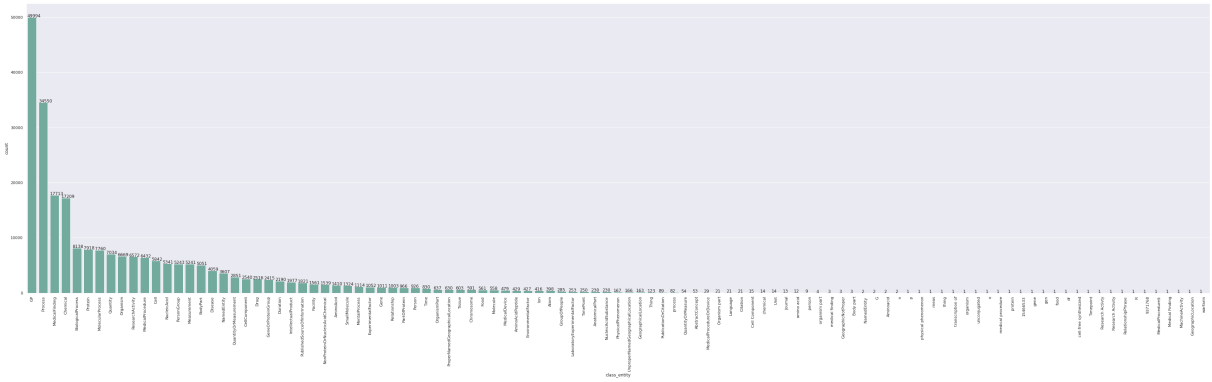
	sentenceText	entity	semantic_class_entity	procset
0	(A) Amph2 SH3 and Amph2-SH3deltaDAPS domains ...	Amph2	GP	bind
4	(A) and an NH -terminal deletion mutant Bcl-...	Bcl-2	GP	bind
5	(A) and an NH -terminal deletion mutant Bcl-...	Bax	GP	bind
10	(A) and cephalexin (B) binding to WT BclI (f...	cephalexin	GP	bind
12	(A) and cephalexin (B) binding to WT BclI (f...	cephalexin	Chemical	bind
...	...	...	...	...
224596	The metabolism of pteroylglutamic acid by the ...	pteroylglutamic acid	NonProteinOrNucleicAcidChemical	manualset1
224597	The metal sites on sarcoplasmic reticulum memb...	lanthanide ions	Ion	manualset1
224599	The methoxy methyl group of MTBE was oxidized ...	CO2	NonProteinOrNucleicAcidChemical	manualset1
224600	The methyl substitution at positions C9 or C10...	T antigen	Protein	manualset1
224601	The mice in the early postnatal period were mo...	radiation	EnvironmentalFactor	manualset1

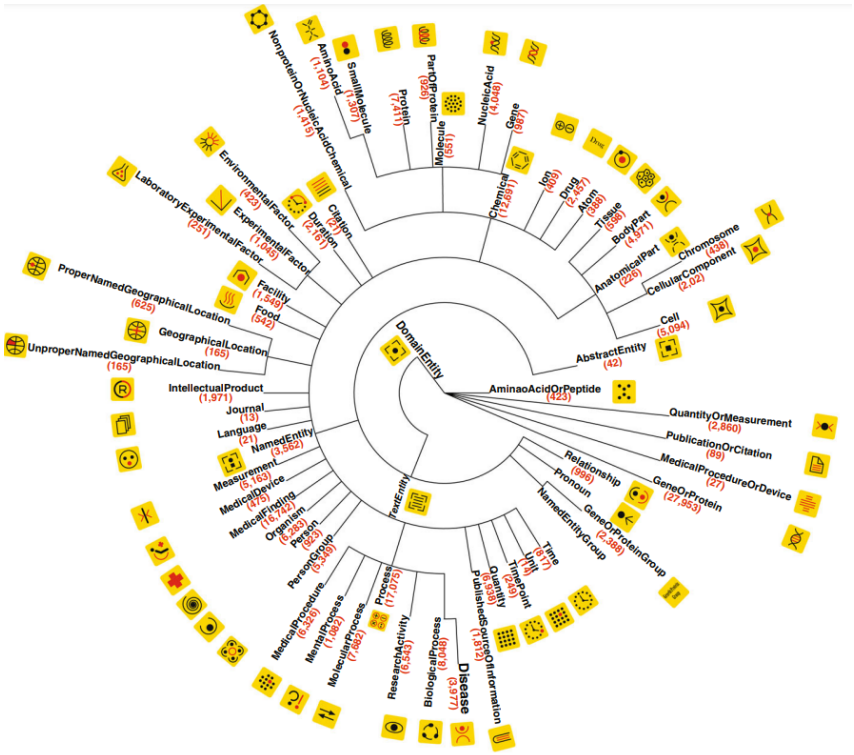
Nous regroupons ensuite les textes. Nous souhaitons avoir, pour chaque texte, la liste de tuples (entités,annotations) pour faciliter la manipulation.

	text	annotation
0	MET (Mesenchymal epithelial transition fac...	[(MET, GP), (HGFR, GP), (MET, GP), (MET, GP), ...
1	A proinflammatory cytokine is a cytokine whic...	[(proinflammatory cytokine, GP), (proinflammat...
2	Additionally, IL-27 itself may mimic the func...	[(IL-27, GP), (STAT1, GP), (IFN- Î³, GP), (STA...
3	After surgery and standard radiotherapy, an e...	[(radiotherapy, MedicalProcedure), (breast can...
4	Antiangiogenic effect of silibinin was couple...	[(silibinin, Chemical), (silibinin, Chemical),...
...	...	...
35319	â□□Transcriptomeâ□□ refers to the entirety of ...	[(Transcriptome, NucleicAcid), (mRNA, NucleicA...
35320	â□□Atypical antipsychotic drugsâ□□ target othe...	[(Atypical antipsychotic drugs, Chemical), (At...
35321	â□□The analysis shows that early childhood tel...	[(television viewing, Process), (autism, Medic...
35322	â□¢\tAnticonvulsants. These mood stabilizing m...	[(valproic acid , Chemical), (divalproex, Chem...
35323	â□¢\tBenzodiazepines. These anti-anxiety medic...	[(anti-anxiety medications, Chemical), (anti-a...

35324 rows × 2 columns

En calculant le nombre d'entités annotées par chaque label, nous constatons que le label GP (Gene or Protein) est le plus attribué. Et que le corpus compte **98 labels**.





L'ensemble des étapes relatives aux différents traitements et les résultats sont disponibles sur : [Notebook-Nero-Traitements]

3 Présentation de l'architecture de modèle SpaCy from scratch

Après avoir effectué tous les traitements nécessaires sur le corpus NERO, tels que la suppression des mots vides, la normalisation et la lemmatisation et le regroupement des texte par entités. Maintenant nous sommes prêts à utiliser ces données pour entraîner un modèle Spacy qui peut être utilisé pour effectuer la tâches de la reconnaissance d'entités nommées.

Spacy accepte un format de données spécifique pour l'entraînement des modèles de reconnaissance d'entités nommées, qui consiste en une liste de tuples contenant le texte de

l'exemple et un dictionnaire des positions de début et de fin de chaque entité nommée dans le texte.
Cette structure de données est essentielle pour que le modèle puisse apprendre à détecter les entités nommées avec précision dans les textes.
Pour cette raison, nous avons dû adapter les données de notre corpus NERO pour qu'elles correspondent au format attendu par Spacy, en incluant les positions de début et de fin de chaque entité nommée. Ce processus nous permettra d'entraîner un modèle Spacy précis pour la reconnaissance d'entités nommées dans nos données.

Après avoir appliqué cette étape de préparation des données, nous disposons maintenant d'un ensemble de données formatées pour l'entraînement d'un modèle Spacy pour la reconnaissance d'entités nommées. Ces données incluent une liste de textes d'exemple et un dictionnaire des positions de début et de fin de chaque entité nommée dans chaque texte.

Nous pouvons maintenant utiliser ces données pour entraîner un modèle Spacy performant et adapté à notre cas d'utilisation.

text	annotation	positions
MET (Mesenchymal epithelial transition fac...	[(MET, GP), (Mesenchymal epithelial transition...	[(4, 7, GP), (9, 49, GP), (66, 99, GP), (101, ...
A proinflammatory cytokine is a cytokine whic...	[(cytokine, GP), (systemic inflammation, Medic...	[(19, 27, GP), (57, 78, MedicalFinding), (3, 2...
Additionally, IL-27 itself may mimic the func...	[(IFN- Î³, GP), (JAK/STAT signalling molecules...	[(54, 61, GP), (96, 125, GP), (15, 20, GP), (1...
After surgery and standard radiotherapy, an e...	[(radiotherapy, MedicalProcedure), (breast can...	[(28, 40, MedicalProcedure), (114, 127, Medica...
Antiangiogenic effect of silibinin was couple...	[(hypoxia-inducing factor-1 alpha, GP), (NOS3,...	[(162, 193, GP), (116, 120, GP), (151, 156, GP...
...	...	...
xperiments involving specific inhibitors and s...	[(p38 mitogen-activated protein kinase, GP), (...	[(103, 139, GP), (182, 186, GP), (232, 235, GP...
yhl021c mutants have no detectable phenotype i...	[(yhl021c, NucleicAcid)]	[(0, 7, NucleicAcid)]
Î²-Lactam antibiotics are a broad class of ant...	[(Î²-lactam nucleus, Chemical), (Î²-Lactam ant...	[(108, 125, Chemical), (0, 21, Chemical)]
Î²-Lactam antibiotics are a broad class of ant...	[(Î²-lactam nucleus, Chemical), (penicillin, C...	[(108, 125, Chemical), (168, 178, Chemical), (...
Î³-Aminobutyric acid (GABA) is the chief inhib...	[(Î³-Aminobutyric acid, Chemical), (GABA, Chem...	[(0, 20, Chemical), (22, 26, Chemical), (41, 6...

Une fois que nos données ont été préparées au format requis pour Spacy, nous pouvons créer un modèle Spacy vide pour y entraîner notre modèle de reconnaissance d'entités nommées. Le modèle vide est initialisé avec des poids aléatoires et ne contient aucune connaissance préalable sur la langue ou les entités nommées que nous souhaitons reconnaître. Nous allons donc devoir entraîner le modèle à partir de zéro en utilisant nos données préparées. Nous avons ensuite procédé à l'entraînement du modèle en utilisant les données préparées en amont.

- Pour chaque exemple dans notre ensemble de données d'entraînement, nous avons extrait le texte et les annotations des entités nommées associées.
- Ensuite, nous avons initialisé les entités nommées de l'objet Doc pour qu'il contienne les valeurs de nos données en utilisant les positions de début et de fin et les labels fournis dans

les annotations de chaque exemple.
- Après, nous avons filtré les entités nommées à l'aide de la fonction filter_spans pour nous assurer que chaque entité est unique et qu'elle ne chevauche pas d'autres entités.
- Enfin, nous avons ajouté chaque document à un objet DocBin et l'avons sauvegardé sur le disque pour une utilisation ultérieure lors de l'entraînement du modèle. Cette étape est cruciale pour entraîner un modèle Spacy performant et adapté à notre cas d'utilisation.

Après avoir préparé les données d'entraînement pour notre modèle Spacy, nous avons rencontré un défi majeur dû à la taille considérable de nos données. En effet, notre ensemble de données était très volumineux et nous avons constaté que l'entraînement du modèle sur notre ordinateur local etait impossible dû au manque de ressources : celui-ci prenait beaucoup de temps et d'espace de stockage.
Pour résoudre ce problème, nous avons opté pour l'utilisation des ressources de calcul offertes par le serveur de l'université. Nous avons donc téléchargé les données d'entraînement sur le serveur et utilisé les capacités de calcul disponibles pour entraîner notre modèle. Cette approche nous a permis de gagner beaucoup de temps et d'obtenir des résultats plus rapidement.
Nous avons donc réussi à entraîner un modèle Spacy performant et adapté à notre cas d'utilisation, en utilisant efficacement les ressources informatiques disponibles.

A la fin de l'entrainement de notre modèle Spacy sur le serveur de l'université, nous avons téléchargé les résultats obtenus et nous les avons stockés sur notre compte Google Drive.

Chapitre 3

Résultats

1 Analyse des performances du modèle entraîné

Les résultats du modèle entraînés sont à première vue très prometteurs. Avec un F1-score de 0.88 et une variété de labels, n'ayant pas encore fait l'objet d'études pour certains : process, Gene or Protein, Medical Procedure, Research Activity, Body Part...etc, le modèle permet de reconnaître une grande panoplie de termes biomédicaux et de les annoter correctement. Nous pouvons le voir sur ces différents textes :

2 Présentation des problèmes rencontrés avec la détection des facteurs environnementaux

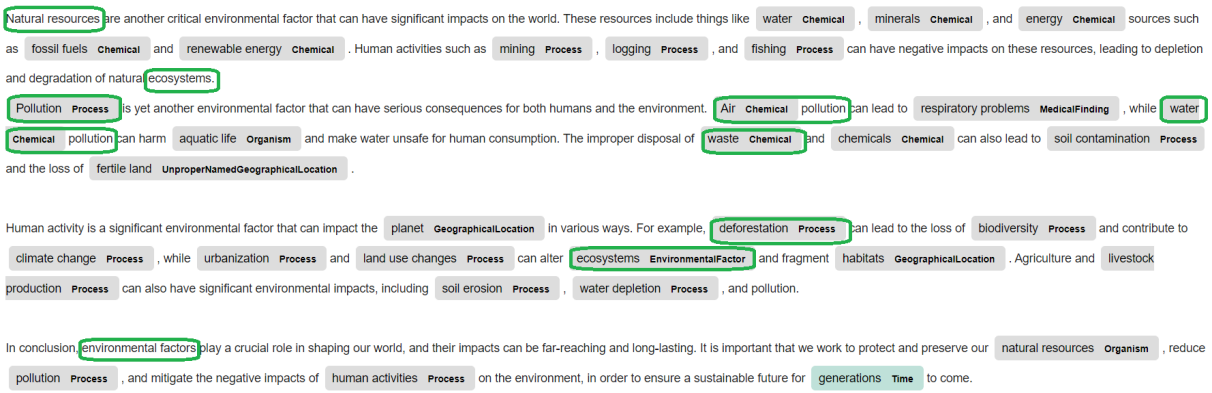
L'analyse des performances, nous a permis de constater que le modèle présentait d'assez bons résultats avec un F1-score de 0.88. Quelques tests effectués sur différents textes ont également montré que le modèle pouvait détecter l'ensemble des entités du corpus et les annoter correctement.

Cependant, suite à une analyse approfondie et quelques recherches nous nous sommes aperçus de quelques problèmes au niveau de la détection des facteurs environnementaux.

A titre d'exemple, dans ce texte les termes "Natural resources", "ecosystems", "environnemental factors" sont des facteurs environnementaux qui n'ont pas été reconnus.

Autre problème, les termes "Pollution", "deforestation", étant des facteurs environnementaux (Environnemental Factor) sont annotés comme des procédés (Process), ceci est dû au fait que le corpus NERO les considère ainsi.

Nous avons également, les termes "water pollution", "air pollution" qui ne sont pas annotés comme un facteurs environnmentaux (Environnemental Factor) au complet, seul les mots "water", et "air" sont reconnus en tant qu'entité chimique (Chemical).



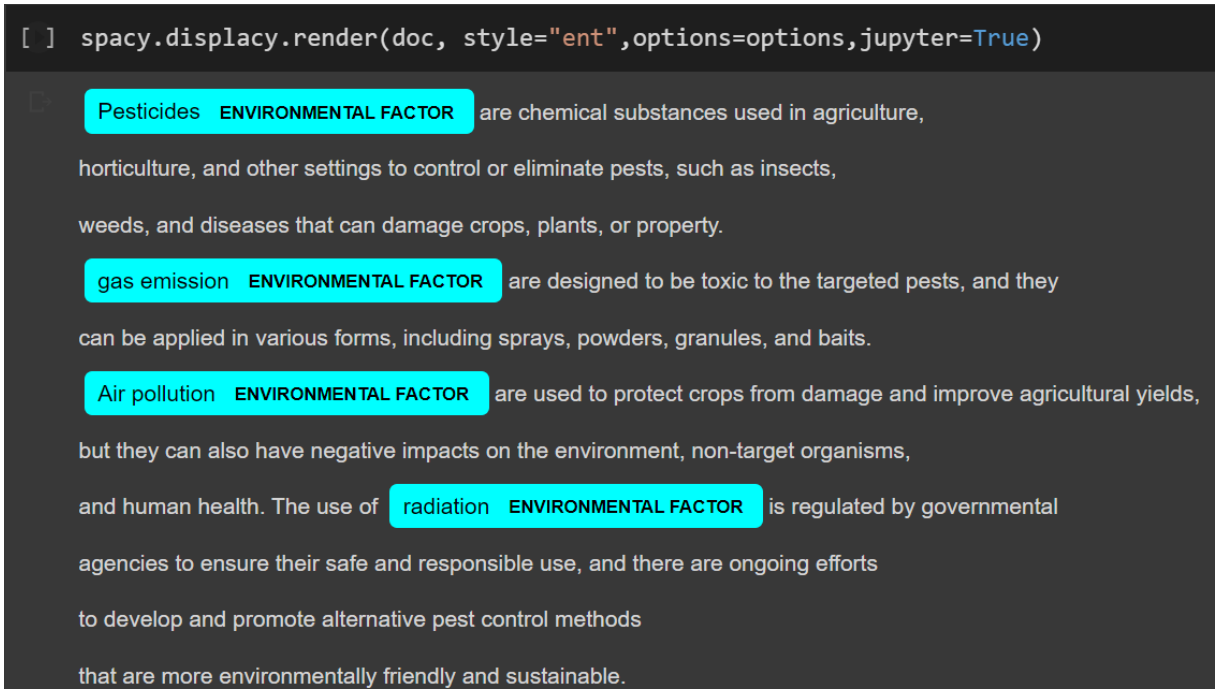
Pour palier à ces problèmes, nous avons dans un premier temps, entraîné un tout nouveau modèle à travers SpaCy à partir de données que nous avons nous même générées relatives aux facteurs environnementeux. En effet, comme aucun modèle n'a encore été entraîné pour détecter ces facteurs, nous étions contraints de développer le nôtre. Cela n'était pas suffisant, nous avons, ensuite combinés les deux modèles de sorte à compléter la détection du modèle NERO.

3 Description de la méthode utilisée pour créer un deuxième modèle pour la reconnaissance des facteurs environnementaux

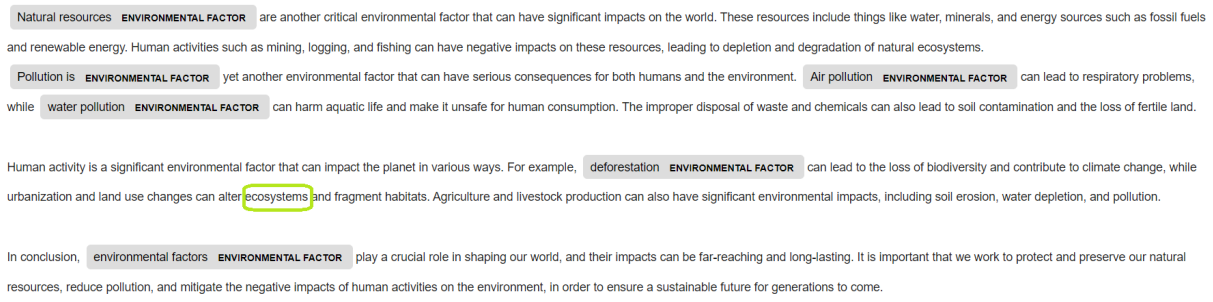
Après avoir utilisé le corpus Nero pour la détection d'entités nommées, nous avons remarqué que celui-ci ne contenait que six phrases de type facteur environnemental. Nous avons donc conclu que notre modèle ne serait pas en mesure de détecter les entités de type facteur environnemental en raison du manque de données. Afin de remédier à cette situation, nous avons envisagé de générer plus de données contenant ce type d'entités. Pour y parvenir, nous avons utilisé un générateur de texte appelé Chatgpt, en lui fournissant un exemple de phrase provenant du corpus Nero. Grâce à cet outil, nous avons été en mesure de générer 75 phrases différentes que nous avons stockées dans le jeu de données ci-dessous.

	Text	Entity	Type_Entity
0	The use of pesticides in fruit and vegetable c...	water	Environmental factor
1	The use of pesticides in fruit and vegetable c...	pesticides	Environmental factor
2	The emission of greenhouse gases from vehicles...	climate change	Environmental factor
3	The emission of greenhouse gases from vehicles...	gases	Environmental factor
4	Deforestation for agricultural expansion can r...	Deforestation	Environmental factor

L'entraînement de ce modèle a été réalisé sur Google Colab en utilisant les données générées par ChatGPT et les mêmes étapes de préparation de données que celles utilisées pour le corpus NERO. Nous avons ainsi pu obtenir un modèle capable de détecter les entités de type Environmental factor



Ainsi, sur le texte précédent, nous obtenons les résultats suivants :



On peut rapidement remarquer que les facteurs environnementaux non détectés ont bel et bien été détectés par le second modèle, à l'exception de "ecosystems" qui ne devrait pas poser problème sachant que nous allons combiner les deux modèles afin de reconnaître et d'annoter l'ensemble des entités.

4 Présentation des résultats combinés des deux modèles

Nous avons préféré combiner les deux modèles plutôt que d'affiner le premier modèle avec les données portants sur les facteurs environnementaux. Cette approche nous permettra d'avoir un meilleur contrôle sur la détection des entités et leur annotation. Pour se faire, nous nous sommes basés sur l'architecture de SpaCy et nous avons créé notre propre classe.

```
empty = spacy.blank("en")

class NlpCombined:
    def __init__(self, text):
        self.text = text

    def entities_dict(self):
        var_env_fact = model_fact_env(self.text)
        var_de_best = model_best(self.text)

        entities_dict = {}
        for ent in var_env_fact.ents:
            entities_dict[ent.text] = ent.label_

        keys_str = ' '.join(entities_dict.keys())

        for ent2 in var_de_best.ents:
            if ent2.text in keys_str:
                pass
            elif ent2.label_=='EnvironmentalFactor':

              entities_dict[ent2.text] ='ENVIRONMENTAL FACTOR'
            else:
              entities_dict[ent2.text] = ent2.label_

        return entities_dict

    def display_entities(self):
        doc = empty(self.text)
        html = displacy.render(doc, style="ent")

        dic = self.entities_dict()
        for key, entity in dic.items():
            html = html.replace(key, f"<span style='background-color:{col

        return html

    def display_text(self):
        display(HTML(self.display_entities()))
```

Cette classe comprendra les méthodes :

`__init__(self, text)`

qui n'est rien d'autre que le constructeur.

`entities_dict(self)`

qui se chargera de combiner les annotations des deux modèles, en annotant dans un premier temps le texte à travers le modèle de détection des facteurs environnementaux **model-fact-env** et en complétant l'annotation en reparcourant le texte avec le modèle **model-best** entraîné sur NERO.

Les méthodes

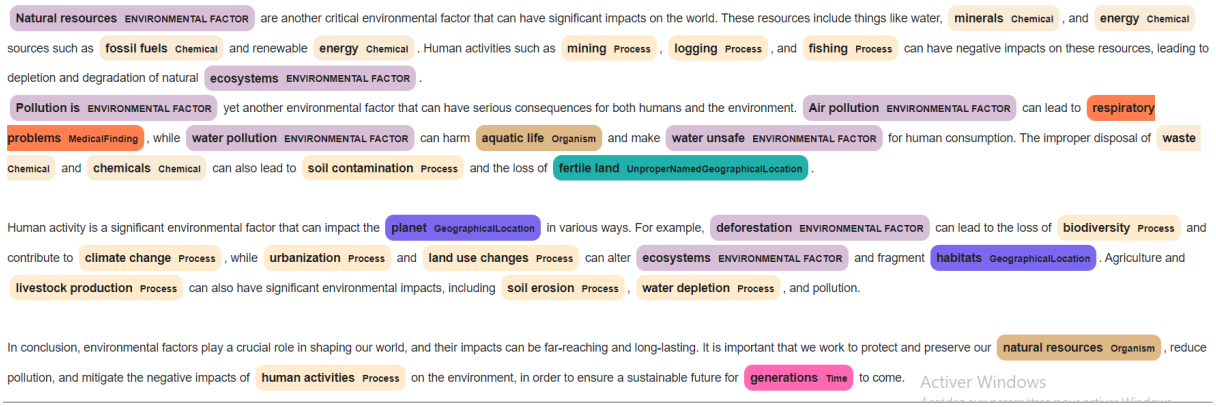
`display_entities(self)`

et

`display_text(self)`

permettront d'afficher le texte annoté en mettant en évidence les termes reconnus et les labels qui leur ont été affectés avec un code couleur en utilisant la bibliothèque displacy. Le texte est coloré selon la nature de l'entité détectée et les entités liées à l'environnement sont spécifiquement mises en évidence en utilisant une couleur différente et une étiquette spécifique.

Nous obtenons donc le résultat ci-dessous :



Chapitre 4

Application streamlit

1 Introduction

Afin de pouvoir présenter nos résultats de manière concrète, de faciliter la présentation et de bénéficier du travail accompli, nous avons eu l'idée d'implémenter une application simple qui permettrait d'insérer un texte en entrée et d'afficher les termes reconnus et annotés à l'utilisateur. Cette application permet de choisir le modèle souhaité entre les trois modèles : **le modèle entraîné sur le Corpus NERO**, **le modèle entraînés sur les données relatives aux facteurs environnementaux** et **le modèle qui combine les deux précédents modèles**. Nous avons pour cela utilisé **streamlit**.

2 Description de l'application streamlit créée pour exposer les résultats

Tout d'abord, les bibliothèques nécessaires sont importées, notamment Streamlit pour la création de l'interface utilisateur, Pandas pour la gestion des données, Matplotlib pour les couleurs et Spacy pour la reconnaissance d'entités nommées. Les modèles de reconnaissance d'entités nommées, qui ont été entraînés précédemment, sont également chargés dans le code. Un fichier CSV contenant les entités et leurs classes sémantiques est également chargé pour fournir des couleurs à chaque entité.

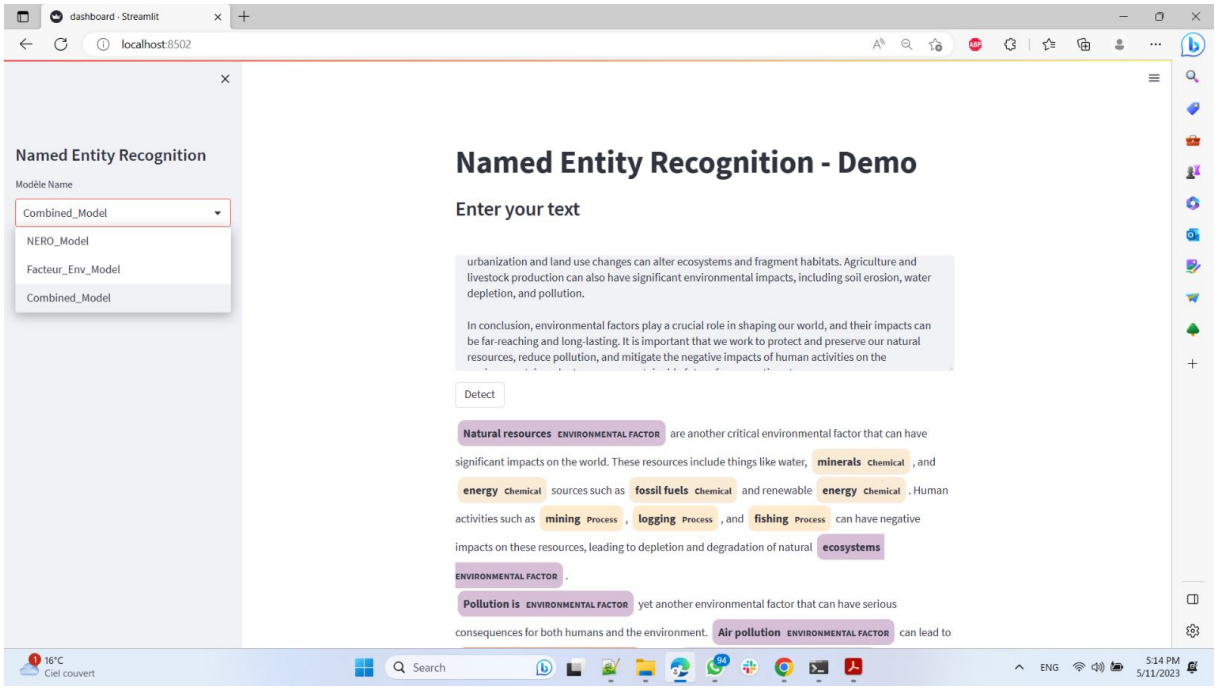
Ensuite, une classe NlpCombined est créée pour combiner les résultats des deux modèles NER. Cette classe utilise les modèles de reconnaissance d'entités nommées pour extraire les entités du texte entré. Elle combine ensuite les résultats des deux modèles et renvoie un dictionnaire contenant toutes les entités extraites et leurs classes sémantiques.

Une méthode pour afficher les entités nommées est également définie. Cette méthode prend le texte entré, extrait les entités nommées et les colore en fonction de leur classe sémantique. Les entités sont ensuite affichées dans une interface utilisateur.

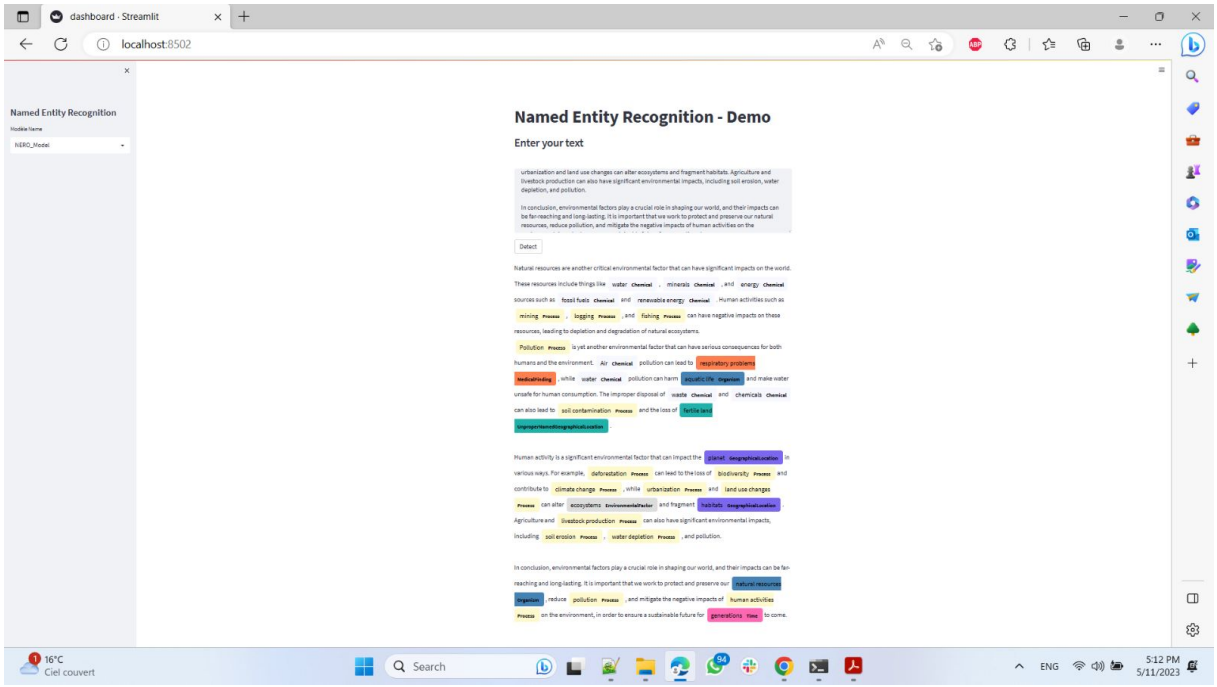
La partie sidebar de l'interface utilisateur permet de choisir l'un des trois modèles de reconnaissance d'entités nommées disponibles. Ensuite, l'utilisateur entre le texte dans la zone de texte et appuie sur le bouton "Detect". Si l'utilisateur a choisi l'un des deux modèles individuels, les entités détectées seront affichées dans une couleur unique. Si l'utilisateur a choisi le modèle combiné, les entités seront affichées avec leur classe sémantique associée.

En somme, cette application permet de visualiser les entités nommées extraites du texte en utilisant l'un des trois modèles de reconnaissance d'entités nommées disponibles.

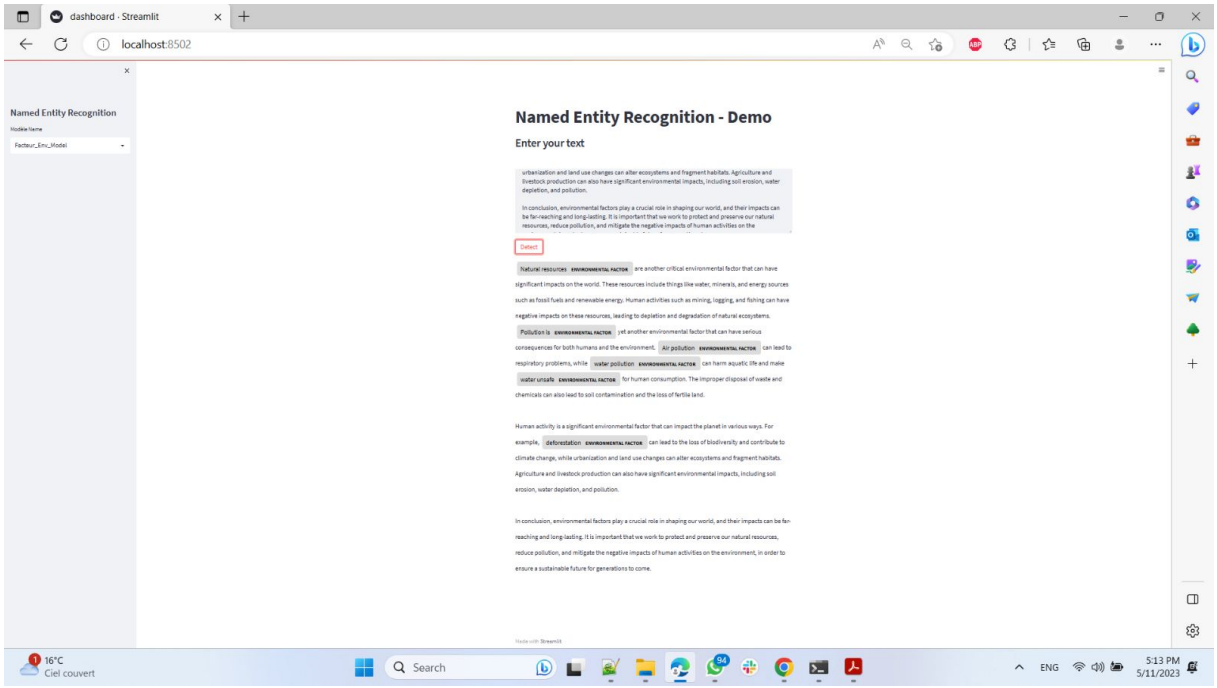
Dans ce qui suit, nous allons présenter l'application streamlit développée.



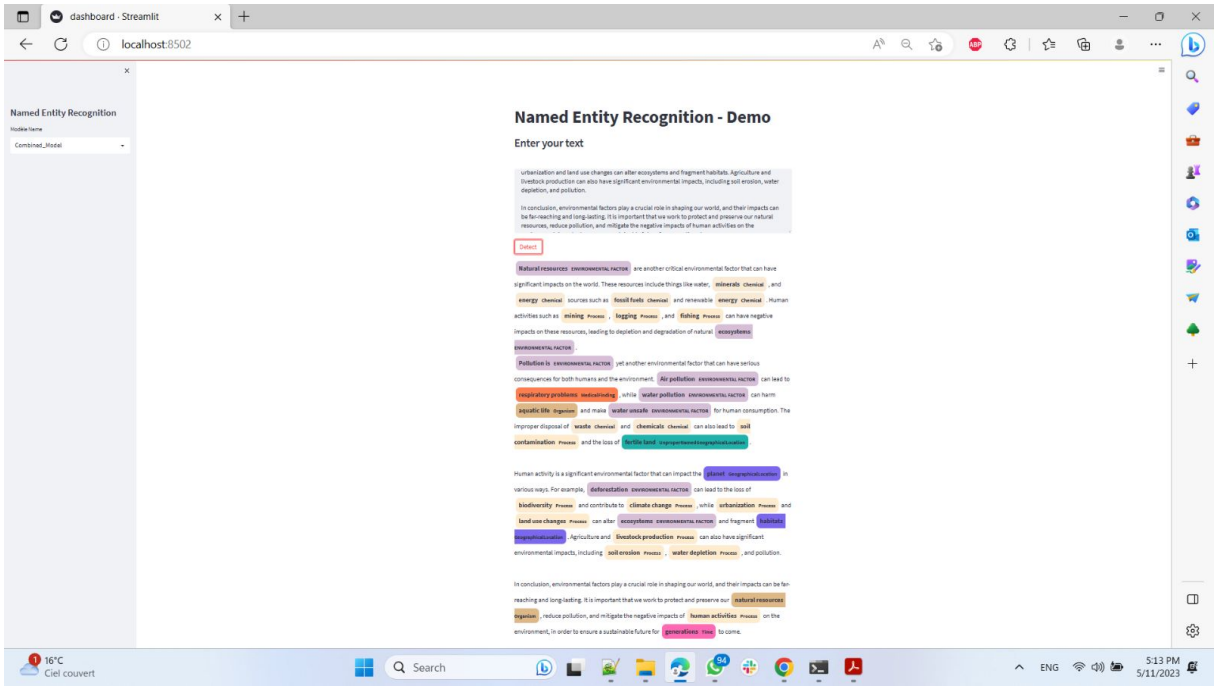
L'application contient une zone de texte où il est possible d'insérer son texte. En cliquant sur bouton "Detect", la reconnaissance d'entité nommées est lancée et le texte annoté est affiché à l'utilisateur :



Sur la précédente image, le modèle sélectionné est le modèle NERO-Spacy, entraîné sur le corpus NERO. Il est possible de choisir entre les trois modèles déployés. Le modèle de détection des facteurs environnementaux est sélectionné dans ce qui suit :



Sur cette dernière image, c'est le modèle combiné que nous sélectionnons :



Conclusion

En conclusion, notre modèle d'entités a démontré des performances prometteuses.
Nous avons obtenu de très bons résultats dans la reconnaissance précise des entités lors des tests que nous avons effectué, ce qui atteste de sa robustesse dans ces domaines.
Nous avons réussi à avoir un F1-score de 0.88 et une variété de label, ce score ayant pu être amélioré grace à la combinaison d'un autre modèle concernant les facteurs environnementaux, au modèle entrainé sur les données de NERO.

Grâce à l'utilisation des modèles combinés, nous avons pu améliorer et enrichir la reconnaissance des entités nommées. Notre modèle a bénéficié d'améliorations significatives en exploitant des techniques avancées d'apprentissage automatique. Avec cela, nous avons renforcé sa capacité à identifier avec précision les entités nommées des entités maladies, symptômes, facteurs environnementaux, gènes et bien plus encore. Cette approche multi-modèles nous a permis de tirer parti des forces de chaque modèle, tout en comblant les lacunes spécifiques de chacun d'entre eux.

Son application présente des implications significatives dans le domaine biomédical et la recherche scientifique. Il peut accélérer l'analyse des textes médicaux, faciliter l'extraction d'informations pertinentes et contribuer à la découverte de nouvelles connaissances. Il ouvre la voie à des améliorations futures, notamment en enrichissant les modèles existants, en les adaptant à des domaines spécifiques et en explorant de nouvelles techniques.

Dans l'ensemble, ce projet nous a permis de découvrir et d'approfondir nos connaissances en matière de traitement automatique du langage naturel et d'entraînement de modèles d'entités nommées. Nous avons également appris à gérer les défis liés au nettoyage des données, au choix des caractéristiques et à l'évaluation des performances, nous avons été confronté à la réalité des ressources pour l'entrainement du modèle, ainsi qu'à la transformation des données au format adéquat à l'entrainement. Ces apprentissages seront précieux pour de futurs projets dans le domaine du machine learning et du traitement automatique du langage naturel, ainsi que des éventuels projets en bioinformatique.

Bibliographie

[AI2, 2022] AI2, A. (2022). scispacy.

[Akbik et al., 2021] Akbik, A., Bergmann, T., Blythe, D., Rasul, K., Schweter, S., and Vollgraf, R. (2021). Nero. *npj Digital Medicine*, 4(1) :1–9.

[Akbik and contributors, cess] Akbik, A. and contributors, F. (Year of access). Flair. GitHub repository.

[BioPortal, cessa] BioPortal (Year of accessa). Bioportal.

[BioPortal, cessb] BioPortal (Year of accessb). Nero root classes.

[Mooney and Kaggle, cess] Mooney, P. and Kaggle (Year of access). Pretrained models for scispacy.

[Name, cessa] Name, A. (Year of accessa). Information artifact ontology.

[Name, cessb] Name, A. (Year of accessb). Nero-nlp.

[Name, cessc] Name, A. (Year of accessc). Obobibliothèque.

[Rédac, 2022] Rédac, T. (2022). spaCy : la bibliothèque Python Open Source de NLP.

[Spacy, 2023] Spacy, D. (2023). Doc · spaCy API Documentation.