Projet de Reconnaissance d’Entités Nommées (NER) avec WikiNER

# I-Qu’est-ce que la reconnaissance d’entités nommées ?

La Named Entity Recognition (NER), également appelée segmentation d’entités ou extraction d’entités, est un composant du [traitement automatique du langage naturel (NLP)](https://www.ibm.com/fr-fr/think/topics/natural-language-processing) qui identifie des catégories prédéfinies d’objets dans un corps de texte.

Ces catégories incluent, entre autres, les noms d’individus, d’organisations et de lieux, les expressions temporelles, les quantités, les codes médicaux, les valeurs monétaires et les pourcentages. Concrètement, la NER désigne le processus qui consiste à examiner une chaîne de texte (c’est-à-dire une phrase, un paragraphe ou un document entier) pour identifier et classer les entités catégorisables.

Lorsque le terme « NER » a été inventé lors de la Sixth Message Understanding Conference (MUC-6), l’objectif était de rationaliser les tâches d’extraction d’informations, qui nécessitaient le traitement de grandes quantités de texte non structuré et l’identification des informations clés. Depuis, la NER s’est développée et a évolué, en grande partie grâce aux progrès des techniques de [machine learning](https://www.ibm.com/fr-fr/think/topics/machine-learning) et d’[apprentissage profond](https://www.ibm.com/fr-fr/think/topics/deep-learning).

## Techniques de NER

Selon une enquête menée en 2019, près de 64 % des entreprises s’appuient sur des données structurées provenant de ressources internes, mais moins de 18 % exploitent les données non structurées et les commentaires sur les réseaux sociaux pour éclairer les décisions métier.

Les organisations qui utilisent la NER pour l’extraction de données non structurées s’appuient sur tout un éventail d’approches, mais la plupart se répartissent en trois grandes catégories : les approches basées sur des règles, les approches de machine learning et les approches hybrides.

* Les **approches basées sur des règles** consistent à définir un ensemble de règles correspondant à la grammaire d’une langue. Ces règles sont ensuite utilisées pour identifier des entités dans le texte en fonction de leurs caractéristiques structurelles et grammaticales. Ces méthodes peuvent demander beaucoup de temps et ne pas être adaptées aux données encore inconnues.
* **Les approches de machine learning** consistent à entraîner un modèle de machine learning piloté par l’[IA](https://www.ibm.com/fr-fr/think/topics/artificial-intelligence) sur un jeu de données étiqueté à l’aide d’algorithmes tels que les champs aléatoires conditionnels et l’entropie maximale (deux types de modèles de langage statistique complexes). Ces techniques peuvent aller des méthodes de machine learning traditionnelles (par exemple, les decision trees et les machines à vecteurs de support) à des approches d’apprentissage profond plus complexes, comme les réseaux neuronaux (RNN) récurrents et les transformers. Ces méthodes s’adaptent davantage aux données encore inconnues, mais elles nécessitent une grande quantité de données d’entraînement étiquetées et peuvent être gourmandes en calcul.
* Les **approches hybrides** combinent des méthodes basées sur des règles et des méthodes de machine learning afin de concilier les avantages de chacune. Il est ainsi possible d’utiliser un système basé sur des règles pour identifier rapidement les entités faciles à reconnaître et un système de machine learning pour identifier les entités plus complexes.

## Méthodologies de NER

Depuis les débuts de la NER, d’importances avancées méthodologiques ont été réalisées, en particulier celles reposant sur des techniques basées sur l’apprentissage profond. Les nouvelles itérations incluent les éléments suivants :

* **Réseaux neuronaux récurrents (RNN)** et **mémoire à long et court terme(LSTM)**. Les RNN sont un type de réseau neuronal conçu pour les problèmes de prédiction de séquences. Les LSTM, un type particulier de RNN, peuvent apprendre à reconnaître des formes au fil du temps et conserver des informations en « mémoire » sur de longues séquences, ce qui les rend particulièrement utiles pour comprendre le contexte et identifier des entités.
* **Champs aléatoires conditionnels(CRF)**. Les CRF sont souvent associés aux LSTM pour les tâches de NER. Ils permettent de modéliser la probabilité conditionnelle d’une séquence complète d’étiquettes, plutôt que de simples étiquettes individuelles, ce qui les rend particulièrement pratiques pour les tâches où l’étiquette d’un mot dépend de celle des mots qui l’entourent.
* **Transformers et BERT**. Les réseaux transformers, en particulier le modèle BERT (Bidirectionnel Encoder Representations from Transformers), ont eu un impact significatif sur la NER. À l’aide d’un mécanisme d’auto-attention qui évalue l’importance des différents mots, BERT prend en compte la totalité du contexte d’un mot en examinant les mots qui le précèdent et le suivent.

## Le processus de NER

### Étape 1. Collecte des données

La première étape de la NER consiste à agréger un jeu de données de texte annoté. Ce jeu de données doit contenir des exemples de texte où les entités nommées sont étiquetées ou marquées, indiquant leur type. Les annotations peuvent être effectuées manuellement ou à l’aide de méthodes automatisées.

### Étape 2. Prétraitement des données

Une fois le jeu de données collecté, le texte doit être nettoyé et formaté. Il faut parfois supprimer des caractères inutiles, normaliser le texte et/ou le diviser en phrases ou en tokens.

### Étape 3. Extraction de caractéristiques

À ce stade, les caractéristiques pertinentes sont extraites du texte prétraité. Il peut s’agir du marquage des parties du discours (Part of speech, POS), des plongements lexicaux ou des informations contextuelles, par exemple. Le choix des caractéristiques dépendra du modèle de NER choisi par l’organisation.

### Étape 4. Entraînement du modèle

L’étape suivante consiste à entraîner un modèle de machine learning ou d’apprentissage profond à l’aide du jeu de données annoté et des caractéristiques extraites. Le modèle apprend à repérer les formes et les relations entre les mots du texte, ainsi que les étiquettes d’entités nommées correspondantes.

### Étape 5. Évaluation du modèle

Une fois celui-ci entraîné, les performances du modèle NER doivent être évaluées. Vous pouvez alors mesurer des paramètres tels que la précision, le rappel et le score F1, qui indiquent dans quelle mesure le modèle identifie et classe correctement les entités nommées.

### Étape 6. Ajustement du modèle

Sur la base des résultats de l’évaluation, vous affinerez le modèle pour améliorer ses performances. Cela peut inclure l’ajustement des hyperparamètres, la modification des données d’entraînement et/ou l’utilisation de techniques plus avancées (par exemple, l’assemblage ou l’adaptation de domaine).

### Étape 7. Inférence

À ce stade, vous pouvez commencer à utiliser le modèle pour effectuer l’inférence sur un texte inconnu. Le modèle prend en charge le texte, applique les étapes de prétraitement, extrait les caractéristiques pertinentes et prédit les étiquettes d’entités nommées pour chaque token ou étendue de texte.

### Étape 8. Post-traitement

Des étapes de post-traitement peuvent être nécessaires pour affiner les résultats produits par le modèle et/ou ajouter des informations contextuelles. Ces tâches peuvent inclure la liaison d’entités, qui consiste à relier les entités nommées à des bases de connaissances ou à des bases de données à des fins d’enrichissement.

## Mise en œuvre du processus NER

Le moyen le plus simple de mettre en œuvre un système de reconnaissance d’entités nommées est de s’appuyer sur une [interface de programmation des applications](https://www.ibm.com/fr-fr/think/topics/api) (API). Les API NER sont des interfaces Web ou locales qui donnent accès aux fonctionnalités de NER. Voici quelques exemples d’API NER couramment utilisées :

### Natural Language Toolkit (NLTK)

NLTK est une plateforme [open source](https://www.ibm.com/fr-fr/think/topics/open-source) de premier plan pour la création de programmes [Python](https://www.ibm.com/fr-fr/think/topics/python-vs-r) qui permet de travailler avec des données en langage humain. Cet outil fournit des interfaces faciles à utiliser pour plus de 100 modèles d’extraction entraînés2. NLTK intègre aussi des bibliothèques de traitement de texte pour la classification, la tokenisation, la racinisation, l’étiquetage, l’analyse et le raisonnement sémantique. NLKT possède son propre classificateur pour la reconnaissance des entités nommées, appelé ne\_chunk, mais fournit également un encapsuleur qui permet d’utiliser l’étiqueteur Stanford NER en langage Python.

### Stanford Named Entity Recognizer

Développé par l’Université de Stanford, Stanford NER est une implémentation Java considérée comme la bibliothèque standard en matière d’extraction d’entités. Cet outil s’appuie sur les CRF et fournit des modèles pré-entraînés pour l’extraction d’entités nommées.

### SpaCy

Écrite en Python et réputée pour sa rapidité et sa facilité d’utilisation, SpaCy est une bibliothèque logicielle open source pour le NLP avancé. Elle s’appuie sur les toutes dernières recherches et a été conçue pour être employée avec de vrais produits. Elle dispose également d’un système statistique perfectionné qui permet aux utilisateurs de créer des extracteurs NER personnalisés.

## Applications de la NER

À mesure que les technologies continuent d’évoluer, les systèmes NER deviendront de plus en plus omniprésents et aideront les organisations à donner un sens aux données qu’elles rencontrent au quotidien. Cette technologie a déjà prouvé son utilité dans de nombreux secteurs, de la santé et la finance au service client, en passant par la cybersécurité.

Voici quelques-uns des cas d’utilisation ayant le plus d’impact :

Extraction d’informations

La NER est une première étape cruciale pour l’extraction d’informations utiles et structurées à partir de grandes bases de données non structurées. Les moteurs de recherche emploient la NER pour améliorer la pertinence et la précision de leurs résultats de recherche.

Agrégation automatisée des actualités

Les agrégateurs d’actualités utilisent la NER pour classer les articles en fonction des entités nommées qu’ils contiennent, pour présenter les actualités de manière plus organisée et plus efficace. La NER permet ainsi d’automatiser le processus de classification, en regroupant des articles similaires et en fournissant une vue plus complète des événements associés.

Surveillance des réseaux sociaux

Avec la prolifération des plateformes de réseaux sociaux, la quantité de données textuelles à analyser devient écrasante. La NER joue alors un rôle important, en identifiant les entités clés dans les publications et les commentaires pour comprendre les tendances et les opinions du public sur différents sujets (en particulier les opinions sur les marques et les produits). Ces informations aident les entreprises à analyser les sentiments, à élaborer des stratégies marketing, à rédiger des réponses dans le cadre du service client et à accélérer le développement des produits.

Chatbots et assistants virtuels

Les assistants virtuels et les chatbots basés sur l’IA générative utilisent la NER pour comprendre de manière précise les requêtes des utilisateurs et les demandes de support client. En identifiant les entités critiques dans les requêtes des utilisateurs, ces outils alimentés par l’IA peuvent fournir des réponses précises et adaptées au contexte. Par exemple, dans la requête « Trouver des restaurants italiens près du bois de Boulogne », la NER aide l’assistant à comprendre « italiens » comme un type de cuisine, « restaurants » comme un type d’établissement et « bois de Boulogne » comme un lieu.

Cybersécurité

Dans le domaine de la cybersécurité, la NER aide les entreprises à repérer les menaces potentielles et les anomalies dans les journaux de réseau et d’autres données liées à la sécurité. Par exemple, elle peut repérer les adresses IP, les URL, les noms d’utilisateur et les noms de fichier suspects dans les journaux de sécurité du réseau. La NER peut ainsi permettre d’approfondir les enquêtes sur les incidents de sécurité et améliorer la sécurité globale du réseau.

## Les défis liés à l’utilisation de la NER

La NER a fait du chemin depuis sa création. Elle intègre désormais des technologies innovantes et son utilisation s’est considérablement étendue. Cependant, il existe quelques défis notables à prendre en compte lors de l’évaluation de ces technologies.

Bien que de nombreux progrès aient été réalisés pour des langues comme l’anglais, la NER n’offre pas le même niveau de précision pour toutes les langues. Cela est principalement dû au manque de données étiquetées dans certaines langues. La NER multilingue, qui implique le transfert de connaissances d’une langue à une autre, est un domaine de recherche actif qui pourrait contribuer à combler ce fossé.

Certaines entités peuvent aussi être imbriquées dans d’autres, ce qui complique leur analyse. Par exemple, dans la phrase « Le campus Pierre-et-Marie-Curie de l’université de la Sorbonne été inauguré en 1959 », « université de la Sorbonne » et « Le campus Pierre-et-Marie-Curie de l’université de la Sorbonne » sont deux entités valides.

En outre, si les modèles de NER classiques sont capables d’identifier des entités courantes telles que les noms et les lieux, ils peuvent rencontrer des difficultés avec les entités propres à un domaine spécifique. Par exemple, dans le domaine médical, il peut être difficile d’identifier des termes complexes tels que des noms de maladies ou de médicaments. Des modèles de NER spécifiques peuvent être entraînés sur des données spécialisées, mais encore faut-il parvenir à rassembler ces informations.

Les modèles de NER peuvent également rencontrer des difficultés en cas d’ambiguïté (par exemple, « Apple » pourrait faire référence à un fruit, la pomme en anglais, ou à l’entreprise technologique), de variantes d’un nom d’entité (par exemple, « USA », « U.S.A. », « États-Unis » et « États-Unis d’Amérique » font tous référence au même pays) ou d’informations contextuelles limitées (car les textes et/ou les phrases ne contiennent pas suffisamment de contexte pour identifier et catégoriser les entités avec précision).

Bien que la NER présente certains défis, les avancées continues améliorent constamment sa précision et son applicabilité, et contribuent donc à minimiser l’impact des lacunes technologiques existantes.

## L’avenir de la NER

Même si la NER est un champ bien établi, il reste encore beaucoup de travail à faire.

L’un des domaines qui s’annonce prometteur est celui des techniques d’apprentissage non supervisé pour la NER. Même si les techniques d’apprentissage supervisé fonctionnent bien, elles nécessitent une grande quantité de données étiquetées, qui peuvent être difficiles à obtenir. Les techniques d’apprentissage non supervisé ne nécessitent pas de données étiquetées et peuvent aider les organisations à surmonter les défis liés à la disponibilité des données.

Une autre perspective intéressante est l’intégration de la NER à d’autres tâches de NLP. Par exemple, des modèles conjoints pour la NER et la liaison d’entités (qui consiste à relier des entités aux entrées correspondantes dans une base de connaissances), ou la NER et la résolution de coréférences (qui implique de déterminer quand deux ou plusieurs expressions d’un texte font référence à la même entité), pourraient donner naissance à des systèmes capables de mieux comprendre les textes et ainsi de pousser plus loin leur analyse.

L’apprentissage few-shot et la NER multimodale développent également les capacités des technologies NER. Avec l’apprentissage few-shot, les modèles sont entraînés pour effectuer des tâches avec seulement quelques exemples, ce qui peut s’avérer particulièrement utile lorsque les données étiquetées sont rares. La NER multimodale, quant à elle, suppose l’intégration de texte avec d’autres types d’entités. Une image ou un fichier audio, par exemple, peuvent fournir du contexte supplémentaire permettant de reconnaître des entités.

# II-Cas pratique de notre étude

# 1. Introduction

Ce projet a pour objectif d'implémenter et comparer plusieurs modèles de reconnaissance d’entités nommées (NER) sur des textes issus du corpus WikiNER. Nous avons appliqué des modèles pré-entraînés, converti les résultats au format BIO et évalué les performances.

# 2. Présentation du corpus

Le corpus utilisé est WikiNER en langue française, au format JSON. Il contient des annotations de type : PER (personne), ORG (organisation), LOC (lieu), MISC (autres).

Exemple d’annotation :

Phrase : Emmanuel Macron est président de la République française.  
Tags : [B-PER, I-PER, O, O, O, B-LOC, I-LOC]

# 3. Méthodologie

3.1 Pipeline général

- Chargement des données JSON  
- Sélection de phrases  
- Conversion au format BIO  
- Application des modèles spaCy, Stanza et CamemBERT  
- Évaluation avec sklearn.metrics.classification\_report

3.2 Modèles utilisés

Nous avons utilisé trois modèles différents :  
- spaCy (fr\_core\_news\_md)  
- Stanza (pipeline NER pour le français)  
- CamemBERT (Jean-Baptiste/camembert-ner sur HuggingFace)

# 4. Résultats et évaluation

4.1 Exemple de prédiction

Phrase : « Barack Obama est né à Honolulu. »  
Vraies étiquettes : [B-PER, I-PER, O, O, O, B-LOC]  
spaCy : [B-PER, I-PER, O, O, O, B-LOC]  
Stanza : [B-PER, I-PER, O, O, O, B-LOC]  
CamemBERT : [B-PER, I-PER, O, O, O, I-LOC] → erreur

4.2 Scores globaux

Modèle | Précision | Rappel | F1-score  
-------|-----------|--------|----------  
spaCy | 0.98 | 0.94 | 0.96  
Stanza | 0.97 | 0.93 | 0.95  
CamemBERT | 0.94 | 0.90 | 0.92

# 5. Analyse des résultats

Tous les modèles détectent correctement les entités principales (PER, LOC). spaCy est rapide et précis. Stanza est robuste pour les phrases complexes. CamemBERT est prometteur mais parfois moins précis pour les entités internes.

# 6. Difficultés rencontrées

- Format non-BIO du corpus initial  
- Alignement des prédictions des modèles avec les tokens d’origine  
- Nettoyage des données bruitées

# 7. Conclusion

Les modèles ont offert de bons résultats globaux. spaCy et Stanza ont bien fonctionné sans entraînement spécifique. CamemBERT serait plus performant avec un fine-tuning sur WikiNER.

# 8. Perspectives

- Fine-tuning de CamemBERT sur WikiNER  
- Interface utilisateur (ex. Streamlit)  
- Utilisation de modèles avec CRF pour affiner les transitions BIO

# Annexes : Extraits de code

Extrait : conversion au format BIO

def convert\_to\_bio(tokens, entities):  
 tags = ['O'] \* len(tokens)  
 for ent in entities:  
 start, end, label = ent  
 tags[start] = f'B-{label}'  
 for i in range(start+1, end):  
 tags[i] = f'I-{label}'  
 return tags