

## Bootstrap - Explication

*Ce projet explore plusieurs concepts importants du Natural Language Processing (NLP) et du machine learning. En utilisant des outils comme **spaCy** et les **Transformers**, nous pouvons construire des modèles sophistiqués capables de reconnaître des entités dans des textes. De plus, nous avons couvert les techniques d'évaluation et de visualisation des résultats qui sont essentielles pour s'assurer que le modèle est performant et interprétable*

### 1. Reconnaissance des Entités Nommées (NER)

#### a. Qu'est-ce que la NER ?

La **Reconnaissance des Entités Nommées** (NER, pour Named Entity Recognition) est une technique qui permet à un programme de trouver des mots spécifiques dans un texte, comme des noms de personnes, des lieux ou des organisations.

Par exemple, dans la phrase *"Marie travaille à Paris pour Google"*, le programme NER reconnaîtrait que "Marie" est une personne, "Paris" est un lieu, et "Google" est une organisation.

#### b. Applications de la NER :

La NER est utile dans plein de domaines :

- **Extraction d'informations** : Pour extraire automatiquement des données importantes dans des documents.
- **Recherche avancée** : Pour permettre à des moteurs de recherche de mieux comprendre les entités que l'on recherche.
- **Analyse de texte** : Comme dans des journaux ou des articles où il est utile de savoir qui sont les personnes ou les entreprises mentionnées.

### 2. Bibliothèques Utilisées : spaCy et Transformers

#### a. spaCy

**spaCy** est une bibliothèque qui aide à traiter du texte en langage naturel (comme du français ou de l'anglais). Elle est rapide et facile à utiliser pour des tâches comme la reconnaissance d'entités.

**Ce que fait spaCy :**

- **Modèles pré-entraînés** : spaCy a déjà des modèles entraînés qui peuvent reconnaître des entités comme des personnes, des lieux ou des organisations.
- **Pipeline de traitement** : spaCy divise le texte en plusieurs parties (comme les mots), puis passe le texte dans différents outils pour comprendre ce qu'il dit.

## b. Transformers

Les **Transformers** sont une technologie plus récente et très performante pour comprendre le texte. Un modèle très connu de Transformers est **BERT**. Ce modèle peut lire et comprendre des phrases et est très bon pour des tâches comme la NER.

Ce que font les Transformers :

- **Modèles NER** : Avec **Hugging Face**, tu peux utiliser des modèles pré-entraînés qui savent déjà reconnaître des entités comme des noms, des lieux, etc.
- **Personnalisation** : Tu peux entraîner ces modèles avec tes propres données pour améliorer les résultats si nécessaire.

## 3. Entraînement de Modèles NER avec spaCy

### a. Ajouter des Nouvelles Entités

Avec spaCy, il est possible d'ajouter de nouvelles catégories d'entités. Par exemple, tu peux entraîner un modèle pour reconnaître des lieux de départ et d'arrivée dans une phrase, comme "DEP" pour le départ et "ARR" pour l'arrivée.

### b. Comment entraîner un modèle avec spaCy ?

1. **Charger un modèle pré-entraîné** : On commence avec un modèle qui sait déjà reconnaître certaines entités.
2. **Ajouter de nouveaux labels** : Tu peux ajouter des étiquettes personnalisées pour que le modèle reconnaisse d'autres entités, comme des destinations de voyage.
3. **Entraînement avec des données annotées** : Tu utilises des exemples de phrases avec les entités marquées, puis tu entraînes le modèle avec ces données pour qu'il apprenne à reconnaître ces nouvelles entités.

### c. Exemple de Code

Le modèle est entraîné avec des phrases comme *"Je veux aller de Paris à Monaco"*, où "Paris" est l'entité de départ ("DEP") et "Monaco" est l'entité d'arrivée ("ARR"). Le modèle apprend à reconnaître ces entités à partir de ces exemples.

## 4. Évaluation de Modèles de Machine Learning

### a. Rapport de Classification

Le **rapport de classification** est un tableau qui montre la qualité des prédictions du modèle. Il affiche des métriques comme :

- **Précision** : Cela mesure combien des prédictions faites étaient correctes.
- **Rappel** : Cela mesure combien des entités réelles ont été trouvées par le modèle.
- **Score F1** : C'est un mélange de la précision et du rappel pour donner une mesure globale de la performance.

**Exemple :**

Un modèle NER pourrait être évalué en comparant les prédictions du modèle avec les vrais labels pour chaque entité.

### b. Exemple de Code

Le code utilise `classification_report` pour afficher un tableau qui montre les scores de précision, de rappel et le score F1 pour chaque catégorie d'entités (comme "DEP" et "ARR").

## 5. Visualisation des Métriques de Performance

### a. Visualisation avec Matplotlib

Pour mieux comprendre les performances du modèle au fil du temps, on utilise des graphiques avec **Matplotlib**. Cela permet de visualiser les scores comme la précision, le rappel, et le score F1 à chaque étape d'entraînement.

**Pourquoi c'est important ?**

Cela aide à voir si le modèle s'améliore avec le temps, ou s'il y a des problèmes pendant l'entraînement.

### b. Exemple de Code

La fonction `plot_metrics` prend des données sur les performances du modèle (par exemple, la précision à chaque époque) et trace un graphique. Cela permet de visualiser facilement les améliorations (ou les baisses) du modèle pendant l'entraînement.