Rapport sur l'Exploration des Méthodes de Reconnaissance d'Entités Nommées

Johanna Roll, Mouad Touzani, Mélanie Yakoub

Présentation du sujet

Ce rapport vise à explorer les méthodes de Reconnaissance d'Entités Nommées (NER), une tâche essentielle du traitement automatique du langage naturel (TALN) qui consiste à identifier et classifier des entités telles que les personnes, les lieux et les organisations dans des textes non structurés. Le NER, par son rôle crucial dans l'extraction précise des informations clés d'un texte, facilite la recherche d'informations, le traitement des langues multilingues, l'analyse de données textuelles et la génération de réponses pour les systèmes de Questions-Réponses. Il est également utilisé dans divers domaines spécialisés tels que la biomédecine, le droit et la finance, pour des applications variées. Ainsi, dans notre projet, nous évaluons plusieurs modèles pour la tâche NER, notamment BERT et DeBERTa, connus pour leur capacité à capturer le contexte des mots de manière bidirectionnelle. Notre benchmark comprend BERT, une ariante de BERT et un modèle qui associe BERT à un LSTM bidirectionnel pour la vectorisation et la classification des tokens.
Cette analyse comparative nous permettra d'évaluer leur efficacité dans la reconnaissance précise des entités nommées.

Présentation des données

Nous utilisons le jeu de données CoNLL-2003, qui contient des actualités de Reuters. Chaque observation est une phrase pré-tokenisée rédigé en anglais avec des étiquettes NER. Les étiquettes indiquent si un mot correspond à une entité nommée de type personne, organisation, localisation, ou divers, ou s'il ne correspond à aucune entité. Lorsque le mot d'une entité est composé, l'étiquette (B) est attribué au premier mot de cette entité et l'étiquette (I) pour les termes suivant dit à "l'interieur". Plus précisément la base de données est constitué de trois esembles pour évaluer les performances prédictives avec : un ensemble d'entrainement composé de 14,041 observations, un ensemble de validationn de 3,250 données et un ensemble de test de 3,453 données.

EDA

La distribution du nombre de mots par phrase est homogène dans tous les ensembles, avec environ 14 mots par phrase en moyenne. De plus, les différentes catégories NER présentent une longueur moyenne similaire en caractères à travers les trois ensembles, avec environ 5 caractères en moyenne.
En outre, un déséquilibre naturel en termes de classes est observé entre les catégories NER, en particulier en raison de la catégorie 0 qui regroupe les autres tokens, mais également parmi les autres catégories. Cette répartition est similaire entre les ensembles ; cependant, il convient de noter que l'ensemble de validation ne comporte pas d'entités nommées (NER) pour l'organisation.

Tokenisation

Nous avons utilisé un tokeniseur selon le modèle. La tokenisation avec BERT consiste à découper les mots en sous-unités plus petites appelées "subword tokens", généralement réalisée par la méthode WordPiece basée sur un critère probabiliste. Cette approche permet de préserver la signification des mots, même lorsqu'ils sont composés ou ambigus. De plus, des tokens spéciaux sont ajoutés pour marquer le début et la fin des phrases, ainsi que pour indiquer les séparateurs entre les phrases.
Cependant, une fois les mots tokenisés et découpés en plusieurs sous-mots pour que le modèle de vectorisation puisse les reconnaître dans le vocabulaire, cela peut entraîner des problèmes de dimensions. Cette situation est due à une inadéquation entre le nombre des étiquettes NER associées aux mots d'origine, qui n'a pas changé, et le nombre de tokens qui a augmenté avec le subword tokenizer, impliquant une division en sous-mots et l'ajout de tokens spéciaux. Ainsi, le nombre de tokens ne correspond plus au nombre d'étiquettes. Il est donc nécessaire de re-étiqueter nos données en fonction de ces changements, ce qui constitue l'objectif de notre fonction de prétraitement.

Pré-processing

Comme mentionné précédemment, une étape de néttoyage post-tokenisation est nécessaire pour la tâche de NER afin de résoudre les problèmes de dimensions. Cette étape implique l'attribution d'une étiquette de -100 à nos tokens spéciaux de début et fin de phrase (-100 étant un index ignoré dans la fonction de perte : entropie croisée), ainsi que l'ajout d'étiquettes supplémentaires pour les tokens divisés en sous-mots. Il faut ensuite également remplaçer les "B" par des "I" selon la position des tokens (lorsque l'étiquette n'est pas 0, c'est-à-dire "autre"). Cette approche nous permet d'assurer une correspondance correcte entre les tokens et leurs étiquettes NER.

Les modèles

DataCollator

Les données sont dynamiquement remplies pour uniformiser la taille des lots, avec des ensembles de 8 lots pour l'entraînement. Les données d'entraînement sont mélangées avant d'être chargées dans le DataLoader, tandis que les ensembles de validation et de test restent non mélangés.

Optimizer & Learning Rate Scheduler

L'optimiseur ajuste itérativement les poids du modèle en fonction du gradient de la fonction de perte pour minimiser l'erreur lors de l'entraînement. Dans notre cas, nous avons utilisé l'optimiseur Adam avec un learning rate de 2e-5.
Le learning rate Scheduler permet de modifier dynamiquement le taux d'apprentissage pendant l'entraînement afin d'optimiser la convergence du modèle vers un minimum global de la fonction de perte. Nous avons utilisé une méthode linéaire pour les ajustements et entraîné sur 3 epochs.

Accelerator

L'utilisation de la classe Accelerator permet de tirer parti des ressources matérielles disponibles pour accélérer le processus d'entraînement du modèle. En effet le modèle DeBERTa étant très lourd nous avons dû rajouter un Accelerator.

Post-Processing & Fine-Tuning

Le post-process permet de filtrer les étiquettes de "padding" et de les exclure des résultats finaux. En résumé, elle prépare les résultats du modèle pour une évaluation facile et compréhensible.

A la suite du post-processing nous avons fait le fine-tuning afin d'améliorer les performances des tâches de TALN en ajustant le modèle aux particularités spécifiques du domaine cible.

Évaluation

Après le fine-tuning nous procédons à l'évaluation de la performance des 3 modèles utilisés (BERT, DeBERTa et LSTM) en utilisant 4 métriques et la perte sur l'ensemble de test.

	BERT	DeBERTa	LSTM
Test Loss	0.1151	0.1064	0.1056
Precision	0.9180	0.9347	0.9193
Recall	0.9050	0.9224	0.9044
F1	0.9115	0.9285	0.9118
Accuracy	0.9827	0.9856	0.9822

Table 1: Résultats sur l'ensemble test

D'après nos résultats, le modèle DeBERTa s'est avéré être le meilleur avec nos 4 métriques les plus élevées. En outre, l'ajout de la couche LSTM au modèle BERT n'a pas significativement améliorer la capacité prédictif du modèle.