TETIS @ Challenge TextMine 2024 : "Reconnaissance d'entités géographiques dans un corpus des Instructions nautiques"

Rémy Decoupes, Roberto Interdonato, Rodrique Kafando, Mathieu Roche, Mehtab Alam Syed, Maguelonne Teisseire, Sarah Valentin

TETIS @ Challenge TextMine 2024 : "Reconnaissance d'entités géographiques dans un corpus des Instructions nautiques"

Rémy Decoupes[*,**] Roberto Interdonato[*,***]
Rodrique Kafando[*,**] Mathieu Roche[*,***] Mehtab Alam Syed [*,***] Maguelonne Teisseire[*,**] Sarah Valentin[*,***]

[*]TETIS, Univ. Montpellier, AgroParisTech, CIRAD, CNRS, INRAE, Montpellier 34090, France
[**]INRAE, F-34398 Montpellier, France
[***]CIRAD, F-34398 Montpellier, France

1 Introduction

Notre équipe, issue de l'unité TETIS (Territoires, environnement, télédétection et information spatiale), s'intéresse à différents tâches du domaine du Traitement Automatique du Langage Naturel (TALN) en se concentrant sur les problématiques associées à l'information spatiale. La reconnaissance des entités spatiales dans les données textuelles représente une étape importante dans la majorité des chaînes de traitements. Aussi, les données relatives à l'information spatiale nécessitent une attention particulière. Par exemple, nos travaux proposent (i) des méthodes d'extraction d'information spatiale à partir de données non standards (Zenasni et al., 2018), (ii) des méthodes et outils de *geocoding* et désambiguïsation d'entités spatiales (Syed et al., 2023; Kafando et al., 2023), (iii) des algorithmes d'augmentation de données fondés sur l'information géographique pour l'entraînement de modèles de type Transformers (Decoupes et al., 2023).

La participation de notre équipe TETIS au défi TextMine 2024 (Guille, 2023) est dans la continuité de ces travaux. Nous résumons notre approche dans la section suivante et mettons à disposition nos codes [1] et modèles [2].

2 Approche

L'approche a consisté à comparer différents modèles pré-entraînés ajustés sur les données du défi. Afin de commenter les différents résultats, nous présentons la distribution du jeu de données.

1. Dépôt logiciel : `https://github.com/tetis-nlp/tetis-challenge_textmine_2024`
2. HuggingFace hub : `https://huggingface.co/rdecoupes/tetis-textmine-2024-camembert-large-based`

2.1 Données d'apprentissage

Le jeu de données d'apprentissage est extrêmement déséquilibré (Figure 1) : les exemples associés à des labels géographiques (*geogFeat*, *geogName*, *name GeogName* et *geogFeat GeogName*) ne représentent que 18% (n=7194/39857) des données labellisées.

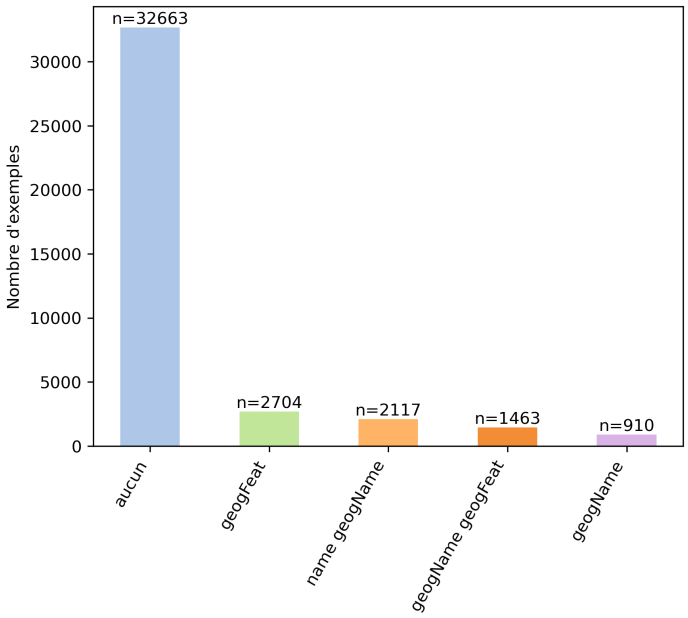

FIG. 1 – Distribution du nombre d'exemples par classe dans le jeu de données d'entraînement

La classe *geogFeat* se distingue des autres par l'absence de noms propres. En effet, les mots ou groupes de mots annotés avec ce label désignent une caractéristique géographique sans nom propre comme *phare* ou *port*. Nous avons converti la tâche de classification multi-labels en une classification simple, en considérant les multi-classes *name GeogName* et *geogFeat GeogName* comme des labels à part entière. Pour la suite des expérimentations, le jeu de données a été séparé en phrases.

2.2 Méthodes et résultats

Dans un premier temps, nous avons ré-entraîné un modèle en langue française proposé par la librairie spaCy (pipeline *fr core news lg* [3]). Le micro F1-score sur le jeu de données de validation (20 % du jeu de données mis à disposition) et celui calculé par la plateforme Kaggle sont de 0.950 et de 0.927, respectivement. Ceci a constitué notre base de référence.

Notre deuxième approche a visé à ré-ajuster des modèles de type Transformers pré-entraînés (*fine-tuning* en anglais). Nous avons comparé trois modèles (Figure 2) : deux modèles en

3. https://github.com/explosion/spacy-models/releases/tag/fr_core_news_lg-3.7.0

langue française (CamemBERT-base et CamemBERT-large) et un modèle multi-langues (XLM-RoBERTa). Sans optimisation des hyperparamètres, CamemBERT-large offre, sur le jeu de données calculé par la plateforme Kaggle, les meilleurs résultats. Nous avons ensuite fait varier certains hyperparamètres tels que les tailles de batch d'entraînement ($batch\ size$), le taux d'apprentissage ($learning\ rate$) et le nombre d'epochs ($epochs$) sur plusieurs entraînements différents. La meilleure combinaison d'hyperparamètres sur le jeu de données de validation est : $batch\ size = 16$, $learning\ rate = 1e-05$, $epochs = 10$, alors que pour les données calculées par la plateforme Kaggle, la meilleure combinaison est : $batch\ size = 8$, $learning\ rate = 2e-05$, $epochs = 10$ (micro F1-score : 0.978). Nous avons, cependant, constaté une grande variabilité entre deux entraînements ayant des hyperparamètres identiques (Figure 2).

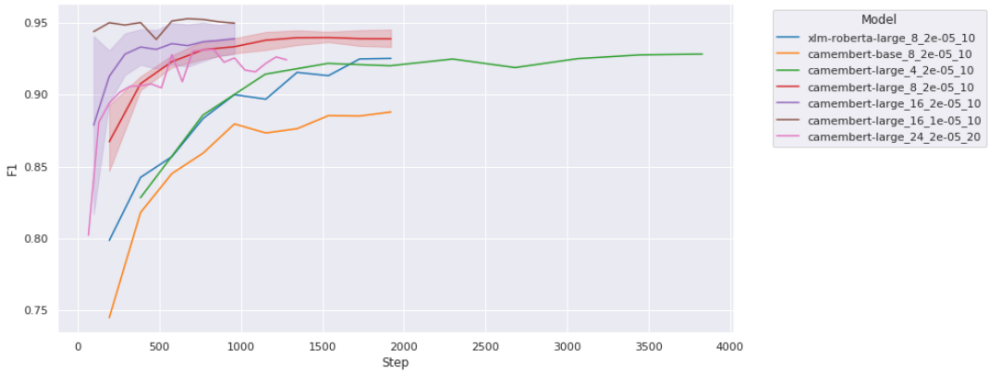

FIG. 2 – Comparaison des F1-score pendant l'entraînement des 3 modèles avec différents hyperparamètres. Les traces sous forme d'enveloppe correspondent aux écarts types des différents entraînements ayant des hyperparamètres identiques. La règle de nommage est : le nom du modèle pré-entraîné, le $batch\ size$, le $learning\ rate$ et le nombre $epochs$

Sur les données de validation, i.e. 20% du jeu de données mis à disposition, les performances du modèle issu du modèle spaCy sont très inégales : la classe *aucun* obtient le meilleur F1-score (0.978), tandis que le F1-score des autres classes ne dépassent pas 0.877 (Tableau 1). Ces performances modérées sont dues à une confusion des instances géographiques avec la classe *aucun*, ce qui diminue leur rappel (Figure 3). La classe *geogFeat*, qui obtient le score le plus bas, souffre également d'un manque de précision. La diminution des performances globales du modèle sur le jeu de données test indique un probable sur-apprentissage avec un biais vers la classe sur-représentée.

TAB. 1 – Comparaison des F1-scores par classe en fonction du modèle pré-entrainé

Modèle	**moyenne pondérée *micro***	**aucun**	**geogFeat**	**name geogName**	**geogFeat geogName**	**geogName**
spaCy	**0.950**	**0.978**	0.779	0.877	0.823	0.804
xlm-roBERTa-large	0.926	0.927	0.914	0.932	0.931	0.914
CamemBERT-large	**0.950**	0.947	**0.915**	**0.969**	**0.980**	**0.971**

Les modèles de type transformers, bien que moins performants pour la classe *aucun*, obtiennent de meilleurs résultats pour les classes géographiques. De plus, leurs performances sur

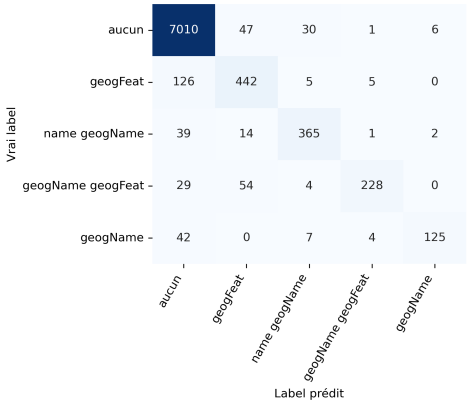

FIG. 3 – Matrice de confusion obtenue à partir du modèle spaCy

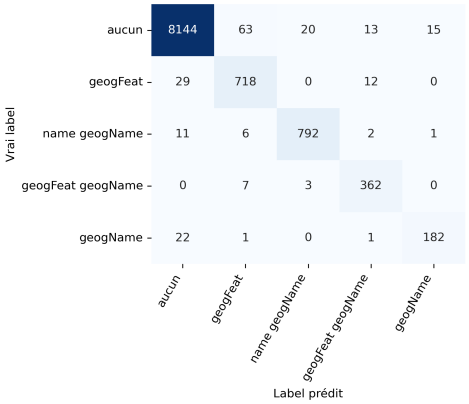

FIG. 4 – Matrice de confusion obtenue à partir du modèle CamemBERT-large

le jeu de validation sont égales voire supérieures aux performances sur le jeu d'entraînement, indiquant une meilleure généralisabilité que le modèle spaCy. Cependant ils rencontrent également des difficultés avec la classe *geogFeat* comme illustré par le Tableau 1. Cette classe est complexe à distinguer de *aucun* (Figure 4). Les modèles détectent plus efficacement les classes contenant un nom propre i.e. *geogName*, *name geogName* et *geogFeat geogName*.

Références

Decoupes, R., M. Roche, et M. Teisseire (2023). GeoNLPlify : A spatial data augmentation enhancing text classification for crisis monitoring. *Intelligent Data Analysis*, 1–25.

Guille, A. (2023). Défi TextMine 2024. Kaggle. `https://kaggle.com/competitions/defi-textmine-2024`.

Kafando, R., R. Decoupes, M. Roche, et M. Teisseire (2023). SNEToolkit : Spatial named entities disambiguation toolkit. *SoftwareX 23*, 101480.

Syed, M. A., E. Arsevska, M. Roche, et M. Teisseire (2023). GeospatRE : extraction and geocoding of spatial relation entities in textual documents. *Cartography and Geographic Information Science*, 1–16.

Zenasni, S., E. Kergosien, M. Roche, et M. Teisseire (2018). Spatial information extraction from short messages. *Expert Systems with Applications 95*, 351–367.