Florent Giauna
Master AMSD/MLSD
Université Paris Cité

Détection d’entités nommées géographiques et Parameter-Efficient Fine-Tuning

La représentation de Marseille dans les chansons de JuL

Introduction

La détection d'entités nommées, Named Entity Recognition (NER) en anglais, est un problème de traitement du langage naturel, Natural Language Processing (NLP) en anglais. La NER consiste à extraire d'un texte des entités, c'est-à-dire des mots ou des expressions, et à leurs attribuer des catégories représentant un concept tel qu'une personne, une organisation ou une localisation (Figure 1).

Figure 1 – Exemple de détection d'entités nommées ou Named Entity Recognition (NER)

En géographie, la NER est ainsi utilisée pour d'extraire de l'information géographique de données non structurées. L'apprentissage automatique, *machine learning* en anglais, et son sous domaine, l'apprentissage profond, *deep learning* en anglais, suscitent l'intérêt de la discipline afin d'extraire de l'information de données de plus en plus volumineuses. C'est le cas pour les données textuelles avec la récupération d'information géographique contenue dans des articles Wikipedia[1], dans la presse contemporaine[2], dans des sources historiques[3] ou encore sur les réseaux sociaux[4].

Dans la continuité de ces travaux nous proposons d'entraîner un modèle pour la reconnaissance d'entités nommées géographiques adapté à un type de corpus non encore étudié : celui des paroles de chansons et spécifiquement des chansons de rap. L'intérêt est de parvenir à extraire des connaissances de bases documentaires non traditionnelles. Cela peut servir en sciences sociales afin d'étudier la représentation de l'espace d'une classe sociale particulière.

Notre corpus de base n'étant pas étiqueté et le temps imparti étant restreint, nous allons annoter une partie du corpus et travailler avec un petit jeu de données. Or les gros modèles de langage ont besoin de beaucoup de données pour s'adapter. Les méthodes Parameter-Efficient Fine-Tuning (PEFT) ont justement été proposées pour améliorer l'efficacité des méthodes de fine-tuning dans des contextes de données et de ressources limitées. Nous proposons de tester leurs performances sur notre corpus et de les comparer avec d'autres techniques.

1 Cillian Berragan, Alex Singleton, Alessia Calafiore & Jeremy Morley (2023) Transformer based named entity recognition for place name extraction from unstructured text, International Journal of Geographical Information Science, 37:4, 747-766, DOI: 10.1080/13658816.2022.2133125

2 Alejandro Molina-Villegas, Victor Muñiz-Sanchez, Jean Arreola-Trapala, Filomeno Alcántara, Geographic Named Entity Recognition and Disambiguation in Mexican News using word embeddings, Expert Systems with Applications, Volume 176, 2021, 114855, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2021.114855.

3 Katherine McDonough, Ludovic Moncla & Matje van de Camp (2019) Named entity recognition goes to old regime France: geographic text analysis for early modern French corpora, International Journal of Geographical Information Science, 33:12, 2498-2522, DOI: 10.1080/13658816.2019.1620235

4 Tao, Liufeng & Xie, Zhong & Xu, Dexin & Ma, Kai & Qiu, Qinjun & Pan, Shengyong & Huang, Bo. (2022). Geographic Named Entity Recognition by Employing Natural Language Processing and an Improved BERT Model. ISPRS International Journal of Geo-Information. 11. 598. 10.3390/ijgi11120598.

I – Corpus

Le choix du corpus s'est porté sur JuL, un rappeur français originaire du quartier de Saint-Jean-du-Désert dans le 12ème arrondissement de Marseille. Trois raisons motivent ce choix.

La première est la place qu'occupe ce rappeur dans le paysage sonore et social français. En effet en 2020, il est devenu le plus gros vendeur de disque de l'histoire du rap français, entre 2010 et 2020 il était le deuxième vendeur de disque toutes catégories confondues, derrière Johnny Hallyday, et il est l'artiste français le plus écouté en streaming. Ses textes font donc écho à une grande partie de la population qui s'y retrouve.

La deuxième est le volume de sa discographie : 29 albums, certains de plus de quarante titres, et de très nombreux morceaux exclusifs produits à l'occasion d'émissions radio. Parmi ces derniers plusieurs titres durent plus de 10 minutes sans refrain et deux titres durent même plus de 30 et 40 minutes. Au total notre corpus est constitué de 835 titres et 54 493 vers.

La troisième est que le style de JuL est caractéristique du rap marseillais. Ce style est très descriptif et raconte des scènes de la vie de tous les jours, par exemple : « Je dis toujours ce que je vois, d'ailleurs, là je suis dans les bouchons Avenue Saint-Antoine[5] ». Au travers de ses textes il cite ainsi de nombreux lieux de Marseille, qui sont intéressant pour connaître la représentation de ce territoire et qui constituent autant d'entités nommées géographiques à reconnaître.

II – Difficultés rencontrées

La première des difficultés a été la constitution du corpus puisqu'il n'existe pas de jeu de données prêt à l'emploi. Les paroles ont été collectées par *web scraping*, une technique d'extraction du contenu de sites Internet. Elles ont été récupérées sur le site Genius à l'aide d'un script rédigé en R. Une première limite est donc liée à cette source. Genius est un site collaboratif où ce sont les utilisateurs qui retranscrivent et annotent les paroles. Ainsi les erreurs ou fautes des utilisateurs sont possibles. Cependant un système de modération limite ces erreurs. Il a ensuite fallu filtrer les chansons pour ne conserver que les paroles chantées par JuL et retirer les parties chantées par d'autres artistes à l'occasion de collaboration (*featuring*).

Artiste très prolifique, JuL, a sorti un album au début de notre travail et un autre pendant. Les paroles de ce dernier album non récupérées dans le corpus initial ont été utilisées pour tester notre modèle sur de nouvelles données et évaluer sa robustesse au changement (*data drift* et *concept drift*).

Une autre difficulté liée à la constitution du jeu de donnée a été l'absence d'étiquetage. En effet, aucune catégorie n'était associée aux paroles récupérées. Le volume du corpus et le temps imparti ne permettait pas un étiquetage manuel de la totalité des entités. Nous nous sommes donc servi d'un modèle pré-existant pour générer des étiquettes. Les résultats de NER peuvent être exploités en amont, pour le pré-traitement. Cela n'a pas été trivial car il a fallu jongler entre différents formats (csv, DataFrame, txt au format IOB, Dataset HuggingFace) sans perdre les associations entité-

5 JuL, Freestyle "Têtes cassées" [Part 8]

étiquette. Évidemment les erreurs des utilisateurs sont une source d'erreur et ainsi on peut obtenir l'entité « Ventiny », étiquetée comme une personne, en lieu et place de « Ventimille » une localisation[6]. Une correction manuelle a été nécessaire.

Un autre problème est l'utilisation abondante de l'argot et du verlan. C'est une langue d'avantage parlée que écrite, et un même mot peut avoir plusieurs orthographes : le verlan de « quartier » peut être écrit « tiers-quar » ou « tiekar ». La pratique propre à un groupe de renommer les lieux afin de se les approprier est un autre exemple, comme la ville de Saint-Jean-du-Désert rebaptisée « Saint-Jean-La-Puenta ».

En ce qui concerne les entités elles-mêmes, nous nous concentrons sur la catégorie localisation, relatif à un lieu. Cependant la performance sur les autres étiquettes joue puisque à titre d'exemple l'entité « Pablo Picasso » peut faire référence à une personne mais aussi à un quartier de la ville de Nanterre. A l'inverse, Tokyo est évidemment la capitale du Japon mais également le nom d'un personnage de la série *La Casa de Papel* très cité par JuL. Dans cet ordre d'idée ce qui est bien une localisation peut ne pas faire référence à un lieu comme être dans une voiture (Clio, RS6, BM, etc.) ou sur un deux roues (TDM, T-Max etc.). Ces entités ont tendance a être catégorisées comme localisation mais nous avons pris de la parti de les considérer comme des organisations étant donné qu'elles font plutôt référence à des marques. En revanche nous considérons être « au tram » ou « sur la place » comme une localisation.

III – Constitution du corpus

Concernant la constitution des jeux de données, seul 20 % des 54 493 vers récupérés ont été annotés soit 10898 vers.

Les prédictions du modèle Camembert-ner ont permis d'obtenir des étiquettes de catégorie, puis une opération de correction manuelle a du être effectuée. En effet les entités ou les catégories prédites n'étaient pas forcément bonnes. Nous avons donc corrigé des exemples contradictoires, *adversarial examples* en anglais. Ce procédé est parfois nommé *Hard Mining* dans la littérature.

Un *adversarial example* est une observation difficile à comprendre pour le modèle[7]. Il est possible de l'améliorer en corrigeant spécifiquement ces observations mal prédites. Par exemple, dans notre cas le modèle n'étiquetait pas les entités « 135 », « 1-3-5 », « 1.3.5 » ou « un-trois-cinq » faisant référence au code postal de Saint-Jean-du-Désert. Les vers annotés ont ensuite été découpés classiquement : 60 % des données ont été allouées au jeu d'entraînement, 20 % au jeu de validation et 20 % au jeu de test.

Au final le jeu d'entraînement comporte 590 entités de la catégorie localisation, le jeu de validation 195 et le jeu test 212.

6 JuL, Freestyle "Têtes cassées" [Part.6]

7 Hendrycks, Dan, et al. "Natural adversarial examples." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2021.

IV – Modèles comparés

La plupart des modèles de langage prêt à l'emploi et notamment les modèles NER ont été entraînés sur le corpus de l'encyclopédie participative Wikipedia. Cependant la manière dont JuL s'exprime n'est pas la même que celle des contributeurs de Wikipedia

Une intuition a motivé notre démarche : l'importance du contexte de notre corpus. L'architecture Transformer, reposant sur le mécanisme d'attention[8] permis par l'utilisation d'encodeurs et de décodeurs, semblait la plus adaptée. L'attention fait référence au mécanisme qui permet de se concentrer sur un élément ou quelques éléments à la fois, c'est-à-dire le mot important dans chaque phrase. Le processus est non directionnel parce que ces modèles regardent tout le texte en même temps. Cette intuition doit être confrontée à un modèle d'une architecture différente.

Le problème est que chaque framework utilise une tokenization différente. Par exemple Flair, spaCy et Camembert utilisent trois tokenization différentes. Or nous générons une partie des étiquettes avec Camembert-ner et cette tokenization n'est pas compatible avec les formats attendus par Flair et spaCy. Nous voulions que Flair et spaCy soient nos baselines mais la tokenization étant différente avec Camembert, il n'est pas possible d'avoir une base de comparaison strictement similaire. D'autant que la tokenization implémentée joue un rôle important dans la prédiction des catégories NER et donc des résultats. Nous avons donc défini Camembert-ner comme notre baseline et nous avons comparé quelques inférences avec le meilleur modèle français de spaCy (fr_core_news_lg).

Nous avons choisi de tester Camembert-ner. CamemBERT[9] est un modèle à l'état de l'art pour le français basé sur l'architecture RoBERTa. Camembert-ner est sa version optimisé pour la tâche de NER, pré-entraîné sur WikiNER-fr un corpus de Wikipedia en Français. Il faudra donc l'affiner (*fine-tuning*) sur notre corpus en évaluant ses performances avec différents hyper-paramètres.

Le problème du fine-tuning est qu'il faut stocker une copie des paramètres du modèle pour chaque tâche. Cela peut être très coûteux en ressource car les plus gros modèles comptes plusieurs milliards de paramètres. Une solution est celle des méthodes Parameter-Efficient Fine-Tuning[10] (PEFT). Elles ont justement été proposées pour améliorer l'efficacité des méthodes de fine-tuning dans des contextes de données et ressources limitées. Elles sont basées sur l'hypothèse que les couches supérieures du modèle sont plus sensibles aux différences entre les tâches, tandis que les couches inférieures sont plus sensibles aux caractéristiques générales du langage. Les méthodes PEFT consistent à figer les couches inférieures du modèle et à ne fine-tuner que les couches supérieures, celles pour la tâche spécifique. Cette approche permet de réduire considérablement le nombre de paramètres à entraîner, ce qui peut améliorer les performances du modèle tout en réduisant les besoins en puissance de calcul et en mémoire. Nous proposons d'étudier trois méthodes PEFT : Adapter, LoRA et le *prefix-tuning*.

8 Vaswani, Ashish, et al. "Attention is all you need." *Advances in neural information processing systems* 30 (2017).
9 Martin, Louis, et al. "CamemBERT: a tasty French language model." *arXiv preprint arXiv:1911.03894* (2019).
10 https://huggingface.co/blog/peft

Les *adaptaters*[11] sont une alternative légère au *fine-tuning* d'un modèle complet, consistant seulement en un ensemble réduit de paramètres nouvellement introduits à chaque couche du transformer. Ils sont efficaces en termes de paramètres, ils accélèrent les itérations d'entraînement et ils sont partageables et modifiables en raison de leur modularité et de leur taille compacte. Tout en maintenant des performances généralement comparables à celles de l'état de l'art du fine-tuning.

En réduisant la dimensionnalité des matrices de poids, la méthode LoRA (Low rank adaptation of large language models)[12] permet de réduire le nombre de paramètres du modèle, tout en préservant sa capacité à capturer les relations importantes entre les mots dans le texte.

Parmi les avancées les plus récentes en PEFT figure l'affinage de préfixe (*prefix-tuning*)[13]. Il s'agit de geler les paramètres du Transformer et de n'optimiser que le préfixe. Le préfixe est une séquence de vecteurs spécifiques à une tâche et correspond à une instruction. Avec le préfixe il est ainsi en quelque sorte possible de personnaliser le modèle pour l'adapter à une utilisation particulière sur un jeu de donnée spécifique.

IV – Résultats

La comparaison des résultats obtenus par les différents modèles expérimentés sur un GPU gratuit de Google Colab, sur 10 epochs indique que la méthode PEFT de Prefix-tuning obtient le meilleur F1 score et Recall sur la catégorie LOC. Le modèle qui obtient la meilleur précision est le Camembert-ner fine-tuned (Tableau 1). Le Prefix-tuning obtient même les meilleurs résultats sur toutes les catégories. (Tableau 2). Le temps d'entraînement est considérablement réduit et le nombre de paramètres est 120 fois moindre que pour le deuxième meilleur modèle, le Camembert-ner fine-tuned (Tableau 3). On constate également sur quelques exemples de paroles de JuL n'ont présentes dans notre corpus d'entraînement ou de test que Camembert-ner fine-tuned et Camembert avec Prefix-tuning obtiennent les résultats attendus (Annexe 1).

Tableau 1 – Comparaison des résultats des modèles sur la catégorie localisation (LOC)

Modèle	Precision	Recall	F1 score
Camembert-ner	0.308	0.287	0.297
Camembert-ner fine-tuned	**0.731**	0.755	0.743
Camembert-ner + Prefix-tuning	0.723	**0.773**	**0.747**
Camembert-ner + LoRA	0.589	0.63	0.609
Camembert-ner + Adapter	0.681	0.741	0.71

11 Pfeiffer, J., Rücklé, A., Poth, C., Kamath, A., Vulic, I., Ruder, S., Cho, K., & Gurevych, I. (2020). AdapterHub: A Framework for Adapting Transformers. ArXiv, abs/2007.07779.

12 Hu, Edward J., et al. "Lora: Low-rank adaptation of large language models." *arXiv preprint arXiv:2106.09685* (2021).

13 Li, Xiang Lisa, and Percy Liang. "Prefix-tuning: Optimizing continuous prompts for generation." arXiv preprint arXiv:2101.00190 (2021).

Tableau 2 – Comparaison des résultats des modèles toutes catégories confondues (LOC, PER, MISC, ORG)

Modèle	Precision	Recall	F1 score
Camembert-ner	0.161	0.308	0.211
Camembert-ner fine-tuned	0.771	0.761	0.766
Camembert-ner + Prefix Tuning	**0.781**	**0.77**	**0.775**
Camembert-ner + LoRA	0.673	0.664	0.669
Camembert-ner + Adapter	0.731	0.739	0.735

Tableau 3 – Comparaison du temps d'entraînement et du nombre de paramètres des modèles

Modèle	Train runtime (seconds)	Number of trainable parameters
Camembert-ner fine-tuned	836,0618	~110 000 000
Camembert-ner + Prefix Tuning	367,8535	921 600
Camembert-ner + LoRA	**334,2136**	**298 757**
Camembert-ner + Adapter	338,7567	2 383 109

Conclusion

La NER est donc utile pour extraire des connaissances de bases documentaires non traditionnelles et peut être particulièrement utilisée en géographie pour étudier la représentation de l'espace d'une classe sociale particulière. La NER peut également être utile pour la création de cartes en géocodant les entités géographiques détectées. Des cartographies quantitatives pour représenter les lieux dont parle le plus un auteur, ou des cartographies qualitatives utilisant une analyse de sentiment pour représenter la perception associée à un lieu à l'aide d'un code couleur sont envisageables.

L'intérêt des méthodes PEFT sur un petit jeu de données et avec des ressources limitées. Sur notre jeu de données les meilleurs scores ont été obtenu avec la méthode du *Prefix-tuning*. Il serait intéressant de voir si cela est généralisable à d'autres corpus atypiques.

L'idée de départ était de générer un texte expliquant la manière dont l'auteur parle d'un lieu mais le temps imparti n'a pas permis de réaliser ce projet.

Annexe 1 – Comparaison des inférences sur quelques exemples

1) « C'est Jul , Saint Jean la Puenta mon pote ». Résultat attendu : Jul = personne [PER], Saint Jean la Puenta = localisation [LOC]

<table>
<tr><th>Modèle</th><th>Résultat</th></tr>
<tr><td>spaCy (fr_core_news_lg)</td><td>C'est Jul PER , Saint Jean la Puenta PER mon pote
JSON Output Maximize
<pre>[
  {
    "entity_group": "PER",
    "word": "Jul",
    "start": 6,
    "end": 9,
    "score": 1
  },
  {
    "entity_group": "PER",
    "word": "Saint Jean la Puenta",
    "start": 12,
    "end": 32,
    "score": 1
  }
]</pre></td></tr>
<tr><td>Camembert-ner</td><td>C'est Jul PER , Saint Jean la Puenta PER mon pote
JSON Output Maximize
<pre>[
  {
    "entity_group": "PER",
    "score": 0.9944803714752197,
    "word": "Jul",
    "start": 5,
    "end": 9
  },
  {
    "entity_group": "PER",
    "score": 0.9975001215934753,
    "word": "Saint Jean la Puenta",
    "start": 11,
    "end": 32
  }
]</pre></td></tr>
<tr><td>Camembert-ner

finetuned</td><td>C'est Jul PER , Saint Jean la Puenta LOC mon pote
JSON Output Maximize
<pre>[
  {
    "entity_group": "PER",
    "score": 0.998735785484314,
    "word": "Jul",
    "start": 5,
    "end": 9
  },
  {
    "entity_group": "LOC",
    "score": 0.9870782494544983,
    "word": "Saint Jean la Puenta",
    "start": 11,
    "end": 32
  }
]</pre></td></tr>
<tr><td>Camembert-ner
+ Prefix Tuning</td><td><pre>[{'entity': 'I-PER', 'score': 0.99805605, 'index': 4, 'word': '▁J', 'start': 5, 'end': 7},
 {'entity': 'I-PER', 'score': 0.9980519, 'index': 5, 'word': 'ul', 'start': 7, 'end': 9},
 {'entity': 'I-LOC', 'score': 0.709264, 'index': 8, 'word': '▁Saint', 'start': 11, 'end': 17},
 {'entity': 'I-LOC','score': 0.7955982, 'index': 9, 'word': '▁Jean', 'start': 17, 'end': 22},
 {'entity': 'I-LOC', 'score': 0.88177663, 'index': 10, 'word': '▁la', 'start': 22, 'end': 25},
 {'entity': 'I-LOC', 'score': 0.93947303, 'index': 11, 'word': '▁Pu', 'start': 25,'end': 28},
 {'entity': 'I-LOC', 'score': 0.94713134, 'index': 12, 'word': 'ent', 'start': 28, 'end': 31},
 {'entity': 'I-LOC', 'score': 0.9483872, 'index': 13, 'word': 'a', 'start': 31, 'end': 32}]</pre></td></tr>
</table>

2) « 1.3.5, l'ovni a l'flow de la tess ». Résultat attendu : 1.3.5 = localisation [LOC], ovni = personne [PER], tess = localisation [LOC]

Modèle	Résultat
spaCy (fr_core_news_lg)	No token was detected Computation time on Intel Xeon 3rd Gen Scalable cpu: 0.016 s JSON Output Maximize []
Camembert-ner	1.3.5, l'ovni a l'flow de la tess MISC JSON Output Maximize [{ "entity_group": "MISC", "score": 0.4477802515029907, "word": "tess", "start": 28, "end": 33 }]
Camembert-ner finetuned	1.3.5, LOC l'ovni PER a l'flow de la tess LOC JSON Output Maximize [{ "entity_group": "LOC", "score": 0.9942300319671631, "word": "1.3.5,", "start": 0, "end": 6 }, { "entity_group": "PER", "score": 0.9885226488113403, "word": "l'ovni", "start": 6, "end": 13 }, { "entity_group": "LOC", "score": 0.9956393241882324, "word": "tess", "start": 28, "end": 33 }]
Camembert-ner + Prefix Tuning	[{'entity': 'I-LOC', 'score': 0.9830177, 'index': 1, 'word': '_1.', 'start': 0, 'end': 2}, {'entity': 'I-LOC', 'score': 0.9867018, 'index': 2, 'word': '3.5', 'start': 2, 'end': 5}, {'entity': 'I-LOC', 'score': 0.9863934, 'index': 3, 'word': ',', 'start': 5, 'end': 6}, {'entity': 'I-PER', 'score': 0.9120515, 'index': 4, 'word': '_l', 'start': 6, 'end': 8}, {'entity': 'I-PER', 'score': 0.924347, 'index': 5, 'word': "'", 'start': 8, 'end': 9}, {'entity': 'I-PER', 'score': 0.94280994, 'index': 6, 'word': 'ov', 'start': 9, 'end': 11}, {'entity': 'I-PER', 'score': 0.94391835, 'index': 7, 'word': 'ni', 'start': 11, 'end': 13}, {'entity': 'I-LOC', 'score': 0.5509928, 'index': 14, 'word': '_tes', 'start': 28, 'end': 32}, {'entity': 'I-LOC', 'score': 0.550422, 'index': 15, 'word': 's', 'start': 32, 'end': 33}]

3) « Les gens ils restent au tiek parce que c'est trop cher à Cannes ». Résultat attendu : tiek = localisation [LOC], Cannes = localisation [LOC]

Modèle	Résultat
spaCy (fr_core_news_lg)	Les gens ils restent au tiek parce que c'est trop cher à Cannes MISC JSON Output Maximize `[ { "entity_group": "MISC", "word": "Cannes", "start": 57, "end": 63, "score": 1 } ]`
Camembert-ner	Les gens ils restent au tiek parce que c'est trop cher à Cannes MISC JSON Output Maximize `[ { "entity_group": "MISC", "score": 0.7080168724060059, "word": "Cannes", "start": 56, "end": 63 } ]`
Camembert-ner finetuned	Les gens ils restent au tiek LOC parce que c'est trop cher à Cannes MISC JSON Output Maximize `[ { "entity_group": "LOC", "score": 0.9971707463264465, "word": "tiek", "start": 23, "end": 28 }, { "entity_group": "MISC", "score": 0.9945573806762695, "word": "Cannes", "start": 56, "end": 63 } ]`
Camembert-ner + Prefix Tuning	`[{'entity': 'I-LOC', 'score': 0.99123317, 'index': 6, 'word': '_ti', 'start': 23, 'end': 26}, {'entity': 'I-LOC', 'score': 0.9928318, 'index': 7, 'word': 'ek', 'start': 26, 'end': 28}, {'entity': 'I-MISC', 'score': 0.9872465, 'index': 16, 'word': '_Cannes', 'start': 56, 'end': 63}]`

Bibliographie

Alejandro Molina-Villegas, Victor Muñiz-Sanchez, Jean Arreola-Trapala, Filomeno Alcántara, Geographic Named Entity Recognition and Disambiguation in Mexican News using word embeddings, Expert Systems with Applications, Volume 176, 2021, 114855, ISSN 0957-4174, https://doi.org/10.1016/j.eswa.2021.114855

Cillian Berragan, Alex Singleton, Alessia Calafiore & Jeremy Morley (2023) Transformer based named entity recognition for place name extraction from unstructured text, International Journal of Geographical Information Science, 37:4, 747-766, DOI: 10.1080/13658816.2022.2133125

Hendrycks, Dan, et al. “Natural adversarial examples.” *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition.* 2021

Hu, Edward J., et al. “Lora: Low-rank adaptation of large language models.” *arXiv preprint arXiv:2106.09685* (2021).

Katherine McDonough, Ludovic Moncla & Matje van de Camp (2019) Named entity recognition goes to old regime France: geographic text analysis for early modern French corpora, International Journal of Geographical Information Science, 33:12, 2498-2522, DOI: 10.1080/13658816.2019.1620235

Li, Xiang Lisa, and Percy Liang. “Prefix-tuning: Optimizing continuous prompts for generation.” arXiv preprint arXiv:2101.00190 (2021)

Martin, Louis, et al. "CamemBERT: a tasty French language model." *arXiv preprint arXiv:1911.03894* (2019)

Pfeiffer, J., Rücklé, A., Poth, C., Kamath, A., Vulic, I., Ruder, S., Cho, K., & Gurevych, I. (2020). AdapterHub: A Framework for Adapting Transformers. ArXiv, abs/2007.07779

Tao, Liufeng & Xie, Zhong & Xu, Dexin & Ma, Kai & Qiu, Qinjun & Pan, Shengyong & Huang, Bo. (2022). Geographic Named Entity Recognition by Employing Natural Language Processing and an Improved BERT Model. ISPRS International Journal of Geo-Information. 11. 598. 10.3390/ijgi11120598

Vaswani, Ashish, et al. “Attention is all you need.” *Advances in neural information processing systems* 30 (2017)