Évaluation des Grands Modèles de Langage pour la Reconnaissance d'Entités Nommées

Hédi Zeghidi, Ludovic Moncla

ÉVALUATION DES GRANDS MODÈLES DE LANGAGE POUR LA RECONNAISSANCE D'ENTITÉS NOMMÉES

A PREPRINT

Hédi Zeghidi
INSA Lyon,
CNRS, Universite Claude Bernard Lyon 1,
LIRIS, UMR5205,
69621 Villeurbanne, France

Ludovic Moncla
INSA Lyon,
CNRS, Universite Claude Bernard Lyon 1,
LIRIS, UMR5205,
69621 Villeurbanne, France
ludovic.moncla@insa-lyon.fr

ABSTRACT

Cet article présente une expérimentation des grands modèles de langage (LLMs) pour la reconnaissance d'entités nommées (NER). Les systèmes NER traditionnels reposent principalement sur des méthodes supervisées et sur des grands ensembles de données annotées, qui sont coûteux et longs à obtenir. Dans cette étude, nous nous sommes intéressé à évaluer la capacité des LLMs pour la tâche de NER en faisant appel au Few-Shot Prompting, ou apprentissage contextuel, qui permet aux modèles de répondre à une requête avec un minimum d'exemples. Les modèles GPTs d'OpenAI ont été évalué et comparé ainsi que certains modèles open source. Les résultats montrent que, bien qu'il y ait un écart de performance avec des modèles supervisés, les grands modèles excellent dans l'adaptation à de nouveaux types d'entités et de domaines avec des données très limitées. Nous explorons également les effets du format de sortie imposé pour les tâches de classification de tokens ou de segments (ou *spans*). Cette étude souligne le potentiel du *few-shot learning* (ou *prompting*) à réduire le besoin de grands ensembles de données annotées, améliorant ainsi l'évolutivité et l'accessibilité du NER et met également en lumière les difficultés pour les modèles génératifs à réaliser ce type de tâche.

Keywords Reconnaissance d'entités nommées, Grands modèles de langages, Few-shot prompting

1 Introduction

La reconnaissance d'entités nommées (NER) est une tâche fondamentale du traitement automatique du langage naturel (TAL), consistant à identifier et classifier des entités telles que des noms, des lieux et des dates dans un texte. Les systèmes de NER traditionnels nécessitent généralement de vastes ensembles de données annotées pour un entraînement efficace, ce qui peut être coûteux et long à constituer. Cette limitation freine le déploiement rapide et l'évolutivité des applications de NER à travers divers domaines et langues. L'apprentissage par *Few-Shot Learning* (FSL) offre une alternative prometteuse en permettant aux modèles d'effectuer des tâches de NER avec un nombre minimal d'exemples annotés. En exploitant les capacités des grands modèles de langage (LLMs), le FSL peut réduire de manière significative la dépendance à des données annotées en grande quantité. Cet article étudie les performances des LLMs dans les tâches de NER en utilisant le FSL, en comparant leur efficacité aux méthodes conventionnelles entièrement supervisées. À travers cette évaluation, nous visons à souligner le potentiel du FSL à transformer le paysage du NER, le rendant ainsi plus accessible et évolutif.

Malgré leurs performances impressionnantes dans de nombreuses tâches de TAL, la NER reste un défi pour les LLMs [Lu et al., 2024, González-Gallardo et al., 2024]. Cela est principalement dû au fait qu'elle est intrinsèquement une tâche d'étiquetage de séquences, tandis que les LLMs sont conçus pour la génération de texte. Les méthodes existantes utilisant les LLMs se contentent généralement d'extraire une liste d'entités à partir du texte, accompagnée de leurs classes respectives. Nous visons à améliorer ce processus en identifiant la position de chaque entité dans le texte et en la restituant dans un format structuré JSON, en incluant à la fois les positions des tokens et des caractères.

Cette étude vise à répondre à la question suivante : Quelle est l'efficacité des techniques de FSL associées aux LLMs pour accomplir des tâches de reconnaissance d'entités nommées ? Nos expérimentations sont réalisées à l'aide du jeu de données GeoEDdA[1] [Moncla et al., 2024], et plusieurs LLMs sont évalués et comparés aux modèles plus traditionnels et en particulier basés sur BERT [Devlin et al., 2018].

2 Etat de l'art

Certains travaux ont étudié les capacités des modèles génératifs pour la NER. Par exemple, Xie et al. [2023] proposent la décomposition de cette tâche en un ensemble de sous-problèmes plus simples par classe et d'y associer une tâche de questions-réponses en plus d'une augmentation syntaxique. Cependant, la plupart des travaux portent sur le FLS. Li and Zhang [2023] étudient les performances des LLMs et des modèles de langage pré-entraînés plus petits (par exemple, BERT, T5) pour une NER spécifique à un domaine. Ashok and Lipton [2023] proposent d'utiliser un LLM pour produire une liste d'entités potentielles avec des explications correspondantes justifiant leur compatibilité avec les définitions de type d'entité fournies. Wang et al. [2023] proposent d'ajouter des jetons spéciaux pour marquer les entités à extraire. L'évaluation montre que cette méthode permet d'obtenir des résultats comparables à ceux des méthodes supervisées.

Pour les modèles plus récents ayant des fenêtres contextuelles élargies, Agarwal et al. [2024] proposent une étude expérimentale comparant le FLS au *many-shot learning* (des centaines ou des milliers d'exemples). Les résultats indiquent une amélioration significative des performances lorsque le nombre d'exemples fournis passe de très peu à beaucoup. Les résultats indiquent également que l'ordre des exemples dans le prompte affecte les performances de l'apprentissage et que l'ajout d'un nombre d'exemples supérieur à l'optimum peut parfois dégrader les performances pour certaines tâches. En outre, le fait de ne fournir que des questions sans réponse dans les exemples donne de bons résultats dans certaines configurations. Ces observations dépendent du modèle et sont basées sur des tests effectués avec Gemini 1.5, GPT-4 et Claude 3.5.

3 Méthodologie

L'objectif de ce travail préliminaire est d'évaluer les capacités des LLMs sur la tâche de NER pour l'annotation au niveau des tokens et des spans (ou entités). Pour ces deux tâches, la sortie doit être au format JSON avec des informations pour chaque token ou span détecté. Un span fait référence à une entité et peut être composé de plusieurs tokens.

Notre méthodologie repose sur un prompt contenant une description de la tâche, du jeu d'étiquettes, ainsi qu'un exemple (avec à la fois l'entrée et la sortie). Une première expérimentation a montré que les LLMs ont du mal à extraire ou calculer avec précision les positions des tokens ou des spans à partir du texte brut. Bien que le format de sortie soit correct, les valeurs numériques de position générées étaient inexactes, ressemblant à des hallucinations ou des nombres aléatoires plutôt qu'à des positions réelles de tokens. Pour résoudre ce problème, nous proposons une solution incluant les informations sur la tokenisation dans les données d'entrée. En particulier, les détails de position des tokens pour la détection des spans. Ainsi l'exemple de texte donné en entrée visible dans l'Exemple 3.1 devient la version tokenisée visible en Exemple 3.2.

Exemple 3.1 *PIRANO, (Géog. mod.) ville d'Italie dans l'Istrie, environ à 14 milles de Capo d'Istria, en tirant vers le midi occidental. Elle est sur une petite presqu'île formée par le golfe Largone, & celui de Trieste. Les Vénitiens en sont les maîtres depuis 1583. Long. 31. 46. lat. 45. 48.*

Exemple 3.2 *('PIRANO' ,0) (',' ,1) ('(' ,2) ('Géog' ,3) ('.' ,4) ('mod' ,5) ('.' ,6) (')' ,7) ('ville' ,8) ('d" ,9) ('Italie' ,10) ('dans' ,11) ('l" ,12) ('Istrie' ,13) (',' ,14) ('environ' ,15) ('à' ,16) ('14' ,17) ('milles' ,18) ('de' ,19) ('Capo' ,20) ('d" ,21) ('Istria' ,22) (',' ,23) ('en' ,24) ('tirant' ,25) ('vers' ,26) ('le' ,27) ('midi' ,28) ('occidental' ,29) ('.' ,30) ('Elle' ,31) ('est' ,32) ('sur' ,33) ('une' ,34) ('petite' ,35) ('presqu" ,36) ('île' ,37) ('formée' ,38) ('par' ,39) ('le' ,40) ('golfe' ,41) ('Largone' ,42) (',' ,43) ('&' ,44) ('celui' ,45) ('de' ,46) ('Trieste' ,47) ('.' ,48) ('Les' ,49) ('Vénitiens' ,50) ('en' ,51) ('sont' ,52) ('les' ,53) ('maîtres' ,54) ('depuis' ,55) ('1583' ,56) ('.' ,57) ('Long' ,58) ('.' ,59) ('31' ,60) ('.' ,61) ('46' ,62) ('.' ,63) ('lat' ,64) ('.' ,65) ('45' ,66) ('.' ,67) ('48' ,68) ('.' ,69)*

[1] `https://huggingface.co/datasets/GEODE/GeoEDdA`

4 Expérimentations et résultats

Les expérimentations[2] sont réalisées en utilisant le jeu de données GeoEDdA [Moncla et al., 2024], qui contient des annotations sémantiques (aux niveaux des tokens et des spans) pour les entités nommées (Spatial, Person, et Misc), les entités nominales, les relations spatiales et les coordonnées géographiques. Les entités nommées étendues ou imbriquées [Gaio and Moncla, 2017], également présentes dans ce jeu de données, n'ont pas été prises en compte dans cette étude. Un exemple[3] provenant du jeu d'entraînement, contenant au moins une entité de chaque classe, est utilisé pour le FSL (inclus dans le prompt) et l'ensemble du jeu de test (200 articles de l'Encyclopédie de Diderot) est utilisé pour l'évaluation.

Les modèles GPT de l'API d'OpenAI[4] (gpt-3.5-turbo-0125, gpt-4-0613 et gpt-4o-2024-05-13) ont été évalués via le framework Python LangChain. Les scores de la classification au niveau des tokens sont présentés dans le Tableau 1. La micro précision moyenne, le rappel et le F1-score sont utilisés comme métriques d'évaluation avec correspondance stricte des entités (correspondance exacte des frontières en terme de tokens annotés). Bien que la performance soit inférieure à celle d'un modèle BERT fine-tuné[5] (avec le jeu d'entraînement entièrement supervisé contenant 1 800 entrées de l'encyclopédie), les expérimentations ont montré une variation significative entre les différentes classes. Au niveau des spans, certains problèmes inhabituels (par rapport au NER traditionnel) apparaissent, car le format de sortie est plus complexe qu'avec une annotation par token. Notre étude a confirmé l'amélioration des performances entre les versions GPT-4 et GPT-3.5. Avec GPT-3.5, seulement 28% des spans prédits sont corrects (comprenant à la fois les frontières et les labels) contre 49% avec GPT-4o, tandis que 13% sont partiellement corrects (frontières partielles et labels corrects) avec GPT-3.5 et 9% avec GPT-4o. De plus, avec GPT-3.5, certaines réponses ne se réfèrent pas au document d'entrée, mais plutôt à l'exemple du prompt. Dans de très rares cas, certaines réponses ne respectent pas strictement le format de sortie JSON prédéfini, et il peut manquer certains attributs de tokens (comme *start*, *end*, *text* ou *label*).

Modèle	**Précision**	**Rappel**	**F-score**
GPT-3.5	0.81	0.36	0.50
GPT-4	0.75	0.62	0.67
GPT-4o	0.68	0.72	0.70
Fine-tuned BERT	0.93	0.94	0.93

Table 1: Scores pour la tâche de classification des tokens

Le diagramme de la Figure 1 montre les micro f-scores obtenus pour chacune des classes pour les différentes versions de GPT testées et pour le modèle BERT fine-tuné. On observe que l'amélioration des performances entre les différentes versions de GPT est confirmé pour l'ensemble des classes à l'exception de la classe `Relation`. Cette classe contient un plus grand nombre que les autres de spans de longueurs très variables (en nombre de tokens). On remarque également des différences de performances entre les classes. Certaines obtiennent de très bon résultats (autour de 80%) avec les modèles GPT-4 ou GPT-4o, telles que les classes `Domain-mark`, `NP-Spatial`, `NP-Person`, `LatLong`. D'autres sont en dessous de 40%, comme par exemple `NP-Misc` et `Relation`.

Nous avons également expérimenté des LLMs plus petits et locaux en utilisant LM Studio[6], tels que Phi3 mini-4k (Microsoft), Gemma 2-9B (Google), Mistral v0.2-7B (MistralAI), Qwen 2-7B (équipe Qwen, affiliée au groupe Alibaba) et Llama 3.1-8B (Meta). Les modèles ont été exécutés sur une GPU Nvidia RTX 3500 ADA et ont obtenu des résultats très variables. Certains LLMs (Llama 3.1-8B et Qwen 2-7B) fournissent la syntaxe correcte du format de sortie JSON mais ne comprennent pas vraiment l'ensemble des labels définis. Ils inventent des tags (ou noms de classe) comme par exemple `Geography`, `City`, `Building`, etc. Ils proposent des entités qui n'existent pas dans le texte d'entrée. Ils en traduisent certaines comme par exemple `Italie` qui devient `Italy`. Ils trouvent plusieurs occurrences d'une même entités là où une seule était présente dans le texte donné en entrée. Enfin, bien que l'entrée soit tokenisée ils inventent des valeurs numériques pour les positions en token des entités. Les autres modèles testés ne comprennent absolument pas la tâche et font une description du lieu dont l'article est donné en entrée, ou répètent simplement la phrase d'entrée.

[2] Le code et les résultats sont disponibles sur Github : `https://github.com/GEODE-project/ner-llm`

[3] `https://huggingface.co/datasets/GEODE/GeoEDdA/viewer/default/train?row=53`

[4] `https://openai.com`

[5] `https://huggingface.co/GEODE/bert-base-french-cased-edda-ner`

[6] `https://lmstudio.ai`

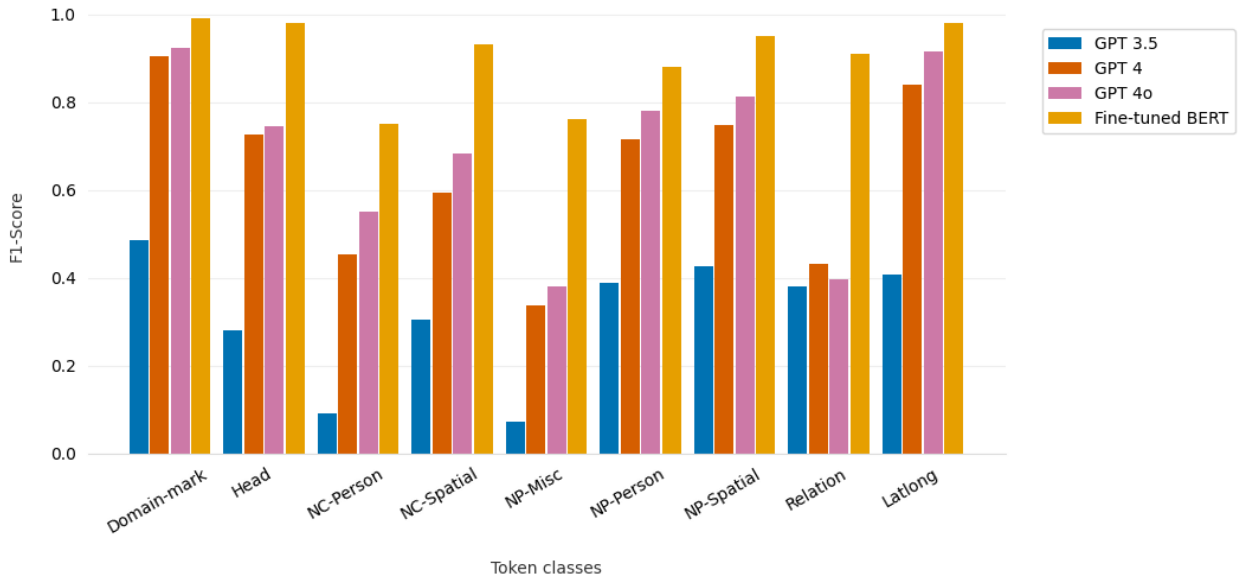

Figure 1: F-scores par classe pour la tâche de classification des tokens

Remerciements

Les auteurs remercient le LABEX ASLAN (ANR-10-LABX-0081) de l'Université de Lyon pour son soutien financier dans le cadre du programme français "Investissements d'Avenir" géré par l'Agence Nationale de la Recherche (ANR).

Bibliographie

Qiuhao Lu, Rui Li, Andrew Wen, Jinlian Wang, Liwei Wang, and Hongfang Liu. Large language models struggle in token-level clinical named entity recognition. *arXiv preprint arXiv:2407.00731*, 2024.

Carlos-Emiliano González-Gallardo, Tran Thi Hong Hanh, Ahmed Hamdi, and Antoine Doucet. Leveraging open large language models for historical named entity recognition. In *The 28th International Conference on Theory and Practice of Digital Libraries*, Ljubljana, Slovenia, 2024.

Ludovic Moncla, Denis Vigier, and Katherine Mcdonough. GeoEDdA: A Gold Standard Dataset for Geo-semantic Annotation of Diderot & d'Alembert's Encyclopédie. In *Second International Workshop on Geographic Information Extraction from Texts (GeoExT)*, Glasgow, Scotland, 2024.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Tingyu Xie, Qi Li, Jian Zhang, Yan Zhang, Zuozhu Liu, and Hongwei Wang. Empirical study of zero-shot ner with chatgpt. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7935–7956, Singapore, 2023.

Mingchen Li and Rui Zhang. How far is language model from 100% few-shot named entity recognition in medical domain. *arXiv preprint arXiv:2307.00186*, 2023.

Dhananjay Ashok and Zachary C Lipton. Promptner: Prompting for named entity recognition. *arXiv preprint arXiv:2305.15444*, 2023.

Shuhe Wang, Xiaofei Sun, Xiaoya Li, Rongbin Ouyang, Fei Wu, Tianwei Zhang, Jiwei Li, and Guoyin Wang. Gpt-ner: Named entity recognition via large language models. *arXiv preprint arXiv:2304.10428*, 2023.

Rishabh Agarwal, Avi Singh, Lei M Zhang, Bernd Bohnet, Stephanie Chan, Ankesh Anand, Zaheer Abbas, Azade Nova, John D Co-Reyes, Eric Chu, et al. Many-shot in-context learning. *arXiv preprint arXiv:2404.11018*, 2024.

Mauro Gaio and Ludovic Moncla. Extended named entity recognition using finite-state transducers: An application to place names. In *The ninth international conference on advanced geographic information systems, applications, and services*, Nice, France, 2017.

Les auteurs

Hédi Zeghidi est diplomé du Master Machine Learning and Data Mining de l'Université Jean Monnet de Saint-Etienne. Il a réalisé ce travail dans le cadre de son stage de fin d'étude au sein du projet interdisciplinaire GEODE financé par le LabEx ASLAN.

Ludovic Moncla est maître de conférences en informatique à l'INSA de Lyon et au laboratoire LIRIS. Ses travaux se situent à l'interface du traitement automatique du langage naturel, du traitement de l'information géographique et des humanités numériques.