Couplage d'Approches LLM et SLM pour le Déploiement de Solutions d'Extraction d'Entités Nommées

Samuel Kierszbaum[1], Nicolas Heulot[2]

[1]Airbus Protect
[2]IRT SystemX

samuel.kierszbaum@airbus.com

Résumé

La reconnaissance d'entités nommées (NER) requiert de gros volumes de données annotées qui sont complexes et coûteux à obtenir dans des domaines spécialisés. Cette étude compare deux approches adaptées à ce contexte : une méthode few-shot exploitant un grand modèle de langage (LLM) et une approche hybride combinant des annotations LLM avec un fine-tuning sur un petit modèle (SLM). Nos résultats confirment l'intérêt d'une approche hybride pour permettre de déployer à l'échelle des systèmes NER nécessitant très peu d'exemples annotés manuellement en entrée.

Mots-clés

TAL, NER, LLM.

Abstract

Named Entity Recognition (NER) effectiveness is often limited by the scarcity of annotated data, in particular in specialized domains. This study compares two NER approaches designed for low-resource settings : (1) a few-shot approach leveraging a large language model (LLM) for annotation and (2) a hybrid method that combines LLM-generated annotations with fine-tuning on a small language model (SLM). Our findings suggest that hybrid strategies can alleviate the challenges of manual annotation while maintaining high-quality entity recognition.

Keywords

NLP, NER, LLM.

1 Introduction

La reconnaissance des entités nommées (NER) est une tâche de traitement du langage naturel (TAL) qui classifie les entités nommées dans le texte, telles que des personnes, des organisations, des lieux, etc. Les travaux récents utilisent un modèle de langage tel que BERT (Bidirectional Encoder Representational Transformer) [7] affiné pour reconnaître des entités spécifiques à un domaine. Cependant, cet ajustement (*fine-tuning*) requiert d'avoir accès à un grand nombre d'exemples étiquetés pour donner de bons résultats. La collecte d'une grande quantité de données de haute qualité pour la tâche NER est très difficile et coûteuse, ce qui limite l'applicabilité de cette tâche.

Dans cet article, nous proposons une approche basée sur l'utilisation d'un LLM pour l'augmentation des données d'annotation, afin de réduire les coûts d'étiquetage manuel et de permettre l'adaptation rapide à des entités spécifiques d'un domaine. Les coûts d'utilisation d'un LLM à l'inférence étant très important, nous proposons de conserver une approche basée sur l'ajustement d'un modèle BERT, car cette dernière permet un passage à l'échelle en termes de nombres de documents traités. En utilisant le LLM comme pseudo-annotateur et l'ajustement d'un modèle BERT sur des annotations générées par le LLM, nous cherchons à évaluer si les performances peuvent égaler celle d'un apprentissage supervisé traditionnel à base de données annotées manuellement.

2 Travaux Antérieurs

La qualité de la reconnaissance d'entités nommées en utilisant de petits modèles de langage (SLMs) dépend fortement à la fois de la qualité des données annotées et de leur quantité. L'obtention de jeux de données avec une haute qualité des annotations demeure un défi majeur, en particulier dans les domaines spécialisés [20, 24].

Pour atténuer ces limitations, de nombreuses approches ont été explorées dans la littérature. L'une d'elles est l'amélioration itérative, qui vise à accroître l'efficacité de l'annotation en s'appuyant sur des méthodes avancées, des outils spécialisés et des techniques semi-automatisées [10, 12, 3]. Ces approches intègrent souvent des mécanismes d'interaction humaine, permettant d'affiner progressivement la qualité des données annotées. On notera certaines de ces approches sont rendues facilement accessibles et reutilisables via des frameworks comme FLAIR [2].

La supervision distante est une approche qui utilise des heuristiques basées sur des règles pour annoter automatiquement les données, réduisant ainsi le besoin d'annotation manuelle [15, 19]. Bien que cette méthode permette de générer efficacement des jeux de données annotés à grande échelle, son efficacité est souvent limitée par la précision et la capacité de généralisation des règles prédéfinies.

La reconnaissance d'entités nommées en apprentissage par faible nombre d'exemples (*few-shot* NER) est une alternative prometteuse car les grands modèles de langage ont démontré une robustesse face aux contextes à faibles

ressources avec des approches comme GliNER [25] par exemple, notamment en termes de disponibilité de données annotées, y compris dans les domaines spécialisés [14, 9].

Bien que l'apprentissage par faible nombre d'exemples (*few-shot learning*) ait déjà été appliqué directement aux tâches de reconnaissance d'entités nommées, son potentiel pour la génération de jeux de données annotés dans un cadre de supervision distante reste peu exploré. Des travaux récents, tels que [18, 21], exploitent les capacités de génération des grands modèles de langage afin de produire des jeux de données annotés à partir de zéro, en générant à la fois les phrases et leurs annotations correspondantes.

Dans notre étude, nous explorons une approche hybride en utilisant l'annotation basée sur les grands modèles de langage en apprentissage par faible nombre d'exemples (*few-shot*) comme étape intermédiaire avant le *fine-tuning*. Notre approche vise à générer des données annotées en exploitant des données existantes mais non annotées, ce qui la distingue des méthodes déjà explorées dans la littérature. Bien que ces dernières soient particulièrement prometteuses dans des contextes où les données disponibles sont limitées, notre méthode semble mieux adaptée aux situations où une quantité importante de données brutes est disponible, mais sans annotations.

3 Méthode

Afin d'évaluer l'impact de l'annotation automatique par les grands modèles de langage sur la reconnaissance d'entités nommées, nous comparons trois approches distinctes. Ces approches représentées dans la la figure 1 ont été choisies pour mesurer les bénéfices et les limites de l'utilisation des LLMs dans un contexte à faibles ressources.

Approche 1 : L'approche *few-shot* NER basée sur les LLM dans un contexte à faibles ressources, où le contexte à faibles ressources fait référence à la disponibilité limitée de données annotées manuellement (30 documents annotés).

Approche 2 : La seconde approche suit le paradigme classique du *fine-tuning* d'un petit modèle de langage sur des données annotées manuellement, ce qui constitue une référence pour comparer l'impact de l'annotation automatique utilisée pour les approches 1 et 3.

Approche 3 : L'approche hybride, également conçue pour un contexte à faibles ressources. Dans un premier temps, la méthode *few-shot* NER basée sur les LLM est utilisée pour générer des données annotées. Ces nouvelles données sont ensuite combinées avec les données annotées manuellement utilisées pour l'apprentissage *few-shot* afin de constituer un jeu de données, qui est ensuite utilisé pour le *fine-tuning* d'un SLM.

En comparant ces trois approches, nous cherchons à voir la viabilité de notre approche hybride pour la tâche de reconnaissance d'entités nommées dans des contextes spécialisés avec peu de données annotées manuellement.

Dans les sections suivantes, nous détaillons le protocole expérimental utilisé pour comparer ces trois approches. Nous commençons par une description du jeu de données, suivie d'un aperçu des procédures d'entraînement spécifiques à chaque approche, afin d'assurer une compréhension approfondie de leur mise en œuvre.

3.1 Jeu de Données

Nous considérons ici le jeu de données AeroBERT-NER [1][22]. Ce jeu de données est composé de 1 432 phrases en anglais issues du domaine de l'ingénierie des exigences en aérospatiale. Chaque phrase est annotée pour la reconnaissance d'entités nommées selon le schéma d'étiquetage BIO, avec des entités réparties en cinq catégories :

- SYS : systèmes et matériels
- VAL : valeurs numériques
- ORG : entreprises et organisations
- DATETIME : expressions de date et d'heure
- RES : ressources documentaires

Comme expliqué dans la section 3.2.2, nous nous concentrons exclusivement sur l'entité SYS pour diverses raisons. Le corpus comprend un nombre total de 1855 entités SYS. Parmi le corpus, 999 phrases contiennent au moins une mention de l'entité système. Les 433 phrases restantes sont conservées dans notre corpus afin de vérifier que nos approches ne créent pas de faux positifs.

Comme nous allons le voir dans les sections 3.3 et 3.4, nous procédons pour les approches utilisant un SLM à une classique validation croisée en 5 partitions. Les phrases sans mentions d'entités SYS sont réparties de manière homogènes parmi les 5 partitions.

Cette validation croisée permet une évaluation robuste, en agrégeant les performances sur les différentes partitions, donc sur l'ensemble du corpus à l'exception des 30 exemples annotés. Nous calculons la performance du modèle LLM de la même manière, sur l'ensemble du jeu de donnée à l'exception des 30 exemples. Les résultats et la méthode d'évaluation sont présentés avec plus de détails dans la section 3.5 le tableau 1.

3.2 Approche 1 - Few-Shot NER LLM

Dans cette section, nous décrivons le protocole expérimental de notre approche *few-shot* NER utilisant un grand modèle de langage dans un contexte à faibles ressources. La conception de cette approche repose sur plusieurs choix méthodologiques clés, chacun ayant des implications sur les performances du modèle, l'efficacité computationnelle et la facilité de mise en œuvre. Dans ce qui suit, nous présentons ces choix et en expliquons la justification.

3.2.1 Choix du Modèle

Pour notre étude, nous avons choisi GPT-4 [16], car il s'agit d'un des modèles les plus fréquemment étudiés dans la littérature et il démontre de manière constante des performances à l'état de l'art pour les tâches de reconnaissance d'entités nommées en apprentissage par faible nombre d'exemples.

D'autres modèles alternatifs existent et, bien que potentiellement moins performants, ils peuvent présenter d'autres avantages en termes de transparence, coût et flexibilité de

1. Il est disponible sur Hugging Face à l'adresse suivante : https ://huggingface.co/datasets/archanatikayatray/aeroBERT-NER.

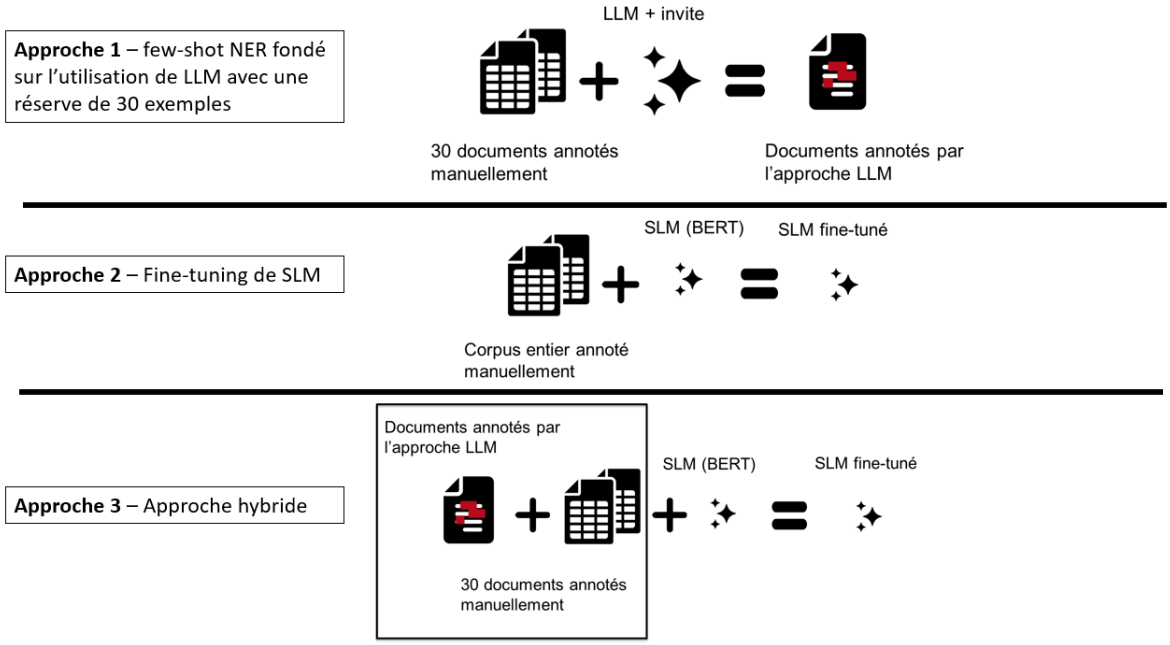

FIGURE 1 – Synthèse des trois approches comparées

déploiement. En particulier, contrairement à GPT-4, qui est un modèle propriétaire à boîte noire, des alternatives open source comme Mixtral [11] ou DeepSeek [6] offrent une meilleure transparence et peuvent être facilement affinés à des domaines spécifiques. De plus, ces modèles sont souvent moins coûteux, et certains peuvent être déployés localement, évitant ainsi les frais liés aux API et réduisant la dépendance aux fournisseurs externes.
Malgré ces considérations, notre choix de GPT-4 est motivé par ses performances supérieures. Étant donné le rôle critique des capacités du modèle dans l'apprentissage avec peu d'exemples, utiliser un modèle hautement performant permet d'établir une référence robuste pour la génération d'annotations afin d'évaluer à la fois notre approche de reconnaissance d'entités nommées en apprentissage par faible nombre d'exemples et l'approche hybride.

3.2.2 Extraction Multi-Entités vs. Mono-Entité

Une décision clé dans la conception d'un système de reconnaissance d'entités nommées basé sur les LLM concerne le choix entre extraire plusieurs entités simultanément ou une seule entité à la fois. Bien que la littérature ne fournisse pas de preuves solides en faveur d'une approche plutôt que l'autre, ces deux stratégies présentent des avantages et des compromis distincts.
L'extraction multi-entités permet d'effectuer une seule inférence par phrase, mais elle complexifie la sélection des exemples, car il est nécessaire d'assurer une représentation équilibrée de toutes les catégories d'entités dans le prompt. L'extraction mono-entité, en revanche, simplifie la structure du prompt et la sélection des exemples, mais elle nécessite plusieurs inférences par phrase (une par type d'entité) ainsi qu'un post-traitement pour consolider les résultats.
Nous avons opté pour l'approche mono-entité, qui présente plusieurs avantages dans notre contexte. Elle réduit la longueur de la section de définition des entités dans le prompt, ce qui diminue la consommation de tokens et les coûts induits. Elle simplifie la sélection des exemples, puisque chaque prompt ne traite qu'un seul type d'entité. Afin d'éviter le post-traitement nécessaire pour fusionner les résultats des différentes entités, nous avons restreint notre étude à une seule catégorie d'entités. Nous avons choisi de nous concentrer sur l'entité SYS, car elle est la plus spécifique au domaine. En mettant l'accent sur les entités SYS, nous nous assurons que nos approches sont évaluées sur l'aspect le plus complexe et spécialisé du jeu de données, là où les modèles généralistes sont les plus susceptibles d'échouer.

3.2.3 Structure du Prompt

Nous proposons d'utiliser un prompt suivant une structure couramment utilisée pour la reconnaissance d'entités nommées en *few-shot learning* avec les LLM, s'inspirant de travaux antérieurs tels que [23, 9].
Le prompt est composé des éléments suivants :
Rôle du système : Définit le modèle comme un assistant spécialisé en NER.
Lignes directrices : Spécifie les règles à suivre pour assurer une annotation cohérente des entités.
Exemples *few-shot* : Illustre la manière dont les entités doivent être annotées.
Phrase test : Contient la phrase à analyser pour l'inférence.

Le contenu des sections Rôle du système et Lignes directrices a été affiné par essais et erreurs à l'aide de LLM plus petits et moins coûteux, afin d'optimiser la clarté et la cohérence de la reconnaissance des entités.

Un prompt complet est présenté ci-dessous :

```
System: You are an AI-assistant tasked with
    identifying and annotating hardware terms
    related to spacecraft, satellites, and
    aeronautical systems in a given text. These
    terms should be enclosed within double "at"
    symbols (@@) at the beginning and double hash
     symbols (##) at the end.

**Guidelines**:
1. Identify Hardware Systems and Components:
Focus on highlighting terms related to physical
    spacecraft, satellites, aeronautical systems,
     or their components. This includes specific
    names (e.g., "CubeSats," "Deep Space Network
    ") and technical terms (e.g., "landing gear,"
     "fuel system," "airplane," "satellite").
    Only hardware systems and their components
    should be annotated.

2. Annotation Format:
Enclose each identified hardware system or
    component within @@ and ##. For example, the
    term "CubeSats" should be annotated as
    @@CubeSats##.

3. Consistency:
Ensure that all instances of similar terms are
    annotated consistently throughout the
    document. For example, if "fuel system" is
    annotated in one sentence, ensure that all
    instances of "fuel system" are similarly
    annotated.

4. Annotation of Compound Terms:
Annotate only the aerospace-related hardware
    systems or components within a compound term.
     Do not enclose the entire phrase unless it
    consists solely of technical elements.
For example, in the phrase "turbine engine
    powered airplane," only @@turbine engine##
    and @@airplane## should be annotated, as "
    powered" is not part of the system or
    component.

5. Contextual Understanding:
Annotate based on relevance to aerospace hardware
     technology. Terms not directly related to
    hardware aerospace systems, components, or
    technologies should not be annotated.

6. Avoid Over-annotation:
Do not annotate people, titles, regulations, or
    operational terms:
Do not annotate roles, personnel, or
    organizational titles, even if they are part
    of aerospace organizations (e.g., "
    administrator," "NASA Administrator," "NASA,"
     "NSF," "Planet Labs").
Avoid annotating regulations, legal references,
    or general phrases (e.g., "requirements of
    part 34," "Sections 25-1181").
Avoid annotating general operational terms like "
    flight," "landing," "takeoff," "stall," "
    surge," "flameout," and "navigation" unless
    they are directly part of a specific
    aerospace hardware technology.
Finally, avoid general scientific terms (e.g., "
    snow," "science," "jamming," "ice accretion")
    .
These are not related to aerospace hardware
    systems and should be left unannotated.

Human: Failure of structural elements of the drag
     limiting systems need not be considered if
    the probability of this kind of failure is
    extremely remote.
AI: Failure of structural elements of the @@drag
    limiting systems## need not be considered if
    the probability of this kind of failure is
    extremely remote.
Human: It must be shown by analysis or test, or
    both, that each operable reverser can be
    restored to the forward thrust position.
AI: It must be shown by analysis or test, or both
    , that each operable @@reverser## can be
    restored to the forward thrust position.
Human: This is an example of input sentence with
    2 example before.
```

3.2.4 Sélection du Réservoir d'Exemples Disponibles

Dans cette étude, nous imposons une contrainte sur le nombre d'exemples annotés pouvant être utilisés pour constituer le réservoir à partir duquel les exemples du prompt seront sélectionnés. Cette contrainte est motivée par l'hypothèse que nous opérons dans un contexte à faibles ressources, où l'accès aux données annotées spécifiques au domaine est limité. Par conséquent, nous limitons ce réservoir à 30 exemples annotés.

À notre connaissance, peu de travaux antérieurs se sont intéressés à la stratégie optimale de sélection des exemples pour constituer ce réservoir dans le cadre du *few-shot* NER basé sur les LLM. Bien que la recherche ait largement exploré la sélection des exemples à inclure dans le prompt, les critères déterminant quels exemples doivent être annotés et ajoutés au réservoir restent encore peu étudiés.

3.2.5 Mécanisme de Sélection des Exemples

Le mécanisme de sélection des exemples à inclure dans le prompt joue un rôle crucial dans les performances du modèle, en particulier dans notre contexte à faibles ressources. En effet, la littérature indique qu'une sélection d'exemples de haute qualité à partir d'un petit réservoir de 30 instances surpasse une approche de prélèvement aléatoire dans un ensemble de données bien plus vaste [1].

Ces résultats renforcent l'hypothèse selon laquelle, dans les contextes à faibles ressources, une curation minutieuse des exemples et une structuration rigoureuse du prompt peuvent compenser les limitations imposées par la taille réduite du jeu de données. En sélectionnant stratégiquement les exemples, il est ainsi possible d'optimiser les performances de l'apprentissage *few-shot*, même lorsque la disponibilité des données est fortement restreinte.

Il est également souligné dans [14] que l'efficacité des grands modèles de langage dépend fortement de la sélection rigoureuse des exemples utilisés pour l'inférence. De plus, la stratégie de sélection optimale est spécifique à chaque jeu de données : différentes approches donnent des résultats variables en fonction des caractéristiques du corpus.

Parmi les diverses stratégies de sélection d'exemples décrites dans la littérature, on distingue :

Le prompting statique : approche la plus simple, où un même ensemble d'exemples est utilisé systématiquement pour toutes les entrées.

La sélection aléatoire : qui introduit de la variabilité à chaque étape d'inférence.
Les méthodes basées sur la similitude : où les exemples sont sélectionnés en fonction de leur pertinence par rapport à la phrase d'entrée.

D'autres approches plus avancées ont également été proposées, telles que le guidage de la sélection des exemples avec un score de complexité [1], ou la sélection des exemples les plus pertinents grâce à des *embeddings* au niveau des entités [23]. Mais ces techniques requièrent un volume de données annotées important en contradiction avec notre contexte à faibles ressources.
Dans notre étude, nous avons choisi d'utiliser la mesure de similitude la plus courante dans la littérature : les *embeddings* de phrases avec calcul par cosinus de similitude. Plus précisément, nous utilisons le modèle all-MiniLM-L6-v2 [2] de sentence transformers [17], comme dans [1], pour le calcul du score de similitude entre phrases.
Bien que cette méthode offre un mécanisme de récupération simple et efficace sur le plan computationnel, cette approche est limitée par la similitude au niveau des phrases qui ne correspond pas toujours à la pertinence au niveau des entités nommées [23]. Par exemple, on peut avoir une similarité sémantique importante entre une phrase dont le sujet correspond à une entité nommée et la même phrase dont le sujet est un pronom faisant référence à cette entité. L'utilisation de cette deuxième phrase comme exemple est peu pertinente pour l'annotation des entités. Dans notre étude, cette limitation ne nous affecte pas car la réserve d'exemples disponibles ne contient que des phrases comportant au moins une occurrence de l'entité cible (SYS).

3.2.6 Format de Sortie

Le format de sortie en *in-context learning (ICL)* influence également le comportement du modèle. Différentes stratégies de formatage existent :
Le format utilisé dans [13] structure les sorties sous forme de dictionnaires (ex. : 'Chemical' : ['apomorphine'], 'Disease' : ['hypothermia']). Toutefois, cette approche ne fournit pas la position des entités dans la phrase, ce qui peut poser des problèmes d'extraction lorsque plusieurs mentions similaires apparaissent.
Une alternative est l'annotation en format BIO, utilisée dans [1], où chaque token est étiqueté comme Beginning (B), Inside (I) ou Outside (O) d'une entité.
D'autres formats, tels que le BMES tagging ou l'extraction basée sur la position des entités, ont été testés mais n'offrent pas de résultats supérieurs [23].
Alternativement, des symboles spéciaux peuvent être utilisés pour mettre en évidence les entités au sein du texte (par exemple, "Carcinoma @@ductal de mama derecha"), une méthode employée avec succès dans plusieurs travaux [14, 23, 9].
Le Chain-of-Thought (CoT) prompting, exploré dans [4], constitue une autre approche visant à améliorer l'interprétabilité. Cependant, l'intégration du raisonnement CoT introduit une charge supplémentaire pour les équipes d'annotation manuelle et complique la mise en œuvre.
Dans notre approche, nous avons privilégié la simplicité et l'efficacité en adoptant le format à base de symboles spéciaux [23, 9], pour plusieurs raisons.
Ce format permet de distinguer efficacement les différentes occurrences de l'entité cible sans accroître la complexité de l'annotation. Il offre l'avantage d'une conversion simple et réversible avec le format BIO, qui constitue à la fois le format original du corpus et celui requis pour le SLM dans l'approche hybride. Pour passer du format @@... au format BIO, il suffit de repérer les tokens encadrés par les balises spéciales : le premier token précédé de @@ reçoit l'étiquette B-SYS, les suivants jusqu'à sont étiquetés I-SYS, et tous les autres tokens sont marqués O. Inversement, pour convertir un corpus BIO en format à balises, on détecte les séquences d'étiquettes B-SYS/I-SYS consécutives et on encadre les tokens correspondants avec @@ au début et à la fin. Ce choix octroie une flexibilité appréciable dans les chaînes de traitement.

3.3 Approche 2 - Fine-tuning

L'approche traditionnelle de *fine-tuning* suit la méthodologie standard d'entraînement des modèles basés sur les transformeurs pour les tâches de reconnaissance d'entités nommées. Plus précisément, nous affinons un modèle BERT, à l'instar de l'étude [22], en utilisant des données annotées manuellement issues du domaine des exigences aérospatiales.
Afin d'assurer une évaluation robuste, nous adoptons une validation croisée en 5 partitions (5-fold cross-validation). À chaque itération, l'ensemble de données est divisé en cinq sous-ensembles, dont quatre sont utilisés pour l'entraînement et un pour la validation. Il est important de noter que les 30 instances sélectionnées pour l'approche NER en apprentissage par faible nombre d'exemples sont systématiquement incluses dans l'ensemble d'entraînement, garantissant ainsi leur présence dans les cinq partitions.
Nous explorons le même espace d'hyperparamètres que celui recommandé dans l'article original sur BERT pour le *fine-tuning* [8] (*Batch size* : 16, 32, *Learning rate* : 2e-5, 3e-5, 5e-5) Le modèle est entraîné sur 5 époques, avec des évaluations périodiques pour suivre ses performances. Plus précisément, la performance du modèle est évaluée à chaque tiers d'époque sur l'ensemble de validation. À l'issue de l'entraînement, nous conservons la meilleure performance obtenue au cours de ces évaluations. Dans la suite, les hyperparamètres ayant permis d'obtenir les meilleures performances l'affinage des différentes approches sont : *batch size* : 16 et *learning rate* : 5e-5.

3.4 Approche 3 - Hybride LLM SLM

L'approche hybride combine la méthode NER en apprentissage par faible nombre d'exemples basée sur un modèle de langage à grande échelle avec le paradigme de *fine-tuning*. Plutôt que de s'appuyer uniquement sur des données annotées manuellement pour le *fine-tuning*, nous utilisons la

2. le modèle est disponible ici : https ://huggingface.co/sentence-transformers/all-MiniLM-L6-v2

TABLE 1 – Évaluation des résultats pour les différentes approches

Approche	F1 Score	Exact	Partial	Missing	Spurious
fine-tuning	**0.90**	**1563**	179	**113**	**238**
LLM	0.83	1362	244	249	242
Hybride	0.85	1408	**256**	191	332

méthode NER basée sur un LLM en *few-shot* pour générer des instances annotées, qui servent ensuite de données d'entraînement pour le processus d'affinage.

Plus précisément, nous reproduisons l'ensemble de données utilisé dans l'approche d'affinage avec une validation croisée en 5 partitions, mais avec une modification essentielle : à l'exception des 30 instances annotées manuellement, qui sont conservées dans les cinq jeux d'entraînement, chaque partition d'entraînement est constituée d'instances annotées par le LLM plutôt que par des annotateurs humains. Hormis cette substitution, la procédure de *fine-tuning* reste inchangée, en maintenant la même architecture de modèle et la même procédure d'exploration et de sélection d'hyperparamètres.

3.5 Évaluation

Pour évaluer la performance, nous utilisons la bibliothèque Python nervaluate [5], qui permet une analyse détaillée des performances en reconnaissance d'entités nommées. Plus précisément, nous rapportons les métriques suivantes :

Score F1 : Moyenne harmonique de la précision et du rappel, fournissant une mesure globale de la capacité du modèle à identifier correctement les entités tout en minimisant les faux positifs et les faux négatifs.

Nombre d'entités manquantes (Missing Count) : Nombre de mentions d'entités présentes dans la vérité terrain mais non prédites par le modèle, mettant en évidence les erreurs liées au rappel.

Nombre d'entités superflues (Spurious Count) : Nombre de prédictions d'entités ne correspondant à aucune entité dans la vérité terrain, reflétant les faux positifs.

Nombre de correspondances exactes (Exact Match Count) : Nombre d'entités prédites correspondant exactement aux annotations de la vérité terrain, à la fois en termes de frontières et d'étiquettes. Cette métrique stricte évalue la capacité du modèle à identifier précisément les entités sans aucune déviation.

Nombre de correspondances partielles (Partial Match Count) : Nombre de cas où une entité prédite chevauche partiellement une entité de la vérité terrain sans correspondance exacte. Cette métrique tient compte des prédictions approximativement correctes, offrant une évaluation plus nuancée que les seules correspondances exactes.

4 Résultats

Nous observons que l'approche traditionnelle de *fine-tuning* sur des données manuellement annotées surpasse les deux autres approches, tandis que l'approche hybride dépasse légèrement la méthode basée uniquement sur le LLM (voir Table 1).

Nos résultats indiquent que l'approche hybride se rapproche des performances de la méthode traditionnelle. Cela permet ainsi de réduire considérablement l'effort d'annotation qui est particulièrement précieuse dans des contextes à faibles ressources, où l'annotation manuelle est coûteuse et chronophage. Les performances des modèles BERT fournissent une indication indirecte de la qualité de l'annotation, les ensembles de données bien annotés contribuant généralement à de meilleures performances des modèles. Cela suggère que les LLMs peuvent être exploités pour accélérer le processus d'annotation. En automatisant ou en augmentant le processus d'annotation, les LLMs ont le potentiel de réduire la dépendance à un effort humain intensif tout en maintenant une annotation de haute qualité, accélérant ainsi le développement de jeux de données annotés dans des environnements à ressources limitées.

Bien que l'approche hybride surpasse la méthode basée sur un LLM dans notre cas, l'écart de performance reste relativement faible. Toutefois, le choix entre ces deux approches ne doit pas se limiter à des critères de performances. D'autres facteurs peuvent être pris en compte.

Premièrement, l'utilisation des LLMs introduit un risque d'hallucination, un problème absent dans la phase d'inférence fondé sur le modèle BERT de la méthode hybride. Un autre aspect critique est le stockage et le coût computationnel : alors que les méthodes d'annotation basées sur les LLMs nécessitent souvent des modèles de grande envergure, exigeant d'importantes ressources de stockage et de mémoire, l'approche hybride exploite les LLMs uniquement pour générer l'ensemble d'entraînement. Par la suite, seul BERT ou un autre modèle de langage de taille réduite est utilisé, diminuant ainsi les besoins computationnels et de stockage à long terme. Cette efficacité est particulièrement pertinente dans les environnements où les contraintes de calcul et de stockage peuvent affecter la faisabilité du déploiement d'une solution de reconnaissance d'entités nommées.

Une autre distinction clé réside dans le compromis entre précision et rappel, comme le reflètent les métriques de nombre d'entités manquantes (Missing Count) et d'entités superflues (Spurious Count). L'approche hybride identifie un plus grand nombre d'entités mais génère également un nombre plus élevé de prédictions superflues, ce qui suggère une inclination vers le rappel au détriment de la précision. À l'inverse, l'approche basée sur les LLMs se montre plus conservatrice.

Le choix entre ces méthodes doit donc être guidé par les exigences opérationnelles spécifiques, en fonction de la

priorité accordée soit à un rappel plus élevé, soit à la réduction des faux positifs. Par ailleurs, les coûts computationnels et les contraintes de stockage doivent également être pris en compte, car ces facteurs peuvent influencer de manière significative la faisabilité de chaque approche selon l'environnement de travail.

4.1 Gestion des Hallucinations

Les sorties générées par le LLM peuvent contenir des hallucinations [26] car le modèle génère des réponses qui ne correspondent pas à l'intention de l'utilisateur. Dans notre cas, les instances d'hallucination sont facilement repérables en comparant la sortie du LLM (un document annoté, auquel on soustrait les annotations pour notre analyse) à son entrée (le document à annoter). Ainsi, nous avons identifié quatre cas d'hallucinations, affectant soit la ponctuation, soit la casse des caractères. Nous les reproduisons ci-dessous :

- **Modification de la casse** :
  - **Entrée 1** : **cubesat** attitude determination techniques have significantly advanced in the past decade , with many of the techniques found on larger spacecraft now also available on CubeSats .
  - **Sortie 1** : **CubeSat** attitude determination techniques have significantly advanced in the past decade , with many of the techniques found on larger spacecraft now also available on CubeSats .
  - **Entrée 2** : **cubesat** instrument builders are also reimagining their instruments based on commercial off-the-shelf (cots) parts .
  - **Sortie 2** : **CubeSat** instrument builders are also reimagining their instruments based on commercial off-the-shelf (cots) parts .

Explication : Le LLM a modifié la casse de "cubesat" en "CubeSat". On constate que cette erreur peut s'expliquer par l'inconsistance au niveau de la capitalisation de ce terme dans le corpus, particulièrement visible dans l'Entrée 1, où l'on trouve les deux manières d'écrire le terme dans la même phrase.

- **Ponctuation** :
  - **Entrée 1** : each fuel storage system must be designed to prevent significant loss of stored fuel from any vent system due to fuel transfer between fuel storage or supply **systems , or** under likely operating conditions .
  - **Sortie 1** : each fuel storage system must be designed to prevent significant loss of stored fuel from any vent system due to fuel transfer between fuel storage or supply **systems, or** under likely operating conditions .
  - **Entrée 2** : the exhaust system , including exhaust heat exchangers for each powerplant or auxiliary power **unit , must** provide a means to safely discharge potential harmful material .
  - **Sortie 2** : the exhaust system, including exhaust heat exchangers for each powerplant or auxiliary power **unit, must** provide a means to safely discharge potential harmful material .

Explication : Suppression d'espace superflu après la virgule, probablement due à un ajustement automatique du modèle pour respecter la convention typographique standard assimilée lors de son apprentissage de la modélisation du langage.

Ces modifications sont mineures et n'affectent que 0,4% du jeu de données. Nous avons décidé de ne pas les prendre en compte. Ainsi, le *fine-tuning* de l'approche hybride a été réalisé sur l'ensemble des données annotées produites par le LLM, y compris celles affectées par des hallucinations.

4.2 Limitations

Pour des raisons de praticité, notre étude s'est concentrée exclusivement sur un seul type d'entité dans un seul ensemble de données. Étendre l'analyse à plusieurs types d'entités et à divers ensembles de données permettrait d'évaluer plus largement les approches proposées et leur capacité de généralisation. Également, il aurait été intéressant d'explorer comment le choix du LLM impacte la performance de notre approche hybride, par exemple en utilisant des modèles comme GPT-3.5 ou un modèle open-source type Mistral. De plus, notre implémentation de l'approche basée sur les LLMs n'a pas exploré de manière exhaustive toutes les techniques d'optimisation possibles, notamment l'auto-vérification (*self-verification*), comme démontré dans [23]. Cette technique introduit une étape d'inférence supplémentaire où le modèle réévalue et affine ses prédictions, renforçant ainsi sa robustesse et sa fiabilité. L'intégration de telles améliorations pourrait encore optimiser les performances de l'approche basée sur les LLMs et réduire les erreurs potentielles.

5 Conclusion

Dans cet article, nous avons comparé une approche classique à une nouvelle approche hybride utilisant un LLM pour une tâche de reconnaissance d'entités nommées. Notre approche hybride se base sur un LLM pour l'augmentation des données d'annotation en partant d'un faible nombre d'exemples, puis l'utilisation de ces données pour affiner un SLM dans le but d'éviter les coûts prohibitifs d'utilisation d'un LLM à l'inférence. Nous avons comparé ces approches sur un jeu de données d'exigences en aérospatiale. Nos résultats, même si ils restent préliminaires, indiquent que l'approche hybride se rapproche des performances d'une approche classique avec toutes des données annotées manuellement (F1 > 0.80). Ainsi l'intégration des LLMs dans le processus d'annotation peut aider à réduire les coûts de déploiement à l'échelle de solutions d'extraction d'entités nommées sur des domaines spécifiques.

5.1 Travaux futurs

Nos résultats suggèrent que l'intégration des LLMs dans le processus d'annotation offre des avantages significatifs. Étant donné la complexité et le coût associés à la construction de corpus annotés de haute qualité, nous estimons qu'une exploration approfondie de cette approche est justifiée. Plus précisément, l'exploitation des LLMs pourrait rationaliser le flux de travail d'annotation en facilitant l'affinement itératif des consignes d'annotation et en assurant une cohérence accrue entre les ensembles de données.
En outre, les LLMs pourraient être utilisés pour appliquer rétroactivement des consignes mises à jour à des corpus précédemment annotés, garantissant ainsi leur alignement avec l'évolution des standards d'annotation. Il serait intéressant d'approfondir ces perspectives afin d'optimiser et d'élargir le rôle des LLMs dans l'annotation des données.

6 Remerciements

Ce travail a obtenu le soutien du gouvernement français dans le cadre du programme "France 2030", au sein de l'Institut de Recherche Technologique SystemX.

Références

[1] Rishabh Adiga, Lakshminarayanan Subramanian, and Varun Chandrasekaran. Designing informative metrics for few-shot example selection, 2024.

[2] Alan Akbik, Tanja Bergmann, Duncan Blythe, Kashif Rasul, Stefan Schweter, and Roland Vollgraf. FLAIR : An easy-to-use framework for state-of-the-art NLP. In *NAACL 2019, 2019 Annual Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 54–59, 2019.

[3] Sachi Angle, Pruthwik Mishra, and Dipti Mishra Sharma. Automated error correction and validation for POS tagging of Hindi. In *Proceedings of the 32nd Pacific Asia Conference on Language, Information and Computation*, Hong Kong, 1–3 December 2018. Association for Computational Linguistics.

[4] Dhananjay Ashok and Zachary C. Lipton. Promptner : Prompting for named entity recognition, 2023.

[5] David Batista and Matthew Antony Upson. nervaluate, October 2020.

[6] DeepSeek-AI. Deepseek-v3 technical report, 2024.

[7] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert : Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics : human language technologies, volume 1 (long and short papers)*, pages 4171–4186, 2019.

[8] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert : Pre-training of deep bidirectional transformers for language understanding, 2019.

[9] Á. García-Barragán, A. González Calatayud, O. Solarte-Pabón, et al. Gpt for medical entity recognition in spanish. *Multimedia Tools and Applications*, 2024.

[10] Nancy Ide, Christian Chiarcos, Manfred Stede, and Steve Cassidy. *Designing Annotation Schemes : From Model to Representation*, pages 73–111. Springer Netherlands, Dordrecht, 2017.

[11] Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mixtral of experts, 2024.

[12] Jan-Christoph Klie, Bonnie Lynn Webber, and Iryna Gurevych. Annotation error detection : Analyzing the past and present for a more coherent future, 2022. arXiv preprint.

[13] Mingchen Li, Yang Ye, Jeremy Yeung, Huixue Zhou, Huaiyuan Chu, and Rui Zhang. W-procer : Weighted prototypical contrastive learning for medical few-shot named entity recognition, 2023.

[14] Mingchen Li and Rui Zhang. How far is language model from 100

[15] Shifeng Liu, Yifang Sun, Bing Li, Wei Wang, and Xiang Zhao. Hamner : Headword amplified multi-span distantly supervised method for domain specific named entity recognition. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05) :8401–8408, Apr. 2020.

[16] OpenAI. Gpt-4 technical report, 2024.

[17] Nils Reimers and Iryna Gurevych. Sentence-bert : Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019.

[18] Joan Santoso, Patrick Sutanto, Billy Cahyadi, and Esther Setiawan. Pushing the limits of low-resource NER using LLM artificial data generation. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics : ACL 2024*, pages 9652–9667, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[19] Jingbo Shang, Liyuan Liu, Xiang Ren, Xiaotao Gu, Teng Ren, and Jiawei Han. Learning named entity tagger using domain-specific dictionary, 2018. arXiv preprint.

[20] Amber C Stubbs. *A methodology for using professional knowledge in corpus annotation*. Brandeis University, 2013.

[21] Ruixiang Tang, Xiaotian Han, Xiaoqian Jiang, and Xia Hu. Does synthetic data generation of llms help clinical text mining ?, 2023.

[22] Archana Tikayat Ray, Olivia J Pinon-Fischer, Dimitri N Mavris, Ryan T White, and Bjorn F Cole. aerobert-ner : Named-entity recognition for aerospace requirements engineering using bert. In *AIAA SCITECH 2023 Forum*, page 2583, 2023.

[23] 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, 2023.

[24] Qiang Wei, Amy Franklin, Trevor Cohen, and Hua Xu. Clinical text annotation–what factors are associated with the cost of time ? In *AMIA Annual Symposium Proceedings*, volume 2018, page 1552. American Medical Informatics Association, 2018.

[25] Urchade Zaratiana, Nadi Tomeh, Pierre Holat, and Thierry Charnois. GLiNER : Generalist model for named entity recognition using bidirectional transformer. In Kevin Duh, Helena Gomez, and Steven Bethard, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies (Volume 1 : Long Papers)*, pages 5364–5376, Mexico City, Mexico, June 2024. Association for Computational Linguistics.

[26] Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. Siren's song in the ai ocean : A survey on hallucination in large language models, 2023.