Des pipelines faciles à réutiliser pour comparer les performances d'outils de reconnaissance d'entités nommées sur les textes cliniques en français

T. Hubert[1,2], G. Vaillant[1,2], O. Birot[1,2], C. Arias[1,2], A. Neuraz[1,2,3], B. Rance[1,2,4] et A. Coulet[1,2,⋆]

[1] Inria Paris, Paris
[2] Centre de Recherche des Cordeliers, Inserm, Université Paris Cité, Sorbonne Université, Paris
[3] Hôpital Necker-Enfants malades, AP-HP, Paris
[4] Hôpital Européen Georges Pompidou, AP-HP, Paris

⋆ Email : adrien.coulet@inria.fr

Résumé

La prise en considération du contenu de textes cliniques des dossiers patients informatisés est centrale pour la conduite d'études observationnelles secondaires, et cela commence en général par la Reconnaissance d'Entités Nommées (NER en anglais). Cependant, les textes cliniques sont très hétérogènes notamment entre les pays, les centres de soins, les services, ce qui fait qu'il est difficile de savoir à l'avance comment les outils existants vont se comporter sur un corpus particulier. Dans cet article nous décrivons la comparaison des performances de quatre approches de NER sur trois corpus français à l'aide de pipelines d'analyse. Les résultats de la comparaison permettent d'observer une supériorité attendue des performances des modèles de langues par rapport aux approches par dictionnaire et questionne sur la nécessité d'affiner les modèles déjà préentraînés sur des textes biomédicaux. Les pipelines développés le sont avec la bibliothèque medkit, sont partagés et ont l'avantage d'être faciles à réutiliser et à adapter à de nouveaux environnement de travail, corpus ou outil de NER. Cette approche de partage de pipelines et de leurs composants nous semble être une alternative intéressante dans un contexte où les textes cliniques peuvent difficilement être partagés pour des raisons de confidentialité.

Mots-clés

Textes cliniques, Reconnaissance d'Entités Nommées, Evaluation, Open science

Abstract

The task of Named Entity Recognition (NER) is central for leveraging the content of clinical texts of Electronic Health Records in observational studies. However, clinical texts are highly heterogeneous between healthcare services and institutions, between countries and languages, making it hard to predict how existing tools may perform on a particular corpus. We compared four NER approaches on three French corpora and share our benchmarking pipeline in an open and easy-to-reuse manner, using the medkit Python library. We include in our pipelines fine-tuning operations with either one or several of the considered corpora. Our results illustrate the expected superiority of language models over a dictionary-based approach, and question the necessity of refining models already trained on biomedical texts. Beyond benchmarking, we believe sharing reusable and customizable pipelines for comparing fast-evolving NLP tools is a valuable contribution, since clinical texts themselves can hardly be shared for privacy concerns.

Keywords

Clinical texts, Named Entity Recognition, Benchmark, Open science

1 Introduction

La reconnaissance d'entités nommées (notée NER pour *Named Entity Recongnition* en anglais) est une tâche classique de traitement automatique des langues (TAL) qui consiste à identifier certains types d'entités dans des textes. Par exemple des mentions de personnes ou de lieux dans des textes du domaine général, ou des mentions de médicaments ou des pathologies dans des textes biomédicaux. Cette tâche est importante dans la conduite d'études observationnelles qui s'appuient sur l'utilisation secondaire de données de santé. En effet, ce genre d'études a souvent besoin de prendre en considération les textes cliniques des dossiers patients informatisés, comme les comptes-rendus d'examens ou d'hospitalisation, car ceux-ci contiennent une grande partie des informations cliniques des patients indisponibles par ailleurs. Les entités mentionnées dans ces textes peuvent par exemple être nécessaire à l'inclusion/exclusion de patients dans l'étude, ou à l'extraction des valeurs associées aux variables réponses ou co-variables associées des patients inclus. Cependant, les tâches d'extraction d'information, comme le NER, sont rendues complexes par l'hétérogénéité des textes cliniques, à la fois dans leur forme et leur contenu. En effet une information identique peut être formulée de façon différente selon les services, les établissements de santé, les pays et la langue locale, rendant difficile de savoir à l'avance comment un outil existant va se comporter sur un corpus particulier.

	QUAERO			CASM2			E3C		
	train	**val**	**test**	**train**	**val**	**test**	**train**	**val**	**test**
Documents	844	844	848	424	106	133	36	18	45
Phrases	1 569	1 514	1 454	6 320	1 613	2 173	509	241	642
LMP (cara.)	96	92	93	140	140	136	146	149	139
Entités (#)	**4 513**	**4 121**	**4 084**	**14 566**	**3 628**	**4 744**	**596**	**272**	**731**
Disorder (%)	29	25	24	48	46	47	100	100	100
Chemical (%)	21	25	25	23	26	23			
Procedure (%)	20	18	19	29	28	29			
Autres (%)	30	32	32						

TABLE 1 – Taille et contenu des trois corpus. LMP est la longueur moyenne des phrases en nombre de caractères.

Nous avons développé des pipelines expérimentaux permettant de comparer quatre outils de NER sur trois corpus annotés manuellement en français, incluant une approche par dictionnaire, deux *transformers*, et une approche générative. Au delà de la comparaison de performances relativement naïve, nous partageons les pipelines programmatiques suivant les principes de la science ouverte pour faciliter la réutilisation, l'ajustement et la comparaison des outils de NER sur d'autres corpus de textes cliniques. Ces pipelines sont implémentés avec medkit [1], une bibliothèque Python open source spécialement conçue pour ce genre d'usage.

2 Matériel et Méthodes

2.1 Trois corpus de textes cliniques

Nous avons considéré trois corpus de référence, dont le contenu est proche des textes cliniques. Ces corpus se composent de descriptions de cas cliniques, de notices de médicaments et de titres d'articles scientifiques. Un ou plusieurs types d'entités y sont annotés manuellement. Le tableau 1 présente leur contenu respectifs en termes de taille, de types d'entités annotés et de leur répartition en ensembles d'entraînement, de validation et de test.

Le corpus QUAERO [15] est composé de titres d'articles de la base documentaire MEDLINE et de notices de médicaments de l'Agence du Médicament Européenne (ou EMA pour *European Medicines Agency*). QUAERO contient des annotations manuelles pour des entités dont le type correspond à certains groupes sémantiques de l'UMLS [12], notamment Chemical and Drugs, Disorder, Procedures, plus d'autres (Anatomy, etc.).

Le corpus CASM2 est construit à partir du corpus CAS [6], auquel est ajouté des annotions faites de manière collaborative par des étudiants de Master de l'Université Paris Cité, avec les types d'entités suivants : problème, test et traitement. Nous proposons ici un alignement de ces trois types d'entités avec les groupes sémantiques UMLS utilisés dans QUAERO pour assurer l'homogénéité des annotations entre les corpus. Ainsi, problème est aligné avec Disorder, test avec Procedures et traitement avec Chemical and Drugs. Nous reconnaissons que cette alignement est simpliste et inadapté dans certains cas, mais ceux-ci sont relativement rares au sein de CASM2.

Le corpus E3C [10] regroupe des cas cliniques (énoncés de cas cliniques, motifs de consultations cliniques, descriptions d'examens physiques ou évaluations de l'état de patients) en cinq langues, annotés selon diverses méthodes (manuelle, semi-automatisée ou entièrement automatisée). Dans ce travail, nous n'avons gardé que le sous-ensemble français et annoté manuellement du corpus. Ce corpus est annoté avec un seul type d'entités : Disorder, que nous alignons avec le groupe sémantique Disorder de l'UMLS.

2.2 Quatre outils de NER

Les outils de NER analysent un texte et l'annotent automatiquement avec des étiquettes faisant référence à divers types d'entités. La bibliothèque medkit dispose d'implémentations originales, ainsi que d'encapsulations d'outils existants de NER et vise à faciliter leur interfaçage et la comparaison de leurs performances. Nous avons utilisé les quatre outils suivants implémentés ou encapsulés avec medkit.

UMLS matcher est un outils de NER qui utilise une mesure de similarité floue et un dictionnaire construit à partir de l'UMLS et ses groupes sémantiques pour trouver des entités [16]. Dans cet article, nous avons utilisé un seuil de similarité à 0.9 et une transformation en lettres minuscules des termes du dictionnaire et du texte à annoter.

Deux modèles de type BERT sont également comparés. Il s'agit de deux modèles basés sur RoBERTa [9], tous deux pré-entraînés sur des textes biomédicaux, puis affinés finement sur une tâche de NER. En effet, ce type de modèles de langues peut être pré-entraîné de façon globale, puis spécialisé pour une tâche particulière ou un type de textes particuliers. Le premier modèle est **DrBERT** [8], qui est pré-entraîné sur un corpus open source français de textes du Web appelé NACHOS. Parmi les différentes versions disponibles de DrBERT, nous avons utilisé la version 4GB. Le second modèle est **CamemBERT-bio** [17] qui lui est construit par un pré-entraînement de façon continue à partir de CamemBERT-base [11] sur trois corpus composés au total de 417 millions de mots.

GPT matcher utilise le modèle conversationnel ChatGPT 3.5-turbo en passant par la bibliothèque spaCy-llm [2].

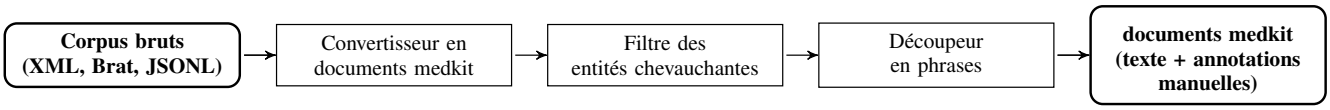

FIGURE 1 – Pipeline de prétraitement

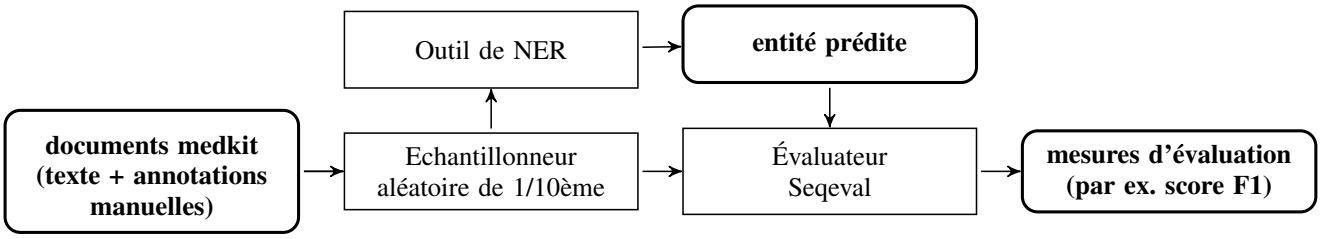

FIGURE 2 – Pipeline d'évaluation

Cette bibliothèque, associée à des instructions adpatées, permettent de reconnaître des entités nommées à partir d'un schéma d'annotation, c'est à dire les différents types d'entités possibles, accompagnées de leur définitions en langage naturel, et bien entendu du texte à annoter. Des éléments de contexte supplémentaires peuvent également être fournis au modèle en utilisant le principe du cheminement de pensés (*chain of thoughts* en anglais) [3] . Nous avons utilisé cette approche pour fournir au GPT matcher 12 phrases annotées manuellement issues de l'ensemble de test de QUAERO.

2.3 Les pipelines réutilisables

Nous avons composé des pipelines à l'aide d'opérations implémentés avec la bibliothèque medkit. Ces opérations constituent les étapes d'une chaîne de traitement définie par un pipeline. Elles peuvent par exemple permettre la transformation des données en entrée ou en sortie pour s'adapter à divers standards, ou de traiter les données. Une opération medkit peut être une implémentation originale, ou l'encapsulation d'un outil externe, tel qu'un outil de NER développé par un tiers ou un modèle partagé sur les portails Hugging Face ou spaCy [18, 7]. Pour ce travail nous avons développé trois pipelines : les pipelines de prétraitement, d'entraînement et d'évaluation.

Pipeline de prétraitement Ce pipeline est composé de trois opérations. La première convertit le corpus en entrée depuis son format d'origine (XML, Brat ou JSONL) dans le format de représentation interne medkit. La seconde opération filtre les entités chevauchantes en éliminant l'annotation la plus petite (c'est à dire l'annotation inclue dans l'autre dans le cas d'inclusions). Cette étape est nécessaire car l'UMLS Matcher et nos deux modèles de type BERT ne peuvent pas prendre en entrée des entitées chevauchantes. La troisième opération découpe le texte en phrase pour faciliter les phases suivantes d'entraînement et d'évaluation. La Figure 1 représente graphiquement l'enchaînement de ces opérations.

Pipeline d'entraînement Ce pipeline est utilisé seulement par les modèles BERT, pour leur ajustements fins. medkit dipose d'opérations pour permettre cet ajustement fin de modèles de classification de tokens disponibles sur Hugging Face. Pour les tâches de NER, ce type d'opération prend en entrée un ensemble prédéfini de types d'entités, un modèle pré-entrainé et un ensemble d'entraînement avec des annotations. Nous distinguons deux utilisations différents de ce pipeline pour l'ajustement fin des modèles BERT. La première utilisation que nous dénommons *spécifique*, utilise un ensemble d'entraînement issu d'un seul corpus ; tandis que la deuxième utilisation, dénommée *générique*, utilise l'agrégation des ensembles d'entraînement des trois corpus QUAERO, CASM2 et E3C. En terme d'expérimentation avec ce pipeline, nous proposons d'ajuster finement chacun des deux modèles BERT avec chaque corpus isolé, en utilisant les étiquettes du corpus concerné comme les étiquettes à reconnaître. Ensuite, les 3 corpus sont fusionnés en un seul pour affiner chacun des deux modèles avec les mêmes paramètres d'entraînement. L'objectif de cette expérience est de comparer l'effet d'un ajustement fin relativement *spécifique* à partir d'un seul corpus, avec un affinement fin relativement *générique* à partir d'un corpus plus grand résultant de la combinaison de nos trois corpus.

Pipeline d'évaluation Un dixième du corpus de test est échantillonné de manière aléatoire, puis soumis à un outils de NER. Les entités prédites (c'est-à-dire, la sortie de l'outil de NER), ainsi que les entités originales (c'est-à-dire, les annotations manuelles du corpus d'origine), sont converties suivant le schéma IOB2. Ensuite, les métriques d'évaluation (score F1, précision, rappel) sont calculées avec la bibliothèque seqeval [14]. La Figure 2 présente de façon graphique le pipeline d'évaluation des outils de NER. Les scores F1 sont calculés d'abord pour chaque type d'entité et ensuite agrégés par une pondération par l'annotation pour chaque type dans le corpus. Les annotations spécifiques "not in any chunk" (c'est-à-dire, les tokens annotés comme "O", signifiant qu'ils ne sont associés à aucune annotation) sont exclues du calcul du score F1 pondéré, diminuant ainsi le score final par rapport à d'autres travaux qui le prennent en compte dans l'évaluation. La variabilité des métriques est évaluée en exécutant plusieurs fois le même pipeline et en calculant la moyenne et l'écart type des performances.

	QUAERO	CASM2	E3C	Moyenne
UMLS matcher	0,48 ± 0,02	0,31 ± 0,02	**0,61 ± 0,03**	0,41
DrBERT spécifique	0,42 ± 0,01	0,41 ± 0,02	0,4 ± 0,04	0,41
DrBERT générique	0,44 ± 0,02	0,42 ± 0,02	0,43 ± 0,04	0,43
CamemBERT-bio spécifique	0,57 ± 0,02	0,57 ± 0,0	0,56 ± 0,02	0,57
CamemBERT-bio générique	**0,59 ± 0,02**	**0,58 ± 0,01**	0,52 ± 0,04	**0,58**
GPT matcher	0,52 ± 0,03	0,34 ± 0,03	0,55 ± 0,04	0,43

TABLE 2 – Scores F1 pondérés, par outil de NER et par ensemble d'entraînement de corpus.

	Chemical		Disorder		Procedure	
	QUAERO	CASM2	QUAERO	CASM2	QUAERO	CASM2
UMLS matcher	0,55 ± 0,06	0,35 ± 0,05	**0,58 ± 0,05**	0,32 ± 0,04	0,29 ± 0,06	0,25 ± 0,05
DrBERT générique	0,67 ± 0,05	0,43 ± 0,06	**0,58 ± 0,03**	0,4 ± 0,03	0,58 ± 0,04	0,45 ± 0,05
CamemBERT-bio gén,	**0,69 ± 0,04**	**0,66 ± 0,07**	0,55 ± 0,05	**0,61 ± 0,04**	**0,6 ± 0,08**	**0,67 ± 0,03**
GPT matcher	0,62 ± 0,04	0,33 ± 0,05	**0,58 ± 0,03**	0,42 ± 0,04	0,4 ± 0,02	0,24 ± 0,02

TABLE 3 – Scores F1 par type d'annotations sur les corpus QUAERO et CASM2. Le corpus E3C a un seul type d'annotation et pour cette raison, les performance pour ce type unique sont celles rapportées dans le Tableau 2.

Les implémentations de ces pipelines, ainsi que leur documentation, sont disponibles pour réutilisation à l'adresse suivante : `https://medkit.readthedocs.io/en/stable/cookbook/ner_benchmark/`.

3 Résultats

Le pipeline de prétraitement a été exécuté pour chaque partition (train, test et validation) de chaque corpus. Le pipeline d'entraînement a été exécuté plusieurs fois, résultant en trois modèles ajustés finement , dit spécifiques (un par corpus) et un dit générique pour chacun des deux modèles BERT ; ce qui résulte en quatre modèles pour DrBERT et quatre pour CamemBERT-bio. Le pipeline d'évaluation a été exécuté sur chaque corpus pour chaque outil de NER, c'est à dire UMLS matcher, les quatre modèles affinés de DrBERT, les quatre de CamemBERT-bio et GPT matcher. L'exécution du pipeline d'évaluation a été répétée 10 fois pour chaque outil de NER, mais seulement 3 fois pour le GPT matcher pour éviter un coût excessif.

Le Tableau 2 donne les scores F1 pondérés pour chaque outil de NER, ainsi que la moyenne pondérée par la taille des corpus (dernière colonne). De façon générale, la version affinée générique de CamemBERT-bio produit les meilleurs F1, avec la meilleure moyenne (0.58). Elle produit également les meilleures performances sur QUAERO et CASM2 avec des F1 à 0.59 ± 0.02 and 0.58 ± 0.01, respectivement. L'UMLS matcher, qui s'appuie sur similarité floue entre chaînes de caractères obtient les meilleures performances sur le corpus E3C avec un F1 de 0.61 ± 0.03, bien qu'il obtienne des performances relativement faibles avec une moyenne de 0.41. Le GPT matcher donne des performances assez inégales, selon les corpus, avec une moyenne de 0.43.

Le Tableau 3 donne le détail des scores F1 pour les 3 types d'entités les plus fréquents dans QUAERO et CASM2, c'est à dire Chemical, Disorder et Procedure. Les modèles BERT affiné finement sur un seul corpus sont omis par soucis de simplification. CamemBERT-bio produit les meilleures performances pour chaque type d'annotation sauf pour le type Disorder de QUAERO, pour lequel les trois autres outils donnent des performances équivalentes.

4 Discussion

Premièrement, nous n'observons pas de grandes différences entre les performances des modèles BERT lorsqu'ils sont ajustés finement avec des corpus plus grands et plus diversifiés plutôt qu'avec un seul corpus. Cela pourrait s'expliquer par le fait que DrBERT et CamemBERT-bio ont déjà "vu" des ensembles de textes biomédicaux grands et divers, ce qui peut expliquer un phénomène de saturation. Dans nos expériences, CamemBERT-bio présente de meilleures performances que DrBERT ce qui est assez consistants avec les observations rapportés dans études comme Naguib *et al.* [13]. En comparaison, nous remarquons que le niveau des métriques et relativement bas, par rapports aux performances attendues pour une tâche de NER, notamment avec des modèles de type BERT [13, 4]. Ici, notre réutilisation naïve des modèles disponibles, notamment sans ajustement des hyper-paramètres, participe à expliquer cette différence. Les phases de prétraitements, comme la tokenisation, appliquées dans les autres travaux seraient également à investiguer pour vérifier si elles différent des nôtres. Enfin notre homogénéisation des types d'entités entre corpus a probablement un impact, mais moindre que notre non considération de l'étiquetage "not in any chunk". Une observation partagée avec d'autres études est l'avantage pour la NER d'utiliser les modèles de langue masqués (*i.e.*, les modèles de type BERT), par rapport aux grands modèles de langage et au *prompting*. Nous modérons cette observation par le fait que nous n'avons considéré qu'un seul LLM, qui n'est pas le plus récent et une approche de prompting plutôt simple. Il serait intéressant de voir quel serait l'impact de fournir davantage d'exemples et des exemples plus divers. Nous notons également que l'outil de NER basé sur

un dictionnaire de l'UMLS obtient les meilleures performances avec E3C et son type unique d'annotations (Disorder). Ceci va dans le sens des études qui illustrent que les approches par apprentissage profond ne sont pas toujours les meilleures [5].

Nous observons des tendances non cohérentes des résultats sur les trois corpus considérés, ce qui rend difficile de tirer des conclusions générales sur plusieurs aspects, par exemple les résultats sur E3C ou avec les Disorder de QUAERO sont incohérents avec les autres. Cela souligne notre constat initial concernant l'hétérogénéité des textes cliniques et le besoin d'outils pour une évaluation flexible des approches et accessibles à des non-experts en TAL. À cette fin les pipelines que nous partageons à l'aide de la bibliothèque medkit sont faciles à adapter dans divers environnements et faciles à adapter pour inclure un nouveaux corpus ou de nouveaux outils de NER.

Remerciements

Ce travail a bénéficié de financement d'Inria, Inria Paris et d'une subvention gouvernementale gérée par l'Agence Nationale de la Recherche dans le cadre du programme France 2030, référence ANR-22-PESN-0007, ShareFAIR.

Références

[1] Répertoire de la bibliothèque medkit. `https://github.com/medkit-lib/medkit`, 2024.

[2] Répertoire de la bibliothèque SpaCy-LLM. `https://github.com/explosion/spacy-llm`, 2024.

[3] Dhananjay Ashok and Zachary C Lipton. PromptNER : Prompting for Named Entity Recognition. *arXiv preprint arXiv :2305.15444*, 2023.

[4] Aman Berhe, Guillaume Draznieks, Vincent Martenot, Valentin Masdeu, Lucas Davy, and Jean-Daniel Zucker. AliBERT : A pre-trained language model for French biomedical text. In *The 22nd Workshop on Biomedical Natural Language Processing and BioNLP Shared Tasks*, pages 223–236, Toronto, Canada, July 2023.

[5] David Chen, Sijia Liu, Paul Kingsbury, Sunghwan Sohn, Curtis B Storlie, Elizabeth B Habermann, James M Naessens, David W Larson, and Hongfang Liu. Deep learning and alternative learning strategies for retrospective real-world clinical data. *NPJ digital medicine*, 2(1) :43, 2019.

[6] Natalia Grabar, Clément Dalloux, and Vincent Claveau. CAS : corpus of clinical cases in French. *Journal of Biomedical Semantics*, 11(1) :1–10, 2020.

[7] Matthew Honnibal and Ines Montani. spaCy 2 : Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.

[8] Yanis Labrak, Adrien Bazoge, Richard Dufour, Mickael Rouvier, Emmanuel Morin, Béatrice Daille, and Pierre-Antoine Gourraud. DrBERT : A robust pretrained model in French for biomedical and clinical domains. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, pages 16207–16221, July 2023.

[9] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. RoBERTa : A robustly optimized BERT pretraining approach. *arXiv preprint arXiv :1907.11692*, 2019.

[10] Bernardo Magnini, Begona Altuna, Alberto Lavelli, Manuela Speranza, and Roberto Zanoli. The E3C project : European clinical case corpus. *Language*, 1(L2) :L3, 2021.

[11] Louis Martin, Benjamin Muller, Pedro Javier Ortiz Suárez, Yoann Dupont, Laurent Romary, Éric de La Clergerie, Djamé Seddah, and Benoît Sagot. CamemBERT : a tasty French language model. *arXiv preprint arXiv :1911.03894*, 2019.

[12] Alexa T McCray, Anita Burgun, and Olivier Bodenreider. Aggregating UMLS semantic types for reducing conceptual complexity. *Studies in health technology and informatics*, 84(0 1) :216, 2001.

[13] Marco Naguib, Xavier Tannier, and Aurélie Névéol. Few shot clinical entity recognition in three languages : Masked language models outperform LLM prompting, 2024.

[14] Hiroki Nakayama. seqeval : A Python framework for sequence labeling evaluation. *Software available from https ://github. com/chakki-works/seqeval*, 2018.

[15] Aurélie Névéol, Cyril Grouin, Jeremy Leixa, Sophie Rosset, and Pierre Zweigenbaum. The QUAERO French medical corpus : A ressource for medical entity recognition and normalization. In *Proc of BioText-Mining Work*, pages 24–30, 2014.

[16] Naoaki Okazaki and Jun'ichi Tsujii. Simple and efficient algorithm for approximate dictionary matching. In *Proceedings of the 23rd International Conference COLING 2010*, pages 851–859, August 2010.

[17] Rian Touchent, Laurent Romary, and Eric de La Clergerie. CamemBERT-bio : a tasty French language model better for your health. *arXiv preprint arXiv :2306.15550*, 2023.

[18] Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, and Pierric Cistac et al. HuggingFace's transformers : State-of-the-art natural language processing, 2020.