



**HAL**  
open science

## Étude sur la classification d'entités nommées

I Keraghel, S Morbieu, M Nadif

► **To cite this version:**

I Keraghel, S Morbieu, M Nadif. Étude sur la classification d'entités nommées. Les 28èmes rencontres de la Société Francophone de Classification (SFC), Jul 2024, Strasbourg, France. hal-04848145

**HAL Id: hal-04848145**

**<https://hal.science/hal-04848145v1>**

Submitted on 19 Dec 2024

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Public Domain

# Étude sur la classification d'entités nommées

I. Keraghel<sup>1,2</sup>, S. Morbieu<sup>2</sup>, M. Nadif<sup>1</sup>

<sup>1</sup> Centre Borelli UMR9010, Université Paris Cité, 75006 Paris

<sup>2</sup> Kernix, 75014, Paris

## Résumé

*La reconnaissance et la classification d'entités nommées au sein de documents est un processus permettant d'extraire des informations pertinentes pour une tâche donnée. Cependant, les méthodes disponibles pour effectuer ce processus varient selon leur nature et leur mode de fonctionnement. Dans cet article, nous nous intéressons principalement aux performances des frameworks de reconnaissance d'entité nommée. Une étude comparative entre les frameworks les plus connus aujourd'hui est proposée afin de sélectionner l'approche qui s'adapte le mieux aux différents jeux de données.*

## Mots-clés

*Reconnaissance d'entités nommées, classification, apprentissage profond, transformers.*

## Abstract

*Named entity recognition and classification is a process for extracting relevant information for a given task. However, the methods available to perform this process vary in nature and mode of operation. In this paper, we are mainly interested in the performance of named entity recognition frameworks. A comparative study between the most known frameworks is proposed in order to select the approach that best fits the different datasets.*

## Keywords

*Named entity recognition, classification, machine learning, deep learning, transformers.*

## 1 Introduction

La reconnaissance et la classification d'entités nommées (NERC) est une technique qui consiste à localiser des concepts clés susceptibles de représenter des entités nommées dans des textes, afin de les classer dans un ensemble prédéfini de classes dépendant de la problématique à traiter. Cette technique apparaît aujourd'hui comme une composante principale dans plusieurs domaines de traitement automatique du langage naturel dont la traduction automatique [1], les systèmes de questions-réponses [2] et la recherche d'information [3].

De nombreux systèmes de NERC [4] ont été développés en particulier dans le monde anglo-saxon, puis dans d'autres langues comme le français. Dans [5], les auteurs présentent un système basé sur des règles permettant d'extraire des

noms d'entreprises. Les premiers systèmes de NERC utilisaient des algorithmes basés sur des règles manuscrites, des lexiques et des caractéristiques orthographiques. Ces systèmes ont été d'abord suivis par des algorithmes basés sur l'apprentissage automatique comme [6], et ensuite par des réseaux de neurones [7] et récemment des transformers [8].

Pour la NERC, cinq familles de méthodes sont souvent évoquées :

- celles basées sur des règles, qui ne nécessitent pas l'acquisition d'un jeu de données annoté car les règles sont élaborées par un expert ;
- l'apprentissage automatique non supervisé ;
- l'apprentissage automatique supervisé, avec une ingénierie pour créer les caractéristiques ;
- l'apprentissage profond, qui s'appuie sur l'utilisation des réseaux de neurones comme les LSTM et CNN ;
- et enfin celles basées sur les transformers qui s'appuient sur l'utilisation du mécanisme d'attention.

Dans la littérature, il existe un bon nombre d'études dans le domaine des NERC. En particulier, l'étude [6] présente une vue d'ensemble des techniques de NERC, depuis l'apparition des systèmes à base de règles jusqu'aux systèmes basés sur l'apprentissage automatique. Les auteurs dans [9] ont analysé les travaux pertinents de la NERC dans les textes biomédicaux, sur la période de 2007-2009. Marrero et al. [10] ont résumé les travaux sur la NERC d'un point de vue théorique et pratique, et ont démontré que cette tâche est loin d'être résolue en 2013. Des études récentes ont inclus dans leurs analyses les méthodes basées sur des réseaux de neurones et les transformers [11].

Dans cet article, nous nous intéressons principalement aux performances des frameworks de NERC sur différents jeux de données. Une étude comparative est proposée afin de sélectionner l'approche qui s'adapte le mieux sur les données testées. Pour ce faire, nous présentons dans un premier temps les approches et les frameworks de NERC les plus connus que nous avons retenus (Hugging Face, Stanford CoreNLP, NLTK, OpenNLP, spaCy et Flair) afin de dresser notre étude comparative. Ensuite, nous présenterons nos données à tester ainsi que notre méthodologie d'expérimentation que nous évaluerons en se basant les métriques appropriées.

## 2 Méthodes

Plusieurs types de méthodes permettent la reconnaissance et l'extraction des entités nommées. Nous discutons dans cette partie des différentes approches.

### 2.1 Règles et base de connaissances

Les systèmes de reconnaissance d'entités nommées à base de connaissances s'appuient sur des ressources lexicales et des connaissances spécifiques au domaine [12]. Ces approches impliquent de confectionner des règles d'extraction, à la manière d'une expression rationnelle. Elles se fondent sur l'extraction des entités nommées en utilisant des marqueurs lexicaux [13] et des ressources terminologiques de noms propres [12].

Les marqueurs lexicaux sont des indices qui encadrent l'entité nommée et qui permettent de dévoiler sa présence. Par exemple, le système CasEN [14] est destiné à reconnaître les organisations politiques.

Les ressources terminologiques regroupent généralement une collection d'entités nommées les plus fréquentes dans un domaine d'application. Par exemple, le système *ProMiner* [15] utilise un dictionnaire de noms des protéines et de gènes.

### 2.2 Apprentissage automatique

Les méthodes classiques d'apprentissage automatique permettent de résoudre les lacunes des méthodes basées sur des règles. Généralement, ces méthodes sont divisées en trois catégories : non supervisées, semi-supervisées et supervisées.

Les méthodes non supervisées exploitent les ressemblances dans les données afin d'inférer le bon modèle. Elles regroupent des syntagmes selon les propriétés communes qu'ils présentent [16].

Les méthodes semi-supervisées apprennent à partir d'un petit nombre d'exemples annotés, les données non annotées étant annotées en fonction de la similitude entre les échantillons non annotés et ceux annotés. Ceci augmente efficacement le volume de données d'apprentissage [17].

En utilisant les méthodes supervisées, la tâche de NERC a été transformée en deux sous problèmes : la classification et l'étiquetage de séquences. On cherche alors à reproduire un schéma d'annotation appris d'un corpus déjà annoté. Les prédictions sont faites en utilisant le modèle déjà formé sur les données annotées. Les algorithmes les plus courants dans cette catégorie sont le modèle de Markov caché (HMM) [18, 19], le modèle d'entropie maximale (ME) [20], les machines à vecteurs de support (SVM) [21] et les champs aléatoires conditionnels (CRF) [22].

### 2.3 Apprentissage profond

Ces dernières années, les méthodes d'apprentissage profond ont été largement utilisées pour la NERC dans plusieurs domaines, et ont donné de bons résultats sur la plupart des corpus en dépassant les modèles traditionnels [23]. Cette famille de méthodes résout la tâche de NERC en exploitant (1) la sémantique en s'appuyant sur des word embeddings tels que Word2vec [24], GloVe [25], fastText [26],

(2) les caractéristiques des entités nommées en s'appuyant sur les embeddings de caractères tels que les réseaux neuronaux convolutifs (CNN) [27], (3) l'ordre des mots par le biais des réseaux récurrents (RNN) [28], (4) l'encodage du contexte en vue de la détection des entités nommées en utilisant des RNN ou des CNN, (5) la modélisation probabiliste des étiquettes en utilisant un CRF [28].

### 2.4 Modèles de langage

Depuis 2017, le traitement automatique du langage a fait un grand pas en avant avec l'avènement des *transformers* [29]. Dans le domaine de NERC, ces modèles peuvent être utilisés afin d'extraire des vecteurs de caractéristiques qui servent d'entrée pour d'autres modèles tels que le CRF ou le LSTM, comme ils peuvent être entraînés directement pour la tâche de classification d'entités nommées. Plusieurs auteurs ont étudié l'influence de ces méthodes dans des tâches de NERC [30], et ont observé que leur utilisation, que ce soit comme modèle de représentation de caractères ou comme classifieur, améliore considérablement les performances de leur modèle.

## 3 Frameworks

Nous listons dans cette partie les principaux frameworks de NERC.

**spaCy**<sup>1</sup> : une bibliothèque libre et open source dédiée au traitement automatique du langage naturel en Python. Parmi les fonctionnalités disponibles dans SpaCy, on peut citer : tokenisation, classification, Parts-of-Speech tagging et NER. Il existe plusieurs modèles pré-entraînés dans spaCy que nous pouvons utiliser pour des tâches telles que NER, extraction d'informations, etc.

**NLTK**<sup>2</sup> : une suite de modules Python dédiés au traitement naturel du langage [31]. NLTK intègre plus de 50 corpus et ressources lexicales telles que WordNet. Au contraire de spaCy qui regroupe des algorithmes adaptés à différentes problématiques, et qui sont gérés par la bibliothèque, NLTK laisse la liberté aux utilisateurs de choisir parmi une large gamme d'algorithmes.

**Stanford CoreNLP**<sup>3</sup> : une bibliothèque qui implémente en Java des modèles de reconnaissance d'entités nommées basés sur des CRF [32]. Malheureusement, les modèles proposés ne sont pas disponibles pour toutes les langues.

**Apache OpenNLP**<sup>4</sup> : une bibliothèque Apache [33] qui prend en charge les tâches NLP les plus courantes, comme l'extraction d'entités nommées, la détection de langue, l'étiquetage morpho-syntaxique, le découpage, etc. Au contraire des autres frameworks qui détectent les entités nommées avec le même modèle, OpenNLP propose un modèle spécialisé pour chaque type d'entité nommée.

**Flair**<sup>5</sup> : une bibliothèque open source qui permet de mettre en place un pipeline de traitement automatique de langage

1. <https://spacy.io/>

2. <https://www.nltk.org/>

3. <https://stanfordnlp.github.io/CoreNLP/>

4. <https://opennlp.apache.org/>

5. <https://github.com/flairNLP/flair>

naturel supportant des applications multilingues [34]. Flair permet d'utiliser de nombreux modèles de langage, comme Flair, BERT et CamemBERT. Elle offre aussi la possibilité de les combiner.

**Hugging Face** : une bibliothèque créée en 2015<sup>6</sup>. Elle fournit des technologies de traitement du langage naturel open source. Hugging Face propose deux types de services : des services gratuits et des services payants destinés aux entreprises. Hugging Face doit sa popularité à sa bibliothèque transformers, qui offre une API permettant d'accéder à plusieurs modèles pré-entraînés notamment pour la reconnaissance d'entités nommées comme BERT, CamemBERT, etc. Hugging Face propose également une plateforme de collaboration permettant aux utilisateurs de créer, entraîner et partager leurs modèles de deep learning.

## 4 Évaluation

Nos expériences ont été menées sur deux jeux de données issues de sources diverses, et permettant de rechercher des entités nommées dans des classes telles que les noms de personnes, les noms d'organisations, les noms de lieux.

### 4.1 Jeux de données

**CoNLL-2003** : un jeu de données constitué principalement d'articles de presse provenant de Reuters, et se concentre sur quatre types d'entités nommées : les personnes (PER), les lieux (LOC), les organisations (ORG) et les noms d'entités diverses (MISC). CoNLL-2003 couvre deux langues (anglais et allemand), et est utilisé pour étalonner les modèles sur la tâche de NERC. Dans notre étude, nous n'avons utilisé que l'anglais. Pour chaque langue, nous disposons de trois ensembles de données : apprentissage, test, et validation.

**WikiANN** : un jeu de données composé d'articles Wikipédia annotés avec trois types d'entités nommées : les personnes, les lieux et les organisations. Il a été construit en utilisant les entités liées dans les pages Wikipédia pour 282 langues différentes. Nous avons utilisé la version de [35], qui prend en charge 176 des 282 langues du corpus WikiANN original. Cette version contient 40 000 exemples dans sa base anglaise, mais nous nous sommes limités à 10 000 exemples (6 000 pour l'apprentissage, 2 000 pour la validation et 2 000 pour les tests) afin d'évaluer les algorithmes sur un jeu de données moins fourni.

Les tables 1 et 2 résument les caractéristiques de ces jeux de données.

### 4.2 Mesures

L'évaluation des systèmes de reconnaissance d'entités nommées repose sur la comparaison des prédictions avec une base de référence en utilisant une évaluation exacte ou relâchée. Dans l'évaluation exacte, les contours et la classe de l'entité nommée doivent correspondre à la base de référence. En revanche, l'évaluation relâchée repose sur un système de score où chaque entité avec la bonne classe est

TABLE 1 – Description des jeux de données

| Données | Ensemble      | Articles | Phrases | Tokens  |
|---------|---------------|----------|---------|---------|
| CoNLL   | apprentissage | 946      | 14 041  | 203 621 |
|         | validation    | 216      | 3 250   | 51 362  |
|         | test          | 231      | 3 453   | 46 435  |
| WikiANN | apprentissage | -        | 20 000  | 160 394 |
|         | validation    | -        | 10 000  | 80 536  |
|         | test          | -        | 10 000  | 80 326  |

TABLE 2 – Nombres d'entités nommées

| Données | Ensemble      | Catégories d'entités nommées |       |       |       |
|---------|---------------|------------------------------|-------|-------|-------|
|         |               | PER                          | LOC   | ORG   | MISC  |
| CoNLL   | apprentissage | 6 600                        | 7 140 | 6 321 | 3 438 |
|         | validation    | 1 842                        | 1 837 | 1 341 | 922   |
|         | test          | 1 617                        | 1 668 | 1 661 | 702   |
| WikiANN | apprentissage | 9 164                        | 9 345 | 9 422 | -     |
|         | validation    | 4 635                        | 4 834 | 4 677 | -     |
|         | test          | 4 556                        | 4 657 | 4 745 | -     |

créditée, même si les contours ne sont pas exacts. Les entités sont aussi créditées si les contours sont bons, malgré l'inexactitude de la classe.

Les campagnes d'évaluation de MUC [36] et ACE [37] reposent sur des méthodes d'évaluation relâchée. Cependant, CoNLL-2003 [38] adopte l'évaluation exacte, qui représente la méthode la plus utilisée aujourd'hui afin d'évaluer les systèmes de reconnaissance d'entités nommées.

Pour l'évaluation des entités nommées, on utilise souvent des métriques classiques comme la précision, le rappel et le F1-score que nous rappelons ci-après.

- La précision donne le nombre d'entités nommées bien reconnues par le modèle rapporté au nombre total d'entités nommées

$$Precision = \frac{VP}{VP + FP} \quad (1)$$

où VP désigne le nombre de Vrais Positifs et FP de Faux Positifs.

- Le rappel mesure le nombre d'entités nommées pertinentes retrouvées par le modèle au regard du nombre total d'entités nommées pertinentes :

$$Rappel = \frac{VP}{VP + FN} \quad (2)$$

où FN est le nombre de Faux Négatifs.

- Le f1-score permet d'évaluer la capacité du modèle à détecter efficacement les entités nommées, en faisant un compromis entre la précision et le rappel.

$$F1 = 2 \times \frac{precision \times rappel}{precision + rappel} \quad (3)$$

## 5 Expériences

Les trois étapes suivantes ont été réalisées :

- **Mise sous un format commun des jeux de données** : Chaque portion de texte est décrite par sa

6. <https://huggingface.co/>

TABLE 3 – Comparaison des frameworks de reconnaissance d’entités nommées

| Frameworks       | Algorithmes                 | CoNLL-2003   |              |              | WikiANN      |              |              |
|------------------|-----------------------------|--------------|--------------|--------------|--------------|--------------|--------------|
|                  |                             | Precision    | Rappel       | F1           | Precision    | Rappel       | F1           |
| Spacy            | en_core_web_sm              | 82.72        | 81.94        | 82.33        | 73.44        | 72.71        | 73.07        |
|                  | en_core_web_md              | 84.96        | 84.54        | 84.75        | 80.12        | 79.05        | 79.58        |
|                  | en_core_web_lg              | 86.53        | 86.63        | 86.58        | 81.25        | 80.16        | 80.70        |
|                  | en_core_web_trf             | 89.47        | 89.70        | 89.58        | <b>85.05</b> | <b>85.44</b> | <b>85.24</b> |
| Hugging Face     | xlm-roberta-large           | <b>92.10</b> | <b>93.50</b> | <b>92.80</b> | 84.27        | 84.29        | 84.28        |
|                  | electra-large-discriminator | 92.02        | 92.12        | 92.07        | 83.21        | 84.11        | 83.66        |
|                  | bert-large-cased            | 91.23        | 92.37        | 91.80        | 84.80        | 84.19        | 84.49        |
|                  | bert-base-NER               | 90.65        | 91.93        | 91.29        | 83.58        | 85.40        | 84.48        |
| NLTK             |                             | 88.10        | 87.00        | 87.55        | 75.19        | 78.02        | 76.58        |
| Flair            |                             | 91.03        | 90.53        | 90.72        | 81.67        | 79.36        | 80.29        |
| Stanford CoreNLP |                             | 88.12        | 87.20        | 87.66        | 74.50        | 77.81        | 76.00        |
| Apache OpenNLP   |                             | 91.57        | 74.84        | 81.64        | 83.16        | 68.08        | 74.85        |

classe (type d’entité nommée ou classe autre) ainsi que sa position dans le texte. Chaque framework étant accompagné de sa propre représentation de données une conversion de ce format pivot vers chacun des formats spécifiques à chaque framework est effectué.

- **Recherche des valeurs d’hyper-paramètres** : Les valeurs par défaut, celles des modèles pré-entraînés librement disponibles ainsi qu’une grille ont été considérées. Les valeurs des modèles pré-entraînés se sont avérées être les meilleures excepté pour les modèles Hugging Face sur les données WikiANN. Les valeurs retenues sont `learning_rate = 2e-5`, `batch_size = 16`, `number_of_epochs = 20` et `weight_decay = 0.01`.
- **Évaluation** : Les métriques discutées précédemment ont été utilisées en comparant les labels prédits avec la base de référence. Le protocole exact basé sur l’évaluation CoNLL-2003 a été retenu, c’est-à-dire que les contours et la classe de l’entité nommée doivent correspondre à la base de référence (cf. la section 4.2). Les résultats sont donnés dans Table 3.

Pour le corpus CoNLL-2003, nous constatons que les modèles issus du framework Hugging Face, essentiellement basés sur des transformers, surpassent nettement les autres méthodes. Contrairement à ce qui a été observé dans l’étude [39], nous avons remarqué que OpenNLP surpasse Stanford CoreNLP.

Sur un jeu de données bien équilibré comme wikiANN où les proportions des entités nommées sont égales, nous constatons aussi que les architectures basées sur des transformers donnent toujours les meilleurs résultats, avec une petite différence en faveur du modèle transformer de spaCy, qui donne des résultats légèrement meilleurs que ceux d’Hugging Face. Il convient également de préciser que les hyper-paramètres utilisés pour entraîner un modèle sur un jeu de données spécifique ne donnent pas nécessairement les meilleurs résultats sur un autre jeu de données, même si les deux semblent proches et partagent les mêmes classes. Par exemple pour le modèle *lm-roberta-large*, nous avons remarqué que l’optimisation de certains hyper-paramètres

a un impact important sur les performances de ce modèle. Cette optimisation nous a permis de gagner environ 4 points en F1-score.

## 6 Conclusion et perspectives

Cet article présente les principales méthodes souvent utilisées pour la reconnaissance d’entités nommées. Nous avons évalué les frameworks les plus populaires permettant de faire de la reconnaissance d’entités nommées sur deux jeux de données. Les résultats de nos expériences ont montré que les méthodes basées sur des transformers sont significativement meilleures que les autres méthodes basées sur les réseaux de neurones ou les méthodes probabilistes. Cependant, les performances dépendent des valeurs des hyper-paramètres. Les valeurs d’hyper-paramètres des modèles pré-entraînés donnent de bons résultats même sur un jeu de données (WikiANN) sur lequel ils n’ont pas été réglés. Une étude plus complète des valeurs des hyper-paramètres et une évaluation sur d’autres jeux de données constituent une piste de futurs travaux.

## Références

- [1] Bogdan Babych and Anthony Hartley. Improving machine translation quality with automatic named entity recognition. In *Proceedings of the 7th International EAMT workshop on MT and other language technology tools, Improving MT through other language technology tools, Resource and tools for building MT at EACL 2003*, 2003.
- [2] Diego Mollá, Menno Van Zaanen, and Daniel Smith. Named entity recognition for question answering. In *Proceedings of the Australasian language technology workshop 2006*, pages 51–58, 2006.
- [3] Jiafeng Guo, Gu Xu, Xueqi Cheng, and Hang Li. Named entity recognition in query. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval*, pages 267–274, 2009.
- [4] Ralph Grishman and Beth M Sundheim. Message understanding conference-6 : A brief history. In *CO-*

- LING 1996 Volume 1 : The 16th International Conference on Computational Linguistics*, 1996.
- [5] Lisa F Rau. Extracting company names from text. In *Proceedings the Seventh IEEE Conference on Artificial Intelligence Application*, pages 29–30. IEEE Computer Society, 1991.
- [6] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Linguisticae Investigationes*, 30(1) :3–26, 2007.
- [7] Ronan Collobert. Deep learning for efficient discriminative parsing. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, pages 224–232. JMLR Workshop and Conference Proceedings, 2011.
- [8] Kai Labusch, Preußischer Kulturbesitz, Clemens Neudecker, and David Zellhöfer. Bert for named entity recognition in contemporary and historical german. In *Proceedings of the 15th conference on natural language processing, Erlangen, Germany*, pages 8–11, 2019.
- [9] Rodrigo Rafael Villarreal Goulart, Vera Lúcia Strube de Lima, and Clarissa Castellã Xavier. A systematic review of named entity recognition in biomedical texts. *Journal of the Brazilian Computer Society*, 17 :103–116, 2011.
- [10] Mónica Marrero, Julián Urbano, Sonia Sánchez-Cuadrado, Jorge Morato, and Juan Miguel Gómez-Berbís. Named entity recognition : fallacies, challenges and opportunities. *Computer Standards & Interfaces*, 35(5) :482–489, 2013.
- [11] Xiaole Li, Tianyu Wang, Yadan Pang, Jin Han, and Jin Shi. Review of research on named entity recognition. In *Advances in Artificial Intelligence and Security : 8th International Conference on Artificial Intelligence and Security, ICAIS 2022, Qinghai, China, July 15–20, 2022, Proceedings, Part II*, pages 256–267. Springer, 2022.
- [12] Satoshi Sekine and Chikashi Nobata. Definition, dictionaries and tagger for extended named entity hierarchy. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC’04)*, Lisbon, Portugal, May 2004. European Language Resources Association (ELRA).
- [13] Shaodian Zhang and Noémie Elhadad. Unsupervised biomedical named entity recognition : Experiments with clinical and biological texts. *Journal of Biomedical Informatics*, 46(6) :1088–1098, 2013. Special Section : Social Media Environments.
- [14] Denis Maurel, Nathalie Friburger, Jean-Yves Antoine, Iris Eshkol, and Damien Nouvel. Cascades de transducteurs autour de la reconnaissance des entités nommées. *Revue TAL*, 52(1) :69–96, 2011.
- [15] BMC Bioinformatics, Daniel Hanisch, Katrin Fundel, Heinz Theodor Mevissen, Ralf Zimmer, and Juliane Fluck. Prominer : rule-based protein and gene entity recognition, 2005.
- [16] Yusuke Shinyama and Satoshi Sekine. Named entity discovery using comparable news articles. In *COLING 2004 : Proceedings of the 20th International Conference on Computational Linguistics*, pages 848–853, Geneva, Switzerland, Aug 23–Aug 27 2004. COLING.
- [17] Michael Collins and Yoram Singer. Unsupervised models for named entity classification. In *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999.
- [18] Sudha Morwal, Nusrat Jahan, and Deepti Chopra. Named entity recognition using hidden Markov model (hmm). *International Journal on Natural Language Computing (IJNLC) Vol, 1*, 2012.
- [19] Daniel M. Bikel, Richard Schwartz, and Ralph M. Weischedel. An algorithm that learns what’s in a name. *Mach. Learn.*, 34(1–3) :211–231, Feb 1999.
- [20] Andrew Eliot Borthwick. *A maximum entropy approach to named entity recognition*. New York University, 1999.
- [21] Hideki Isozaki and Hideto Kazawa. Efficient support vector classifiers for named entity recognition. In *COLING 2002 : The 19th International Conference on Computational Linguistics*, 2002.
- [22] Burr Settles. Biomedical named entity recognition using conditional random fields and rich feature sets. In *Proceedings of the International Joint Workshop on Natural Language Processing in Biomedicine and its Applications (NLPBA/BioNLP)*, pages 107–110, Geneva, Switzerland, August 28th and 29th 2004. COLING.
- [23] Maryam Habibi, Leon Weber, Mariana Neves, David Luis Wiegandt, and Ulf Leser. Deep learning with word embeddings improves biomedical named entity recognition. *Bioinformatics*, 33(14) :i37–i48, 2017.
- [24] Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. Distributed representations of words and phrases and their compositionality. In C.J. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc., 2013.
- [25] Jeffrey Pennington, Richard Socher, and Christopher D Manning. Glove : Global vectors for word representation. In *EMNLP*, volume 14, pages 1532–1543, 2014.
- [26] Piotr Bojanowski, Edouard Grave, Armand Joulin, and Tomas Mikolov. Enriching word vectors with subword information. *arXiv preprint arXiv :1607.04606*, 2016.
- [27] Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, pages 1064–1074, Berlin, Germany, August 2016. Association for Computational Linguistics.

- [28] Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. Neural architectures for named entity recognition. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, pages 260–270, San Diego, California, June 2016. Association for Computational Linguistics.
- [29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.
- [30] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert : Pre-training of deep bi-directional transformers for language understanding, 2018. cite arxiv :1810.04805Comment : 13 pages.
- [31] Steven Bird. Nltk : the natural language toolkit. In *Proceedings of the COLING/ACL 2006 Interactive Presentation Sessions*, pages 69–72, 2006.
- [32] Christopher D Manning, Mihai Surdeanu, John Bauer, Jenny Rose Finkel, Steven Bethard, and David McClosky. The stanford corenlp natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics : system demonstrations*, pages 55–60, 2014.
- [33] Ted Kwartler. *The OpenNLP Project*, pages 237–269. 05 2017.
- [34] 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 *Proceedings of the 2019 conference of the North American chapter of the association for computational linguistics (demonstrations)*, pages 54–59, 2019.
- [35] Afshin Rahimi, Yuan Li, and Trevor Cohn. Massively multilingual transfer for ner. *arXiv preprint arXiv :1902.00193*, 2019.
- [36] Ralph Grishman and Beth Sundheim. Message Understanding Conference- 6 : A brief history. In *COLING 1996 Volume 1 : The 16th International Conference on Computational Linguistics*, 1996.
- [37] George Doddington, Alexis Mitchell, Mark Przybycki, Lance Ramshaw, Stephanie Strassel, and Ralph Weischedel. The automatic content extraction (ACE) program – tasks, data, and evaluation. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Lisbon, Portugal, May 2004. European Language Resources Association (ELRA).
- [38] Erik F Sang and Fien De Meulder. Introduction to the conll-2003 shared task : Language-independent named entity recognition. *arXiv preprint cs/0306050*, 2003.
- [39] Xavier Schmitt, Sylvain Kubler, Jérémy Robert, Mike Papadakis, and Yves LeTraon. A replicable comparison study of ner software : Stanfordnlp, nltk, opennlp, spacy, gate. In *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*, pages 338–343. IEEE, 2019.