Institut National des Langues et Civilisations Orientales

Département Textes, Informatique, Multilinguisme

Reconnaissance d'entités nommées dans les tweets

MASTER

TRAITEMENT AUTOMATIQUE DES LANGUES

Parcours :

Ingénierie Multilingue / Traductique et Gestion de l'Information

par

Yizhou XU

Directrice de mémoire :

Frédérique SEGOND

Encadrante :

Paule LECUYER

Année universitaire 2018/2019

TABLE DES MATIÈRES

Liste des figures

Liste des tableaux

INTRODUCTION

Contexte

La recherche effectuée dans ce mémoire porte sur la reconnaissance d'entités nommées dans des tweets en anglais.

La reconnaissance d'entités nommées (ci-après REN) est considérée comme un composant crucial pour de nombreuses applications du traitement automatique du langage naturel (ci-après TAL). Elle est, par nature, une sous-tâche de l'extraction d'information (ci-après EI) qui porte sur la détection et la catégorisation des entités nommées dans des données textuelles structurées ou non structurées. Cette tâche est la première étape pour la plupart des tâches d'EI, telles que l'extraction de relation [Hoffmann et al., 2011], l'extraction d'événements [Ritter et al., 2012], etc. Elle sert aussi de base à tant d'autres tâches du TAL comme traduction automatique [Babych and Hartley, 2003], résumé automatique [Babych and Hartley, 2003], peuplement d'ontologies [Tanev and Magnini, 2006], etc. Étant donné son importance, la REN est un sujet bien étudié dans la communauté du TAL. Comme les technologies correspondantes sont relativement matures, cette tâche s'oriente désormais vers de nouvelles directions de recherche [Ehrmann, 2008]. Nous nous concentrons, dans ce mémoire, sur deux pistes spécifiques : la typologie enrichie et le traitement de textes bruités (tweets [1]).

Outre sa popularité dans le milieu académique, la reconnaissance d'entités nommées a également suscité un intérêt certain dans le cadre commercial. Entre autres scénarios d'applications commerciales remarquables, citons MediaCentric et AMI Enterprise Intelligence, les deux plate-formes de Bertin IT [2] dans lesquels le système de REN conçu dans ce travail est intégré. MediaCentric est une plate-forme qui intègre de multiples fonctionnalités d'analyse en profondeur des contenus multimédias et multilingues pour fournir un support à la prise de décision. Parmi d'autres fonctionnalités, la plate-forme crée pour chaque document textuel traité une fiche dont le remplissage se base sur la REN. AMI Entreprise Intelligence est un logiciel de la veille stratégique qui consiste à collecter puis analyser les informations les plus à jours sur son environnement afin de prendre les meilleurs décisions possibles. La première étape d'une telle analyse est la reconnaissance d'entités nommées.

Problématique

Au niveau théorique, nous avons identifié deux problèmes principaux qui empêche le développement d'un système de REN : le traitement de textes bruités et la disponibilité des ressources nécessaires. Différents des textes standard, les textes bruités

1. Message court posté sur le service Twitter.
2. https://www.bertin-it.com/en/

sont souvent marqués par des caractéristiques comme langage familier, fautes d'orthographe, mauvais usage des majuscules, etc. [Ritter et al., 2011] constate que la performance [3] des systèmes de REN conçus pour des textes standard est gravement dégradé sur des tweets, à savoir des textes courts et bruités. La conception et l'implémentation du système devraient se baser sur des méthodes et des ressources spécifiques aux textes bruités, ce qui nous confronte au deuxième problème : la disponibilité de ressources. [Derczynski et al., 2016] indique que l'absence d'un corpus annoté de grande taille et de haute qualité, analogue au corpus comme CoNLL 2003, est l'un des obstacles majeurs au développement de systèmes de REN dans des textes bruités. En ce qui concerne notre travail chez Bertin IT, aucun corpus annoté disponible (comme corpus de Ritter, de W-NUT, etc) ne peut répondre à nos besoins particuliers.

Au niveau pratique, la tâche de la REN est souvent compliquée par les besoins spécifiques dans les différents scénarios d'utilisation en milieu industriel. À l'égard de Bertin IT, la tâche est requise par tous les deux systèmes mais les besoins de MediaCentric peuvent très bien se distinguer de ceux d'AMI EI. Pour le remplissage d'une fiche, les catégories d'entités pertinentes sont essentiellement les types canoniques comme personne, lieu et organisation ; en revanche, pour la veille stratégique, où les documents à traiter porte le plus souvent sur des sujets économiques ou technologiques, il convient de prendre en compte des entités nommées concernant des produits et des expressions numériques. D'ailleurs, MediaCentric exige un faible temps de traitement (par exemple, 1000 documents par seconde) car le système est censé d'être capable de fournir des alertes en temps réel ; au contraire, AMI IE a une préférence pour la qualité de résultat. Un système peut rarement satisfaire à la fois ces deux besoins avec des ressources limitées.

À la lumière de l'analyse qui précède, nous pouvons décomposer notre problématique en 3 aspects afin de définir les orientations de notre travail :

1. Comment concevoir et implémenter un système polyvalent de REN qui puisse couvrir des besoins variés et qui puisse fournir une solution globale pour répondre aux exigences différentes (en l'occurrence, faible temps de traitement et résultat de bonne qualité) ?
2. Comment réaliser un tel système en face du manque de ressources ?
3. Comment assurer la performance d'un tel système sur des tweets ?

Organisation du mémoire

Pour répondre à ces questions, nous articulons ce mémoire autour de quatre chapitres :

1. Le premier chapitre clarifie les définitions des notions de base de l'étude ;
2. Le deuxième chapitre est consacré à la revue des ressources existantes et à la description des ressources créées dans ce travail ;
3. Le troisième chapitre est dédié à l'étude systématique des méthodes existantes et à la présentation des méthodes adoptées dans ce mémoire ;
4. En fin, le quatrième chapitre présente les expérimentations menées.

3. Le terme « performance » est ici à entendre au sens de « résultats obtenus (précision, rappel et F-mesure). »

CHAPITRE 1

ENTITÉS NOMMÉES ET RECONNAISSANCE D'ENTITÉS NOMMÉES

Sommaire

1.1 Introduction

1.2 Historique

La notion d'entité nommée a été introduite pour la première fois dans les campagnes d'évaluation américaines MUC (*Message Understanding Conferences*), plus précisément MUC-6 [Grishman and Sundheim, 1995] organisée en 1995. Les conférences ont pour objectif d'encourager le développement de nouvelles et meilleures méthodes d'extraction d'information. Dans ce cadre, la notion d'entité nommée couvre trois types d'expressions : ENAMEX (les noms propres incluant les noms de personnes, d'organisation et de lieux), TIMEX (les expressions temporelles) et NUMEX (les expressions numériques). Sous l'influence de ces campagnes, les premiers travaux en reconnaissance d'entités nommées ont essentiellement été réalisés sur des textes journalistiques[Poibeau, 2011].

À la fin des années 1990, la tâche de la REN a été héritée par le programme ACE (*Automatic Content Extraction*). Tout au long du programme, la définition de la tâche a été enrichie et la typologie d'entités nommées a connu une évolution constante. De plus, la tâche n'était plus limitée à l'anglais, des textes en d'autres langues (par exemple, le chinois et l'arabe) ont également été pris en compte.

Ces dernières années, avec l'essor des réseaux sociaux, des contenus produits par les utilisateurs attirent de plus en plus l'attention de la communauté du TAL. Une série d'ateliers W-NUT(*Workshop on Noisy User-generated Text*) sont organisés depuis

2015 pour appliquer les technologies du TAL aux textes bruités créés par les utilisateurs. Les entités nommées font l'objet d'une attention constante de ces ateliers et diverses ressources et méthodes sont proposées à l'égard de la tâche REN.

1.3 Entités nommées

Bien que la reconnaissance d'entités nommées fasse l'objet d'études approfondies depuis plus de 20 ans, aucun consensus n'a cependant pu être établi sur la définition des entités nommées, de sorte que les tentatives de définir, ou redéfinir, cette notion ne sont pas finies jusqu'à aujourd'hui.

[Ehrmann, 2008] a inventorié de nombreuses définitions proposées par les travaux antérieurs [Chinchor and Robinson, 1997, Chinchor, 1998, Sang and De Meulder, 2003, Meur et al., 2004, Friburger, 2002, Tran, 2006, Weissenbacher, 2003, Poibeau, 2003, Daille et al., 2000, Enjalbert, 2005, Vicente, 2005, Sekine et al., 2002]; [Nouvel et al., 2016] a enrichi davantage cette liste en prenant en compte les travaux récents [Rosset et al., 2011, Galibert et al., 2011, Nouvel, 2012].

Dans ce travail, nous adoptons la définition des entités nommées d'ESTER [Le Meur et al., 2004] selon notre compréhension de la tâche basée sur l'analyse des besoins des systèmes cibles. La définition est comme suit :

> *Même s'il n'existe pas de définition standard, on peut dire que les EN sont des types d'unités lexicales particuliers qui font référence à une entité du monde concret dans certains domaines spécifiques notamment humains, sociaux, politiques, économiques ou géographiques et qui ont un nom (typiquement un nom propre ou un acronyme).*

1.4 Reconnaissance d'entités nommées

La tâche de la reconnaissance d'entités nommées consiste à trouver toutes les mentions d'entités nommées dans un texte donné et les catégoriser [Jurafsky and Martin, 2009]. Selon [Nouvel et al., 2016], elle se compose de trois sous-tâches :

1. détection des entités ;
2. catégorisation des entités selon une typologie prédéfinie ;
3. résolution référentielle des entités.

Dans notre travail, nous ne prenons en considération que les deux premières étapes ; la troisième étape, souvent connue sous le nom anglais de « *entity linking* », est considérée comme une tâche indépendante en aval. Les termes « extraction d'entités nommées » [Carreras et al., 2002, Asahara and Matsumoto, 2003, Etzioni et al., 2005] et « reconnaissance d'entités nommées » [Nadeau and Sekine, 2007, Lample et al., 2016] sont utilisés de façon interchangeable dans la littérature existante. Dans ce mémoire, nous utilisons uniquement ce dernier.

1.5 REN dans des textes bruités

Les données textuelles bruitées sont omniprésentes dans les communications du monde réel. Nombreux travaux [Knoblock et al., 2007, Subramaniam et al., 2009, Vogel and Tresner-Kirsch, 2012, Derczynski et al., 2013] ont déjà été accomplis dans le domaine du TAL à cet égard mais les auteurs fournissent rarement une définition claire du « texte bruité ». Les textes bruités étudiés dans ce travail sont limités aux tweets qui sont généralement marquées par les caractéristiques suivantes :

— langage familier ;
— fautes d'orthographe ;
— abréviation personnalisée ;
— mauvais usage des majuscules et des ponctuations ;
— emoticons et emojis ;
— @-mention, hashtag, URL.

D'autres textes qui satisfont à au moins une de ces caractéristiques sont également considérés comme « textes bruités » ; en revanche, un texte qui ne présente aucune des caractéristiques *supra.* est un « texte standard ».

En raison de ces caractéristiques, la performance des systèmes du TAL conçus pour des textes standard est gravement dégradée sur des tweets [Ritter et al., 2011]. En ce qui concerne la REN, la situation est encore pire car le défi le plus important réside dans des entités rares et émergentes. Les « entités rares » sont celles dont la forme de surface est unique (par exemple « picton rd [1] ») ; ce type d'entité est aussi connue sous le nom anglais de « *singleton entity* ». Les « entités émergentes » sont de nouvelles entités qui sont (quasiment) jamais vues auparavant et qui deviennent soudainement populaires (par exemple « kktny [2] »). Ces entités sont ci-après dénommées « entités hors vocabulaire ».

1.6 Métrique d'évaluation

1.6.1 Rappel, Précision et F-Mesure

Le rappel (formule 1.1) mesure le nombre d'éléments correctement étiquetés par le système (vrais positifs) rapporté au nombre d'éléments étiquetés dans la référence (vrais positifs et faux négatifs).

$$\text{Rappel} = \frac{\text{vrais positifs}}{\text{vrais positifs + faux négatifs}} \tag{1.1}$$

La précision (formule 1.2) mesure le nombre d'éléments correctement étiquetés par le système (vrais positifs) rapporté au nombre total d'éléments étiquetés par le système (vrais et faux positifs).

$$\text{Précision} = \frac{\text{vrais positifs}}{\text{vrais positifs + faux positifs}} \tag{1.2}$$

La F-mesure (formule 1.3) est la moyenne harmonique pondérée du rappel et de la précision. La valeur accordée à β permet de pondérer le rappel ou la précision, ou d'équilibrer les deux mesures (avec $\beta = 1$).

$$\text{F-mesure} = \frac{(1+\beta^2) \times \text{ précision } \times \text{ rappel}}{\beta^2 \times \text{ précision } + \text{ rappel}} \tag{1.3}$$

1. Picton Road, voir https://en.wikipedia.org/wiki/Picton_Road,_New_South_Wales
2. Abréviation pour « *Kourtney and Kim Take New York* »

1.6.2 Performance

Comme ce travail est effectué dans le milieu industriel, nous évaluons également la performance au niveau de système. Théoriquement, la performance peut être décomposée en plusieurs éléments tels que :

— temps de traitement ;
— efficience (utilisation des ressources informatiques comme processeur, mémoire, stockage, réseau) ;
— haute disponibilité.

Nous ne prenons en compte que le temps de traitement dans ce travail. La performance est simplement mesurée en terme de temps de traitement de 300 documents.

1.7 Conclusion

Dans ce chapitre, nous avons d'abord examiné l'évolution de la notion d'entité nommé, puis clarifié les définitions des notions de base dans ce mémoire. À la fin du chapitre, nous avons vu les mesures d'évaluation de la tâche en question.

CHAPITRE 2

RESSOURCES

Sommaire

2.1 Introduction

La tâche de la reconnaissance d'entités nommées implique diverses ressources linguistiques, telles que les typologies, les corpus annotés, et les lexiques et bases de connaissances [Ehrmann et al., 2016, Nouvel et al., 2016]. Chaque type de ressource joue un rôle spécifique : les typologies sont utilisées pour définir un cadre sémantique des entités considérées ; les corpus servent à illustrer un objectif et peuvent être utilisés comme base d'apprentissage ou comme référence d'évaluation ; les lexiques et les bases de connaissances fournissent des informations, linguistiques ou encyclopédiques, sur les entités en question [Nouvel et al., 2016]. Nous nous concentrons ici sur le corpus [1] et la typologie [2]. Ce chapitre est dédié à la revue des ressources existantes associées aux entités nommées et à la description de celles utilisées dans notre projet.

2.2 Ressources existantes

2.2.1 Typologie

Le terme « typologie » peut désigner les systèmes de types qui permet l'analyse, la description et la classification d'une réalité complexe [3]. En ce qui concerne les entités nommées, il s'agit d'une description formalisée et structurée des classes sémantiques

1. Pour une description plus détaillée des corpus associés aux entités nommées, veillez consulter le site de Damien Nouvel à l'adresse http://damien.nouvels.net/resourcesen/corpora.html
2. Pour une description plus détaillée des typologies associées aux entités nommées, veillez consulter le site de Damien Nouvel à l'adresse http://damien.nouvels.net/resourcesen/typologies.html
3. Voir https://www.cnrtl.fr/lexicographie/typologie

à considérer (les objets du monde réel qui présentent un intérêt) et d'une définition de leur portée (leur réalisation dans des textes) [Ehrmann et al., 2016].

Les typologies peuvent comporter, pour les plus simples, 4 types de base comme celle des campagnes d'évaluation CoNLL [Sang and De Meulder, 2003]. Néanmoins, une telle typologie n'est cependant pas suffisante pour couvrir de nombreuses applications potentielles. On a souvent besoin d'en enrichir les catégories, d'en étendre la hiérarchie et d'en augmenter la granularité pour couvrir de nouveaux besoins. Le choix est souvent guidé par le domaine d'application, qui peut être général (comme l'extraction d'information dans le domaine des médias) ou spécifique (comme celle dans le domaine des sciences biologiques) [Ehrmann et al., 2016]. Par exemple, pour le domaine de la biologie moléculaire, nous pouvons ajouter les catégories comme «*proteins*», «*DNAs*» et «*RNAs*» [Ohta et al., 2002]. Les financiers du projet, ainsi que les clients ou donneurs d'ordre, exercent également une influence sur cette décision [Ehrmann et al., 2016]. Tel est le cas, par exemple, de la campagne ACE 05 où l'on a ajouté une catégorie « arme ».

Il existant tant de typologies proposées par différents auteurs ou projets, dans divers domines et en multiples langues [Sekine and Isahara, 2000, Fleischman and Hovy, 2002, Li and Roth, 2002, Harabagiu et al., 2003, Rosset et al., 2007, Magnini et al., 2008, Benikova et al., 2014]. On ne vise pas ici à fournir une description exhaustive mais se concentre sur ce qui constitue le fondement de ce domaine et qui a une influence directe sur notre travail.

2.2.1.1 Typologies de base : MUC-6 et CoNLL

La notion d'entité nommée a été introduite lors de la conférence MUC-6 [Grishman and Sundheim, 1996]. Dans ce cadre, 7 types d'entités, regroupés en 3 catégories, sont définies :

- ENAMEX (Entity NAMe EXpression) : personnes, organisations, lieux ;
- TIMEX (TIMe EXpression) : expressions temporelles (les dates et les heures) ;
- NUMEX (NUMeric EXpression) : expressions numériques (les unités monétaires et les pourcentages).

Basé sur cette typologie primitive, les campagnes d'évaluation CoNLL-2002 et CoNLL-2003 [Sang and De Meulder, 2003] [4] ont proposé une typologie de 4 catégories [5] : PER (personne), ORG (organisation), LOC (lieu) et MISC (divers). Ce dernier type contient des entités qui n'appartiennent pas aux trois groupes précédents.

2.2.1.2 Typologie canonique : ACE

Suite aux campagnes MUC-6 et MUC-7[Chinchor, 1998], la typologie d'entités nommées a connu une évolution constante durant les campagnes ACE [Ace, 2002b, Ace, 2002a, Ace, 2004, Ace, 2005b, Ace, 2008]. Par rapport aux typologies proposées par MUC et CoNLL, qui sont « simples »[Ehrmann et al., 2016], celles introduites par ACE [6] sont plus complexes. À part les 3 types de base proposés par MUC (PER, ORG et LOC), 4 catégories supplémentaires sont ajoutées :

4. Voir https://www.clips.uantwerpen.be/conll2003/ner/

5. Les expressions temporelles et numériques ne sont pas prises en compte dans les campagnes d'évaluation CoNLL ; il n'existe pas de correspondance entre la catégorie MISC et les catégories d'entités de MUC.

6. le directive d'annotation (version 6.6 2008.06.13) est disponible à l'adresse https://www.ldc.upenn.edu/sites/www.ldc.upenn.edu/files/english-entities-guidelines-v6.6.pdf

— GPE (*Geo-political entity*)[7] : régions géographiques définies par des groupes politiques et / ou sociaux.
— FAC (Facility) : les installations.
— WEA (Weapon) : les armes.
— VEH (Vehicle)[8] : les véhicules.

ACE a organisé les entités en 2 niveaux avec 7 types et 45 sous-types. Par ailleurs, les entités nommées annotées ne sont plus limitées aux noms propres, les descriptions définies et d'autres expressions référentielles sont également inclues dans la campagne. Cet aspect était hérité par la typologie utilisée pour la campagne ESTER-2 [Galliano et al., 2009] et la typologie du projet QUAERO [Grouin et al., 2011]. D'autres typologies, définies pour des campagnes d'évaluation, ont été essentiellement développées sur la base de la typologie ACE, avec l'ajout de nouvelles catégories [Ehrmann et al., 2016].

2.2.1.3 Typologies étendues et hiérarchiques

A l'instar d'ACE, la campagne ESTER [Galliano et al., 2009] a proposé une typologie enrichie et hiérarchique. Celle-ci a redéfinit la typologie en modifiant la hiérarchie et ajoutant plusieurs catégories (avec nombreuses sous-catégories) telles que « fonction » et « production humaine ». Se fondant sur les travaux antérieurs [Galliano et al., 2009, Ehrmann, 2008, Group et al., 2009, Tran, 2006, Rosset et al., 2011, Sekine and Nobata, 2004, Nadeau and Sekine, 2007], le projet QUAERO[Grouin et al., 2011] a développé davantage cette hiérarchie en structurant les entités et introduisant la notion de composition.

2.2.2 Corpus

Les jeux de données du langage naturel sont appelés corpus; un jeu de données annotées selon la même spécification est appelé corpus annoté [Pustejovsky and Stubbs, 2012]. Quant à la tâche de la reconnaissance d'entités nommées, le corpus annoté correspondent à un ensemble de documents enrichis par les étiquettes d'entités nommées selon une typologie donnée [Ehrmann et al., 2016].

Selon [Ehrmann et al., 2016], le corpus est la ressource plus riche parmi les trois types de ressources inventoriés. On peut distinguer différents types de corpus selon la modalité (écrit, oral ou mixte) et la langue (anglais, français, ou multilingue). Dans cette partie, nous nous concentrons uniquement sur les corpus anglais de textes écrits (y compris la transcription des émissions) qui soient pertinent à notre projet.

2.2.2.1 Corpus de textes standard : MUC, ACE, CoNLL

Les corpus utilisés dans les campagnes MUC-6 et MUC-7 sont produits par *Linguistic Data Consortium* (LDC). MUC-6 [Sundheim, 1995][9] contient 318 articles annotés issus du *Wall Street Journal*. MUC-7 [Marsh and Perzanowski, 1998] est com-

7. Au départ, les GPEs ne faisaient pas partie des entités ACE. Cependant, au cours des exercices d'annotation initiaux, il est apparu clairement que le même mot ferait souvent référence à différents types d'entités - parfois lieu (comme dans « *the riots in Miami* »), parfois organisation (comme dans « *Miami imposed a curfew* »), parfois personne (comme dans « *Miami railed against the curfew* »). Ce sont ces expressions métonymiques qui ont donné lieu à la création du type hybride GPE [Ace, 2005a]
8. La catégorie « véhicule » est introduite dans MUC-7 comme une sous-catégorie.
9. Voir https://catalog.ldc.upenn.edu/LDC2003T13

posé d'environ 158 000 articles provenant du *New York Times*. Les corpus d'entraînement et de test utilisés dans la tâche *Named Entity* contiennent chacun 100 articles récupérés par le système *Managing Gigabytes*[Witten et al., 1999].

Une série de corpus sont produits par LDC tout au long du programme ACE organisé entre 1999 et 2008. Ces corpus sont souvent annotés en plusieurs niveaux (entités, expressions temporelles, relations et événements) pour être utilisés dans différentes tâches proposées. Les corpus ACE intègrent des textes de différentes natures tels que les articles journalistiques et les transcriptions de parole.

Le corpus CoNLL 2003 (anglais) [10] est l'un des corpus les plus utilisés dans la communauté REN et devient un standard *de facto* pour la tâche. Il s'agit d'une collection d'articles provenant du corpus *Reuters* [11]. Le corpus est réparti en trois sous corpus : un corpus d'entraînement (946 documents), un corpus de développement (216 documents) et un corpus de test (231 documents). C'est le premier corpus d'entités nommées fourni en format CoNLL avec l'encodage BIO.

Nombreux corpus de textes standard sont également créés hors du contexte des campagnes d'évaluation et d'annotation, tels que le corpus GMB [Bos et al., 2017] et le corpus WikiNER [Ghaddar and Langlais, 2017].

2.2.2.2 Corpus de textes bruités (tweets) : W-NUT

W-NUT (*Workshop on Noisy User-generated Text*) est une série d'altiers portant sur le traitement automatique des textes bruités. Plusieurs corpus sont créés durant ces altiers, prenons comme exemple le corpus W-NUT 2015 [Baldwin et al., 2015]. Les corpus d'entraînement et de développement pour la tâche REN ont été créés sur la base des travaux antérieurs [Ritter et al., 2011] où l'on distingue 10 types d'entités nommées différents. Les données sont divisées en 2 parties : 1795 tweets annotés pour l'entraînement et 599 pour le développement. Les données de test sont échantillonnées de manière aléatoire de décembre 2014 à février 2015. Deux locuteurs natifs anglais sont recrutés pour annoter indépendamment le corpus de test selon le guide d'annotation.

2.2.2.3 Corpus mixte : OntoNotes 5.0

À part les corpus de textes standard et les corpus spécifique aux textes bruités, il existe aussi des corpus mixtes. Entre autres exemples notables, citons le corpus d'OntoNotes 5.0 [12]. La version 5.0[Weischedel et al., 2013] est la version finale du projet OntoNotes, qui vise à annoter un grand corpus comprenant différents genres de textes (articles journalistiques, conversations téléphoniques, *weblogs*, *talk-shows*, Nouveau Testament et Ancien Testament, etc). Le corpus est enrichi par des annotations aux niveaux syntaxique et sémantique. En ce qui concerne les entité nommées, 18 catégories d'entités nommées sont prises en compte.

2.3 Ressources créées dans notre travail

La disponibilité de corpus a constitué le premier obstacle au développement du système dans notre travail car aucune des typologies des corpus existants ne peuvent

10. Voir https://www.clips.uantwerpen.be/conll2003/ner/
11. Voir https://trec.nist.gov/data/reuters/reuters.html
12. Voir https://catalog.ldc.upenn.edu/LDC2013T19

couvrir toutes les catégories dont nous avons besoin. Cela nous conduit à consacrer la première phase de notre travail à la création d'un corpus Twitter avec une typologie conçue sur mesure.

2.3.1 Typologie

Établir un guide d'annotation (en anglais *annotation guideline*) est généralement le premier pas vers un corpus bien annoté [Pustejovsky and Stubbs, 2012]. Dans le cas tout particulier du corpus d'entités nommées, déterminer les catégories à prendre en considération est l'une des parties les plus problématiques et les plus importantes dans la définition du guide. Ainsi, dans cette partie, nous considérons la création du guide d'annotation comme un processus de conception et de validation de la typologie.

2.3.1.1 Démarches de conception et de validation

Nous suivons ici la méthodologie proposée par Cyril Grouin dans le cadre du cours « Méthodes et évaluations en fouille de texte (LTP5A02P) » à l'INaLCO [13].

1. Analyse du phénomène linguistique et de la tâche
    a) Un corpus de 200 tweets (y compris 100 documents du Corpus de Base [14], 50 documents du Corpus Général et 50 documents du Corpus Spécifique) est d'abord construit pour cette étape ;
    b) Basé sur l'observation sur ce corpus-ci, nous avons proposée une première version de typologie en s'appuyant sur une analyse exploratoire des données (tweets), une analyse des besoins des applications cibles (MediaCentric et AMI EI) et une étude systématique des typologies existantes.
2. Annotation initiale, multiples annotations simples [15] et mise à jour du guide d'annotation
    a) Un corpus de 100 tweets est d'abord construit pour cette étape ;
    b) Une annotation manuelle est effectuée sur ce corpus-ci selon la spécification déterminée ; la typologie est mise à jour après les décomptes statistiques des entités annotées (suppression de catégories inutiles et fusion de sous-catégories très rares) ;
    c) Une série d'annotations simples sont effectuées sur le même corpus et le taux d'accords intra-annotateur (coefficient Kappa) est calculé entre les deux annotations consécutives ; la typologie, ainsi que le guide d'annotation, est mise à jour après chaque tour d'annotation ; un intervalle de 2 semaines est garanti pour assurer l'effectivité des taux d'accords intra-annotateur ; l'annotation simple est répétée jusqu'à obtention d'un taux d'accords intra-annotateur supérieur au seuil de 0,8, qui signifie « une fiabilité acceptable » [Krippendorff, 1980].

13. Initialement proposée pour la création du guide d'annotation et du « *golden standard* », cette méthodologie est adaptée ici à notre tâche selon la situation spécifique du projet.

14. Pour les descriptions détailles des corpus (Corpus de Base, Corpus Général et Corpus Spécifique), voir 2.3.2.4

15. Annotation simple signifie que l'annotation est effectuée par un seul annotateur ; idéalement, il convient que l'annotation de cette phase soit faite par multiples personnes sur le même corpus afin d'arriver à une version consensuelle via discussion.

3. Double annotation [16]
   a) Un nouveau corpus de 20 tweets [17] est d'abord construit pour cette étape ;
   b) Une double annotation est exécutée sur le corpus pour vérifier davantage la fiabilité ainsi que la validité de la typologie ; le taux d'accords inter-annotateurs (coefficient Kappa) est calculé ; une fois que l'annotation est finie, la typologie est mise à jour en fonction du consensus obtenu via discussion ;
   c) Les deux étapes précédentes sont répétées jusqu'à obtention d'un taux d'accords inter-annotateurs supérieur au seuil de 0,8.

2.3.1.2 Résultats

— Un total de 3 tours d'annotations simples (y compris l'annotation initiale) sont effectuées durant un mois pour parvenir finalement à un taux d'accords intra-annotateur supérieur à 0,8 ; 2 tours de doubles annotations sont exécutées pour obtenir un taux d'accords inter-annotateur supérieur à 0,8 et 2 tours de doubles annotations supplémentaires sont effectués.
— Typologie finale
  — Les types d'entités sont organisés en hiérarchie à 2 niveaux : 10 types et 25 sous-types ;
  — Les 10 catégories d'entités prises en compte sont : PER(personne), ORG(organisation), LOC(lieu), GSP(entité géo-politique), PROD (produit), EVE(événement), NUM (expression numérique), TIME(expression temporelle), HT(*hashtag*) et AT(*at-mention*) ;
— Pour certaine catégorie, les entités peuvent éventuellement recevoir un attribut :
  — **PER** : fonction
  — **TIME** : relatif et absolute
  — **ORG, PER, EVE** : religion
— Un guide d'annotation est créé à base de cette typologie.

2.3.2 Corpus

Deux corpus annotés sont envisagés dans le cadre du projet : un corpus général et un corpus de « radicalisation ».

2.3.2.1 Objectif

1. Créer un corpus Twitter du domaine général pour l'entraînement, le développement et l'évaluation du modèle qui servira au développement du système automatique de la reconnaissance d'entités nommées.
2. Créer un corpus Twitter du domaine spécifique, en l'occurrence la « radicalisation », pour évaluer la robustesse du modèle général sur des nouveaux domaines [18].

16. Double annotation signifie que l'annotation est exécutée indépendamment par un deux annotateurs.

17. Un corpus de telle taille ne peut pas couvrir toutes les catégories d'entités, notamment pour une typologie raffinée, mais un corpus plus grand n'est pas pratique pour la phase de double annotation dans notre travail.

18. Dans une certaine mesure, cela peut être considéré comme un corpus hors-domaine.

2.3.2.2 Données

1. Le jeu de données *Twitter Stream Grab* [19] provenant du site Internet Archive [20] est utilisé dans ce projet pour la création du corpus Twitter général. *Twitter Stream Grab* est une simple collection de JSON extraite du flux Twitter général (Spritzer [21]) entre 2011 et 2018. Les données sont organisées selon le temps de publication (mois/jour/heure/minute). Nous avons choisi les données de 22 mois : 2 mois de l'année 2016 (mai et juin), 10 mois de 2017 (janvier – avril, juillet – décembre) et 10 mois de 2018 (janvier – octobre).
2. Un ensemble de 170 122 tweets, extraits du Twitter à l'aide d'une liste de mots-clés pertinents à la « radicalisation », sont utilisés pour construire le corpus du domaine spécifique. La collection de données est effectuée pendant 3 jours (du 18 juin 2019 au 20 juin 2019) avec un outil privé.

2.3.2.3 Déduplication, filtrage et sélection

1. Pour évider des documents redondants, nous avons d'abord supprimer les tweets dupliqués en ignorant la casse des caractères et le mot réservé « RT(Retweet) ».
2. Nous avons effectué une série de filtrages selon différents critères :
   — langue (meta-donnée) : anglais ;
   — longueur minimale du texte : 10 caractères ;
   — partie de discours [22] : le texte contient au moins 2 noms propres [23].
3. La représentativité et l'équilibre du corpus sont garantis par un échantillonnage aléatoire et stratifié (en fonction du temps de publication) :
   — 5 tweets par minute ;
   — 300 minutes par jour ;
4. Nous avons ainsi obtenu un corpus Twitter général (brut) contant environ 1 million [24] tweets et un corpus « radicalisation » (brut) contenant 4500 tweets.

2.3.2.4 Pré-traitement

— **Nettoyage** :
  — supprimer des emojis et des mots réservés de Twitter en utilisant tweet-preprocessor [25] ;
  — supprimer des caractères non latins à l'aide de l'expression régulière.
— **Standardisation** :
  — normaliser des caractères non ascii en utilisant textacy [26] et ftfy [27] ;
  — supprimer des espaces supplémentaires ;
  — normaliser des ponctuations à l'aide des expressions régulières.

19. https://archive.org/details/twitterstream
20. Un organisme à but non lucratif consacré à l'archivage du Web qui agit aussi comme bibliothèque numérique, voir https://archive.org/
21. Un échantillon de 1% de l'ensemble des tweets publics.
22. La partie de discours est étiquetée à l'aide de CMU Twiiter PoS Tagger
23. Le seuil est empiriquement déterminé.
24. 5 docs / minute × 300 minutes / jour × (environ) 30 jours / mois × 22 mois ≈ 1 million docs.
25. https://pypi.org/project/tweet-preprocessor/
26. https://pypi.org/project/textacy/
27. https://pypi.org/project/ftfy/

— **Tokenisation** : segmenter des tokens à l'aide d'un tokeniseur conçu sur mesure ;
— **TrueCasing** : rendre des caractères en vraie casse en utilisant truecase [28]

A la fin de cette étape, nous avons obtenu deux corpus pré-traités : un gros corpus de tweets du domaine général (appelé ci-après « Corpus de Base ») ; un corpus de tweets concernant le sujet de la radicalisation. Nous avons sélectionné aléatoirement 1500 documents à partir du Corpus de Base pour construire le corpus à annoter dans la phase suivante et à utiliser dans le projet (appelé ci-après « Corpus Général »). Nous avons également échantillonné 300 documents à partir du corpus de « radicalisation » pour construire le corpus de test hors-domaine (appelé ci-après « Corpus Spécifique »).

2.3.2.5 Annotation : pré-annotation automatique + validation manuelle

Le Corpus Spécifique est manuellement annoté et validé. En contraste, l'annotation du Corpus Général se compose de deux parties : pré-annotation automatique et validation manuelle. Deux types (phases) de pré-annotation sont conçues et réalisés :

1. **Alpha pré-annotation** : la première phase vise à produire un corpus d'amorce [29] avec le moindre effort humain (collection de données, annotation manuelle, programmation, etc).
2. **Beta pré-annotation** : le deuxième tour de pré-annotation consiste à produire un plus grand corpus pré-annoté en utilisant le modèle entraîné sur le corpus d'amorce, ce qui fait partie du pipeline de l'entraînement continuel (voir 3.3.3).

Nous ne décrivons dans cette section que l'Alpha pré-annotation.

Outil d'annotation Un certain nombre d'outils d'annotation sont actuellement disponibles pour l'annotation d'entités nommées, tels que GATE [30], brat [31], WebAnno [32],docanno [33], etc. Après une comparaison systématique (accessibilité et licence, installation, configuration, fonctionnalités, vitesse, formats d'entrée, formats de sortie, etc), nous avons choisi brat [Stenetorp et al., 2012].

Format d'annotation Plusieurs formats d'annotations sont disponibles pour la REN : les premiers corpus d'entités nommées sont essentiellement annotés au format SGML ; les corpus avec une typologie hiérarchique utilisent souvent le format XML ; les corpus avec une typologie simple, en particulier ceux qui sont fournis dans les campagnes d'évaluation, sont généralement annotés au format CoNLL ; plus récemment, il y a aussi des outils qui demandent que le corpus soit annoté au format JSON. Dans ce travail, les annotations produites par l'outil brat sont converties en format CoNLL avec le schéma de préfixes BIOES [34].

28. https://pypi.org/project/truecase/
29. Corpus d'une taille relativement petite (600 documents).
30. https://gate.ac.uk/
31. https://brat.nlplab.org/
32. https://webanno.github.io/webanno/
33. https://doccano.herokuapp.com/
34. Les préfixes signifient respectivement ***B**eginning*, ***I**nside*, ***O**utside*, ***E**nding*, et ***S**ingle*.

Démarches Inspiré par le travail de [Ritter et al., 2011], nous suivons [Collins and Singer, 1999, Downey et al., 2007, Elsner et al., 2009] en considérant la reconnaissance d'entités nommées comme deux sous-tâches indépendantes, détection d'entités nommées et catégorisation d'entités nommées, ce qui nous permet d'appliquer des techniques mieux adaptées à chaque tâche. Nous avons pré-annoté le corpus en suivant les étapes suivantes :

1. pré-annoter les hashtags (HT), les mentions (AT) et les expressions numériques (NUM) à l'aide de l'expression régulière ;
2. pré-annoter les dates et les heures (TIME) à l'aide de pytimeextractor [35] ;
3. ré-annoter le corpus OntoNotes 5 [36] en ne gardant que le préfixe (B, I, O) des étiquettes et ignorant les catégories ;
4. entraîner un modèle discriminant (CRF) sur le corpus ré-annoté à l'étape précédente ;
5. entraîner un classifieur de multi-classe (SVM) sur un corpus d'articles provenant du Wikipédia ;
6. appliquer le modèle CRF sur le corpus pour détecter des entités nommées ;
7. une fois que une entité est détectée, chercher l'article Wikipédia correspondant à l'aide de l'outil wikipedia [37] :
   — si un seul article est trouvé, appliquer le classifieur SVM sur cet article et utilise le résultat de la prédiction pour classer l'entité ;
   — si plusieurs articles sont trouvés, calculer la similarité entre le tweet et l'article, puis appliquer le classifieur SVM sur le article qui est plus proche du tweet et utiliser le résultat de la prédiction pour classer l'entité ;
   — si aucun article n'est trouvé, on ignore cette entité.

Résultats Après la pré-annotation et la validation, nous avons enlevé les documents sans aucune entité et ceux qui ne contient que des hashtags et des mentions. Nous avons également intégré dans le Corpus Général les tweets annotés lors de la validation du guide d'annotation. A la fin de cette phase, le Corpus Général et le Corpus Spécifique contiennent respectivement 1223 tweets [38] et 241 tweets.

2.3.2.6 Répartition de corpus

Nous avons réparti 20 fois le Corpus Génétal [39] de manière aléatoire selon le protocole : 800 documents pour l'entraînement, 200 pour le développement et 200 pour le teste. La meilleure répartition est choisie selon la distribution d'entités.

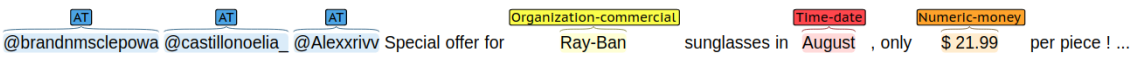

FIGURE 2.1 – Exemple d'annotation

	Train	***Dev***	***Test***	**Spécifique**
Nb de documents	800	200	200	241
Nb de tokens	14 757	3 838	3 690	4 417
Nb d'ENs	3 124	782	753	879
Nb d'ENs (AT et HT exclus)	1 556	401	358	463

TABLE 2.1 – Statistiques descriptives des corpus

2.3.2.7 Statistiques des corpus

2.4 Conclusion

Dans ce chapitre, nous avons d'abord examiné les typologies et les corpus existants ; ensuite, pour répondre aux besoins particuliers des systèmes MediaCentric et AMI EI, nous avons conçu notre propre typologie et créé les corpus correspondants.

35. https://pypi.org/project/pytimeextractor/
36. La typologie du corpus OntoNote 5 est proche de notre choix.
37. https ://pypi.org/project/wikipedia/
38. Les documents annotés lors de l'entraînement continuel sont inclus.
39. Les premiers 1200 documents selon l'identifiant

CHAPITRE 3

MÉTHODES

Sommaire

3.1 Introduction

La reconnaissance d'entités nommées est un sujet bien étudié dans le domaine du TAL [Mai et al., 2018]. Divers approches sont proposées par différents chercheurs pour cette tâche, y compris mais pas limité à : des premières approches à base de règles [Rau, 1991, Appelt et al., 1995, Iwanska et al., 1995], puis des approches d'apprentissage statistique telles que HMM [Zhou and Su, 2002], SVM [Isozaki and Kazawa, 2002, Takeuchi and Collier, 2002] et CRF [McCallum and Li, 2003] ; ensuite des approches à base des réseaux de neurones telles que LSTM+CNN+CRF [Ma and Hovy, 2016, Yang and Zhang, 2018] ou BiLSTM/LSTM-CRF/Pile de LSTMs [Lample et al., 2016, Misawa et al., 2017, Martins et al., 2019] ; et plus récemment, des approches à base des modèles de langue pré-entraînés comme Elmo[Peters et al., 2018], Bert[Devlin et al., 2018] et Flair[Akbik et al., 2018]. Ce troisième chapitre est consacré d'abord à la revue des approches existantes pour la reconnaissance d'entités nommées puis à la présentation des méthodes adoptés dans ce mémoire.

3.2 Méthodes existantes

Les approches de la reconnaissance d'entités nommées peuvent être regroupées en plusieurs catégories sous différentes perspectives. Par exemple, [Jean-Louis, 2011, Poibeau, 2011, Nouvel, 2012, Chiticariu et al., 2013] distinguent deux grands types d'approches : les approches symboliques (linguistiques « de surface », orientées

connaissances, à base de règles) et les approches statistiques (probabilistes, orientées données, à base d'apprentissage automatique) ; [Jurafsky and Martin, 2009, Boros, 2018], reposant sur une distinction supplémentaire entre les deux types d'apprentissage automatique, regroupent les approches en trois catégories : les approches à base de règles (ou patrons), les approches à base de caractéristiques et les approches à base de réseaux de neurones. Il existent des typologies de méthodes encore plus complexes, par exemple on peut distinguer les apprentissages supervisé, semi-supervisé et non supervisé ou ajouter une catégorie hybride (symbolique+statistique). Dans cette partie, nous examinons les méthodes selon la typologie proposée par [Jurafsky and Martin, 2009].

3.2.1 Méthodes à base de règles

Historiquement, les premiers systèmes développés pour la REN étaient essentiellement les systèmes symboliques, ou systèmes à base de règles [Nadeau and Sekine, 2007]. Ces systèmes consistent généralement en un ensemble de règles fabriquées à la main, telles que des expressions régulières, des patrons morphosyntaxiques, des lexiques, des caractéristiques orthographiques et des ontologies [Budi and Bressan, 2003].

Bien qu'il faille attendre la fin de l'année 1995 pour que la notion d'entité nommée soit introduite par MUC-6, [Rau, 1991], qui a proposé un système d'extraction des noms d'entreprises en reposant sur des règles fabriqués à la main, peut être considéré comme l'un des premiers travaux dans cette famille de méthodes. Ensuite, nombreux systèmes à base de règles ont été proposés durant et après les campagnes MUC. Par exemple, [Appelt et al., 1995] a proposé un système basé sur un ensemble d'expressions régulières créés à la main et obtenu une F-mesure de 94 sur le corpus de MUC-6 ; le système UNO [Iwańska, 1995] a utilisé de nombreuses ressources linguistiques telles que des gazetteers et des pages jaunes.

Comme ces approches reposent sur des règles conçus par humaines, les systèmes sont relativement simples à interpréter et peuvent obtenir une précision élevée, notamment dans un domaine spécifique. Néanmoins, les systèmes symboliques sont généralement développés par des linguistes et des experts et donc souvent spécifique à une langue ou un domaine, ce qui peut entraîner des problèmes de portabilité et de robustesse. Autrement dit, la performance de tels systèmes n'est pas forcément garantie pour de nouveaux domaines et de nouvelles langues.

3.2.2 Méthodes à base d'ingénierie de caractéristiques

La reconnaissance d'entités nommées est généralement considérée comme une tâche de l'étiquetage séquentiel [1]. Il s'agit d'attribuer une étiquette, choisie parmi un ensemble fixe d'étiquettes, à chaque élément d'une séquence[Jurafsky and Martin, 2009]. L'étiquetage est l'un des applications canoniques [2] de l'apprentissage supervisé, qui consiste à apprendre une fonction de prédiction à partir d'exemples annotés [Li, 2012]. En ce qui concerne la REN, l'apprentissage automatique peut être utilisé pour remplacer des règles créées par

1. Il existe d'autres interprétations de cette tâche (par exemple, [Collins and Singer, 1999, Carreras et al., 2002])

2. L'apprentissage supervisé est souvent utilisé dans les tâches du classement (en anglais classification), de la régression et de l'étiquetage.

l'homme. Différents algorithmes d'apprentissage automatique, tels que HMM, SVM, CRF sont appliqués à cette tâche :

HMM (*Hidden Markov Model*) est l'un des premiers modèles statistiques appliqués à la tâche de la REN. Par exemple [Zhou and Su, 2002] ont utilisé un HMM et un chunk tagger à base d'HMM pour construire un système de la REN, atteignant une F 96,6 sur le corpus de MUC 6. SVM (*Support Vector Machine*) [Cortes and Vapnik, 1995], souvent utilisé dans la tâhce de classement, est également appliqué à la REN par les chercheurs qui considèrent cette tâche comme une tâche de classement. Tel est le cas de [McNamee and Mayfield, 2002] qui extrait des entités nommées en combinant 8 classifieurs SVM.

Récemment, l'algorithme le plus utilisé est CRF (*Conditional Random Fields*) [Lafferty et al., 2001], qui permet de prendre en compte l'interaction de variables « voisines ». Cet algorithme est souvent utilisé pour traiter des données séquentielles, notamment des données textuelles. Entre d'autres travaux réalisés à base de CRF, citons l'exemple le plus remarquable à cet égard, [Finkel et al., 2005], qui a donné naissance au Stanford Named Entity Recognizer [3]. Grâce à un extracteur de caractéristiques puissant, Standard NER peut obtenir un résultat de F 87,94 sur le corpus de test de CoNLL 2003 [4].

Ces approches d'apprentissage automatique sont essentiellement basées sur des caractéristiques conçues à la main. Une caractéristique est une représentation numérique des données brutes du monde réel ; le processus de conception et d'extraction des caractéristiques les plus appropriées en fonction des données, du modèle et de la tâche est appelé ingénierie de caractéristiques [Zheng and Casari, 2018]. Les caractéristiques peuvent être présentées sous formes de valeurs booléennes, numériques ou nominales. Quant aux caractéristiques pour la REN, nous pouvons citer ici quelques exemples fournis par [Jurafsky and Martin, 2009] :

— forme du mot actuel, forme des mots voisins ;
— plongements du mot actuel, plongements des mots voisins ;
— partie du discours du mot actuel, partie du discours des mots voisins ;
— présence du mot actuel dans le gazetier ;
— étiquette de syntagme du mot actuel et des mots voisins ;
— préfixe du mot actuel ;
— suffixe du mot actuel.

Les approches à base d'ingénierie de caractéristiques ont plusieurs avantages par rapport aux approches à base de règles : ce type d'approches sont particulièrement robuste pour les entrées inconnues [Poibeau, 2011] ; les systèmes sont plus facile à étendre, il s'agit d'un simple ajout de données annotées. En dépit de ces avantages, cette famille d'approches présente aussi des inconvénients, entre d'autres l'exigence des experts humaines pour la conception de caractéristiques.

3. Voir https://nlp.stanford.edu/software/CRF-NER.shtml
4. Voir https://nlp.stanford.edu/projects/project-ner.shtml

3.2.3 méthodes à base de réseaux de neurones

3.2.3.1 Apprentissage Profond

La performance des méthodes d'apprentissage automatique traditionnelles[5] dépend fortement de la représentation des données. Les classifieurs exigent des entrées qui sont mathématiquement faciles à traiter pour la machine mais des données du monde réel tels que des images et des textes sont généralement compliqués et très variables. Il est nécessaire de découvrir des caractéristiques ou des représentations utiles à partir de données brutes. L'extraction manuelle de caractéristiques prend beaucoup de temps. Une solution à ce problème est d'utiliser l'apprentissage automatique pour découvrir non seulement la projection de la représentation à la sortie mais aussi la représentation elle-même ; cette approche est appelée apprentissage de représentation [Bengio et al., 2017].

Néanmoins, cette approche n'est pas en mesure de résoudre unilatéralement tous les problèmes que nous avons relevés. Il existe de nombreux facteurs importants qui sont construits à partir de chaque élément de données observées. Il est difficile d'extraire ce type de caractéristiques, abstraites et à haut niveau, des données brutes, ce qui donne naissance à l'apprentissage profond (en anglais *deep learning*). L'apprentissage profond résout ce problème en introduisant la représentation hiérarchique. Les caractéristiques abstraites sont exprimés en termes de caractéristiques plus simples[Bengio et al., 2017]. C'est-à-dire, l'apprentissage profond permet à la machine de construire des concepts complexes sur la base des simples concepts. [**?**] a proposé l'une des premières architectures de réseau de neurones pour la REN et obtenu une F-mesure de 89,59[6] sur le corpus de test de CoNLL 2003. Après cette première tentative, diverses architectures de réseaux de neurones sont proposées pour la REN au cours des dernières années.

Parmi tant d'autres architectures, RNN (*Recurrent Neural Network*) [Rumelhart et al., 1988], en particulier LSTM (Long Short-Term Memory) [Hochreiter and Schmidhuber, 1997], occupe une place importante dans la REN grâce à sa capacité d'établir une dépendance entre des mots voisins. CNN (*Convolutional Neural Network*) est également appliqué à la REN, en particulier pour extraire des caractéristiques au niveau de caractère.

[Huang et al., 2015] a proposé une architecture de LSTM bidirectionnel pour extraire des caractéristiques au niveau de mot et a montré que l'ajout d'une couche de CRF à LSTM peut améliorer la performance. Le travail a obtenu un résultat de F-mesure 84,26 sur le corpus de test de CoNLL 2003. Par rapport au (Bi)LSTM-CRF, [Ma and Hovy, 2016] a introduit une couche de CNN pour améliorer le plongement de mots en extrayant des représentations au niveau de caractère, attaignant une F-mesure de 91,2 sur le corpus de test de CoNLL 2003.

3.2.3.2 Plongement

Comme nous l'avons dit ci-dessus, dans un système d'apprentissage automatique, les données d'entrée doivent être transformées en forme numérique pour que les modèles puissent les utiliser dans leur calcul. Des représentations numériques tradi-

5. Le terme « méthodes d'apprentissage automatique traditionnelles » doit s'entendre au sens de « apprentissage automatique sans réseaux de neurones ».

6. Ce résultat est obtenu à l'aide d'une ingénierie de caractéristiques spécifiques à la tâche (par exemple, gazetier) ; le résultat sans aucune intervention humaine est de F-mesure 81,47.

tionnelles à base de sac de mots (par exemple *tf-idf* et *one-hot*) peuvent causer la malédiction de la dimension, ce qui donne naissance à la représentation distribuée.

La représentation distribuée, plus connue sous le nom anglais de *embedding* (en français, plongement) [7], est une représentation plus dense qui peut projeter un espace vectorielle de grande dimension dans un vecteur de dimensions limitées. Elle suit essentiellement l'hypothèse distributionnelle [Harris, 1954], selon laquelle les mots qui apparaissent dans les mêmes contextes linguistiques partagent des significations similaires. Les plongements de mots sont souvent utilisés comme première couche dans une architecture d'apprentissage profond.

Nous pouvons distinguer deux types de méthodes de plongement selon les algorithmes utilisés : les plongements statiques et les plongements dynamiques. Les plongements classiques comme word2vec [Mikolov et al., 2013] et Glove [Pennington et al., 2014] sont statiques parce que le même mot aura toujours la même représentation, quel que soit le contexte dans lequel il apparaît. Les plongements dynamiques (ou plongements contextualisés), tels que Elmo [Peters et al., 2018], Bert [Devlin et al., 2018] et Flair [Akbik et al., 2018], peuvent capturer la sémantique des mots dans différents contextes pour résoudre le problème des mots polysémiques.

Les plongements de mots, notamment les plongements dynamiques, sont à l'origine des progrès dans les travaux les plus récents. Par exemple, [Devlin et al., 2018] a obtenu un résultat de F-mesure 92,81 [8] en utilisant le plongement de Bert ; [Akbik et al., 2018] a enregistré une F-mesure de 93,09 à l'aide du plongement de Flair ; [Straková et al., 2019] a proposé une architecture de Embeddings Stack+LSTM+CRF en empilant trois différents plongements (ELMo+BERT+Flair), attaignant une F-mesure de 93,38.

3.2.4 Méthodes pour la REN dans textes bruités

Afin de régler les problèmes rencontrés dans des textes bruités, diverses approches sont proposées par la communauté du TAL au cours de la dernière décennie. Ces méthodes concernent essentiellement le processus de pré-traitement [Kaufmann and Kalita, 2010, Hemalatha et al., 2012, Porta and Sancho, 2013]. Par exemple, pour remédier des mauvais usages des majuscules, [Nebhi et al., 2015] a présenté une approches de *turecasing*, qui vise à rendre les caractères en casse vraie ; en revanche, [Mayhew et al., 2019] a testé différents pré-traitements au niveau de la casse (*truecased*, *caseless*, *truecase+caseless*, etc). Tous les deux travaux témoignent que le processus de *turecasing* (avec outils existants) contribue peu aux résultats finaux.

3.3 Méthodes adoptées dans notre travail

Nous pouvons constater ici que les méthodes à base de réseaux de neurones, notamment celles basées sur les plongements dynamiques, offrent un avantage considérable par rapport aux autres méthodes. Néanmoins, les réseaux de neurones n'ont pas une bonne réputation pour la performance [9]. La performance d'un système à base

7. Il existe aussi des plongements au différents niveaux, tels que plongements de caractères, de mots, de phrases et de textes.

8. Tous les résultats comparés ici sont réalisés sur le corpus de test de CoNLL 2003.

9. Le temps de traitement

de réseaux de neurones dépendent énormément de la disponibilité du GPU. Comme nous avons analysé au début du mémoire, chaque application cible a ses besoins particuliers : AMI EI demande un résultat de précision élevée et MediaCentric exige un faible temps de réponse. Ainsi, nous avons décidé d'adopter différentes méthodes pour les deux applications. Pour MediaCentric, nous allons entraîner un modèle CRF, qui est capable de traiter un grand volume de données en temps réel ; en revanche, pour AMI EI, nous allons utiliser un modèle à base de réseaux de neurones qui peut apporter de meilleurs résultats.

3.3.1 Modèle à base d'ingénierie de caractéristiques

3.3.1.1 CRF

Il existe principalement deux implémentations de CRF : CRF++ [10] et CRFsuite [11]. Nous avons choisi ici CRFsuite car elle nous permet de créer un nombre arbitraire de caractéristiques pour chaque token et qu'il existe un *scikit-learn-like* wrapper [12] pour cette implémentation.

Le paramétrage du modèle est réalisé à l'aide de l'outil Exhaustive Grid Search [13] fourni par scikit-learn.

3.3.1.2 Ingénierie de caractéristiques

L'une des raisons pour lesquelles les méthodes statistiques à base d'ingénierie de caractéristiques, comme CRF, sont toujours populaires est qu'elles réduisent le problème à la recherche d'un ensemble de caractéristiques appropriés. Nous avons conçu les caractéristiques en nous référant à l'extracteur de caractéristiques de Standford NER [14]. Les caractéristiques utilisées dans ce travail sont regroupées en trois catégories [15] :

1. **Caractéristiques de surface**
   — Type : *alpha*,*digit alnum*, *identifier*, *punctuation*, etc ;
   — Forme : *lower*, *upper*, *capital*, *mix*.
2. **Caractéristiques externes**
   — Word cluster : CMU Twitter word cluster ;
   — PoS : partie de discours étiquetée par CMU Twiiter PoS Tagger ;
   — Soundex : code de Soundex généré par jellyfish [16].
3. **Tokens voisins** :
   — les tokens précédents et les tokens suivants ;
   — BOS(début de la phrase) et EOS(fin de la phrase.

10. https://taku910.github.io/crfpp/
11. http://www.chokkan.org/software/crfsuite/
12. https://sklearn-crfsuite.readthedocs.io/en/latest/
13. https://scikit-learn.org/stable/modules/grid_search.html#exhaustive-grid-search
14. https://nlp.stanford.edu/nlp/javadoc/javanlp/edu/stanford/nlp/ie/NERFeatureFactory.html
15. Inspiré de [Grouin, 2013]
16. https://pypi.org/project/jellyfish/

3.3.2 Modèle à base de réseaux de neurones

3.3.2.1 Pile de plongement + BiLSTM + CNN + CRF

Nous avons utilisé deux outils dans ce travail pour produire le modèle final.

Pour tester les différents plongements et générer le pile de plongements, nous avons utilisé le *framework* Flair [17] créé par [Akbik et al., 2018]. Flair est une boîte à outils du TAL qui nous permet de créer une pile de plongements en combinant les différents plongements comme Glove, FastText, Bert, Elmo et Flair. Le *framework* implémente également une architecture BiLSTM+CRF [Huang et al., 2015] pour la tâche de l'annotation séquentielle.

Pour tester les différentes architectures de réseaux de neurones, nous avons utilisé le *framework* NCRFpp [18]. Cet outil nous permet de comparer les différentes architectures comme CNN (au niveau de mot ou de caractère) + LSTM (au niveau de mot ou de caractère). Les réglages d'hyperparamètres sont effectués à l'aide des fonctionalités fournies par le *framework* NCRFpp.

3.3.3 Pipeline de l'entraînement continuel

Pour répondre au problème des entités hors vocabulaire, nous avons proposé un pipeline de l'entraînement continuel qui permet le système de ré-entraîner constamment de nouveaux modèles en profitant de nouvelles données avec le moindre intervention humaine.

3.4 Conclusion

Dans ce chapitre, nous avons d'abord effectué une étude systématique des méthodes existantes concernant la REN ; ensuite, comme les exigences des deux applications cibles sont différentes, nous avons décidé de construire le système de REN en utilisant différentes approches.

17. https://github.com/zalandoresearch/flair
18. https://github.com/jiesutd/NCRFpp

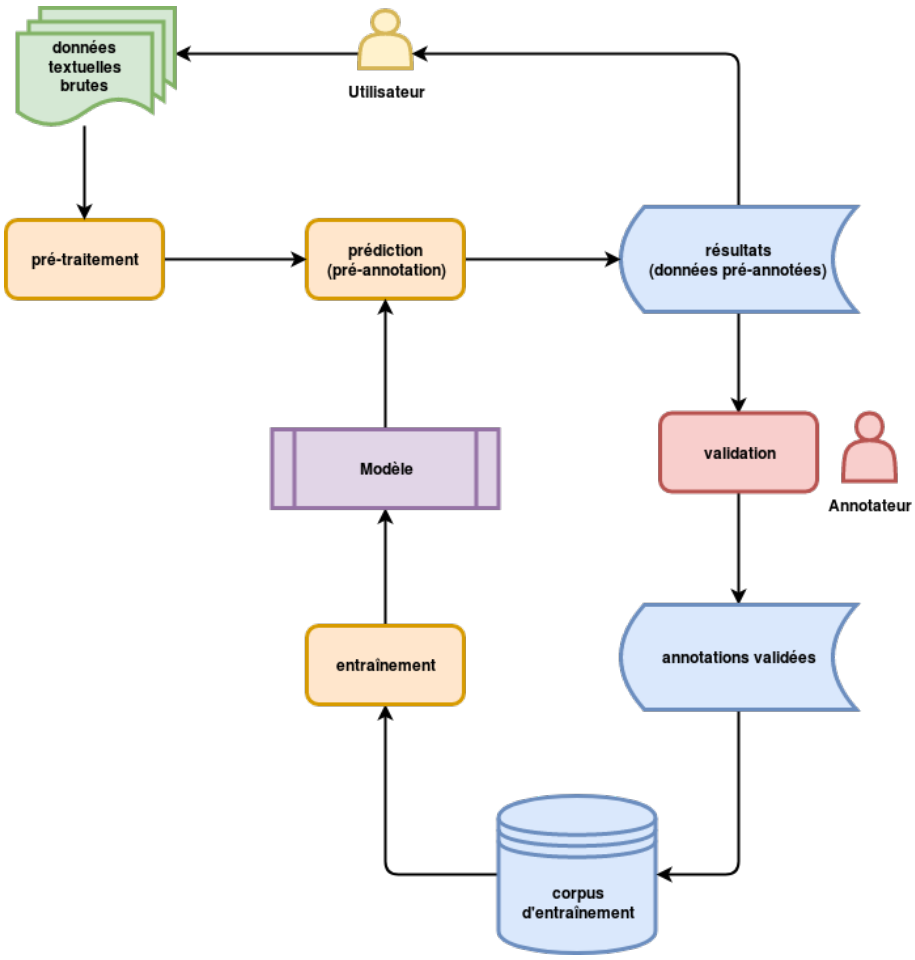

FIGURE 3.1 – Pipeline de l'entraînement continuel

CHAPITRE 4

EXPÉRIMENTATIONS

Sommaire

4.1 Introduction

4.1.1 Configuration globale

4.1.1.1 Corpus

1. **Entraînement** : Corpus Général entraînement
2. **Développement** : Corpus Général développement
3. **Test**
   — *in-domain* test : Corpus Général test
   — *out-domain* test : Corpus Spécifique

4.1.1.2 Environnement Matériel

— **CPU** : Intel(R) Core(TM) i7-4710MQ CPU @ 2.50GHz
— **GPU** : GK208M [GeForce GT 730M] [1]
— **RAM** : 16 GB
— **Disque** : HDD 1T

1. GPU n'est pas utilisé dans ce travail parce que la disponibilité de GPU n'est pas garantie dans l'environnement de production.

4.1.2 Métrique

Basé sur une analyse de la distribution des entités annotées, nous décidons d'évaluer les systèmes en micro F mesure en ignorant les mentions (AT) et les hashtags (HT) car les catégories les plus représentées (sauf AT et HT) sont également les catégories les plus pertinentes (PER, LOC et GPE) pour nos applications.

Dans cette section, sauf mention particulière, les systèmes à base d'ingénierie de caractéristiques sont uniquement évalués au niveau de catégorie (8 classes [2]) et les systèmes à base de réseaux de neurones sont évalué au niveau de sous-catégorie (23 classes).

4.1.3 Baseline

Le baseline est obtenu par un modèle CRF qui n'utilise que la forme des tokens comme caractéristique.

Rappel	**Précision**	**F-mesure**
0,196	0,560	0,290

TABLE 4.1 – Baseline

4.2 Ingénierie de caractéristiques

Ce premier groupe d'expériences visent à découvrir la meilleur combinaison de caractéristiques pour notre modèle CRF.

4.2.1 Caractéristiques de surface

Type	**Forme**	**Rappel**	**Précision**	**F1**
False	False	0,196	0,560	0,290
True	False	0,229	**0,562**	0,325
True	True	**0,396**	0,549	**0,460**

* La détection de forme dépend de celle de type.

TABLE 4.2 – Caractéristiques de surface

2. PER, ORG, LOC, GSP, PROD, EVE, NUM, TIME

4.2.2 Caractéristiques externes

Cluster	PoS	Soundex	Rappel	Précision	F1
False	False	Flase	0,396	0,549	0,460
False	False	True	**0,399**	0,525	0,454
False	True	False	**0,397**	0,601	**0,478**
False	True	True	0,390	0,599	0,472
True	False	False	0,385	**0,614**	**0,473**
True	False	True	0,379	0,551	0,449
True	True	False	**0,401**	0,609	**0,484**
True	True	True	**0,400**	0,607	**0,482**

TABLE 4.3 – Caractéristiques externes

Cluster	Pos	Soundex
0,00409 sec	1,548 sec	0,0114 sec

TABLE 4.4 – Temps de l'extraction de caractéristiques (300 documents)

4.2.3 Tokens voisins

Taille de fenêtre	Rappel	Précision	F1
0	0,385	0,614	0,473
1	0,408	0,610	0,489
2	**0,410**	**0,623**	**0,495**
3	0,385	0,588	0,464

TABLE 4.5 – Caractéristiques de tokens voisins

Les résultats nous permettent de tirer plusieurs conclusions :

1. Tous les trois catégories de caractéristiques aident à améliorer les résultats ; nous pouvons obtenir respectivement une augmentation de F-mesure de 0,17, 0,24 et 0,22 ;
2. Entre les deux caractéristiques de surface, forme exerce une plus grande influence sur le résultat final ;
3. Le word cluster et le PoS servent généralement à augmenter la F-mesure ; en revanche, le code Soundex baisse toujours le résultat ;
4. Le PoS ralentit considérablement le traitement ;
5. La F-mesure n'augmente pas linéairement avec la taille de fenêtre de tokens voisins.

4.3 Réseaux de neurones

	BiLSTM+CRF	BiLSTM+CNN+CRF
Glove	0,549	0,569
FastText (Twitter)	0,573	0,601
Bert	0,624	0,627
Elmo	0,596	0,595
Flair	0,583	0,590
Glove+Bert	0,611	0,616
FastText+Bert	0,630	0,636
Elmo+Bert	0,637	0,625
Flair+Bert	0,624	0,629
FastText+Elmo+Bert	**0,641**	**0,645**
FastText+Flair+Bert	0,631	0,628
FastText+Bert+Elmo+Flair	**0,643**	**0,649**

TABLE 4.6 – Résultats des expériences sur différentes configurations de réseaux (F mesure)

Dans cette expérience, nous avons testé deux architectures de réseaux neurones et différentes combinaisons de plongements. Nous pouvons constater ici que :

1. Les plongements dynamiques surpasse globalement les plongements statiques ;
2. Différents de la constatation des travaux antérieurs [Akbik et al., 2018], les résultats obtenus dans notre étude montre que le plongement Elmo fonctionne généralement mieux que le plongement Flair ;
3. L'empilement de plongement peut généralement améliorer davantage le résultat final ;
4. L'ajout d'une couche de CNN au niveau de caractère peut aider à augmenter la F-mesure.

4.4 Facteurs externes

Comme [Gut and Bayerl, 2004] a indiqué que la fiabilité de l'annotation manuelle et la complexité de la tâche d'annotation sont négativement corrélées, nous voulons tester ici si la complexité de la tâche d'annotation exerce aussi une influence sur l'annotation automatique. Dans cette expérience, la complexité de la tâche est représentée par la granularité de la typologie et la complexité du schéma d'annotation.

4.4.1 Configuration

4.4.1.1 Granularité de typologie

Pour tester les différentes granularités de typologie, nous avons ré-annoté les corpus en fusionnant les catégories et ensuite ré-entraîné le modèle.

— original : 23 catégories + sous-catégories
— 8 catégories : PER, ORG, LOC, GSP, PROD, EVE, NUM, TIME ;
— 5 catégories : PER, ORG, LOC (LOC et GSP), TIME, MISC (PROD, EVE, NUM).

4.4.1.2 Schéma d'annotation

Originalement annotés en schéma BIOES, les corpus sont convertis en schéma BIO dans cette section.

— BIOES : ***B**eginning*, ***I**nside*, ***O**utside*, ***E**nding*, et ***S**ingle* ;
— BIO : ***B**eginning*, ***I**nside*, ***O**utside*.

4.4.2 Résultats

	Bert+Flair+BiLSTM+CRF		
	5 catégories	**8 catégories**	**23 catégories**
BIOES	0,680	0,665	0,624
BIO	0,668	0,666	0,622

TABLE 4.7 – Influences des facteurs externes (réseau de neurones)

	Ingénierie de caractéristiques+CRF		
	5 catégories	**8 catégories**	**23 catégories**
BIOES	**0,544**	0,495	0,447
BIO	0,537	0,494	0,456

TABLE 4.8 – Influences des facteurs externes (ingénierie de caractéristiques)

Nous pouvons remarquer que :

1. Plus le nombre de catégories est grand, plus la F-mesure est faible ; c'est-à-dire la complexité de la tâche et la F-mesure sont négativement corrélées.
2. Pour les modèles à base de réseaux de neurones, le schéma d'annotation exerce peu d'influence ; les modèles à base d'ingénierie de caractéristiques sont plus sensibles au schéma d'annotation.
3. Pour les modèles à base de réseaux de neurones, la performance du schéma BIOES surpasse essentiellement celle de BIO.

4.5 Test sur le corpus du domaine spécifique

Les modèles d'apprentissage sont souvent entraînés sur un volume important de données car l'entraînement sur un petit corpus peut causer le problème de sur-apprentissage. Afin de vérifier la capacité de généralisation (ou robustesse) des modèles, nous les avons testés sur un corpus *out-domain* (le Corpus Spécifique).

	Corpus Général	**Corpus Spécifique**
Ingénierie de caractéristiques	0,495	0,422
Réseaux de neurones	0,649	0,636

TABLE 4.9 – Test sur le Corpus Spécifique

Le modèle à base de réseaux de neurones est plus robuste que le modèle à base d'ingénierie de caractéristiques. Bien que notre corpus d'entraînement soit petit, les décalages entre les résultats de test *in-domain* et ceux de test *out-domain* sont relativement acceptable. C'est-à-dire, nos modèles sont assez robustes. Nous avons utilisé

dans ce travail les plongments tels que Elmo, Flair et Bert. Ces plongements sont, par nature, modèles de la langue pré-entraînés. Dans une certaine mesure, notre travail peut être considéré comme apprentissage par transfert. C'est la raison pour laquelle nous pouvons obtenir un bon modèle avec un petit corpus.

4.6 Conclusion

Dans ce dernier chapitre, nous avons mené une série d'expérimentations. Nous pouvons constater que les plongements améliorer considérablement le résultat de la REN. Les modèles à base de réseaux de neurones surpassent absolument les modèles à base d'ingénierie de caractéristiques.

Conclusion générale

Dans ce mémoire, nous avons tenté de répondre à la problématique « comment concevoir et réaliser un système polyvalent de reconnaissance d'entités nommées pour des textes bruités en face du manque de ressources ? »

Nous avons décomposé cette question en plusieurs sous-questions.

1. Comment concevoir un système qui puisse satisfaire de divers besoins ? Pour couvrir des besoins de différents domaines d'application, nous avons élaboré une typologie enrichie, raffinée, adaptable et extensible, ce qui permet de contenter les besoins actuels et potentiels.
2. Comment réaliser un tel système en face du manque de ressources ? Pour palier ce problème, nous avons annoté un corpus d'amorce et proposé un pipeline de l'entraînement continuel. Comme un sous-produit, le pipeline nous permet d'élargir et d'enrichir le corpus de façon continuelle.
3. Comment garantir la performance du traitement sur tweets ? Nous avons approché ce problème en adoptant la stratégie de « diviser pour régner ». Par exemple, nous avons proposé un pipeline d'entraînement pour résoudre le problème d'entités émergentes ; nous avons également établi une série de pré-traitements pour remédier les problèmes comme mauvais usage des majuscules.
4. Comment réaliser un système qui puisse répondre aux différentes exigences qui s'opposent les unes aux autres (comme faible temps de traitement et bonne qualité de résultat) ? Pour répondre aux différents exigences, nous avons construit deux différents modèles : un modèle à base de réseaux de neurones qui peut produire un résultat de bonne qualité et un modèle de l'apprentissage classique qui peut traiter un grand volume de données en temps réel.

BIBLIOGRAPHIE

[Ace, 2002a] Ace (2002a). EDT Annotation Guidelines v2.5. – Cité page 12.

[Ace, 2002b] Ace (2002b). Entity Detection and Tracking: Phase1 v2.2. – Cité page 12.

[Ace, 2004] Ace (2004). ACE English Annotation Guidelines for Entities v4.2.6. – Cité page 12.

[Ace, 2005a] Ace (2005a). The ACE 2005 (ACE05) evaluation plan. – Cité page 13.

[Ace, 2005b] Ace (2005b). ACE English Annotation Guidelines for Entities v5.6.6. – Cité page 12.

[Ace, 2008] Ace (2008). ACE English Annotation Guidelines for Entities v6.6. – Cité page 12.

[Akbik et al., 2018] Akbik, A., Blythe, D., and Vollgraf, R. (2018). Contextual string embeddings for sequence labeling. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1638–1649. – Cité pages 21, 25, 27 et 32.

[Appelt et al., 1995] Appelt, D. E., Hobbs, J. R., Bear, J., Israel, D., Kameyama, M., Kehler, A., Martin, D., Myers, K., and Tyson, M. (1995). Sri international fastus systemMUC-6 test results and analysis. In *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*. – Cité pages 21 et 22.

[Asahara and Matsumoto, 2003] Asahara, M. and Matsumoto, Y. (2003). Japanese named entity extraction with redundant morphological analysis. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 8–15. Association for Computational Linguistics. – Cité page 8.

[Babych and Hartley, 2003] Babych, B. and Hartley, A. (2003). 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: Resources and Tools for Building MT*, pages 1–8. Association for Computational Linguistics. – Cité page 5.

[Baldwin et al., 2015] Baldwin, T., de Marneffe, M.-C., Han, B., Kim, Y.-B., Ritter, A., and Xu, W. (2015). Shared tasks of the 2015 workshop on noisy user-generated text: Twitter lexical normalization and named entity recognition. In *Proceedings of the Workshop on Noisy User-generated Text*, pages 126–135. – Cité page 14.

[Bengio et al., 2017] Bengio, Y., Goodfellow, I., and Courville, A. (2017). *Deep learning*, volume 1. Citeseer. – Cité page 24.

[Benikova et al., 2014] Benikova, D., Biemann, C., Kisselew, M., and Pado, S. (2014). Germeval 2014 named entity recognition shared task: companion paper. – Cité page 12.

[Boros, 2018] Boros, E. (2018). *Neural Methods for Event Extraction*. PhD thesis. – Cité page 22.

[Bos et al., 2017] Bos, J., Basile, V., Evang, K., Venhuizen, N. J., and Bjerva, J. (2017). The groningen meaning bank. In *Handbook of linguistic annotation*, pages 463–496. Springer. – Cité page 14.

[Budi and Bressan, 2003] Budi, I. and Bressan, S. (2003). Association rules mining for name entity recognition. In *Proceedings of the Fourth International Conference on Web Information Systems Engineering, 2003. WISE 2003.*, pages 325–328. IEEE. – Cité page 22.

[Carreras et al., 2002] Carreras, X., Màrquez, L., and Padró, L. (2002). Named entity extraction using adaboost. In *COLING-02: The 6th Conference on Natural Language Learning 2002 (CoNLL-2002)*. – Cité pages 8 et 22.

[Chinchor and Robinson, 1997] Chinchor, N. and Robinson, P. (1997). Muc-7 named entity task definition. In *Proceedings of the 7th Conference on Message Understanding*, volume 29, pages 1–21. – Cité page 8.

[Chinchor, 1998] Chinchor, N. A. (1998). Overview of muc-7/met-2. Technical report, SCIENCE APPLICATIONS INTERNATIONAL CORP SAN DIEGO CA. – Cité pages 8 et 12.

[Chiticariu et al., 2013] Chiticariu, L., Li, Y., and Reiss, F. R. (2013). Rule-based information extraction is dead! long live rule-based information extraction systems! In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 827–832. – Cité page 21.

[Collins and Singer, 1999] Collins, M. and Singer, Y. (1999). Unsupervised models for named entity classification. In *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*. – Cité pages 19 et 22.

[Cortes and Vapnik, 1995] Cortes, C. and Vapnik, V. (1995). Support-vector networks. *Machine learning*, 20(3):273–297. – Cité page 23.

[Daille et al., 2000] Daille, B., Fourour, N., and Morin, E. (2000). Catégorisation des noms propres: une étude en corpus. *Cahiers de grammaire*, 25(25):115–129. – Cité page 8.

[Derczynski et al., 2016] Derczynski, L., Bontcheva, K., and Roberts, I. (2016). Broad twitter corpus: A diverse named entity recognition resource. In *COLING*. – Cité page 6.

[Derczynski et al., 2013] Derczynski, L., Ritter, A., Clark, S., and Bontcheva, K. (2013). Twitter part-of-speech tagging for all: Overcoming sparse and noisy data. In *Proceedings of the International Conference Recent Advances in Natural Language Processing RANLP 2013*, pages 198–206. – Cité page 9.

[Devlin et al., 2018] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2018). Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*. – Cité pages 21 et 25.

[Downey et al., 2007] Downey, D., Broadhead, M., and Etzioni, O. (2007). Locating complex named entities in web text. In *IJCAI*, volume 7, pages 2733–2739. – Cité page 19.

[Ehrmann, 2008] Ehrmann, M. (2008). *Les Entitées Nommées, de la linguistique au TAL: Statut théorique et méthodes de désambiguïsation*. PhD thesis. – Cité pages 5, 8 et 13.

[Ehrmann et al., 2016] Ehrmann, M., Nouvel, D., and Rosset, S. (2016). Named entity resources-overview and outlook. – Cité pages 11, 12 et 13.

[Elsner et al., 2009] Elsner, M., Charniak, E., and Johnson, M. (2009). Structured generative models for unsupervised named-entity clustering. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 164–172. Association for Computational Linguistics. – Cité page 19.

[Enjalbert, 2005] Enjalbert, P. (2005). L'extraction d'information, chapitre 8. – Cité page 8.

[Etzioni et al., 2005] Etzioni, O., Cafarella, M., Downey, D., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D. S., and Yates, A. (2005). Unsupervised named-entity extraction from the web: An experimental study. *Artificial intelligence*, 165(1):91–134. – Cité page 8.

[Finkel et al., 2005] Finkel, J. R., Grenager, T., and Manning, C. (2005). Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 363–370. Association for Computational Linguistics. – Cité page 23.

[Fleischman and Hovy, 2002] Fleischman, M. and Hovy, E. (2002). Fine grained classification of named entities. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics. – Cité page 12.

[Friburger, 2002] Friburger, N. (2002). *Reconnaissance automatique des noms propres: application à la classification automatique de textes journalistiques*. PhD thesis, Tours. – Cité page 8.

[Galibert et al., 2011] Galibert, O., Rosset, S., Grouin, C., Zweigenbaum, P., and Quintard, L. (2011). Structured and extended named entity evaluation in automatic speech transcriptions. In *Proceedings of 5th International Joint Conference on Natural Language Processing*, pages 518–526. – Cité page 8.

[Galliano et al., 2009] Galliano, S., Gravier, G., and Chaubard, L. (2009). The ester 2 evaluation campaign for the rich transcription of french radio broadcasts. In *Tenth Annual Conference of the International Speech Communication Association*. – Cité page 13.

[Ghaddar and Langlais, 2017] Ghaddar, A. and Langlais, P. (2017). Winer: A wikipedia annotated corpus for named entity recognition. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 413–422. – Cité page 14.

[Grishman and Sundheim, 1995] Grishman, R. and Sundheim, B. (1995). Design of the muc-6 evaluation. In *Proceedings of the 6th conference on Message understanding*, pages 1–11. Association for Computational Linguistics. – Cité page 7.

[Grishman and Sundheim, 1996] Grishman, R. and Sundheim, B. (1996). Message understanding conference-6: A brief history. In *COLING 1996 Volume 1: The 16th International Conference on Computational Linguistics*. – Cité page 12.

[Grouin, 2013] Grouin, C. (2013). *Anonymisation de documents cliniques: performances et limites des méthodes symboliques et par apprentissage statistique*. PhD thesis. – Cité page 26.

[Grouin et al., 2011] Grouin, C., Rosset, S., Zweigenbaum, P., Fort, K., Galibert, O., and Quintard, L. (2011). Proposal for an extension of traditional named entities: From guidelines to evaluation, an overview. In *Proceedings of the 5th linguistic annotation workshop*, pages 92–100. Association for Computational Linguistics. – Cité page 13.

[Group et al., 2009] Group, T. W. et al. (2009). Guidelines for temporal expression annotation for english for tempeval 2010. – Cité page 13.

[Gut and Bayerl, 2004] Gut, U. and Bayerl, P. S. (2004). Measuring the reliability of manual annotations of speech corpora. In *Speech Prosody 2004, International Conference*. – Cité page 32.

[Harabagiu et al., 2003] Harabagiu, R. S., Harabagiu, A., Moldovan, D., Clark, C., Bowden, M., Williams, J., and Bensley, J. (2003). Answer mining by combining extraction techniques with abductive. – Cité page 12.

[Harris, 1954] Harris, Z. S. (1954). Distributional structure. *Word*, 10(2-3):146–162. – Cité page 25.

[Hemalatha et al., 2012] Hemalatha, I., Varma, G. S., and Govardhan, A. (2012). Preprocessing the informal text for efficient sentiment analysis. *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*, 1(2):58–61. – Cité page 25.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780. – Cité page 24.

[Hoffmann et al., 2011] Hoffmann, R., Zhang, C., Ling, X., Zettlemoyer, L., and Weld, D. S. (2011). Knowledge-based weak supervision for information extraction of overlapping relations. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, pages 541–550. Association for Computational Linguistics. – Cité page 5.

[Huang et al., 2015] Huang, Z., Xu, W., and Yu, K. (2015). Bidirectional lstm-crf models for sequence tagging. *arXiv preprint arXiv:1508.01991*. – Cité pages 24 et 27.

[Isozaki and Kazawa, 2002] Isozaki, H. and Kazawa, H. (2002). Efficient support vector classifiers for named entity recognition. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics. – Cité page 21.

[Iwańska, 1995] Iwańska, L. (1995). Wayne state university: Description of the uno natural language processing system as used for muc-6. In *Proceedings of the 6th conference on Message understanding*, pages 263–277. Association for Computational Linguistics. – Cité page 22.

[Iwanska et al., 1995] Iwanska, L., Croll, M., Yoon, T., and Adams, M. (1995). Wayne state university: Description of the uno natural language processing system as used for MUC-6. In *Sixth Message Understanding Conference (MUC-6): Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*. – Cité page 21.

[Jean-Louis, 2011] Jean-Louis, L. (2011). *Approches supervisées et faiblement supervisées pour l'extraction d'événements et le peuplement de bases de connaissances*. PhD thesis, Paris 11. – Cité page 21.

[Jurafsky and Martin, 2009] Jurafsky, D. and Martin, J. H. (2009). *Speech and Language Processing (2Nd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA. – Cité pages 8, 22 et 23.

[Kaufmann and Kalita, 2010] Kaufmann, M. and Kalita, J. (2010). Syntactic normalization of twitter messages. In *International conference on natural language processing, Kharagpur, India*. – Cité page 25.

[Knoblock et al., 2007] Knoblock, C., Lopresti, D., Roy, S., and Subramaniam, L. V. (2007). Special issue on noisy text analytics. *International Journal on Document Analysis and Recognition*, 10(3):127–128. – Cité page 9.

[Krippendorff, 1980] Krippendorff, K. (1980). *Content analysis: an introduction to its methodology*. Sage, London, UK. – Cité page 15.

[Lafferty et al., 2001] Lafferty, J., McCallum, A., and Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. – Cité page 23.

[Lample et al., 2016] Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K., and Dyer, C. (2016). Neural architectures for named entity recognition. *arXiv preprint arXiv:1603.01360*. – Cité pages 8 et 21.

[Le Meur et al., 2004] Le Meur, C., Galliano, S., and Geoffrois, E. (2004). Conventions d'annotations en entités nommées-ester. *Rapport technique de la campagne Ester*. – Cité page 8.

[Li, 2012] Li, H. (2012). Statistical learning method. *Tsinghua university press*, pages 95–115. – Cité page 22.

[Li and Roth, 2002] Li, X. and Roth, D. (2002). Learning question classifiers. In *Proceedings of the 19th international conference on Computational linguistics-Volume 1*, pages 1–7. Association for Computational Linguistics. – Cité page 12.

[Ma and Hovy, 2016] Ma, X. and Hovy, E. (2016). End-to-end sequence labeling via bi-directional lstm-cnns-crf. *arXiv preprint arXiv:1603.01354*. – Cité pages 21 et 24.

[Magnini et al., 2008] Magnini, B., Cappelli, A., Tamburini, F., Bosco, C., Mazzei, A., Lombardo, V., Bertagna, F., Calzolari, N., Toral, A., Lenzi, V. B., et al. (2008). Evaluation of natural language tools for italian: Evalita 2007. In *LREC*. – Cité page 12.

[Mai et al., 2018] Mai, K., Pham, T.-H., Nguyen, M. T., Nguyen, T. D., Bollegala, D., Sasano, R., and Sekine, S. (2018). An empirical study on fine-grained named entity recognition. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 711–722. – Cité page 21.

[Marsh and Perzanowski, 1998] Marsh, E. and Perzanowski, D. (1998). Muc-7 evaluation of ie technology: Overview of results. In *Seventh Message Understanding Conference (MUC-7): Proceedings of a Conference Held in Fairfax, Virginia, April 29-May 1, 1998*. – Cité page 13.

[Martins et al., 2019] Martins, P. H., Marinho, Z., and Martins, A. F. (2019). Joint learning of named entity recognition and entity linking. *arXiv preprint arXiv:1907.08243*. – Cité page 21.

[Mayhew et al., 2019] Mayhew, S., Tsygankova, T., and Roth, D. (2019). ner and pos when nothing is capitalized. *arXiv preprint arXiv:1903.11222*. – Cité page 25.

[McCallum and Li, 2003] McCallum, A. and Li, W. (2003). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the seventh conference on Natural language learning at HLT-NAACL 2003-Volume 4*, pages 188–191. Association for Computational Linguistics. – Cité page 21.

[McNamee and Mayfield, 2002] McNamee, P. and Mayfield, J. (2002). Entity extraction without language-specific resources. In *proceedings of the 6th conference on Natural language learning-Volume 20*, pages 1–4. Association for Computational Linguistics. – Cité page 23.

[Meur et al., 2004] Meur, C., Gallinao, S., and Geoffrois, E. (2004). Conventions d'annotations en entités nommées. – Cité page 8.

[Mikolov et al., 2013] Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119. – Cité page 25.

[Misawa et al., 2017] Misawa, S., Taniguchi, M., Miura, Y., and Ohkuma, T. (2017). Character-based bidirectional lstm-crf with words and characters for japanese named entity recognition. In *Proceedings of the First Workshop on Subword and Character Level Models in NLP*, pages 97–102. – Cité page 21.

[Nadeau and Sekine, 2007] Nadeau, D. and Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26. – Cité pages 8, 13 et 22.

[Nebhi et al., 2015] Nebhi, K., Bontcheva, K., and Gorrell, G. (2015). Restoring capitalization in# tweets. In *Proceedings of the 24th International Conference on World Wide Web*, pages 1111–1115. ACM. – Cité page 25.

[Nouvel, 2012] Nouvel, D. (2012). Reconnaissance des entités nommées par exploration de règles d'annotation. *These de doctorat, Université François Rabelais de Tours, Tours, France*. – Cité pages 8 et 21.

[Nouvel et al., 2016] Nouvel, D., Ehrmann, M., and Rosset, S. (2016). *Named Entities for Computational Linguistics*. Wiley Online Library. – Cité pages 8 et 11.

[Ohta et al., 2002] Ohta, T., Tateisi, Y., and Kim, J.-D. (2002). The genia corpus: An annotated research abstract corpus in molecular biology domain. In *Proceedings of the second international conference on Human Language Technology Research*, pages 82–86. Morgan Kaufmann Publishers Inc. – Cité page 12.

[Pennington et al., 2014] Pennington, J., Socher, R., and Manning, C. (2014). Glove: Global vectors for word representation. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pages 1532–1543. – Cité page 25.

[Peters et al., 2018] Peters, M. E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., and Zettlemoyer, L. (2018). Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*. – Cité pages 21 et 25.

[Poibeau, 2003] Poibeau, T. (2003). *Extraction automatique d'information: Du texte brut au web sémantique*. – Cité page 8.

[Poibeau, 2011] Poibeau, T. (2011). *Traitement automatique du contenu textuel*. Lavoisier. – Cité pages 7, 21 et 23.

[Porta and Sancho, 2013] Porta, J. and Sancho, J.-L. (2013). Word normalization in twitter using finite-state transducers. *Tweet-Norm@ SEPLN*, 1086:49–53. – Cité page 25.

[Pustejovsky and Stubbs, 2012] Pustejovsky, J. and Stubbs, A. (2012). *Natural Language Annotation for Machine Learning: A guide to corpus-building for applications*. " O'Reilly Media, Inc.". – Cité pages 13 et 15.

[Rau, 1991] Rau, L. F. (1991). Extracting company names from text. In *[1991] Proceedings. The Seventh IEEE Conference on Artificial Intelligence Application*, volume 1, pages 29–32. IEEE. – Cité pages 21 et 22.

[Ritter et al., 2011] Ritter, A., Clark, S., Etzioni, O., et al. (2011). Named entity recognition in tweets: an experimental study. In *Proceedings of the conference on empirical methods in natural language processing*, pages 1524–1534. Association for Computational Linguistics. – Cité pages 6, 9, 14 et 19.

[Ritter et al., 2012] Ritter, A., Etzioni, O., Clark, S., et al. (2012). Open domain event extraction from twitter. In *Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 1104–1112. ACM. – Cité page 5.

[Rosset et al., 2007] Rosset, S., Galibert, O., Adda, G., and Bilinski, E. (2007). The limsi participation in the qast track. In *Workshop of the Cross-Language Evaluation Forum for European Languages*, pages 414–423. Springer. – Cité page 12.

[Rosset et al., 2011] Rosset, S., Grouin, C., and Zweigenbaum, P. (2011). *Entités nommées structurées: guide d'annotation Quaero*. LIMSI-Centre national de la recherche scientifique. – Cité pages 8 et 13.

[Rumelhart et al., 1988] Rumelhart, D. E., Hinton, G. E., Williams, R. J., et al. (1988). Learning representations by back-propagating errors. *Cognitive modeling*, 5(3):1. – Cité page 24.

[Sang and De Meulder, 2003] Sang, E. F. and De Meulder, F. (2003). Introduction to the conll-2003 shared task: Language-independent named entity recognition. *arXiv preprint cs/0306050*. – Cité pages 8 et 12.

[Sekine and Isahara, 2000] Sekine, S. and Isahara, H. (2000). Irex: Ir & ie evaluation project in japanese. In *LREC*, pages 1977–1980. Citeseer. – Cité page 12.

[Sekine and Nobata, 2004] Sekine, S. and Nobata, C. (2004). Definition, dictionaries and tagger for extended named entity hierarchy. In *LREC*, pages 1977–1980. Lisbon, Portugal. – Cité page 13.

[Sekine et al., 2002] Sekine, S., Sudo, K., and Nobata, C. (2002). Extended named entity hierarchy. In *LREC*. – Cité page 8.

[Stenetorp et al., 2012] Stenetorp, P., Pyysalo, S., Topić, G., Ohta, T., Ananiadou, S., and Tsujii, J. (2012). Brat: a web-based tool for nlp-assisted text annotation. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 102–107. Association for Computational Linguistics. – Cité page 18.

[Straková et al., 2019] Straková, J., Straka, M., and Hajič, J. (2019). Neural architectures for nested ner through linearization. *arXiv preprint arXiv:1908.06926*. – Cité page 25.

[Subramaniam et al., 2009] Subramaniam, L. V., Roy, S., Faruquie, T. A., and Negi, S. (2009). A survey of types of text noise and techniques to handle noisy text. In *Proceedings of The Third Workshop on Analytics for Noisy Unstructured Text Data*, pages 115–122. ACM. – Cité page 9.

[Sundheim, 1995] Sundheim, B. M. (1995). Overview of results of the muc-6 evaluation. In *Proceedings of the 6th conference on Message understanding*, pages 13–31. Association for Computational Linguistics. – Cité page 13.

[Takeuchi and Collier, 2002] Takeuchi, K. and Collier, N. (2002). Use of support vector machines in extended named entity recognition. In *proceedings of the 6th conference on Natural language learning-Volume 20*, pages 1–7. Association for Computational Linguistics. – Cité page 21.

[Tanev and Magnini, 2006] Tanev, H. and Magnini, B. (2006). Weakly supervised approaches for ontology population. In *11th Conference of the European Chapter of the Association for Computational Linguistics*. – Cité page 5.

[Tran, 2006] Tran, M. (2006). *Prolexbase: un dictionnaire relationnel multilingue de noms propre: conception, implémentation et gestion en ligne*. PhD thesis, Tours. – Cité pages 8 et 13.

[Vicente, 2005] Vicente, M. R. (2005). La glose comme outil de désambiguïsation référentielle des noms propres purs. *Corela. Cognition, représentation, langage*, (HS-2). – Cité page 8.

[Vogel and Tresner-Kirsch, 2012] Vogel, J. and Tresner-Kirsch, D. (2012). Robust language identification in short, noisy texts: Improvements to liga. In *The Third International Workshop on Mining Ubiquitous and Social Environments*, pages 43–50. – Cité page 9.

[Weischedel et al., 2013] Weischedel, R., Palmer, M., Marcus, M., Hovy, E., Pradhan, S., Ramshaw, L., Xue, N., Taylor, A., Kaufman, J., Franchini, M., et al. (2013). Ontonotes release 5.0 ldc2013t19. *Linguistic Data Consortium, Philadelphia, PA*, 23. – Cité page 14.

[Weissenbacher, 2003] Weissenbacher, D. (2003). Etude et reconnaissance automatique des relations de synonymie et de renommage dans les textes de génomique. *Mémoire de DEA, Laboratoire d'informatique de Paris Nord.(sous la direction de Mme. Adeline Nazarenko)*. – Cité page 8.

[Witten et al., 1999] Witten, I. H., Witten, I. H., Moffat, A., Bell, T. C., Bell, T. C., and Bell, T. C. (1999). *Managing gigabytes: compressing and indexing documents and images*. Morgan Kaufmann. – Cité page 14.

[Yang and Zhang, 2018] Yang, J. and Zhang, Y. (2018). Ncrf++: An open-source neural sequence labeling toolkit. *arXiv preprint arXiv:1806.05626*. – Cité page 21.

[Zheng and Casari, 2018] Zheng, A. and Casari, A. (2018). *Feature engineering for machine learning: principles and techniques for data scientists*. " O'Reilly Media, Inc.". – Cité page 23.

[Zhou and Su, 2002] Zhou, G. and Su, J. (2002). Named entity recognition using an hmm-based chunk tagger. In *proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 473–480. Association for Computational Linguistics. – Cité pages 21 et 23.

Index