UNIVERSITÉ DU QUÉBEC À MONTRÉAL

# APPORT D'UNE APPROCHE HYBRIDE À BASE DE RÉSEAUX DE NEURONES ET DE STATISTIQUES POUR LA RECONNAISSANCE DES ENTITÉS NOMMÉES EN DOMAINES GÉNÉRAL ET RESTREINT

MÉMOIRE

PRÉSENTÉ

COMME EXIGENCE PARTIELLE

DE LA MAÎTRISE EN INFORMATIQUE

PAR

GHAITH DEKHILI

JUIN 2020

UNIVERSITÉ DU QUÉBEC À MONTRÉAL
Service des bibliothèques

*Avertissement*

La diffusion de ce mémoire se fait dans le respect des droits de son auteur, qui a signé le formulaire *Autorisation de reproduire et de diffuser un travail de recherche de cycles supérieurs* (SDU-522 – Rév.10-2015). Cette autorisation stipule que «conformément à l'article 11 du Règlement no 8 des études de cycles supérieurs, [l'auteur] concède à l'Université du Québec à Montréal une licence non exclusive d'utilisation et de publication de la totalité ou d'une partie importante de [son] travail de recherche pour des fins pédagogiques et non commerciales. Plus précisément, [l'auteur] autorise l'Université du Québec à Montréal à reproduire, diffuser, prêter, distribuer ou vendre des copies de [son] travail de recherche à des fins non commerciales sur quelque support que ce soit, y compris l'Internet. Cette licence et cette autorisation n'entraînent pas une renonciation de [la] part [de l'auteur] à [ses] droits moraux ni à [ses] droits de propriété intellectuelle. Sauf entente contraire, [l'auteur] conserve la liberté de diffuser et de commercialiser ou non ce travail dont [il] possède un exemplaire.»

# REMERCIEMENTS

Mes vifs remerciements à ma directrice de recherche, Mme Fatiha Sadat, professeure à l'Université du Québec à Montréal (UQAM) d'avoir dirigé ce mémoire, pour son soutien précieux, son aide, ses conseils judicieux et son encouragement durant toute la période de ce travail.

Mes sincères remerciements à toutes les professeures et tous les professeurs du département d'informatique de l'UQAM qui m'ont enseigné durant mes années d'études, pour la qualité de leur enseignement.

Un grand merci aux membres du jury pour l'intérêt qu'ils ont porté à mon travail en acceptant d'examiner ce mémoire et de l'enrichir par leurs propositions.

Je remercie également tous mes collègues étudiants du laboratoire qui sont devenus non seulement de bons collègues, mais en plus de bons amis.

Je dédie ce mémoire à mes parents et à mes sœurs pour leur patience, soutien moral et encouragements tout au long de mes études universitaires.

Enfin un remerciement spécial à mes nièces Aïcha, Khadija, Zeineb, Zahra et Kenza et à mon neuveu Yahya.

TABLE DES MATIÈRES

LISTE DES TABLEAUX

LISTE DES FIGURES

LISTE DES ACRONYMES

| | |
|---|---|
| **BiLSTM** | Réseaux de neurones récurrents bidirectionnels à mémoire court terme et long terme (**Bi**directional **L**ong **S**hort-**T**erm **M**emory) |
| **CNN** | Réseaux de neurones convolutifs (**C**onvolutional **N**eural **N**etworks) |
| **CRF** | Champs aléatoires conditionnels (**C**onditional **R**andom **F**ields) |
| **EI** | **E**xtraction d'**I**nformation |
| **EN** | **E**ntité **N**ommée |
| **LSTM** | Réseaux de neurones récurrents à mémoire court terme et long terme (**L**ong **S**hort-**T**erm **M**emory) |
| **POS** | **P**art-**O**f-**S**peech |
| **REN** | **R**econnaissance des **E**ntités **N**ommées |
| **RNN** | Réseaux de neurones récurrents (**R**ecurrent **N**eural **N**etworks) |
| **TALN** | **T**raitement **A**utomatique du **L**angage **N**aturel |

# RÉSUMÉ

La Reconnaissance des Entités Nommées (REN) est une sous-tâche de l'activité d'extraction d'information et du Traitement Automatique du Langage Naturel. Elle consiste à identifier certains objets textuels tels que les noms de personne, d'organisation et de lieu. Ce travail de maîtrise se concentre sur la tâche de REN pour un domaine restreint, celui de l'électronique, caractérisé par la disponibilité de peu de données annotées, tout en comparant à un domaine général où les données annotées de plus grande taille sont disponibles. En effet, cette tâche pose un certain nombre de difficultés et de défis qui sont inhérents aux caractéristiques du traitement de données d'un domaine particulier restreint et des deux tâches d'annotation et d'apprentissage machine pour la REN.

Dans un premier temps, nous étudions les spécificités de la REN en utilisant trois approches : statistique, à base de réseaux de neurones et hybrides. Dans un second temps, nous proposons d'étudier une méthode d'apprentissage par transfert pour la REN depuis un domaine général pour un apprentissage vers un domaine restreint. Dans un troisième temps, nous étudions la contribution de l'emploi d'un graphe de connaissances basé sur le bon sens dans l'amélioration des performances d'un système REN.

Nous utilisons dans notre architecture des couches du réseau récurrent bidirectionnel à mémoire court et long terme, combinées à une couche de Champ Aléatoire Conditionnel augmenté avec d'autres caractéristiques.

Nous menons différents types d'expérimentations afin d'optimiser et d'évaluer les approches proposées.

D'après les évaluations et résultats obtenus, nous constatons que le modèle basé sur l'apprentissage par transfert donne de meilleurs résultats et augmente les scores de la F-mesure de 15%, 6% et 5% par rapport au modèle statistique de base, au modèle statistique avec caractéristiques orthographiques et au modèle neuronal de base respectivement. Les résultats obtenus avec l'emploi d'un graphe de connaissances basé sur le bon sens ont montré également une amélioration de 2.86% dans la F-mesure du système global.

Mots-clés : Traitement automatique des langues naturelles, reconnaissance des entités nommées, réseaux de neurones, Champ Aléatoire Conditionnel, apprentissage

par transfert, plongements de mots.

# INTRODUCTION

L'information textuelle devient de plus en plus disponible sur le Web, augmentant la nécessité de techniques et moyens pour traiter des données stockées dans une forme structurée ou non structurée. L'extraction d'information a émergé comme une solution pour faire face à ce problème. Cette dernière commence par une collection de texte qu'elle transforme en information plus facile à analyser, en isolant les fragments de texte pertinents, et en extrayant l'information pertinente de ces fragments (Cowie et Lehnert, 1996). Extraire l'information utile à partir de données volumineuses reste un grand défi qui demande de développer de nouvelles technologies pour gérer de telles quantités de données. En effet, nombreux domaines d'extraction d'information et de traitement du langage naturel demandent certains outils de pré-traitement pour analyser la structure lexicale, morphologique, phonétique, syntaxique et sémantique du texte (Goyal *et al.*, 2018).

Dans ce mémoire, nous nous intéressons à l'une des plus importantes sous-tâches en extraction d'information, la Reconnaissance des Entitées Nommées (REN). Cette dernière peut être considérée comme une technique de pré-traitement du texte qui joue un rôle vital dans différentes applications liées au langage naturel telles que le résumé automatique, la traduction automatique, la recherche d'information, les systèmes de questions réponses, etc. Cette tâche consiste à identifier certains objets textuels tels que les noms de personne, d'organisation et de lieu. Le concept d'extraction d'entités nommées a été proposé dans la sixième conférence *Message Understanding Conference* en 1996. Depuis, nombreuses techniques ont été développées par plusieurs chercheurs pour extraire différentes entités à partir

de différentes langues et différents genres de textes. Un intérêt croissant existe encore parmi la communauté de recherche pour développer davantage de nouvelles approches pour extraire diverses entités nommées qui sont utiles dans diverses applications en langage naturel (Goyal *et al.*, 2018).

Les premiers systèmes REN étaient basés sur des règles définies par les experts, des traits orthographiques et des ontologies. Ces systèmes ont été suivis par les systèmes REN basés sur les caractéristiques extraites manuellement et l'apprentissage automatique (Nadeau et Sekine, 2007). Un peu plus tard Collobert *et al.* (2011) ont introduit les systèmes REN basés sur les réseaux de neurones avec un minimum de caractéristiques extraites à la main. Ces modèles ont eu du succès puisqu'ils ne demandent pas de ressources spécifiques au domaine telles que les lexiques et les ontologies, et sont par conséquent plus indépendants du domaine. Différentes architectures neuronales ont été proposées, la plupart basées sur une sorte de réseaux de neurones récurrents (*Recurrent Neural Networks*) ou RNN appliqués sur les mots, les sous mots et/ou les caractères (Yadav et Bethard, 2018).

## 0.1 Problématique

Les systèmes REN à base de connaissances n'exigent pas de données d'entraînement annotées puisqu'ils s'appuient sur le lexique et les connaissances spécifiques aux domaines. Ceux-ci fonctionnent bien lorsque le lexique est exhaustif, mais échouent dans les exemples qui ne figurent pas dans les dictionnaires du domaine par exemple. La précision est généralement élevée pour ces systèmes grâce au lexique, mais le rappel est souvent faible à cause des règles spécifiques aux langues, aux domaines et aux dictionnaires incomplets. Un autre inconvénient des systèmes REN à base de connaissances c'est d'avoir besoin d'experts du domaine pour construire et maintenir les ressources de connaissances (Yadav et Bethard,

2018). Finalement ces systèmes ne peuvent être utilisés que dans les domaines pour lesquels ils ont été conçus (Goyal *et al.*, 2018).

Les modèles supervisés d'apprentissage automatique apprennent à faire des prédictions en passant par un entraînement sur des exemples d'entrées et leurs sorties attendues et peuvent être utilisés pour remplacer les règles établies par les experts (Yadav et Bethard, 2018). Ces méthodes construisent le modèle en l'entraînant sur des exemples d'entraînement et par la suite ce modèle est utilisé pour prédire des exemples similaires du même genre dans les données tests. Les données étiquetées sont l'entrée essentielle de ces méthodes d'apprentissage. Annoter les données d'entraînement est une tâche fastidieuse qui prend du temps et demande l'engagement d'experts du domaine pour annoter les données efficacement en quantité suffisante (Goyal *et al.*, 2018). Cette tâche devient encore plus coûteuse quand il s'agit d'un domaine spécifique d'où l'impossiblié d'utiliser des données libre d'accès (*open source*) déjà annotées.

Des modèles non supervisés d'apprentissage automatique sont apparus également pour remédier au problème du manque de données annotées. L'apprentissage non supervisé est un algorithme qui utilise l'information qui est ni classifiée ni étiquetée. Ces méthodes utilisent des données purement non étiquetées pour la prise de décisions. Le but de l'apprentissage non supervisé est de générer un modèle qui considère les caractéristiques structurelles et distributionnelles des données pour apprendre plus sur les données (Goyal *et al.*, 2018), mais l'inconvénient de ces modèles c'est la difficulté et complexité pour obtenir de bonnes performances, en comparaison à l'apprentissage supervisé, puisque ces derniers vont devoir détecter par eux-même les similarités entre les données.

Mis à part le problème de données annotées, les modèles neuronaux demandent généralement pour être entraînés une grande quantité de données annotées pour

produire des modèles puissants et prévenir le sur-apprentissage. Une hypothèse majeure dans de nombreux algorithmes d'apprentissage automatique est que les données d'entraînement et les données futures doivent être dans le même espace d'attributs et doivent avoir la même distribution, quand la distribution change, la majorité des modèles ont besoin d'être reconstruits à zero, en utilisant de nouvelles données d'entraînement. Par contre, dans plusieurs applications des cas réels, cette hypothèse peut ne pas tenir. Par exemple, nous avons parfois une tâche de classification liée à un certain domaine, mais nous avons suffisamment de données d'entraînement dans un autre domaine, où ces dernières données peuvent être dans un autre espace d'attributs, ou bien elles suivent une distribution de données différente. Dans ce cas, il est difficile d'atteindre la performance des modèles de l'état de l'art basés sur des attributs extraits manuellement en appliquant les modèles neuronaux entraînés sur une petite quantité de données annotées. Dans ce contexte, le transfert de connaissance, si appliqué avec succès, améliorerait la performance de l'apprentissage en nous évitant la tâche coûteuse d'étiquetage des données, et en exploitant des grandes quantités de données annotées des datasets hors-domaine, et utiliser les traits appris pour effectuer une tâche sur un dataset cible (Pan et Yang, 2010; Meftah *et al.*, 2018).

Les systèmes présentés précédemment deviennent encore moins performants dans des situations qui demandent un raisonnement basé sur le bon sens (en anglais : *Commonsense*), où il est souvent requis de résoudre l'ambiguïté causée par l'information implicite dans une phrase. Dans ce cas, développer des modèles capables de raisonner en intégrant des connaissances basées sur le *Commonsense* permetterait à un système REN de mieux comprendre le sens derrière les mots que les utilisateurs emploient, et d'apprendre plus d'entités pour pouvoir améliorer son efficacité.

## 0.2 Objectifs

Les objectifs principaux de ce mémoire sont comme suit :

- L'extraction des ENs en anglais dans un domaine spécifique qui est le domaine de l'électronique en utilisant un modèle statistique ;
- Utilisation d'attributs de base ;
- Amélioration de la performance du système en ajoutant d'autres attributs orthographiques ;
- Étude d'un modèle à base de réseaux de neurones pour la REN ;
- Étude de l'apprentissage par transfert pour transférer la connaissance d'un domaine général avec une quantité de données annotées suffisante vers un domaine spécifique avec une petite quantité de données ;
- Étude de l'emploi du *Commonsense* dans la tâche de la REN dans le domaine général ;
- Évaluation des différents modèles construits par des mesures telles que la précision, le rappel et la F-mesure et la comparaison des résultats obtenus avec les différents modèles.

## 0.3 Contributions

Nos contributions portent sur trois aspects distincts. Le premier aspect consiste au développement d'un système de reconnaissance des entités nommées dans un domaine restreint qui est le domaine de l'électronique. L'enjeu dans notre cas consiste au fait que non seulement nous ne disposons pas de données suffisantes pour bien entraîner nos modèles, mais en plus le nombre d'entités est assez élevé comparativement aux données libres d'accès qui existent déjà dans les autres domaines.

Le deuxième aspect repose sur l'application de la technique de transfert d'apprentissage dans un domaine restreint, en transférant la connaissance d'un domaine source qui est le domaine général, vers un domaine cible qui est le domaine de l'électronique. Au mieux de notre investigation de la littérature, la majorité des méthodes existantes sont désignées pour les problèmes liés à la classification de l'image et du texte au lieu des problèmes de classification des séquences ce qui rend notre tâche plus complexe.

Le troisième aspect consiste à l'emploi du *Commonsense* pour améliorer les performances d'un système REN. À notre connaissance, c'est la première étude qui explore cette piste dans la tâche de la REN dans le domaine général.

## 0.4 Organisation du mémoire

Après l'introduction, la problématique, les objectifs de ce mémoire et les contributions, nous présentons sa structure.

Le chapitre I expose les notions de base de notre travail. Il comporte 5 sections. La première section explique ce que c'est la REN. La deuxième section présente les données annotées pour la REN. La troisième section décrit les plongements de mots. La quatrième section présente le modèle statistique utilisé dans notre travail et finalement, la cinquième partie décrit les architectures de réseaux de neurones utilisées dans nos expérimentations .

Le chapitre II est un état de l'art sur les différentes approches pour l'extraction des ENs.

Le chapitre III décrit notre méthodologie pour l'extraction des ENs dans un domaine spécifique en utilisant un modèle statistique, un modèle neuronal, suivi par une étude de l'apprentissage par transfert pour transférer les connaissances d'un

domaine général vers un domaine spécifique.

Le chapitre IV est consacré à l'évaluation des modèles proposés. Nous commençons par décrire les données que nous avons utilisées. Ensuite, nous présentons les différentes évaluations et expérimentations accomplies. Nous enchaînons par une description de l'implémentation et de l'entraînement de nos modèles, suivie par une discussion sur les résultats obtenus.

Le chapitre V présente une nouvelle piste que nous explorons qui est l'apprentissage à partir d'une base de connaissances externe. Dans les premières parties nous introduisons quelques notions de base. Nous enchaînons par présenter notre approche proposée. En dernier lieu, nous présentons les évaluations des résultats obtenus.

Nous clôturons par une conclusion qui résume notre travail ainsi qu'une présentation de perspectives et travaux futurs.

# CHAPITRE I

# CONCEPTS DE BASE

## 1.1 Reconnaissance des Entités Nommées

La Reconnaissance des Entités Nommées (REN) consiste en un processus d'identification d'un mot ou d'un groupe de mots qui se réfèrent à une entité particulière dans un texte (Le, 2019). La figure 1.1 (Li *et al.*, 2018) montre un exemple où un système REN reconnait 3 entités nommées à partir d'une séquence de mots $s = (w_1, w_2, ..., w_n)$ donnée. Dans l'exmple $I_s \; \epsilon \; [1, N]$ et $I_e \; \epsilon \; [1, N]$ sont respectivement les indexes de début et de fin d'une mention d'une entité nommée, $t$ est le type d'entité à partir d'un ensemble de catégories prédéfini. *Organization*, *person*, *location*, *protein*, *drug* et *disease names* sont des exemples d'entités nommées.

Il existe plusieurs outils REN valables en ligne avec des modèles pré-entraînés. La table 1.1 résume les outils les plus populaires pour la REN en anglais. *StandfordCoreNLP*, *OSU Twitter NLP*, *Illinois NLP*, *NeuroNER*, *NERsuite*, *Polyglot* et *Gimli* sont les résultats de projets académiques. *spaCy*, *NLTK*, *OpenNLP*, *LingPipe*, *AllenNLP* et *IBM Watson* viennent du milieu industriel ou des projets *open source* (Li *et al.*, 2018).

Tableau 1.1 Outils de REN offerts par l'académie et des projets industriels

| **Système REN** | **URL** |
|---|---|
| StanfordCoreNLP | `https://stanfordnlp.github.io/CoreNLP/` |
| OSU Twitter NLP | `https://github.com/aritter/twitter_nlp` |
| Illinois NLP | `http://cogcomp.org/page/software/` |
| NeuroNER | `http://neuroner.com/` |
| NERsuite | `http://nersuite.nlplab.org/` |
| Polyglot | `https://polyglot.readthedocs.io/` |
| Gimli | `http://bioinformatics.ua.pt/gimli` |
| spaCy | `https://spacy.io/` |
| NLTK | `https://www.nltk.org/` |
| OpenNLP | `https://opennlp.apache.org/` |
| LingPipe | `http://alias-i.com/lingpipe-3.9.3/` |
| AllenNLP | `https://allennlp.org/models` |
| IBM Watson | `https://www.ibm.com/watson/` |

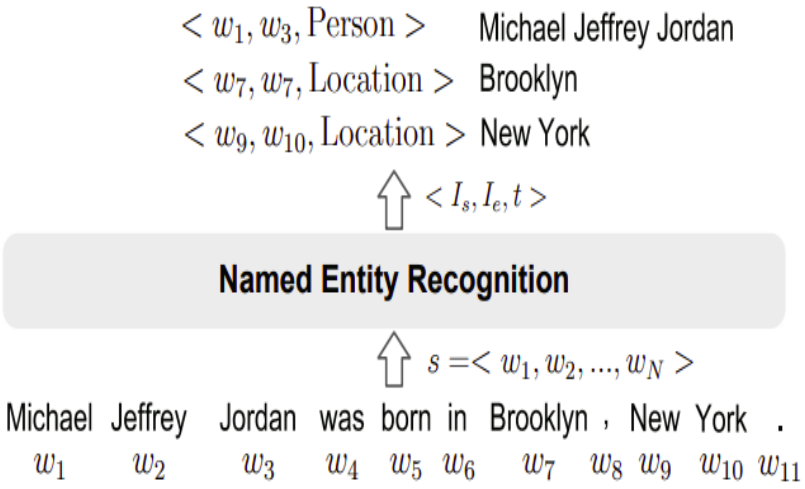


Figure 1.1 Exemple de reconnaissance d'entités nommées

## 1.2 Données annotées pour la REN

Les données annotées de haute qualité sont importantes autant pour l'apprentissage d'un modèle de prédiction que pour son évaluation. La table 1.2 liste quelques *datasets* les plus utilisés avec leurs sources de données et le nombre de types d'entités. Avant 2005, les *datasets* ont été développés en annotant les articles des nouvelles avec un nombre petit de types d'entités. Par la suite, plus de *datasets* ont été développés sur des sources de textes plus variées telles que les articles de Wikipédia, des conversations et du texte généré par des utilisateurs (les tweets et les commentaires sur YouTube par exemple), et le nombre des types des étiquettes est devenu plus large (89 pour OntoNotes). Dans la table 1.2, le nombre de types d'entités varient entre 2 dans GENETAG et 790 dans NCBI-Disease (Li *et al.*, 2018).

| Corpus | Year | Text Source | #Tags | URL |
|---|---|---|---|---|
| MUC-6 | 1995 | Wall Street Journal texts | 7 | https://catalog.ldc.upenn.edu/LDC2003T13 |
| MUC-6 Plus | 1995 | Additional news to MUC-6 | 7 | https://catalog.ldc.upenn.edu/LDC96T10 |
| MUC-7 | 1997 | New York Times news | 7 | https://catalog.ldc.upenn.edu/LDC2001T02 |
| CoNLL03 | 2003 | Reuters news | 4 | https://www.clips.uantwerpen.be/conll2003/ner/ |
| ACE | 2000 - 2008 | Transcripts, news | 7 | https://www.ldc.upenn.edu/collaborations/past-projects/ace |
| OntoNotes | 2007 - 2012 | Magazine, news, conversation, web | 89 | https://catalog.ldc.upenn.edu/LDC2013T19 |
| W-NUT | 2015 - 2018 | User-generated text | 18 | http://noisy-text.github.io |
| BBN | 2005 | Wall Street Journal texts | 64 | https://catalog.ldc.upenn.edu/ldc2005t33 |
| NYT | 2008 | New York Times texts | 5 | https://catalog.ldc.upenn.edu/LDC2008T19 |
| WikiGold | 2009 | Wikipedia | 4 | https://figshare.com/articles/Learning_multilingual_named_entity_recognition_from_Wikipedia/5462500 |
| WiNER | 2012 | Wikipedia | 4 | http://rali.iro.umontreal.ca/rali/en/winer-wikipedia-for-ner |
| WikiFiger | 2012 | Wikipedia | 113 | https://github.com/xiaoling/figer |
| $N^3$ | 2014 | News | 3 | http://aksw.org/Projects/N3NERNEDNIF.html |
| GENIA | 2004 | Biology and clinical texts | 36 | http://www.geniaproject.org/home |
| GENETAG | 2005 | MEDLINE | 2 | https://sourceforge.net/projects/bioc/files/ |
| FSU-PRGE | 2010 | PubMed and MEDLINE | 5 | https://julielab.de/Resources/FSU_PRGE.html |
| NCBI-Disease | 2014 | PubMed | 790 | https://www.ncbi.nlm.nih.gov/CBBresearch/Dogan/DISEASE/ |
| BC5CDR | 2015 | PubMed | 3 | http://bioc.sourceforge.net/ |
| DFKI | 2018 | Business news and social media | 7 | https://dfki-lt-re-group.bitbucket.io/product-corpus/ |

Figure 1.2 Liste des datasets annotés pour la REN en anglais

## 1.3 Les plongements de mots

Les plongements de mots ou *word embeddings* sont des représentations vectorielles des mots qui peuvent être optimisés pour avoir des propriétés reflétant le sens sémantique des mots qu'ils représentent. Les mots sont représentés par un nombre fixe de réels qui forment un vecteur. Pour plusieurs applications, il est désirable que les mots ayant un sens sémantique similaire, aient des vecteurs similaires. Autrement dit, les mots qui ont approximativement le même sens devraient avoir approximativement les mêmes vecteurs. En d'autres termes, les mots peuvent être considérés comme étant placés dans un espace multi-dimentionnel. Le nombre de dimensions étant fixé à l'avance, et les mots reliés sémantiquement sont regroupés ensemble. Puisque les vecteurs de mots partagent les mêmes dimensions, ils peuvent être considérérs comme étant des représentations distribuées de mots, contrairement aux représentations locales, où chaque mot aurait une valeur qui

lui est associée. La figure 1.3 (Kenter, 2017) fournit une visualisation de plongements de mots à 3 dimensions. Une représentation vectorielle d'un mot est notée $\overrightarrow{mot}$. La figure illustre que les vecteurs de mots reliés sématiquement, $\overrightarrow{newspaper}$ et $\overrightarrow{magazine}$, sont plus proches l'un de l'autre que du vecteur du mot $\overrightarrow{biking}$ (Kenter, 2017).

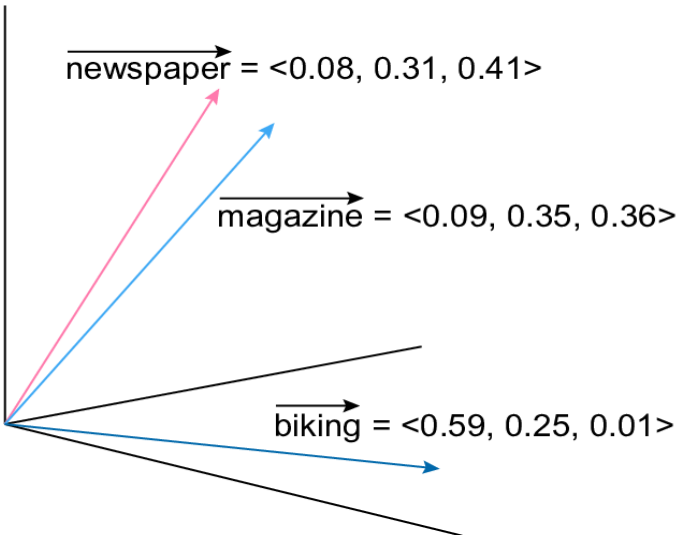


Figure 1.3 Exemple de plongements de mots à trois dimensions

Apprendre ces vecteurs de mots à partir d'un corpus textuel peut être réalisé en exploitant des réseaux de neurones entraînés sur de grandes quantités de données non étiquetées, une approche devenue possible grâce aux avancées récentes dans l'apprentissage non supervisé des plongements de mots sur des quantités massives de données, et grâce aux algorithmes d'entraînement des réseaux de neurones permettant des architectures profondes (Chiu et Nichols, 2015a). Les modèles résultants sont communément appelés des modèles de plongements de mots et il a été démontré qu'ils fournissent une connaissance préalable importante grâce à leur pouvoir de généralisation (Goldberg, 2016; Camacho-Collados et Pilehvar, 2018). Cette propriété s'est révélée déterminante pour atteindre la performance de l'état de l'art dans plusieurs tâches du traitement automatique des langues naturelles

(TALN), quand elle est integrée dans une architecture de réseaux de neurones (Zou *et al.*, 2013; Camacho-Collados et Pilehvar, 2018).

Cette nouvelle branche de prédiction est devenue populaire à travers le modèle Word2vec (Mikolov *et al.*, 2013a). Word2vec est basé sur une architecture simple mais efficace qui fournit des propriétés sémantiques intéressantes (Mikolov *et al.*, 2013b). Deux modèles Word2Vec différents mais reliés ont été proposés : *Continuous Bag-Of-Words* (CBOW) et *Sikp-gram*. L'architecture CBOW est basée sur un modèle de langue basé sur les réseaux de neurones (Bengio *et al.*, 2003) et a pour but de prédire le mot courant en utilisant le contexte des mots qui l'entourent et en minimisant la fonction perte suivante :

$$E = -log(p(\overrightarrow{w_t}|\overrightarrow{W_t})) \tag{1.1}$$

où $w_t$ est le mot cible et $W_t = w_{t-n}, ..., w_t, ..., w_{t+n}$ représente la séquence de mots dans le contexte. Le modèle *Skip-gram* est similaire au modèle CBOW mais dans ce cas le but est de prédire les mots dans le contexte entourant étant donné le mot cible, au lieu de prédire le mot cible lui-même (Camacho-Collados et Pilehvar, 2018). La Figure 1.4 (Mikolov *et al.*, 2013a) montre une simplification de l'architecture générale des modèles CBOW et *Skip-gram* de *Word2vec*. L'architecture consiste en des couches d'entrées, des couches cachées et des couches de sortie. La couche d'entrée a la taille du vocabulaire et encode le contexte comme étant une combinaison de représentations de vecteurs *one-hot* (ou encodage 1 parmi $n$) des mots voisins d'un mot cible donné. La couche de sortie a la même taille que la couche d'entrée et contient un vecteur *one-hot* du mot cible durant la phase d'entraînement.

Une autre architecture importante de plongements de mots est *GloVe* (Pennington *et al.*, 2014). Cette dernière combine la factorisation matricielle globale et les

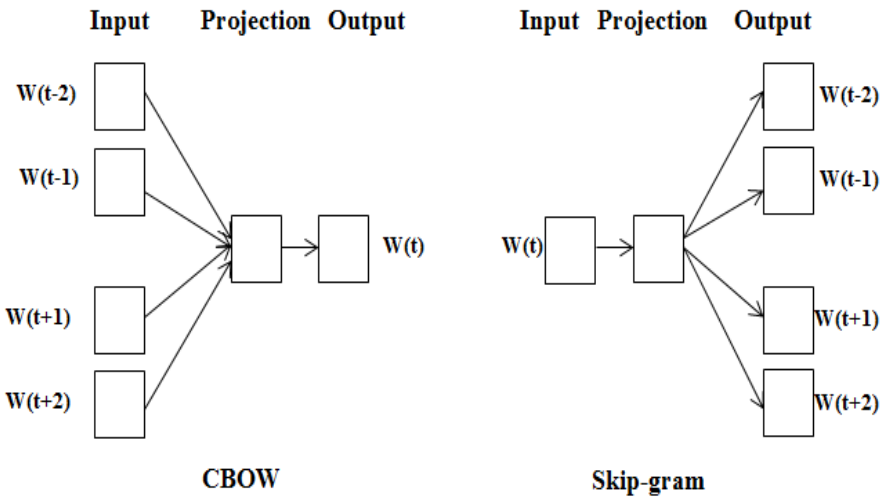


Figure 1.4 Architectures des modèles CBOW et Skipgram de Word2vec

méthodes des fenêtres du contexte local à travers un modèle de regression bilinéaire. Une matrice de co-occurence mot-à-mot est utilisée. *GloVe* évite un grand coût computationnel, en évitant de construire une matrice de co-occurence complète, mais l'entraînement sera plutôt fait directement sur les éléments non nuls (Kenter, 2017).

Ces dernières années ont vu des approches plus complexes qui tentent d'améliorer la qualité des plongements de mots, incluant les modèles exploitant les arbres de dépendance (ou *dependency parse-trees* ) (Levy et Goldberg, 2014) ou les patrons symétriques (Schwartz *et al.*, 2015), les modèles exploitant les n-grammes (Wieting *et al.*, 2016), les caractères (Lample *et al.*, 2016), ceux qui représentent les mots comme des distributions de probabilité (Athiwaratkun et Wilson, 2017), d'autres qui apprennent les plongements de mots dans des espaces vectoriels multilangues (Artetxe *et al.*, 2018), ou ceux qui exploitent les ressources de connaissance (Camacho-Collados et Pilehvar, 2018).

## 1.4 Apprentissage machine pour la REN

Dans cette section, nous présentons un modèle traditionnel d'apprentissage machine, très utilisé pour la REN qui est les Champs Aléatoires Conditionnels (en anglais : *Conditional Random Fields*) ou CRF.

### 1.4.1 Champs Aléatoires Conditionnels

Les CRF sont des modèles graphiques probabilistes utilisés fréquemment pour les tâches d'étiquetage des séquences telles que le POS tagging, la reconnaissance des objets, la reconnaissance des entités nommées, etc. Différents des classifieurs discrets, les CRF ont la propriété particulière de prendre en considération les exemples voisins. Ils prennent en compte les caractéristiques contextuelles lors de la prédiction d'une séquence de mots à l'entrée. Les CRF calculent la probabilité conditionnelle d'une séquence de classe $c = (c_1, c_2, ..., c_N)$ étant donné une séquence d'observation $o = (o_1, o_2, ..., o_N)$. Cette probabilité est calculée comme suit :

$$P(c|o) = \frac{1}{Z_0} \exp(\sum_{n=1}^{N} \sum_{k=1}^{K} \lambda_k * f_k(n, o, c_{n-1}, c_n)) \tag{1.2}$$

où $f_k(n, o, c_{n-1}, c_n)$ est une fonction d'attributs qui considère les classes d'étiquettes courantes et précédentes pour calculer la probabilité. $\lambda_k$ est le poids d'apprentissage qui est calculé durant l'entraînement. $Z_0$ est le facteur de normalisation qui permet d'avoir une somme de probabilités conditionnelles égale à 1 (Goyal *et al.*, 2018).

## 1.5 Apprentissage profond pour la REN

Récemment, les modèles de REN basés sur l'apprentissage profond sont devenus dominants et ont atteint les meilleurs résultats de l'état de l'art (Devlin *et al.*, 2018). Ces modèles ont l'avantage de pouvoir découvrir automatiquement les caractéristiques cachées liées aux textes. Dans cette section nous présentons une brève description de quelques architectures de réseaux de neurones utilisés dans le domaine de l'apprentissage profond et que nous utilisons dans notre travail. Ces architectures sont les réseaux de neurones récurrents à mémoire court et long terme (en anglais : *Long Short-Term Memory* ) ou LSTM, les réseaux de neurones récurrents bidirectionnels à mémoire court et long terme (en anglais : *Bidirectional Long Short-Term Memory* ) ou BiLSTM et les réseaux de neurones à convolution (en anglais : *Convolutional Neural Networks*) ou CNN.

### 1.5.1 Réseau récurrent à mémoire court et long terme

Le LSTM (Hochreiter et Schmidhuber, 1997) est une architecture de réseaux de neurones récurrents artificiels (*Recurrent Neural Networks*) ou RNN utilisé dans le domaine de l'apprentissage profond. Cette puissante famille de modèles connexionnistes peut détecter la dynamique du temps à travers des cycles dans un graphe (Ma et Hovy, 2016b). Les RNNs prennent comme entrée une sequence de vecteurs $(x_1, x_2, ..., x_n)$ à l'instant $t$ et retourne une autre séquence de vecteurs $(h_1, h_2, ..., h_n)$, l'état caché, qui stocke toute information utile à (et avant) l'instant $t$. Bien que les RNNs peuvent, en théorie, apprendre les dépendances à long terme, en pratique ils n'arrivent pas à le faire et ont tendance à devenir biaisés vers les entrées les plus récentes de la séquence (Bengio *et al.*, 1994). Les LSTM ont été conçus pour combattre ce problème en incorporant une cellule mémoire, et il a été démontré qu'ils sont capables de detecter les dépendances à long terme. Cela

est fait grâce à la présence de portails multiplicatifs qui contrôlent la quantité d'informations venant de l'état précédent à oublier et l'information venant des entrées à laisser passer à la cellule mémoire suivante (Hochreiter et Schmidhuber, 1997; Ma et Hovy, 2016a). La figure 1.5 (Ma et Hovy, 2016a) montre le schéma basique d'une unité LSTM.

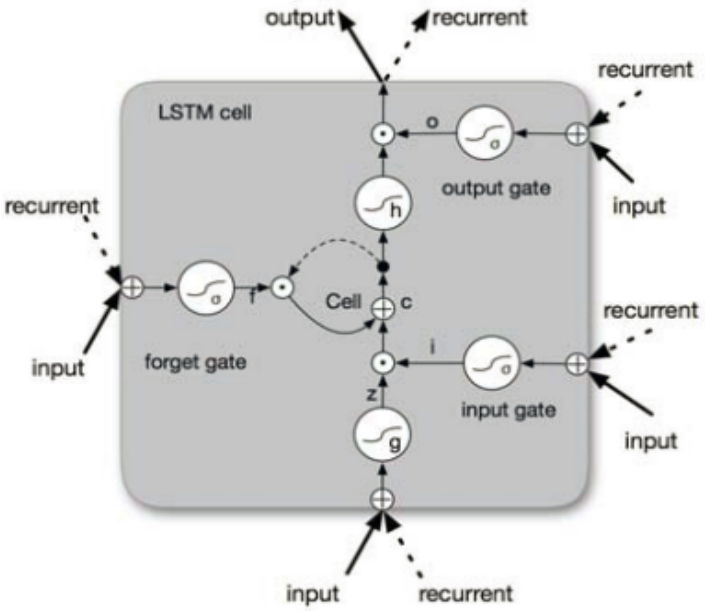


Figure 1.5 Schéma d'une unité LSTM

Les formules pour mettre à jour une unité LSTM à un temps t sont :

$$i_t = \sigma(W_i h_{t-1} + U_i x_t + b_i) \tag{1.3}$$

$$f_t = \sigma(W_f h_{t-1} + U_f x_t + b_f) \tag{1.4}$$

$$\tilde{c_t} = tanh(W_c h_{t-1} + U_c x_t + b_c) \tag{1.5}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c_t} \tag{1.6}$$

$$o_t = \sigma(W_o h_{t-1} + U_o x_t + b_o) \tag{1.7}$$

$$h_t = o_t \odot tanh(c_t) \tag{1.8}$$

où :

- $\sigma$ est la fonction sigmoid
- $\odot$ est la fonction produit
- $x_t$ est le vecteur d'entrée à un instant $t$
- $h_t$ est le vecteur de l'état caché qui stocke toute l'information utile à (et avant) l'instant $t$
- $U_i$ , $U_f$ , $U_c$ , $U_o$ dénotent les poids des matrices des différents portails pour l'entrée $x_t$
- $W_i$,$W_f$,$W_c$,$W_o$ sont les matrices des poids pour l'état caché $h_t$
- $b_i$, $b_f$, $b_c$, $b_o$ dénotent les vecteurs des biais

(Ma et Hovy, 2016a).

### 1.5.2 Réseau récurrent bidirectionnel à mémoire court et long terme

Pour chaque séquence $(x_1, x_2, ..., x_n)$ contenant n mots, chacune représentée comme étant un vecteur de dimension d, un LSTM calcule une represéntation $\overrightarrow{h_t}$ du contexte gauche de la phrase pour chaque mot $t$ (Lample *et al.*, 2016). Étant donné que l'information que nous pouvons obtenir en lisant la même séquence dans le sens inverse peut être bénéfique pour mieux comprendre la phrase, une solution efficace a été prouvée par Dyer *et al.* (2015), est *Bidirectional LSTM* ou BiLSTM. L'idée est de calculer, en utilisant un deuxième LSTM, une représentation $\overleftarrow{h_t}$ du contexte droit du mot, ce qui nous permettrait de détecter l'information passée et l'information future. La représentation d'un mot en utilisant ce modèle est obtenue en concaténant les deux états cachés, pour former la sortie finale ht=[$\overrightarrow{h_t}$ ;$\overleftarrow{h_t}$]. Ces représentations incluent efficacement une représentation d'un mot dans son contexte, qui est utile dans de nombreuses applications d'étiquetage (Lample *et al.*,

2016).

### 1.5.3 Réseau de neurones convolutifs

Les CNN ont été largement utilisés également dans différentes tâches liées au TALN. La figure 1.6 montre une description basique d'un CNN appliqué sur du texte (Yin *et al.*, 2017).

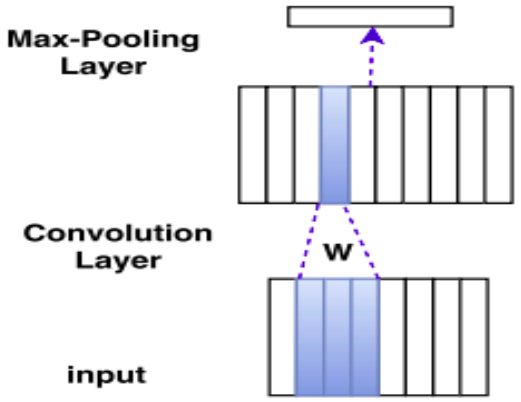


Figure 1.6 Schéma basique d'un CNN

#### 1.5.3.1 La couche d'entrée

Soit une séquence $x$ contenant $n$ entrées. Chaque entrée est représentée par un vecteur dense de dimension $d$, donc l'entrée $x$ est représentée par une matrice d'attributs de dimensions $d$ x $n$.

#### 1.5.3.2 La couche de convolution

La couche de convolution est utilisée pour représenter l'apprentissage à partir des glissements des *w-grams*. Pour une séquence d'entrée avec $n$ entrées : $x_1, x_2, ..., x_n$,

soit le vecteur $c_i \epsilon \mathbb{R}^{wd}$ le résultat de la concaténation des plongements des $w$ entrées $x_{i-w+1}, ..., x_i$ où $w$ est la largeur du filtre et $0 < i < s+w$. Les plongements des $x_i$ avec $i < 1$ ou $i > n$, sont remplacés par des vecteurs de 0. Nous générons par la suite la représentation $p_i \epsilon \mathbb{R}^d$ pour le *w-grams* $x_{i-w+1}, ..., x_i$ en utilisant les poids de convolution $W \epsilon \mathbb{R}^{d*wd}$ :

$$p_i = tanh(W.c_i + b) \tag{1.9}$$

où le biais $b \epsilon \mathbb{R}^d$.

#### 1.5.3.3 La couche de *Maxpooling*

Toutes les représentations $p_i$ des différents *w-grams* ($i = 1...s+w-1$) sont utilisées pour générer la représentation de la séquence d'entrée $x$ en choisissant le maximum comme suit : $x_j = max(p_{1,j}, p_{2,j}, ...)$ avec $(j = 1, ..., d)$.

## 1.6 Conclusion

Dans ce chapitre, nous avons introduit les concepts de base pour la compréhension de la tâche d'extraction des entités nommées qui est la tâche sur laquelle porte ce mémoire.

Nous présentons dans le prochain chapitre un état de l'art sur les travaux de l'extraction des entités nommées.

# CHAPITRE II

# ÉTAT DE L'ART

Tel que mentionné précédemment, la REN consiste en un processus d'identification d'un mot ou d'un groupe de mots qui se réfèrent à une entité particulière dans un texte. La notion des entités nommées (ENs) inclut non seulement les noms propres mais aussi des entités plus complexes telles que les expressions multi-mots. Les entités nommées sont en général catégorisées selon des taxonomies dépendemment du domaine d'application. Elles couvrent en général les noms des personnes, les endroits ou les organisations, mais peuvent parfois référer à des notions plus techniques telles que les gènes et les expressions temporelles et monétaires (Tjong Kim Sang et De Meulder, 2003; Doddington *et al.*, 2004). Le domaine de la REN a un impact important sur de nombreuses applications liées au TALN telles que la traduction automatique, l'extraction d'informations inter-langues, la recherche d'informations, etc.

Les approches principales dans la REN sont basées sur (i) la linguistique computationnelle (TALN) et (ii) l'apprentissage automatique. La première approche (i) utilise des modèles basés sur des règles. Des experts en linguistique définissent manuellement un ensemble de règles pour identifier les entités nommées dans les textes. L'inconvénient majeur de cette approche consiste au fait que le système REN a besoin de plusieurs expériences et une bonne connaissance des différentes

langues ou des différents domaines. En plus, ces systèmes ne peuvent pas être appliqués dans d'autres langues à cause des caractéristiques spécifiques de chaque langue. D'autre part, l'approche basée sur l'apprentissage automatique (ii) utilise un large corpus annoté servant à un corpus d'apprentissage pour bâtir un système basé sur la connaissance linguistique. Plusieurs techniques d'apprentissage automatique sont appliquées, telles que le modèle de Markov caché (*Hidden Markov Model*) ou (HMM), l'entropie maximale (*Maximum Entropy*), les arbres de décision (*decision trees*), les machines à vecteurs de support (*Support Vector Machines*) ou SVM et les champs aléatoires conditionnels (*Conditional Random Fields*) ou CRF (Nadeau et Sekine, 2007). En général les deux approches (i) et (ii) utilisent des informations lexicales à partir de ressources externes telles que les bases de connaissances et les dictionnaires, pour bâtir les systèmes REN, ce qui améliore la performance du système.

## 2.1 Approche REN à base de règles

Dans le domaine biomédical, Hanisch *et al.* (2005) proposent un système appelé *ProMiner* qui utilise un dictionnaire de synonymes pour identifier les mentions des protéines et des gènes dans le texte et leur associer leurs identifiants correspondants à partir du dictionnaire. Quimbaya *et al.* (2016) proposent une approche basée sur un dictionnaire également pour la REN dans le domaine médical. Leurs résultats montrent que leur approche a amélioré le rappel comparativement aux systèmes existants, sans ayant un grand impact sur la précision. D'autres systèmes REN connus à base de règles sont LaSIE-II (Humphreys *et al.*, 1998), NetOwl (Krupka et Hausman, 1998) et FASTUS (Appelt *et al.*, 1995). Ces systèmes sont principalement basés sur des règles sémantiques et syntaxiques pour reconnaître les entités (Li *et al.*, 2018).

## 2.2 Approche REN par apprentissage machine

### 2.2.1 Apprentissage supervisé

Lafferty *et al.* (2001) ont entraîné un CRF et ont inclus des attributs morphologiques, *Part-Of-Speech (POS) Tags*, et les séquences de mots. Joshi *et al.* (2015) utilisent un CRF également, dans leur modèle et montrent qu'utiliser *Word2Vec* comme une représentation distribuée de mots améliore la performance pour la reconnaissance des entités nommées pour le domaine du e-commerce et leur modèle performe bien avec une petite quantité de données du domaine.

En français ou en anglais, la reconnaissance des entités nommées est plus facile que dans les autres langues grâce à la *capitalization* des entités dans le corpus. Dans ce contexte, nous citons le système CasEN (Friburger, 2002), qui a donné 73,9% de F1 dans la campagne d'évaluation ESTER2, le système de Poibeau *et al.* (2003) a donné 90% de F1 dans la campagne d'évaluation MUC-6, le système mXS (Nouvel *et al.*, 2013), utilisant les techniques de fouille de textes basées sur la détection des motifs hiérarchiques pour l'évolution semi-automatique d'une base de connaissance, a donné 79% de F1 dans la campagne d'évaluation ETAPE qui dépend de la campagne ESTER2.

Récemment, les modèles basés sur les réseaux de neurones ont fait preuve d'efficacité dans les tâches de la REN. Ratinov et Roth (2009) comparent différentes approches pour la reconnaissance des entités nommées et ont bâti un modèle supervisé en utilisant *Regularized Average Perceptron* (Freund et Schapire, 1999). Les réseaux de neurones basés sur des LSTM ont été utilisés largement dans différentes applications de reconnaissance des entités nommées grâce à leur aptitude à détecter des dépendances à long terme grâce à leurs cellules mémoires. Ces modèles ont montré de bons résultats comparés aux approches traditionnelles, tels

que SVM et CRF, même s'ils n'ont pas besoin de dictionnaires, Gazetteers ou d'autres informations additionnelles. Chiu et Nichols (2015b) présentent un modèle hybride en combinant des LSTMs bi-directionnels et des CNN. Leur modèle performe mieux que les modèles existants dans quelques applications en ayant 91.62% de F1 dans le *dataset* CoNLL-2003. Gillick *et al.* (2016) décrivent un modèle basé sur des couches LSTM, capable d'analyser des textes dans différentes langues. Leur modèle performe mieux que les modèles existants dans les tâches de POSTagging et REN en n'utilisant que les données d'entraînement fournies. Lample *et al.* (2016) introduisent un modèle neuronal similaire à (Chiu et Nichols, 2015b), basé sur des LSTMs bidirectionnels et combiné avec un CRF, ayant 90.94% de F1 dans le même dataset. Pour détecter l'information orthographique et morphologique, ils utilisent deux modèles de représentations des caractéristiques, basés sur les mots et sur les caractères. Yao et Huang (2016) proposent aussi un modèle basé sur des LSTM bi-directionnels.

Plus récemment l'apprentissage par transfert est apparu. Dong *et al.* (2017) introduisent un réseau de neurones récurrents bi-directionnel (RNN) qui extrait automatiquement la connaissance dans le domaine médical en traitant les bilans des malades. Dans leur modèle ils commencent par entraîner un BiRNN dans le domaine général, et par la suite ils l'utilisent pour le transfert de connaissances du domaine général, pour entraîner un BiRNN plus profond, pour reconnaître les concepts médicaux. Lee *et al.* (2017) incorporent l'apprentissage par transfert dans l'apprentissage profond pour les graphes de données. En transférant l'information géométrique intrinsèque apprise dans le domaine source, leur approche peut construire un modèle pour une nouvelle tâche dans le domaine cible, mais reliée à la tâche source, sans avoir besoin de collecter de nouvelles données et sans entraîner un nouveau modèle de zero. Les auteurs ont testé leur approche avec des données textuelles à grande échelle, et ont confirmé l'efficacité du transfert

d'apprentissage pour l'apprentissage profond sur les graphes. D'après leurs expériences, le transfert d'apprentissage est plus efficace quand le domaine source et le domaine cible ont des similarités structurelles dans les représentations des graphes. Aguilar *et al.* (2017) proposent une nouvelle approche multi-tâches en employant une tâche secondaire plus générale qui est la segmentation des entités nommées avec une tâche primaire qui est la catégorisation des entités nommées. Zhao *et al.* (2018) proposent un nouvel algorithme d'apprentissage par multi-tâches en joignant les deux tâches de normalisation et de reconnaissance des entités nommées. Leur approche performe mieux que les approches existantes sur deux *datasets* du domaine medical. Liu *et al.* (2018) proposent un modèle d'apprentissage multi-tâches basé sur les graphes où les différentes tâches peuvent communiquer entre elles et la relation entre ces dernières est apprise dynamiquement. Les auteurs évaluent leur approche sur les deux tâches de classification des textes et d'étiquetage des séquences et montrent que leur modèle performe mieux que les références qu'ils ont considérées. Khan *et al.* (2020) appliquent également l'apprentissage multi-tâches dans le domaine biomédical, en utilisant une architecture neuronale basée sur les transformeurs. Dans leurs résultats, les auteurs montrent que leur modèle performe mieux que les modèles déjà existants en termes de temps d'entraînement, de mémoire et de prédictions.

Zhu et Wang (2019) proposent un réseau de neurones convolutionnels basé sur le mécanisme d'attention (*Convolutional Attention Network*) ou CAN, pour la REN pour la langue chinoise. Leur modèle est constitué de neurones de convolution appliqués sur les caractères avec une couche d'attention locale, et un réseau de neurones récurrents à portes (*Gated Recurrent Unit*) ou GRU avec une couche d'attention globale pour détecter respectivement l'information à partir des caractères adjacents et le contexte global de la phrase. Luo *et al.* (2019) utilisent le mécanisme d'attention également pour développer un modèle exploitant des repré-

sentations contextualisées hiérarchiques niveau phrase et niveau document pour modéliser une information plus générale sur les mots dans différents contextes. Straková *et al.* (2019) utilisent ELMo, BERT et Flair, dont nous parlons plus en détails dans la section suivante, pour enrichir leur architecture neuronale basée sur des couches BiLSTM et CRF, pour améliorer les résultats de l'état de l'art dans la tâche de reconnaissance des entités nommées imbriquées. Lin *et al.* (2019) entraînent conjointement deux modèles neuronaux. Un premier modèle basé sur le mécanisme d'attention et un modèle auxiliaire entraîné sur des Gazetteers. Les auteurs montrent que l'incorporation de ce dernier dans le premier modèle améliore les performances de leur modèle de base avec l'utilisation de moins de donnés annotées. Jilek *et al.* (2018) proposent une méthode de REN à temps réel basée sur une ontologie pour extraire plus d'informations sur les entités à traiter. Leur système est comparable en termes de rapidité à des systèmes déjà existants tout en donnant de meilleurs scores et sans ayant recours à des architectures sophistiquées qui demandent un entraînement plus coûteux. Sousa et Couto (2020) proposent BiOnt, un système basé sur des ontologies du domaine biomédical. Dans leur travail, les auteurs montrent que l'utilisation des méthodes d'apprentissage profond conjointement avec des ressources de connaissances externes telles que les ontologies améliore l'extraction des relations entre les entités, ce qui donne de l'information supplémentaire sur les associations possibles entre ces dernières. D'après leurs résultats, les auteurs montrent que leur système testé sur trois *datasets* différents, donne de meilleurs scores comparativement aux modèles qui n'utilisent pas de ressources de connaissances externes. Hu *et al.* (2020) proposent un modèle basé sur l'apprentissage multi-tâches et le mécanisme d'attention pour la REN. Les auteurs divisent les entités en des entités simples et des entités composées. Dans leur travail, ils entraînent un modèle séparé pour détecter ces dernières. Les auteurs montrent que l'information liée au contexte est plus utile en considérant un document au complet au lieu de traiter juste la phrase dans laquelle l'entité apparaît.

D'après les résultats, le modèle résultant dépasse en termes de performance les méthodes de la REN niveau phrase et niveau document déjà existantes.

Pour résoudre le problème des systèmes qui ne se basent pas sur le *Commonsense*, Balahur *et al.* (2011) construisent *EmotiNet*, une base de connaissances basée sur le *Commonsense*, pour la détection des émotions implicites, en tenant compte du contexte d'une phrase. Dans leur recherche, ils démontrent qu'utiliser cette ressource améliore les performances des détecteurs d'émotions dans la tâche de détection et de classification des sentiments, en performant mieux dans des contextes où aucune mention explicite d'émotions n'est présente. Amplayo *et al.* (2018) proposent d'améliorer la performance d'un système de résumé automatique en présentant *Entity2Topic* (E2T). E2T est un module qui encode les entités extraites à partir d'un texte par un système d'extraction d'entités. Ces entités fournissent des informations basées sur le *Commonsense* une fois liées à une base de connaissances telle que Wikipedia. Dans leur travail, ils montrent que l'application de E2T à un simple modèle séquentiel avec un mécanisme d'attention, améliore considérablement la performance de leur système de résumé automatique. Wang *et al.* (2018) présentent *Three-way Attentive Networks* (TriAN) pour modéliser les interactions entre le passage, les questions et les réponses, dans la tâche de compréhension automatique de textes. Dans leur travail ils utilisent les plongements construits à partir des arrêtes de ConceptNet (Speer *et al.*, 2017) comme des traits additionnels pour modéliser le *Commonsense*. Zhong *et al.* (2019) pré-entraînent un modèle générique sur une ressource externe basée sur le *Commonsense*. Leurs résultats améliorent l'état de l'art dans deux tâches de question/réponse qui demandent le raisonnement basé sur le *Commonsense*. Bosselut *et al.* (2019) présentent *COMmonsEnse Transformers* (COMET), un modèle qui génère de nouveaux triplets pour enrichir des graphes de connaissances basés sur le *Commonsense*. Leur modèle a été entraîné sur les deux bases de connaissances

ConceptNet et ATOMIC (Sap *et al.*, 2018). Leurs résultats montrent que leur modèle est capable de générer des connaissances que les experts considèrent de haute qualité et que la performance de leur modèle est proche de la performance humaine dans cette tâche. Ma *et al.* (2019) utilisent le mécanisme d'attention en pré-entraînant BERT sur des bases de connaissances basées sur le *Commonsense* dans la tâche de question/réponse, ce qui améliore leurs résultats comparativement à leur modèle de base.

### 2.2.2 Apprentissage non supervisé

Collobert *et al.* (2011) proposent une architecture de réseaux de neurones unifiés qui peut être appliquée dans plusieurs applications de traitement des langues naturelles telles que le POSTagging et la REN. Cette flexibilité est réalisée par le fait que leur système apprend des représentations internes basées sur des *datasets* d'entraînement non étiquetés au lieu d'utiliser des *datasets* pour une application bien spécifique. Dans leur système ils combinent des couches de CNN avec une couche CRF au dessus. Zhang et Elhadad (2013) proposent une approche non supervisée pour extraire les entités nommées à partir du texte dans le domaine biomédical. Leur classifieur basé sur la similarité distributionnelle a montré une performance compétitive dans la classification des entités. Sachan *et al.* (2018) entraînent un modèle de langue bi-directionnel (BiLM) sur des données non annotées et transfèrent ses poids pour pré-entraîner un modèle REN avec la même architecture ce qui aboutit à une meilleure initialisation des paramètres du modèle REN. Les auteurs ont évalué leur approche sur quatre ensembles de données REN dans le domaine médical et montrent une amélioration de F1-score comparé aux approches existantes dans l'état de l'art. Ils ont montré également que le transfert des poids du BiLM mène à un entraînement plus rapide du modèle pré-entraîné, et ce dernier a besoin de moins d'exemples d'entraînement pour atteindre

un certain F1-score. Peters *et al.* (2018) introduisent ELMo, un nouveau type de représentation contextualisée des mots qui modélisent non seulement des caractéristiques complexes des différentes utilisations des mots telles que la syntaxe et la sémantique, mais en plus comment ces utilisations varient dépendemment du contexte. Leur modèle prend en considération l'état interne d'un modèle de langue bidirectionnel profond pré-entraîné sur un large corpus textuel. Dans leur expérimentations, ils montrent que ces représentations peuvent être rajoutées facilement à des modèles exitants pour améliorer significativement les résultats de l'état de l'art sur différentes tâches du TALN y compris la REN. Devlin *et al.* (2018) introduisent *BERT*, acronyme de *Bidirectional Encoder Representations from Transformers*, qui est un modèle de langue désigné pour pré-entraîner des représentations bidirectionnelles profondes à partir de textes non étiquetés. Ce pré-entraînement et trés coûteux mais se fait une seule fois. Par la suite, la partie pré-entraînée de BERT peut être ajustée avec une couche de sortie additionnelle pour ajuster le modèle pour une tâche spécifique, sans apporter de modifications de l'architecture du modèle. BERT a battu le record sur onze tâches du TALN incluant la REN. Akbik *et al.* (2018) proposent un nouveau type de plongements de mots contextualisés, *Falir embeddings*, en tirant partie des états internes d'un modèle de langue entraîné sur les caractères. Ces plongements ont la particularité d'être entraînés en modélisant le mot comme étant une séquence de caractères et sont contextualisés dépendamment des mots qui entourent le mot en question. Bari *et al.* (2020) proposent un modèle REN non supervisé qui transfère la connaissance d'une langue à une autre sans ayant recours à des dictionnaires bilingues ou des données parallèles. Leurs expérimentations sur cinq langues différentes démontrent l'efficacité de leur approche en performant mieux que les modèles déjà existants et en ayant un meilleur score pour chaque paire de langues.

L'apprentissage non supervisé a été utilisé également pour découvrir des relations

basées sur le *Commonsense*. Trinh et Le (2018) présentent une méthode simple d'apprentissage non supervisé. Leurs résultats montrent une amélioration dans l'extraction des traits à partir d'une question liée à une bonne réponse, indiquant une bonne compréhension du contexte. Zhang *et al.* (2018) présentent *ReCoRD*, un *dataset* pour la compréhension automatique de textes en se basant sur le *Commonsense*.

Les graphes de connaissances ont été adoptés également dans l'apprentissage non supervisé et sont devenus omniprésents dans de nombreux domaines tels que la bio-informatique, la chimie et les réseaux sociaux (Bai *et al.*, 2019). Auer *et al.* (2007) proposent DBPedia qui inclut des connaissances extraites à partir de Wikipedia, ce qui fournit un nombre important de faits, liés surtout aux entités nommées contenues dans les articles de Wikipedia. (Google, 2012) présentent le graphe de connaissance de Google, qui est considéré comme étant le graphe le plus large et le plus général, se concentrant sur les entités nommées qui peuvent être désambiguïsées en les manipulant comme étant des objets et non pas des chaînes de caractères, mais son contenu n'est pas libre d'accès. La particularité de ConceptNet comparé aux ressources citées précédemment, c'est que ConcepNet est suffisamment large, libre d'accès et basée sur le sens des mots tels qu'ils sont utilisés dans le langage naturel. Cette concentration sur les mots la rend compatible avec l'idée de représenter le sens des mots sous forme de vecteurs (Speer *et al.*, 2017).

## 2.3 Conclusion

Dans ce chapitre, nous avons présenté un état de l'art sur la tâche étudiée dans notre mémoire qui est l'extraction des entités nommées. Cet état de l'art présente des systèmes basés sur des approches REN à base de règles et par apprentissage

machine. Au mieux de notre investigation de la littérature, nous n'avons pas trouvé de méthodes qui traitent la tâche de la REN dans le domaine de l'électronique. Dans notre travail, nous proposons une approche hybride à base de réseaux de neurones et de statistiques pour la REN en domaine général et restreint qui est le domaine de l'électronique dans notre cas. Dans le chapitre suivant, nous détaillons le fondement de notre approche.

# CHAPITRE III

# MÉTHODOLOGIE

## 3.1 Introduction

La reconnaissance des entités nommées est une tâche où des approches appliquant des modèles statistiques telles que CRF, SVM ou Perceptron ont donné de bonnes performances en utilisant les caractéristiques extraites à la main (Chiu et Nichols, 2015a). Plus tard, plusieurs travaux de recherche (Chiu et Nichols, 2015b; Huang *et al.*, 2015; Lample *et al.*, 2016) ont proposé des modèles plus efficaces basés sur les réseaux de neurones qui nécessitent moins de caractéristiques extraites à la main et apprennent à la place des traits importants à partir des plongements de mots. À part l'utilisation des caractéristiques liées aux mots, ces travaux de recherche ont démontré un intérêt pour les caractéristiques explicites niveau caractères, qui peuvent être utiles surtout avec les mots rares, sur lesquels les plongements de mots n'ont pas été entraînés. Plus récemment l'apprentissage par transfert est apparu pour permettre le transfert de connaissance entre des domaines différents. Dans ce chapitre, nous commençons par mettre l'accent sur l'importance de la sélection des caractéristiques et nous enchaînons par présenter nos approches et les différentes caractéristiques choisies pour l'entraînement de nos modèles.

## 3.2 Importance de la sélection des caractéristiques

La sélection appropriée des caractéristiques est une tâche cruciale dans les systèmes REN basés sur l'apprentissage supervisé. Les caractéristiques sont les propriétés et les attributs des objets textuels dans un modèle computationnel. Les caractéristiques jouent un rôle important pour représenter un aspect multidimentionnel des différentes formes du texte qui sont utilisées par la suite par la méthode d'apprentissage choisie pour générer un modèle. Ce modèle est capable de reconnaître des données similaires et classifier des exemples reçus à l'entrée. Nadeau et Sekine (2007) ont divisé l'espace des caractéristiques en trois groupes qui sont les caractéristiques basées sur des ressources externes, les caractéristiques liées aux documents et au corpus et les caractéristiques basées sur les mots. Les premières caractéristiques sont basées sur les ressources linguistiques telles que le lexique, les dictionnaires, les Gazetteers, etc. Ces caractéristiques determinent si un mot est membre ou non d'une de ces ressources. Les caractéristiques liées aux documents et au corpus sont conçues en se basant sur la structure et le contenu des documents. En dernier lieu, les caractéristiques basées sur les mots incluent les caractéristiques orthographiques, contextuelles et morphologiques (Goyal *et al.*, 2018).

## 3.3 Approche statistique

Tel que mentionné ci-haut, les CRF sont parmi les modèles statistiques qui ont donné de bonnes performances en utilisant des caractéristiques extraites à la main. Dans notre travail nous utilisons *CRFsuite*[1] qui est une implémentation des CRF décrit dans le chapitre II. Parmi toutes les implémentations des CRF, CRFsuite

1. `http://www.chokkan.org/software/crfsuite/`

permet un entraînement du modèle et un étiquetage des données le plus rapide possible.

### 3.3.1 Caractéristiques utilisées

Dans cette partie nous présentons les différentes caractéristiques considérées pour l'entraînement de notre modèle statistique. Ces caractéristiques sont divisiées en deux catégories : les caractéristiques basiques et les caractéristiques orthographiques additionnelles.

#### 3.3.1.1 Les caractéristiques basiques

Les différents *datasets* que nous utilisons se présentent dans un format de 4 colonnes qui sont les tokens, les POSTags (Part Of Speech Tags) et les Chunktag que nous décrivons dans les sections ci-après, et les étiquettes. Ces différentes colonnes représentent les caractéristiques basiques que nous considérons dans l'entraînement de notre modèle. L'exemple ci-après présente un exemple du format des données utilisé dans nos expérimentations.

##### 3.3.1.1.1 Le POSTagging

Le POSTagging est une technique qui associe à chaque mot d'un texte, l'information grammaticale correspondante. L'importance du POSTag pour le traitement de la langue est la grande quantité d'informations qu'il donne sur un mot et son voisinage (Hasan *et al.*, 2007), raison pour laquelle il peut être utilisé dans plusieurs tâches du TALN telles que la recherche d'information, l'extraction d'information, la REN, etc. La table 3.2 montre une liste partielle des POSTags utilisés dans le projet *The PennTreebank* (Marcus *et al.*, 1993), qui est le *Tagset* le plus

Tableau 3.1 Exemple de données dans le format CoNLL

| **Token** | **POSTag** | **chunkTag** | **Label** |
|---|---|---|---|
| EU | NNP | I-NP | I-ORG |
| rejects | VBZ | I-VP | O |
| German | JJ | I-NP | I-MISC |
| call | NN | I-NP | O |
| to | TO | I-VP | O |
| boycott | VB | I-VP | O |
| British | JJ | I-NP | I-MISC |
| lamb | NN | I-NP | O |
| . | . | O | O |

populaire, puisque la plupart des POSTaggers entraînés sur l'anglais ont été entraînés sur ce tagset (Bird *et al.*, 2009). La liste complète (Marcus *et al.*, 1993) de ces derniers se trouvent dans l'annexe de ce mémoire.

#### 3.3.1.1.2 Le *Chunking*

Le *Chunking* est le processus de segmenter et d'étiqueter des séquences de mots. La figure 3.1 montre un exemple de *Chunking* sous forme d'arbre. Les petites cases montrent la tokénisation au niveau des mots et le POSTagging, tandis que les grandes cases montrent la fragmentation de niveau supérieur. Chacune de ces grandes cases est appelée *Chunk*. Lors de la construction des *Chunks*, il est possible de définir les règles grammaticales qui indiquent comment ségmenter la phrase. Dans l'exemple présenté, une règle a été définie en utilisant l'expressions régulière suivante :

grammar = "NP : <DT>?<JJ>*<NN>"

Tableau 3.2 Liste partielle des POSTags utilisés dans le projet The Penn Treebank

| **Tag** | **Description** |
|---|---|
| NNP | Nom propore, singulier |
| VBZ | Verbe, troisième personne, singulier, présent |
| JJ | Adjectif |
| NN | Nom, singulier |
| VB | Verbe, forme basique |
| ... | ... |

Cette règle dit qu'un *Chunk* doit être formé à chaque fois que le *Chunker* trouve un déterminant optionnel (DT) suivi par un nombre quelconque d'adjectifs (JJ), suivi par un nom (NN).

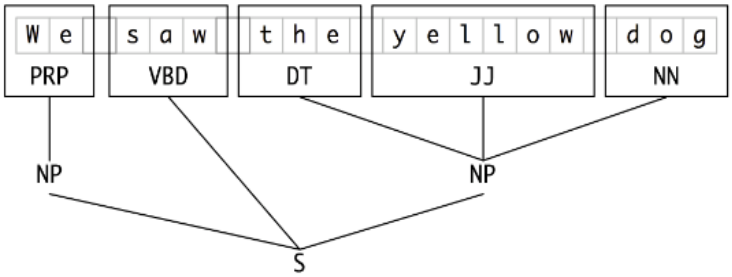


Figure 3.1 Représentation en arbre des structures des chunks

3.3.1.2 Les caractéristiques additionnelles

A part les caractéristiques basiques, nous avons utilisé des caractéristiques orthographiques liées aux mots qui ont amélioré la performance de notre modèle statistique. Parmi ces caractéristiques nous utilisons les *n-grams*, nous vérifions si le mot est en majuscule, si c'est un titre ou bien s'il s'agit d'un caractère spécial.

## 3.4 Approche neuronale

Tel que mentionné précédemment, une solution pour résoudre le problème de mémoire à long terme des réseaux de neurones, ce sont les réseaux de neurones récurrents (Goller et Kuchler, 1996). Récemment, les RNNs ont montré un succès remarquable dans diverses tâches du TALN telles que la reconnaissance de la parole, la traduction automatique, le résumé automatique, etc. Les LSTMs, un type particuler des RNN, grâce à leurs cellules mémoires permettent d'apprendre les dépendances à long terme (Chiu et Nichols, 2015a). Pour les tâches d'étiquetage séquentiel telles que la REN et la reconnaissance de la parole, un modèle basé sur un LSTM bi-directionnel prend en considération le contexte des deux côtés du mot et élimine le problème du contexte limité cité ci-haut (Chiu et Nichols, 2015a). Une méthode simple mais très efficace consiste à utiliser les sorties des couches BiLSTM comme des attributs pour faire des prédictions indépendantes pour chaque sortie $y_t$ (Ling *et al.*, 2015). Pour les tâches d'étiquetage des séquences, il est bénéfique de considérer les corrélations entre les labels dans le voisinage et décoder la meilleure chaîne de labels conjointement pour une phrase à l'entrée, puisque la "grammaire" qui caractérise des séquences de labels interprétables impose certaines contraintes, telles qu'un adjectif est le plus souvent suivi par un nom que par un verbe, qui peut être l'équivalent, dans la tâche de REN avec l'annotation BIO2 standard (Veenstra et Tjong Kim Sang, 1999) à I-ORG ne peut pas suivre I-PER (Ma et Hovy, 2016b). Huang *et al.* (2015) a démontré qu'ajouter une couche de CRF au-dessus du BiLSTM niveau mot améliore la performance du système. Par conséquent, tel que dans le travail présenté par Chiu et Nichols (2015a); Lample *et al.* (2016), nous procédons à l'étiquetage des séquences en utilisant un CRF, au lieu de les étiqueter indépendamment.

### 3.4.1 Caractéristiques utilisées

Parmi les caractéristiques que nous avons utilisées dans notre travail nous distinguons les caractéristiques de bases et les caractéristiques additionnelles.

#### 3.4.1.1 Caractéristiques de base

Dans notre travail, nous avons considéré les plongements de mots et les plongements de caractères comme caractéristiques de base.

##### 3.4.1.1.1 Les plongements de mots

Les couches d'entrées de notre modèle sont des représentations vectorielles de chaque mot. Apprendre des représentations indépendantes pour les types des mots à partir d'une quantité insuffisante de données d'entraînement est un problème difficile puisqu'il y a un nombre important de paramètres à estimer (Lample *et al.*, 2016). Dans notre étude, nous utilisons des plongements de mots pré-entrainés pour initialiser notre *look-up table* et pour enrichir notre *dataset* d'entraînement. Nous utilisons les plongements de mots *Word2Vec* entraînés sur 100 billions mots de *Google News* (Mikolov *et al.*, 2013a) et les plongemets de mots *GloVe 1.2* entraînés sur 840 billions mots du *Common Crawl* (Pennington *et al.*, 2014)). En plus, puisque nous avons supposé que les plongements de mots entraînés sur les données texte du domaine performent mieux, nous avons construit des plongements en utilisant du texte brut du domaine, mais nous n'avons pas reporté les résultats puisque les données n'étaient pas suffisantes par rapport à *Glove* et *Word2Vec*. Suivant Collobert *et al.* (2011), tous les mots sont mis en minuscules avant de les passer à travers la *lookup table* pour les convertir en leurs plongements de mots correspondants.

3.4.1.1.2 Extraction des caractéristiques liées aux caractères

Les systèmes combinant le contexte et les caractères d'un mot ont démontré de bonnes performances comme étant des systèmes REN qui ne demandent pas une grande connaissance du domaine ou une grande quantité de ressources spécifiques au domaine (Yadav et Bethard, 2018). Apprendre les plongements niveau caractère a l'avantage d'apprendre des représentations spécifiques à la tâche et au domaine étudié (Lample *et al.*, 2016). Il exite deux modèles basiques dans cette catégorie, l'un utilise des couches BiLSTM et l'autre des couches de convolution (Yadav et Bethard, 2018).

— **Extraction des caractéristiques liées aux caractères avec BiLSTM**
Ce type de modèle applique un BiLSTM sur chaque mot pour extraire un nouveau vecteur de caractéristiques à partir des vecteurs de caractéristiques liés à chaque caractère composant ce mot. Le plongement du mot déduit à partir des caractères est le résultat de la concaténation des deux vecteurs *foreward* et *backward* du BiLSTM. Ces derniers sont concaténés à leur tour à un plongement de la *look-up table* pour obtenir une représentation de ce mot, le tout est passé à un autre BiLSTM niveau phrase, tel que montré dans la Figure 3.3. $l_i$ représente le mot i et son contexte gauche, $r_i$ représente le mot i et son contexte droit. La concaténation de ces deux vecteurs mène à une représentation du mot i dans son contexte $c_i$ (Lample *et al.*, 2016). Finalement, les étiquettes seront prédites en utilisant une couche softmax ou CRF (Lample *et al.*, 2016). La figure 3.2 (Lample *et al.*, 2016) donne un exemple d'extraction des caractéristiques liées aux caractères du mot *Mars* en utilisant un BiLSTM. L'architecture complète du modèle est présentée par la figure 3.3 (Lample *et al.*, 2016).

— **Extraction des caractéristiques liées aux caractères avec CNN**

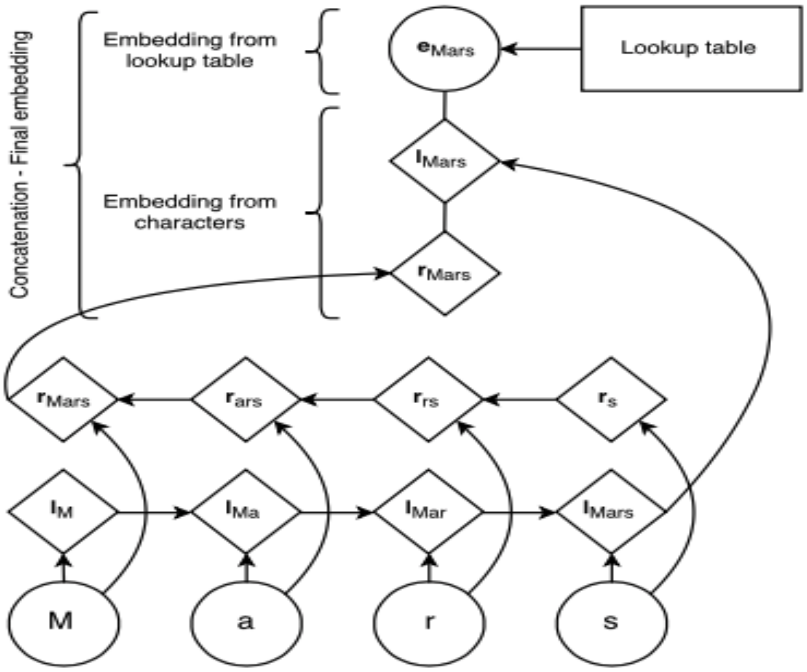


Figure 3.2 Extraction des caractéristiques liées aux caractères du mot "Mars" avec un BiLSTM

Pour chaque mot nous employons une convolution et une couche *maxpooling* pour extraire un nouveau vecteur de caractéristiques à partir des vecteurs de caractéristiques liés à chaque caractère. La figure 3.4 (Chiu et Nichols, 2015a) montre un exemple d'extraction des caractéristiques liées aux caractères du mot *Picasso* en utilisant un CNN. Cette couche de convolution est suivie par une couche Bi-LSTM. Pour chaque direction du BiLSTM (*forward* et *backword*), l'entrée est injectée dans des couches multiples de LSTMs connectées en séquence, c'est-à-dire les unités LSTM de la seconde couche prennent la sortie de la première couche et ainsi de suite, l'architecture complète du modèle est présentée par la figure 3.5 (Chiu et Nichols, 2015a). Dans la figure, une seule unité LSTM est présentée pour simplifier le schéma (Chiu et Nichols, 2015a). Ce modèle représente un mot comme

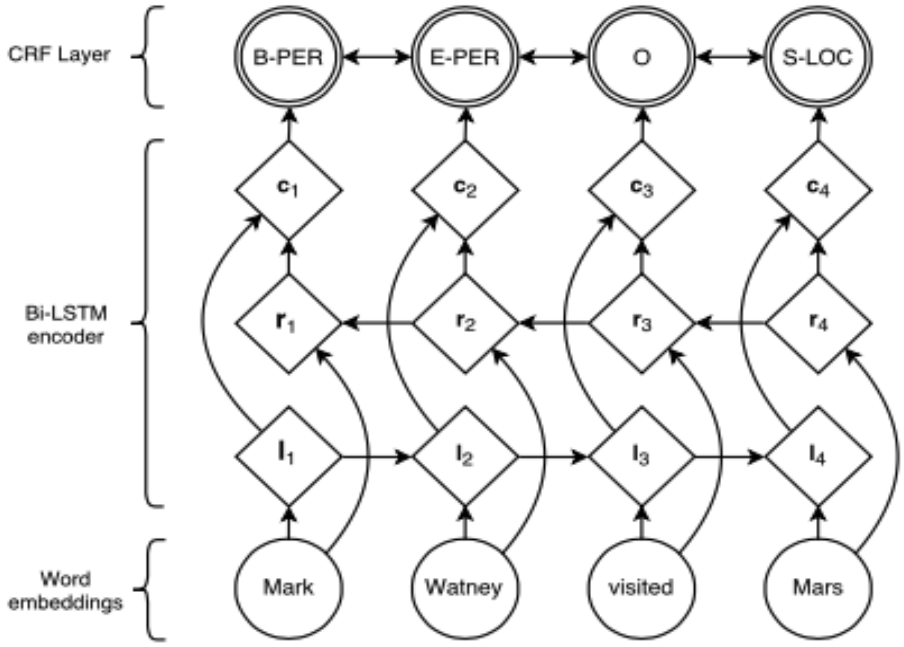


Figure 3.3 L'architecture principale du système de REN en utilisant une combinaison de BiLSTM et CRF

étant la combinaison du plongement du mot et la sortie de la convolution appliquée sur ses caractères. En dernier lieu une couche softmax ou CRF est placée au-dessus des couches BiLSTM pour générer les étiquettes.

Dans notre approche neuronale, suivant Chiu et Nichols (2015a), nous avons initialisé aléatoirement une *lookup table* avec des valeurs entre [-0.5,0.5] pour générer un plongement de caractères de dimension 25 . Le jeu de caractères inclut tous les caractères uniques du dataset CoNLL-2003, plus les *tokens* spéciaux *PADDING* et *UNKNOWN*. Le token *PADDING* est utilisé pour le CNN, et le token *UNKNOWN* est utilisé pour tous les autres caractères.

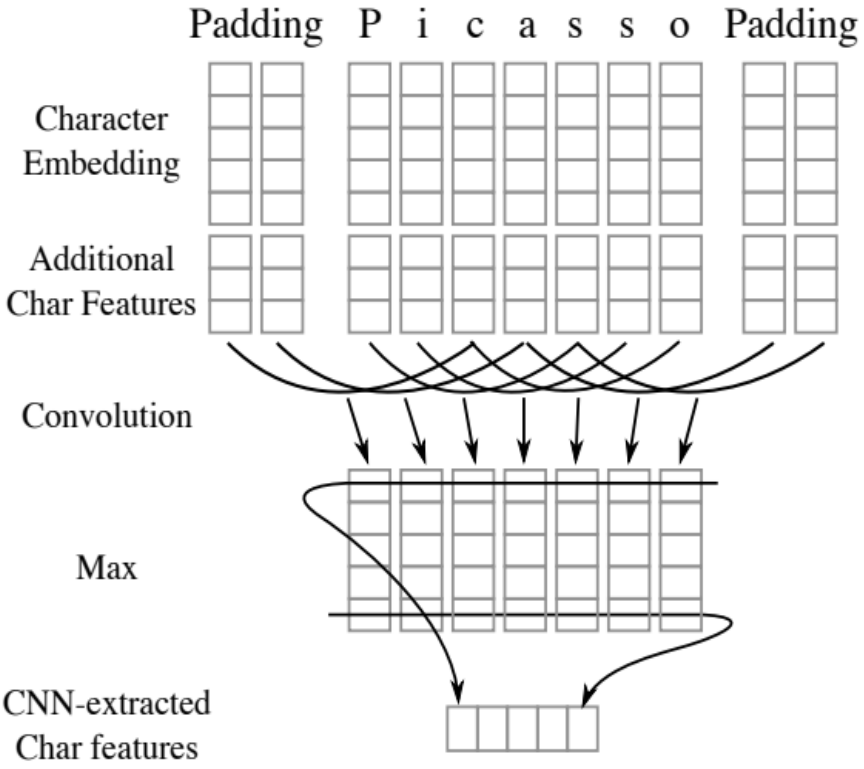


Figure 3.4 Extraction des caractéristiques liées aux caractères avec un CNN

#### 3.4.1.2 Caractéristiques additionnelles

##### 3.4.1.2.1 Caractéristiques additionnelles liées aux mots

Puisque l'information liée aux lettres majuscules a été supprimée lors de la conversion des mots en des plongements de mots, nous utilisons une *lookup table* séparée pour ajouter cette caractéristique avec les options suivantes : *allCaps* (le mot n'est composée que de lettres majuscules), *upperInitial* (la première lettre du mot est en majuscule), *lowercase* (toutes les lettres sont en minuscule) et *mixedCaps* (mot mélangé de lettres majuscules et minuscules )(Collobert *et al.*, 2011; Chiu et Nichols, 2015a).

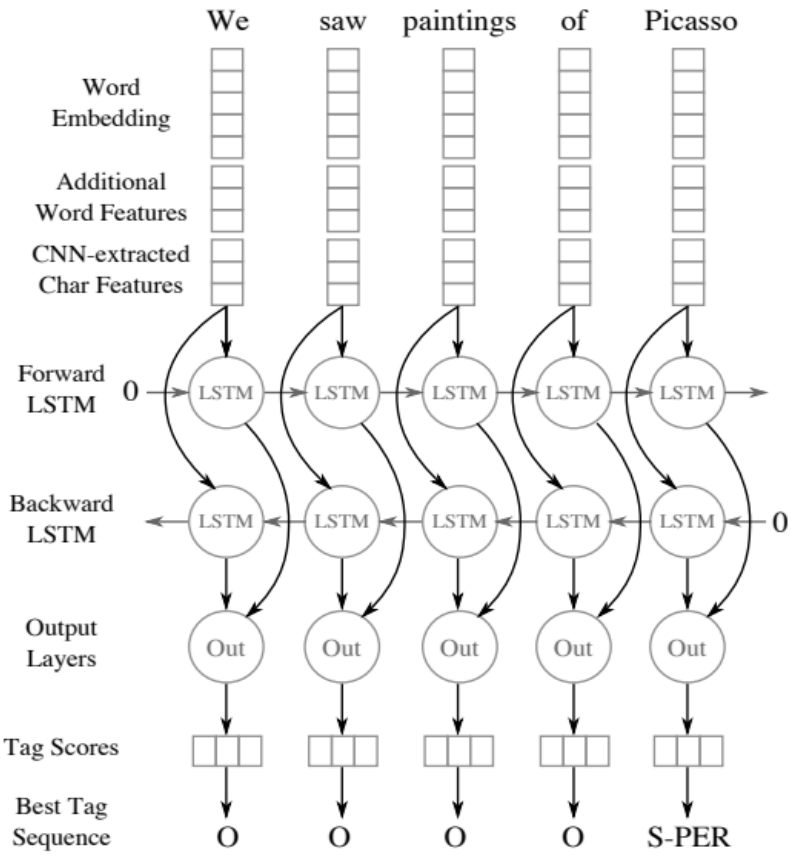


Figure 3.5 L'architecture principale de notre système REN en utilisant BiLSTM et CRF

3.4.1.2.2 Caractéristiques additionnelles liées aux caractères

Pour les caractéristiques additionnelles liées aux caractères, nous avons utilisé une *lookup table* qui génère un vecteur de dimension 4 représentant le type du caractère (majuscule, minuscule, ponctuation, autre).

## 3.5 Approche basée sur l'apprentissage par transfert

Tel que mentionné précédemment, les modèles neuronaux demandent généralement pour être entraînés une grande quantité de données annotées pour produire des modèles puissants et prévenir le sur-apprentissage. En plus, nombreux algorithmes d'apprentissage automatique ont besoin d'être reconstruits à zero, en utilisant de nouvelles données d'entraînement, dès que la distribution des données ou l'espace d'attributs du départ change. Dans ce contexte, le transfert de connaissance, améliorerait la performance de l'apprentissage en nous permettant d'exploiter de grandes quantités de données annotées des *datasets* hors-domaine, et d'utiliser les traits appris pour effectuer une tâche sur un *dataset* cible (Pan et Yang, 2010; Meftah *et al.*, 2018).

### 3.5.1 Définition de l'apprentissage par transfert

Le principe consiste à faire apprendre un réseau de neurones parent sur un problème source avec une quantité suffisante de données annotées et transférer ces connaissances apprises pour initialiser un réseau enfant et représenter les données d'un problème cible avec une quantité insuffisante d'exemples d'entraînements (Pan et Yang, 2010; Meftah *et al.*, 2018). Cette approche a donné de très bons resultats et a amélioré d'une façon incroyable la REN. La figure 3.6 (Pan et Yang, 2010) montre la différence entre le processus d'apprentissage automatique dans les techniques traditionnelles et celles basées sur le transfert de connaissance. Tel que montré dans la figure, dans les techniques traditionnelles d'apprentissage, le modèle doit être ré-entraîné de zéro pour chaque tâche, alors que les techniques basées sur le transfert d'apprentissage essayent de transférer la connaissance à partir des tâches précédentes vers une tâche cible ayant une quantité insuffisante de données d'entraînement.

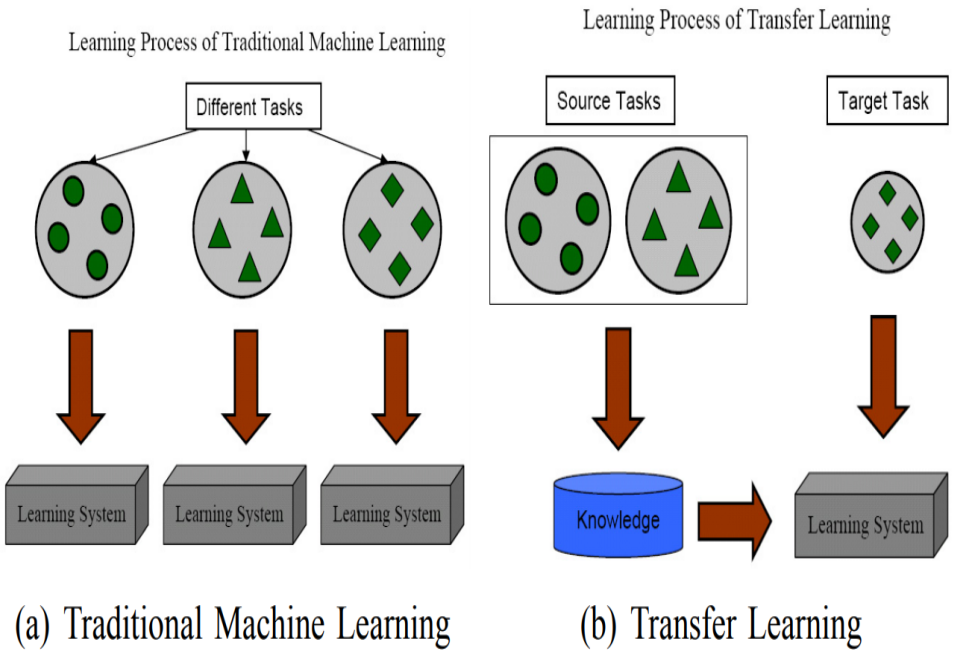


Figure 3.6 Différence entre le processus d'apprentissage dans les techniques traditionnelles et celles basées sur le transfert de connaissance

Deux approches de transfert d'apprentissage ont été proposées pour la reconnaissance des entités nommées multi-classes (Rodriguez *et al.*, 2018) :

3.5.1.0.1 PRED

Dans cette approche, le transfert d'apprentissage est réalisé en entraînant un classifieur sur le domaine source, et utiliser ses prédictions sur le domaine cible comme des traits pour un deuxième classifieur.

#### 3.5.1.0.2 Pré-entraînement

Dans cette approche, nous entraînons un réseau de neurones sur le domaine source, nous utilisons les poids pré-entraînés pour initialiser un nouveau réseau de neurones, et nous faisons par la suite un *fine-tuning* des poids sur le domaine cible.

### 3.5.2 Adaptation

Pour adapter un modèle pré-entraîné pour un modèle cible, différentes directions peuvent être considérées parmi lesquelles nous citons la modification ou non de l'architecture du modèle et les schémas d'optimisation.

#### 3.5.2.0.1 Modification architecturale

Lors du transfert d'apprentissage, dépendamment de la tâche cible, nous avons le choix de changer ou non l'architecture interne du modèle pré-entraîné (Ruder *et al.*, 2019).

- **Garder l'architecture interne du modèle pré-entraîné inchangée :** Cela peut être aussi simple que d'ajouter une ou plusieurs couches linéaires au-dessus d'un modèle pré-entraîné. Par contre, nous pouvons utiliser la sortie du modèle comme entrée pour un modèle séparé, ce qui est parfois bénéfique quand une tâche cible demande des interactions qui ne sont pas valables dans des *embeddings* pré-entraînés.
- **Modifier l'architecture interne du modèle pré-entraîné :** Une des raisons pour lesquelles nous pourrions vouloir faire cela, est d'adapter le modèle pour une tâche cible de structure différente, avec plusieurs séquences d'entrée par exemple. Dans ce cas, nous pouvons utiliser le mo-

dèle pré-entraîné pour initialiser le plus possible le modèle de la tâche cible de structure différente. De plus, modifier les paramètres de la tâche cible pourrait réduire le nombre de paramètres qui nécéssitent un *fine-tuning*.

3.5.2.0.2 Modification des schémas d'optimisation

En terme d'optimisation du modèle lors du transfert d'apprentissage, nous pouvons choisir de mettre à jour ou non les poids pré-entraînés. Dans la pratique, un classifieur linéaire est entraîné au-dessus des représentations pré-entrainées. La figure 3.7 (Ruder *et al.*, 2019) montre l'utilisation d'un modèle pré-entraîné comme attributs dans un modèle séparé en aval.

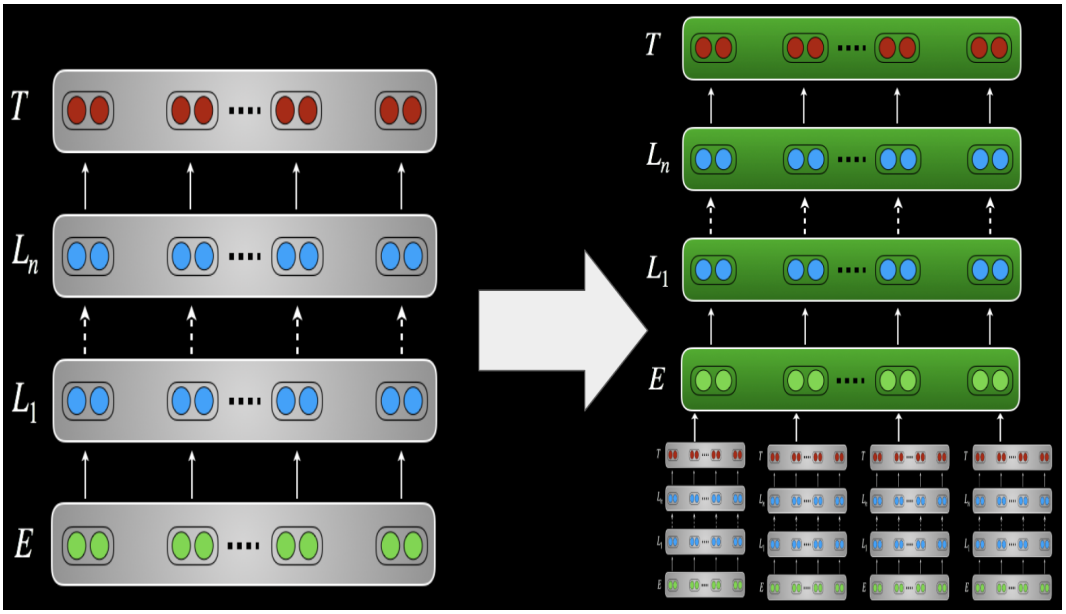


Figure 3.7 Utilisation d'un modèle pré-entraîné comme attribut dans un modèle séparé en aval

— **Ne pas changer les poids pré-entraînés :**

Une bonne performance peut être atteinte en utilisant non seulement la re-

présentation de la dernière couche, mais apprendre la combinaison linéaire des représentations des couches antérieures (Peters *et al.*, 2018). Les représentations pré-entraînées peuvent être utilisées comme des entrées dans un modèle en aval sans modifier les poids (Ruder *et al.*, 2019).

— **Changer les poids pré-entraînés (*fine-tuning*) :**
Dans ce cas, les poids pré-entraînés sont utilisés pour initialiser les paramètres d'un modèle en aval. Toute l'architecture pré-entraînée est ensuite entraînée durant la phase d'adaptation (Ruder *et al.*, 2019).

### 3.5.3 Approche proposée

Dans notre approche basée sur le transfert d'apprentissage, nous entraînons un premier modèle sur les données CoNLL03. Nous transférons les poids pré-entraînés du premier modèle pour initialiser les poids du modèle cible ayant la même architecture que le modèle de départ. Ces poids sont *fine-tunés* par la suite en entraînant le modèle cible sur les données annotées du domaine cible. Les deux figures 3.8 et 3.9 montrent les deux architectures du modèle source et du modèle cible respectivement. Les deux modèles ont la même architecture et les différentes couches ont les mêmes dimensions à l'exception de la dernière couche CRF où la dimension de la couche change dépendamment du nombre d'entités.

## 3.6 Conclusion

Dans ce chapitre, nous avons présenté notre approche hybride pour la REN en domaine général et restreint. Dans notre méthodologie nous commençons par un modèle statistique où nous utilisons des traits de base. Nous enrichissons notre modèle par la suite par des traits orthographiques qui englobent des informations importantes liées à la tâche de la REN. Cette approche statistique est suivie

| Layer (type) | Output Shape | Param # | Connected to |
|---|---|---|---|
| char_input (InputLayer) | (None, None, 52) | 0 | |
| char_embedding (TimeDistributed | (None, None, 52, 30) | 2700 | char_input[0][0] |
| drop_1 (Dropout) | (None, None, 52, 30) | 0 | char_embedding[0][0] |
| Conv1D (TimeDistributed) | (None, None, 52, 30) | 2730 | drop_1[0][0] |
| MaxPooling1D (TimeDistributed) | (None, None, 1, 30) | 0 | Conv1D[0][0] |
| words_input (InputLayer) | (None, None) | 0 | |
| casing_input (InputLayer) | (None, None) | 0 | |
| Flatten (TimeDistributed) | (None, None, 30) | 0 | MaxPooling1D[0][0] |
| words_embed (Embedding) | (None, None, 100) | 2294900 | words_input[0][0] |
| case_embed (Embedding) | (None, None, 8) | 64 | casing_input[0][0] |
| drop_2 (Dropout) | (None, None, 30) | 0 | Flatten[0][0] |
| concatenate_1 (Concatenate) | (None, None, 138) | 0 | words_embed[0][0]<br>case_embed[0][0]<br>drop_2[0][0] |
| bidirectional_1 (Bidirectional) | (None, None, 400) | 542400 | concatenate_1[0][0] |
| crf (CRF) | (None, None, 9) | 3708 | bidirectional_1[0][0] |

Figure 3.8 Architecture du modèle source

par une approche hybride où nous appliquons l'apprentissage par transfert pour transférer la connaissance d'un domaine général vers un domaine restreint. Dans le chapitre suivant, nous présentons les résultats d'évaluations de nos différents modèles.

| Layer (type) | Output Shape | Param # | Connected to |
|---|---|---|---|
| char_input (InputLayer) | (None, None, 52) | 0 | |
| char_embedding (TimeDistributed | (None, None, 52, 30) | 2700 | char_input[0][0] |
| drop_1 (Dropout) | (None, None, 52, 30) | 0 | char_embedding[0][0] |
| Conv1D (TimeDistributed) | (None, None, 52, 30) | 2730 | drop_1[0][0] |
| MaxPooling1D (TimeDistributed) | (None, None, 1, 30) | 0 | Conv1D[0][0] |
| words_input (InputLayer) | (None, None) | 0 | |
| casing_input (InputLayer) | (None, None) | 0 | |
| Flatten (TimeDistributed) | (None, None, 30) | 0 | MaxPooling1D[0][0] |
| words_embed (Embedding) | (None, None, 100) | 2294900 | words_input[0][0] |
| case_embed (Embedding) | (None, None, 8) | 64 | casing_input[0][0] |
| drop_2 (Dropout) | (None, None, 30) | 0 | Flatten[0][0] |
| concatenate_1 (Concatenate) | (None, None, 138) | 0 | words_embed[0][0]<br>case_embed[0][0]<br>drop_2[0][0] |
| bidirectional_1 (Bidirectional) | (None, None, 400) | 542400 | concatenate_1[0][0] |
| crf (CRF) | (None, None, 33) | 14388 | bidirectional_1[0][0] |

Figure 3.9 Architecture du modèle cible

# CHAPITRE IV

# ÉVALUATIONS

Dans ce chapitre, nous commençons par introduire les données que nous utilisons dans nos expérimentations. Nous enchaînons ensuite avec l'introduction des métriques utilisées pour l'évaluation de nos différents modèles. Par la suite, nous présentons les différentes configurations faites pour l'implémentation et l'entraînement de nos modèles. En dernier lieu, nous présentons nos évaluations en utilisant les métriques choisies et nous discutons les différents résultats obtenus.

## 4.1 Les données d'évaluation

### 4.1.1 Données du domaine général

Il est à noter que plusieurs travaux sur la REN reportent les performances de leurs modèles sur les *datasets* CoNLL03 et OntoNotes. Ces deux *datasets* sont considérés comme les *datasets* les plus utilisés pour comparer les performances des différents modèles existants. CoNLL03 contient des annotations des nouvelles de Reuters [1] entre août 1996 et août 1997 dans deux langues : anglais et allemand. Ce *dataset* contient 4 types d'entités nommées : *location* (LOC), *person* (PER), *organization*

1. `http://www.reuters.com/researchandstandards/`

Tableau 4.1 Statistique du dataset CoNLL03 en anglais

| **Datasets** | **#tokens** | **#types** | **#sentences** |
|---|---|---|---|
| **Train** | 204,562 | 23,624 | 14,985 |
| **Dev** | 51,573 | 9,966 | 3,464 |
| **Test** | 46,624 | 9,485 | 3,682 |
| **Total** | 302,759 | 43,075 | 22,131 |

(ORG) et *miscellaneous* (MISC), ce dernier type représente l'ensemble des entités qui n'appartiennent pas aux types d'entités précédents. Le tableau 4.1 présente les statistiques de ce *dataset*. Quant au projet OntoNotes, il a été construit dans le but d'annoter un corpus large incluant des genres de textes plus variés (*weblogs*, nouvelles, *talk shows*, des coversations téléphoniques, etc.).

### 4.1.2 Données du domaine restreint

Les données du domaine restreint que nous utilisons sont des données du domaine de l'électronique fournies par la compagnie Thales. Le dataset contient 33 types d'entités nommées. Le tableau 4.2 présente les différentes entités avec leurs nombres d'occurences. Le tableau 4.3 présente les statistiques du *dataset* de Thales. Les données sont à la base dans le format json. Ces données ont été transformées par la suite dans le format CoNLL tel que présenté dans la table 4.4.

#### 4.1.2.1 Nouvelle taxonomie

Dans le but de réduire le nombre important d'entités nommées dans le *dataset*, nous définissons une nouvelle taxonomie où nous regroupons toutes les entités dont le nombre d'occurence est inférieur à 60 et nous les considérons comme des "O" pour "*Out*". À titre d'exemple les entitées "Networking_Device", "Au-

Tableau 4.2 Les différentes entités du domaine restreint avec leurs nombres d'occurence

| **Entité** | **Fréquence** | **Entité** | **Fréquence** |
|---|---|---|---|
| O | 101502 | Network_Service_Provider | 1134 |
| User_Name(TL) | 823 | Service_Offering | 574 |
| Model_Name | 668 | TV_Box_Device | 168 |
| Networking_Manufacturer | 388 | Software_Manufacturer | 261 |
| Video_Feature | 116 | Version | 114 |
| Location(TL) | 107 | | |
| Networking_Feature | 101 | Networking_Protocol | 92 |
| TV_Box_Manufacturer | 81 | TV_Manufacturer (TL) | 73 |
| Software_Device | 60 | Phone_Device | 46 |
| Networking_Device | 42 | Signal_Strength | 40 |
| Audio_Manufacturer | 29 | Error_Code | 21 |
| Cabling_Device | 15 | Phone_Manufacturer | 14 |
| Computer_Manufacturer | 13 | Hardware_Feature | 12 |
| Manufacturer | 11 | Software_Feature | 10 |
| Network_Device | 9 | Audio_Feature | 9 |
| UID | 9 | Mac_Address | 9 |
| Computer_Device | 6 | IP_Address | 5 |

Tableau 4.3 Statistique du dataset du domaine restreint (en anglais) (nouvelle taxonomie)

| **Datasets** | **#tokens** | **#sentences** |
|---|---|---|
| **Train** | 85,000 | 5,080 |
| **Dev** | 17,000 | 1,016 |
| **Test** | 11,330 | 677 |
| **Total** | 113,300 | 6,773 |

Tableau 4.4 Exemple de données du domaine restreint dans le format CoNLL

| **Token** | **POSTag** | **chunkTag** | **Label** |
|---|---|---|---|
| I | PRP | O | O |
| have | VBP | B-VP | O |
| a | DT | B-NP | O |
| Sony | NNP | I-NP | B-TV_Manufacturer |
| Bravia | NNP | I-NP | B-Model_Name |
| KDL | NNP | I-NP | I-Model_Name |
| 52V4100 | CD | O | I-Model_Name |

dio_Manufacturer", "Cabling_Device" sont tranformées en "*Out*".

## 4.2 Critères d'évaluation

Différentes mesures d'évaluation pour examiner les performances des systèmes REN ont été discutées dans la littérature. L'évaluation est faite pour vérifier la capacité d'un outil à identifier correctement les types d'entités. Pour cela, les prédictions des sytèmes REN sont comparées avec les prédictions faites par des annotateurs experts. Dans notre travail, les métriques d'évaluation intrinsèques utilisées pour la comparaison sont la précision, le rappel, et le *F-score* (Goyal *et al.*, 2018).

### 4.2.1 Précision, rappel et F-score

La précision, le rappel et le F-score sont calculés en se basant sur les vrais positifs (VP), les faux positifs (FP) et les faux négatifs (FN). Les vrais positifs sont les entités qui sont reconnues par les systèmes REN et coïncident avec le *ground truth*. Les faux positifs sont les entités qui sont reconnues par les systèmes REN mais ne coïncident pas avec le *ground truth* et les faux négatifs sont les entités annotées dans le *ground truth* mais qui ne sont pas reconnues par le système REN (Li *et al.*, 2018).

#### 4.2.1.1 La précision

La précision mesure l'aptitude d'un système REN à ne déterminer que les entités correctes . Cette dernière est calculée en faisant le rapport entre les entités prédites correctement et toutes les entités nommées détectées (Li *et al.*, 2018).

Tableau 4.5 Calcul de la précision

| **Predits** / **Réels** | **Negatifs** | **Positifs** |
|---|---|---|
| **Negatifs** | Vrais Negatifs | Faux Positifs |
| **Positifs** | Faux Negatifs | Vrais Positifs |

La précision est calculée par la formule suivante :

$$Precision = \frac{VraisPositifs}{VraisPositifs + FauxPositifs} \tag{4.1}$$

#### 4.2.1.2 Le rappel

Le rappel mesure l'aptitude d'un système REN à déterminer toutes les entités dans le corpus. Cette dernière est calculée en faisant le rapport entre les entités prédites correctement et l'ensemble des entités étiquetées correctement et les étiquettes non détectées (Li *et al.*, 2018).

Tableau 4.6 Calcul du rappel

| **Predits** / **Réels** | **Negatifs** | **Positifs** |
|---|---|---|
| **Negatifs** | Vrais Negatifs | Faux Positifs |
| **Positifs** | Faux Negatifs | Vrais Positifs |

Le rappel est calculé par la formule suivante :

$$Rappel = \frac{VraisPositifs}{VraisPositifs + FauxNegatifs} \tag{4.2}$$

#### 4.2.1.3 F_mesure

La F_mesure est la moyenne pondérée de la précision et du rappel. Cette métrique a comme formule :

$$F1\_measure = 2 * \frac{Precision * Rappel}{Precision + Rappel} \tag{4.3}$$

Puisque la majorité des systèmes REN incluent des types d'entités multiples, il est souvent nécessaire d'évaluer la performance des systèmes à travers toutes les classes d'entités. Deux mesures sont couramment utilisées pour cet objectif : *macro-averaged F-score* et *micro-averaged F-score*. La *macro-averaged F-score* calcule le *F-score* indépendamment de chaque type d'entité, par la suite une moyenne de toutes ces valeurs est calculée. La *Micro-averaged F-score* prend en considération la contribution des entités des autres classes pour calculer la moyenne. Cette dernière peut être affectée par la qualité de reconnaître les entités dans des classes plus larges que d'autres dans le corpus (Li *et al.*, 2018).

## 4.3 Implémentation et entraînement

Tel que dans Chiu et Nichols (2015a), nous utilisons dans nos expérimentations le schéma d'étiquetage "BIO", acronyme de *Begin*, *Inside* et *Outside*, ce qui indique la position du *token* dans une entité composée. Ce schéma d'étiquetage est couramment utilisé dans la REN.

Nous implémentons notre modèle en utilisant la librairie Keras (backend Tensorflow). Comme il est mentionné dans le travail de Chiu et Nichols (2015a), les états initiaux des LSTM sont des vecteurs initialisés à zero. À part les plongements de mots et de caractères dont l'initialisation a été décrite précédemment, toutes les

*lookup tables* ont été initialisées aléatoirement.

Pour entraîner notre réseau, nous utilisons l'algorithme de rétro-propagation, en utilisant l'algorithme *nadam*. Nous utilisons des couches de LSTM de dimension 200. Nous n'utilisons pas des connaissances spécifiques à une langue ou bien des ressources externes tel que les *Gazetteers*. Nous utilisons également des couches *dropout* de 0.5 pour encourager le modèle de dépendre des plongements niveau caractère et des représentations des mots pré-entraînés. Nous introduisons des couches *dropout* aux sorties des couches LSTM qui sont efficaces pour eviter le sur-apprentissage (Lample *et al.*, 2016; Chiu et Nichols, 2015a). L'entraînement se fait par des *mini-batch*, qui sont des ensembles de phrases avec le même nombre de *tokens*. Nous explorons également d'autres algorithmes d'optimisation tels que *momentum* (Nesterov, 1983) et *AdaDelta* (Zeiler, 2012). Nous avons rendu notre code [2] disponible sur GitHub pour fin de référence.

## 4.4 Évaluation du modèle statistique

### 4.4.1 Résultats obtenus

Dans cette section nous présentons les résultats obtenus avec le modèle statistique. Les tables 4.7 et 4.8 présentent les résultats avec les caractéristiques basiques et les caractéristiques orthographiques que nous avons définies précédemment.

### 4.4.2 Discussion

À partir des tables 4.7 et 4.8, nous constatons qu'enrichir le modèle par des caractéristiques orthographiques, a mené à une augmentation remarquable des

2. `https://github.com/GhaithDek?tab=repositories`

Tableau 4.7 Résultats obtenus avec le modèle statistique avec les données du domaine restreint (caractéristiques basiques)

| **Entité** | **Précision** | **Rappel** | **F-mesure** |
|---|---|---|---|
| **Location** | 0.000 | 0.000 | 0.000 |
| **User_Name** | 0.784 | 0.397 | 0.527 |
| **TV_Manufacturer** | 0.667 | 0.250 | 0.364 |
| **Service_Offering** | 0.500 | 0.216 | 0.302 |
| **Video_Feature** | 0.667 | 0.333 | 0.444 |
| **Networking_Manufacturer** | 0.722 | 0.703 | 0.712 |
| **Model_Name** | 0.548 | 0.283 | 0.374 |
| **Software_Manufacturer** | 0.957 | 0.880 | 0.917 |
| **TV_Box_Device** | 0.773 | 0.680 | 0.723 |
| **Software_Device** | 1.000 | 0.500 | 0.667 |
| **Networking_Feature** | 0.500 | 0.333 | 0.400 |
| **Version** | 0.667 | 0.667 | 0.667 |
| **Networking_Protocol** | 0.750 | 0.600 | 0.667 |
| **Network_Service_Provider** | 0.946 | 0.963 | 0.955 |
| **TV_Box_Manufacturer** | 0.000 | 0.000 | 0.000 |
| **Moyenne** | 0.730 | 0.568 | 0.622 |

Tableau 4.8 Résultats obtenus avec le modèle statistique avec les données du domaine restreint (caractéristiques basiques + orthographiques)

| **Entité** | **Précision** | **Rappel** | **F-mesure** |
|---|---|---|---|
| **Location** | 0.000 | 0.000 | 0.000 |
| **User_Name** | **0.910** | **0.836** | **0.871** |
| **TV_Manufacturer** | **0.800** | **0.500** | **0.615** |
| **Service_Offering** | 0.800 | 0.216 | 0.302 |
| **Video_Feature** | 0.600 | **0.500** | **0.545** |
| **Networking_Manufacturer** | 0.676 | 0.622 | 0.648 |
| **Model_Name** | **0.576** | **0.317** | **0.409** |
| **Software_Manufacturer** | 0.957 | 0.880 | 0.917 |
| **TV_Box_Device** | **0.867** | 0.520 | 0.650 |
| **Software_Device** | 1.000 | 0.333 | 0.500 |
| **Networking_Feature** | 0.500 | 0.167 | 0.250 |
| **Version** | 0.667 | 0.667 | 0.667 |
| **Networking_Protocol** | 0.750 | 0.600 | 0.667 |
| **Network_Service_Provider** | 0.944 | 0.936 | 0.940 |
| **TV_Box_Manufacturer** | **0.400** | **0.222** | **0.286** |
| **Moyenne** | **0.794** | **0.661** | **0.712** |

moyennes des précisions et des rappels, ce qui a abouti à une amélioration de la F-mesure globale de 9% par rapport au modèle statistique basique. A part l'amélioration de la performance globale du système, nous remarquons une amélioration sur la majorité des entités prises séparément, telles que *User_ Name*, *TV_ Manufacturer*, *Video_ Feature*, *Model_ Name* et en dernier lieu *TV_ Box_ Manufacturer*. Toutes ces améliorations justifient le rôle important de l'information orthographique dans la tâche de la REN.

## 4.5 Évaluation du modèle neuronal

### 4.5.1 Résultats obtenus

La table 4.9 montre les résultats obtenus avec le modèle neuronal. Toutes les caractéristiques utlisées, les plongements de mots pré-entraînés et les paramètres de notre modèle ont été décrits précédemment.

### 4.5.2 Discussion

D'après les résultats présentés dans les tableaux 4.7 et 4.9, nous remarquons que notre approche neuronale améliore la F_mesure de 9 catégories d'entités nommées qui sont *Location*, *User_ name*, *TV_ Manufacturer*, *Service_ Offering*,*Video_ feature*, *Networking_ Manufacturer*, *Model_ name*, *Software_ Manufacturer* et *TV_ Box_ Manufacturer*. L'amélioration des résultats des ces entités, aboutit à une augmentaion de la F_mesure globale de toutes les catégories d'entités par rapport aux résultats obtenus avec le modèle statistique basique.

Si nous comparons maintenant les deux tableaux 4.8 et 4.9, nous constatons que le modèle neuronal améliore les types d'entités *Location*, *Service_ Offering*,

Tableau 4.9 Résultats obtenus avec le modèle neuronal avec les données du domaine restreint (sans transfert d'apprentissage)

| Entité | Précision | Rappel | F-mesure |
|---|---|---|---|
| **Location** | **0.375** | **0.375** | **0.375** |
| **User_Name** | **0.859** | **0.838** | **0.848** |
| **TV_Manufacturer** | 0.385 | **0.500** | **0.435** |
| **Service_Offering** | **0.667** | **0.286** | **0.400** |
| **Video_Feature** | **1.000** | **0.455** | **0.625** |
| **Networking_Manufacturer** | **0.667** | **0.839** | **0.743** |
| **Model_Name** | 0.474 | **0.614** | **0.535** |
| **Software_Manufacturer** | 0.947 | **0.900** | **0.923** |
| **TV_Box_Device** | 0.556 | 0.278 | 0.370 |
| **Software_Device** | 0.500 | 0.250 | 0.333 |
| **Networking_Feature** | 0.000 | 0.000 | 0.000 |
| **Version** | 0.600 | 0.545 | 0.571 |
| **Networking_Protocol** | 0.692 | **0.643** | 0.667 |
| **Network_Service_Provider** | 0.885 | 0.920 | 0.902 |
| **TV_Box_Manufacturer** | **0.800** | **0.308** | **0.444** |
| **Moyenne** | **0.743** | **0.704** | **0.723** |

*Video_feature*, *Networking_Manufacturer*, *Model_name*, *Software_Manufacturer* et *TV-Box-Manufacturer*.

Toutes ces améliorations prouvent l'efficacité de l'architecture de notre modèle neuronal et des différentes caractéristiques utilisées pour l'entraînement, notamment les plongements de mots pré-entraînés sur des larges quantités de données brutes et les plongements liés aux caractères extraits à partir de nos données du domaine restreint. Par conséquent, le système neuronal arrive à extraire les connaissances nécessaires à partir des données d'entraînement, sans avoir recours à des caractéristiques extraites à la main.

## 4.6 Évaluation de l'approche basée sur l'apprentissage par transfert

Dans cette section, nous commençons par présenter le *mapping* pour transférer les connaissances des données CoNLL03 vers les données Thales, nous enchaînons par la suite par les résultats obtenus, suivi par une comparaison des performances des différents modèles développés.

### 4.6.1 Mapping

Pour faire la correspondance de l'indexage des étiquettes entre le *dataset* source et le *dataset* cible, nous définissons un *mapping* entre les types d'entités nommées. Ce *mapping* est une sorte d'équivalence entre les types d'entités des deux *datasets*.

### 4.6.2 Résultats obtenus

La table 4.11 montre les résultats obtenus avec l'approche neuronale en appliquant l'apprentissage par transfert des données CoNLL vers les données Thales .

Tableau 4.10 Mapping utilisé entre les entités sources et les entités cibles

| **CoNLL dataset** | | **Thales dataset** | |
|---|---|---|---|
| **Entité** | **Indice** | **Entité** | **Indice** |
| O | 0 | O | 0 |
| B-LOC | 1 | B-Location | 1 |
| B-PER | 2 | B-User_Name | 2 |
| B-ORG | 3 | B-TV_Manufacturer | 3 |
| I-PER | 4 | I-User_Name | 4 |
| I-ORG | 5 | I-TV_Manufacturer | 5 |
| I-LOC | 6 | I-Location | 6 |
| B-MISC | 7 | B-MISC | 7 |
| I-MISC | 8 | I-MISC | 8 |
| | | ... | ... |

### 4.6.3 Discussion

D'après les résultats présentés dans les tables 4.7, 4.8, 4.9 et 4.11, nous remarquons que le transfert d'apprentissage des données CoNLL vers les données Thales améliore les moyennes des précisions, des rappels et des F_mesure de toutes les entités évaluées. L'application de l'apprentissage par transfert a montré une amélioration significative dans la F_mesure, 15% comparée aux résultats obtenus avec le premier modèle statistique, 6% comparée à ceux obtenus avec le deuxième modèle statistique et de 5% comparée aux résultats obtenus avec le modèle neuronal basique.

Si nous comparons les deux tableaux 4.9 et 4.11, nous remarquons que le transfert d'apprentissage améliore la précision, le rappel et par conséquent la F_mesure des entités *Location*, *TV_Manufacturer*, *Service_Offering*, *TV_ Box_ Device* et

Tableau 4.11 Résultats obtenus avec le transfert d'apprentissage des données du domaine général vers les données du domaine restreint

| Entité | Précision | Rappel | F-mesure |
|---|---|---|---|
| **Location** | **0.925** | **0.933** | **0.929** |
| **User_Name** | 0.839 | 0.729 | 0.780 |
| **TV_Manufacturer** | **0.688** | **0.733** | **0.710** |
| **Service_Offering** | **0.846** | **0.440** | **0.579** |
| **Video_Feature** | 1.000 | 0.154 | 0.267 |
| **Networking_Manufacturer** | **0.857** | 0.600 | 0.706 |
| **Model_Name** | **0.500** | 0.300 | 0.375 |
| **Software_Manufacturer** | 0.500 | 0.200 | 0.286 |
| **MISC** | 1.000 | 0.400 | 0.571 |
| **TV_Box_Device** | **0.714** | **0.556** | **0.625** |
| **Software_Device** | 0.000 | 0.000 | 0.000 |
| **Networking_Feature** | **0.500** | **0.400** | **0.444** |
| **Version** | **1.000** | 0.250 | 0.400 |
| **Networking_Protocol** | 0.000 | 0.000 | 0.000 |
| **Network_Service_Provider** | 0.000 | 0.000 | 0.000 |
| **TV_Box_Manufacturer** | 0.000 | 0.000 | 0.000 |
| **Moyenne** | **0.824** | **0.727** | **0.773** |

*Networking_Feature* par rapport au modèle neuronal basique, cela est dû essentiellement au *mapping* effectué entre les deux *datasets* source et cible, ce qui a permis au modèle cible de tirer profit de la connaissance liée aux entités du domaine source.

Si nous comparons les deux tableaux 4.8 et 4.11, nous constatons qu'à part les types d'entités *Location*, *TV_Manufacturer*, *Service_Offering* et *Networking_Feature*,

l'apprentissage par transfert améliore aussi *Networking_ Manufacturer*, par contre, pour le reste des catégories d'entités, le modèle statistique, enrichi par les caractéristiques orthographiques performe mieux, cela peut être expliqué par l'importance de l'information syntaxique que rapportent ces caractéristiques et la large portion d'information qu'elles encodent, nécessaire pour le modèle dans la tâche de la REN.

Si nous comparons les deux tableaux 4.11 et 4.7, nous remarquons également une amélioration apportée par le transfert de connaissance, par rapport au modèle statistique basique, pour la valeure de la F_mesure de 5 entités qui sont *Location*, *User_ name*, *TV_ Manufacturer*, *Service_ Offering* et *Networking_ Feature*.

## 4.7 Conclusion

Dans ce chapitre, nous avons commencé par présenter les données et les critères d'évaluations utilisés dans notre étude. Nous avons enchaîné par détailler l'implémentation et l'entraînement des différents modèles et nous avons clôturé par l'évaluation des performances de nos différents modèles dans la tâche de la REN. Dans nos évaluations, nous avons constaté d'une part l'importance des traits orthographiques que nous avons utilisés dans notre modèle statistique, et d'autre part l'efficacité du transfert de connaissances du domaine général avec des quantités suffisantes de données d'entraînement, vers le domaine restreint avec peu de données annotées.

# CHAPITRE V

# APPRENTISSAGE À PARTIR D'UNE BASE DE CONNAISSANCES EXTERNE

## 5.1 Introduction

Le *Commonsense* peut être vital dans certaines applications telles que la compréhension du langage naturel (en anglais : Natural Language Understanding (NLU)), où il est souvent requis de résoudre l'ambiguïté causée par l'information implicite dans une phrase. Parmi les formes sous lesquelles le *Commonsense* peut appraître nous citons les graphes de connaissances. Dans ce chapitre, nous commençons par souligner l'importance du *Commonsense* et son implication dans diverses applications. Nous enchaînons par présenter les graphes de connaissances. Par la suite nous expliquons notre approche, suivie des expérimentations et des évaluations de nos résultats.

## 5.2 Commonsense

Malgré le succès remarquable des approches neuronales dans différentes tâches du TALN, plusieurs deviennent moins performantes dans des situations qui demandent le *Commonsense*. Pour cette raison, les communités du TALN et de l'apprentissage automatique se sont toujours intéressés au développement de mo-

dèles capables de raisonner en utilisant le *Commonsense*. Cela a été accentué par la prolifération des technologies de l'intelligence artificielle telles que les systèmes de dialogue, les systèmes de recommandation et les outils de recherche d'information (Trichelair *et al.*, 2018). Trichelair *et al.* (2018) confirme que le progrès de ces technologies et l'interaction des utilisateurs avec ces systèmes dépendent des avancements du raisonnement basé sur le *Commonsense*, au point que nous pourrions dire que ces systèmes peuvent paraître inintelligents quand ils manquent de *Commonsense*. Dans notre travail, nous exploitons l'idée d'intégrer les connaissances basées sur le *Commonsense* dans l'une des tâches du TALN qui est la REN, pour apprendre plus d'entités et améliorer l'efficacité d'un système REN. Pour cela, nous profitons de la disponibilité du graphe de connaissances multilingue libre d'accès ConceptNet, que nous présentons plus en détail dans une section suivante, en l'utilisant comme une ressource externe dans un système REN. Le raisonnement basé sur le *Commonsense* a été employé dans d'autres études liées aux différentes applications du TALN, telles que les sytèmes de question/réponse, l'analyse des sentiments et les résumés automatiques. Au mieux de notre investigation de la littérature, nous n'avons pas trouvé de méthodes qui emploient le *Commonsense* pour traiter la tâche de la REN.

## 5.3 Les graphes de connaissances

Les dernières années ont vu une croissance rapide dans l'utilisation des graphes de connaissances. Ces derniers sont devenus omniprésents et sont utilisés dans de nombreux domaines tels que la bio-informatique, la chimie et les réseaux sociaux (Bai *et al.*, 2019). Cela a été intensifié par l'apparition d'un grand nombre de graphes de connaissances tels que YAGO (Suchanek *et al.*, 2007), Freebase (Bollacker *et al.*, 2008), DBpedia (Lehmann *et al.*, 2015) et ConceptNet (Speer *et al.*, 2017) qui ont été appliqués avec succès à des applications du monde réel, telles que

l'extraction d'information, les systèmes de question/réponse et la désambiguïsation des entités nommées (Wang *et al.*, 2017; Nguyen, 2019). Ces représentations de connaissances sont des collections de triplets $(h, r, t)$, où chaque triplet représente une relation $r$ (en anglais : *relation*) entre une entité tête $h$ (en anglais : *head entity*) et une entité queue $t$ (en anglais : *tail entity*). Les bases de connaissances peuvent être considérées par conséquent comme étant des graphes directs multi-relationnels, où les noeuds correspondent aux entités, et les arrêtes liant les noeuds encodent différents types de relations (Nguyen, 2019). La Figure 5.1 (Zhong *et al.*, 2019) montre un échantillon de sous-graphe extrait à partir de Conceptnet.

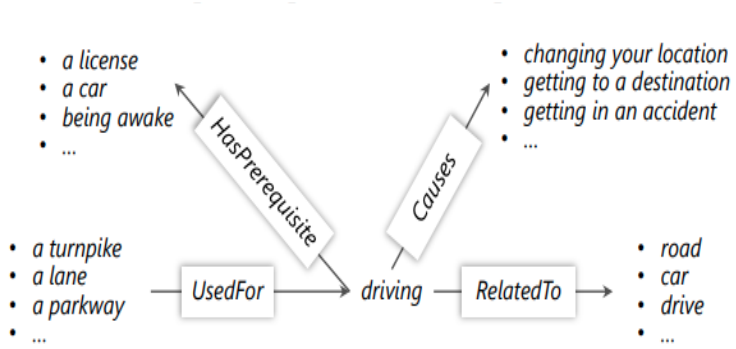


Figure 5.1 Un échantillon de sous-graphe de ConceptNet avec driving comme mot central

Malgré son efficacité dans la représentation des données structurées, la nature symbolique des triplets rend ces graphes de connaissances difficiles à manipuler. Pour remédier à ce problème, les plongements de graphes ont été proposés. La construction de ces plongements consiste à représenter les composants du graphe, incluant les entités et les relations, dans des espaces vectoriels continus, pour simplifier leurs manipulations tout en préservant la structure inhérente à la base de connaissance (Wang *et al.*, 2017). Ces représentations vectorielles peuvent être utilisées par la suite pour entraîner des modèles d'apprentissage automatique.

## 5.4 ConceptNet

ConceptNet est une base de connaissances basée sur le *Commonsense*, qui connecte les mots et les phrases du langage naturel (les termes), par des arrêtes étiquetées (les relations). Sa connaissance a été collectée à partir de plusieurs ressources. Elle contient plus de 21 millions d'arrêtes et plus de 8 milions de noeuds, son vocabulaire anglais contient approximativement 1.5 millions de noeuds. Elle a été désignée pour représenter la connaissance générale nécessaire pour la compréhension des langues, améliorant les applications liées au langage naturel, en leur permettant de mieux comprendre le sens derrière les mots que les utilisateurs emploient (Speer *et al.*, 2017). Cette nouvelle version de ressource de données liées est particulièrement bien adaptée pour les techniques avancées du TALN et les approches telles que les plongements de mots et de graphes. Quand ConceptNet est combinée avec des plongements de mots tels que GloVe et Word2Vec, elle offre aux applications une compréhension qu'elles ne pourraient pas acquérir ni à partir des sémantiques distributionnelles, ni à partir de ressources plus restreintes telles que WordNet et DBPedia. Notre travail repose sur le travail présenté par Speer *et al.* (2017) qui ont bâti des plongements de mots robustes qui représentent non seulement ConceptNet mais en plus les plongement de mots distributionnels appris à partir des textes, en concaténant les colonnes des matrices pré-entraînées fournies par Word2Vec et Glove, avec les plongements de mots extraits à partir de ConceptNet. Cet ensemble de plongements de mots représente différents domaines et regroupe des forces complémentaires.

## 5.5 Approche proposée

Dans notre approche, nous investissons l'idée d'utiliser des plongements de mots pré-entraînés extraits à partir de la ressource externe ConceptNet pour enri-

chir nos données d'entraînement dans la tâche de la REN. Ces plongements de mots sont appelés ConceptNet PPMI (Speer *et al.*, 2017). Nous utilisons par la suite ConceptNet Numberbatch (Speer *et al.*, 2017), qui est la concaténation de ConceptNet PPMI avec les plongements de mots de Word2Vec et ceux de GloVe. Cette étude a pour but de comparer les plongements de mots qui ne représentent que la connaissance relationnelle (ConceptNet PPMI ) et leur combinaison avec les plongements de mots qui ne représentent que les sémantiques distributionnelles (Word2vec et GloVe).

Dans notre architecture nous utilisons des couches BiLSTM, combinées avec une couche CRF augmenté par d'autres traits tels que des couches *dropout*. Tel qu'il a été mentionné précedemment, apprendre les plongements niveau caractère a l'avantage d'apprendre des représentations spécifiques à la tâche et au domaine étudié. Dans notre travail, à part l'utilisation des représentations de mots, nous utilisons également des représentations basées sur les caractères pour détecter l'information morphologique et orthographique. En suivant Lample *et al.* (2016), nous utilisons des couches BiLSTM pour extraire la représentation de chaque mot à partir de ses caractères. Une *look-up table* pour les caractères, initialisée aléatoirement et qui contient un plongement pour chaque caractère a également été utilisée. Le plongement de mot déduit à partir de ses caractères est le résultat de la concaténation des deux vecteurs résultants des couches BiLSTM. L'extraction des représentations de mots à partir de leurs caractères a été expliquée plus en détails dans le chapitre III.

## 5.6 Expérimentations et évaluations

Dans cette section nous décrivons les données utilisées dans cette partie du travail. Nous donnons une vue d'ensemble sur l'implémentation et l'entraînement de notre

modèle, aussi bien que les évaluations et les résultats.

### 5.6.1 Les données d'évaluations

Nos expérimentations sont basées sur le *dataset* du domaine général CoNLL03 présenté dans le chapitre IV. Tel que dans Lample *et al.* (2016), nous utilisons dans nos expérimentations le schéma d'annotations IOBES, une variation des IOB qui sont couramment utilisés dans la REN, qui encode l'information sur les entités singleton (S) et marque explicitement la fin des entités nommées (E).

### 5.6.2 Implémentations et entraînements

Pour entraîner notre réseau, nous utilisons l'algorithme de rétro-propagation, en utilisant l'algorithme du gradient stochastique (en anglais : *stochastic gradient descent* ou SGD) avec un taux d'apprentissage égal à 0.01. Nous utilisons une seule couche pour les deux LSTM de dimension 100 chacune. Nous utilisons un *dropout* de 0.5 pour encourager le modèle à dépendre des plongements niveau caractère et des représentations de mots pré-entraînés. À part remplacer chaque bit avec un zero dans le *dataset*, nous n'appliquons pas de pré-traitement de données.

Dans nos expérimentations, nous avons réimplémenté l'architecture du modèle de Lample *et al.* (2016). [1]

### 5.6.3 Évaluations des résultats

Les tables 5.1, 5.2 et 5.3 montrent une comparaison des résultats obtenus avec notre modèle.

---

1. `https://github.com/glample/tagger`

D'après les résultats présentés dans la table 5.2 nous remarquons d'une part que l'utilisation des plongements de mots de ConceptNet PPMI améliore le F-score des deux catégories d'entités nommées, LOC et PER. D'autre part, l'augmentation de la précision de ces deux entités a abouti à une augmentation de la précision globale de toutes les catégories des entités nommées. Par contre, l'utilisation des plongements de mots de ConceptNet PPMI, n'a pas amélioré les résultats de la catégorie d'entité nommée ORG, qui est un problème commun dans la tâche de la REN puisque les organisations peuvent être exprimées par des acronymes, qui sont extrêmement ambigus. Par contre, si nous analysons la table 5.3, nous remarquons une augmentation de la précision de la catégorie ORG. Cela peut être exliqué par l'utilisation des plongements de mots de GloVe et Word2Vec. En plus, la précision de la catégorie PER a augmenté remarquablement, ce qui mène à une amélioration de la précision globale.

Pour conclure, l'utilisation des plongements de mots de ConcepNet PPMI, en combinaison avec ceux de GloVe et Word2Vec, donne de meilleurs résultats que de les utiliser indépendamment. L'utilisation de ces plongements hybrides a montré une amélioration significative de 2.86% dans la F-mesure du système global. La combinaison des forces des trois plongements de mots pré-entraînés, construits à partir de différentes données et différents domaines, a aidé dans l'amélioration de la performance globale du système. En plus, la petite quantité de données à partir de laquelle les plongements de mots de ConcepNet PPMI ont été extraits (21 millions d'arrêtes et plus de 8 millions de noeuds, son vocabulaire anglais contient approximativement 1.5 million de noeuds) comparé à GloVe (840 billions mots du Common Crawl) et Word2Vec (100 billions mots de Google News), peut expliquer l'amélioration de la performance, surtout lors de l'utilisation de la combinaison des plongements. De plus, tel que montré dans les trois tables, l'utilisation de ConceptNet Numberbatch a amélioré remarquablement, la performance de chaque

entité prise séparément.

Tableau 5.1 Les résultats de la REN sans l'utilisation de plongements de mots pré-entraînés (seulement les plongements construits à partir de nos données ont été utilisés)

| **Entité** | **Précision(%)** | **Rappel(%)** | **F1-score (%)** |
|---|---|---|---|
| LOC | 89.56 | 88.97 | 89.26 |
| MISC | 79.82 | 74.36 | 76.99 |
| ORG | 84.66 | 77.06 | 80.68 |
| PER | 81.10 | 92.08 | 86.24 |
| **Moyenne** | 84.38 | 84.54 | **84.46** |

Tableau 5.2 Les résultats de la REN avec les plongements ConceptNet PPMI + les plongements construits à partir de nos données

| **Entité** | **Précision(%)** | **Rappel(%)** | **F1-score (%)** |
|---|---|---|---|
| LOC | **91.83** | 87.59 | **89.66** |
| MISC | 75.73 | 73.79 | 74.75 |
| ORG | 81.44 | 79.53 | 80.48 |
| PER | **85.81** | 88.62 | **87.19** |
| **Moyenne** | **85.02** | 83.80 | **84.40** |

## 5.7 Conclusion

Dans ce chapitre, nous avons commencé par présenter le *Commonsense*, les graphes de connaissances en général et la base de connaissance ConcepNet que nous avons utilisés pour intégrer le *Commonsense* dans l'entraînement de notre modèle. Nous avons enchaîné par présenter notre approche, et nous avons clôturé par l'évaluation des résultats obtenus. Nos résultats ont démontré le rôle du *Commonsense*

Tableau 5.3 Les résultats de la REN avec les plongements ConceptNet Numberbatch + les plongements construits à partir de nos données

| **Entité** | **Précision(%)** | **Rappel(%)** | **F1-score (%)** |
|---|---|---|---|
| LOC | 89.03 | 91.97 | **90.47** |
| MISC | 76.94 | 77.49 | **77.22** |
| ORG | **86.06** | 81.40 | **83.66** |
| PER | **92.08** | 92.08 | **92.08** |
| **Moyenne** | **87.56** | 87.09 | **87.32** |

dans l'amélioration des performances d'un système REN dans le domaine général.

# CONCLUSION

Dans ce mémoire, nous avons présenté une étude sur la conception et développement d'un système de reconnaissance des entités nommées pour deux domaines, général et restreint. En nous basant sur un domaine spécifique, l'électronique, nous avons appliqué des méthodes statistiques, une méthode neuronale basique et une méthode neuronale basée sur l'apprentissage par transfert. Dans nos expérimentations, nous avons utilisé des caractéristiques basiques et des caractéristiques orthographiques liées à la syntaxe des mots. Nous avons utilisé également des plongements niveau mots et des plongements niveau caractères, ce qui permet de détecter l'information morphologique et orthographique des mots et d'apprendre des représentations spécifiques à la tâche et aux domaines étudiés.

D'après les résultats présentés précédemment, nous avons conclu que l'apprentissage par transfert permet de donner de meilleurs résultats comparativement au modèle neuronal de base et aux deux modèles statistiques basés sur des caractéristiques extraites à la main, malgré l'accès de ces derniers à des règles spécifiques du domaine.

Dans notre travail, nous avons évalué également la contribution d'une base de connaissance basée sur le *Commonsense*, ConceptNet, dans l'amélioration des performances d'un système REN dans le domaine général.

Nos contributions dans ce mémoire reposent sur trois aspects distincts. Le premier aspect consiste au développement d'un système REN pour un domaine restreint. L'enjeu dans notre cas consiste au fait que nous ne disposons pas de quantités suffisantes de données annotées et en plus, le nombre d'entités est assez élevé

comparativement aux données libres d'accès existantes. Le deuxième aspect repose sur l'application de la technique de transfert de connaissance d'un domaine source vers un domaine cible, le domaine de l'électronique. Au mieux de notre investigation de la littérature, la majorité des méthodes existantes sont désignées pour les problèmes liés à la classification de l'image et du texte au lieu des problèmes de classification des séquences ce qui rend notre tâche plus complexe. Shaha et Pawar (2018) appliquent l'apprentissage par transfert dans la tâche de classification d'images en utilisant comme point de départ les systèmes les plus populaires, AlexNet (Krizhevsky *et al.*, 2012), VGG16 et VGG19 (Simonyan et Zisserman, 2015). Les auteurs montrent qu'utiliser ces modèles pré-entraînés donne de meilleures performances comparativement aux systèmes existants. Bhatt *et al.* (2016) et Moon et Carbonell (2016) proposent des modèles de classification de textes où ils transfèrent les connaissances de différents domaines sources vers un domaine cible. Les travaux de Rodriguez *et al.* (2018) et Chen et Moschitti (2019) sont considérés comme étant les travaux les plus proches du nôtre. Les auteurs ont également appliqué l'apprentissage par transfert pour la REN où les classes d'entités dans les domaines source et cible sont différentes. Dans le premier travail les auteurs ont traité différents domaines cibles tels que les domaines général, médical et militaire, mais n'ont pas abordé le domaine de l'électronique. De plus, le nombre d'entités dans les domaines traités est supérieur au nombre d'entités dans le domaine source, mais reste remarquablement inférieur au nombre d'entités dans notre cas, ce qui rend notre tâche plus difficile. Dans le deuxième travail, les auteurs ont traité un seul domaine cible qui est le domaine général mais qui diffère d'une seule entité par rapport au domaine source ce qui rend notre tâche encore plus difficile. Le troisième aspect de nos contributions consiste à l'emploi du *Commonsense* pour améliorer les performances d'un système REN dans le domaine général. À notre connaissance, c'est la première étude qui emploie le *Commonsense* dans la tâche de la REN, en combinant des plongements de mots

extraits à partir des connaissances relationnelles (ConceptNet Numberbatch) avec des plongements sémantiques distributionnels (Word2Vec et GloVe).

Parmi les perspectives, nous proposons d'utiliser d'autres traits ou caractéristiques dans l'architecture hybride et impliquer d'autres bases de connaissances telles que les ontologies et les bases lexicales du *Commonsense*. Yadav *et al.* (2018) ont prouvé qu'enrichir un modèle CNN-BiLSTM-CRF par des affixes aboutit à une amélioration remarquable des performances de leur système. Nous comptons également appliquer le mécanisme d'attention et comparer le modèle résultant aux modèles que nous avons développés. L'application de l'apprentissage multi-tâche ferait partie également de nos travaux futurs pour évaluer l'effet de cette technique de transfert d'apprentissage sur la performance de notre modèle. Il serait pertinent également d'appliquer les techniques d'interprétablité et/ou explicabilité pour mieux expliquer les résultats obtenus grâce à nos différents modèles.

PUBLICATIONS

**Ghaith Dekhili**, Ngoc Tan Le, Fatiha Sadat. **Augmenting Named Entity Recognition with Commonsense Knowledge**. In Proceedings of the 57th Annual Meeting of **ACL2019**, Widening Natural Language Processing Workshop **(WiNLP2019)**, Florence, Italy, from July 28th to August 2nd, 2019.

**Ghaith Dekhili**, Ngoc Tan Le, Fatiha Sadat. **Improving Named Entity Recognition with Commonsense Knowledge Pretraining**. In Proceedings of **PRICAI2019**, Pacific Rim Knowledge Acquisition Workshop **(PKAW2019)**, Yanuca Island, Cuvu, Fiji, August 26-30, 2019.

ANNEXE A

| | | | |
|---|---|---|---|
| 1. CC | Coordinating conjunction | 25. TO | *to* |
| 2. CD | Cardinal number | 26. UH | Interjection |
| 3. DT | Determiner | 27. VB | Verb, base form |
| 4. EX | Existential *there* | 28. VBD | Verb, past tense |
| 5. FW | Foreign word | 29. VBG | Verb, gerund/present participle |
| 6. IN | Preposition/subordinating conjunction | 30. VBN | Verb, past participle |
| 7. JJ | Adjective | 31. VBP | Verb, non-3rd ps. sing. present |
| 8. JJR | Adjective, comparative | 32. VBZ | Verb, 3rd ps. sing. present |
| 9. JJS | Adjective, superlative | 33. WDT | *wh*-determiner |
| 10. LS | List item marker | 34. WP | *wh*-pronoun |
| 11. MD | Modal | 35. WP$ | Possessive *wh*-pronoun |
| 12. NN | Noun, singular or mass | 36. WRB | *wh*-adverb |
| 13. NNS | Noun, plural | 37. # | Pound sign |
| 14. NNP | Proper noun, singular | 38. $ | Dollar sign |
| 15. NNPS | Proper noun, plural | 39. . | Sentence-final punctuation |
| 16. PDT | Predeterminer | 40. , | Comma |
| 17. POS | Possessive ending | 41. : | Colon, semi-colon |
| 18. PRP | Personal pronoun | 42. ( | Left bracket character |
| 19. PP$ | Possessive pronoun | 43. ) | Right bracket character |
| 20. RB | Adverb | 44. " | Straight double quote |
| 21. RBR | Adverb, comparative | 45. ‘ | Left open single quote |
| 22. RBS | Adverb, superlative | 46. “ | Left open double quote |
| 23. RP | Particle | 47. ’ | Right close single quote |
| 24. SYM | Symbol (mathematical or scientific) | 48. ” | Right close double quote |

Figure 5.2 Le Penn Treebank POS tagset

# BIBLIOGRAPHIE


Aguilar, G., Maharjan, S., López-Monroy, A. P. et Solorio, T. (2017). A multi-task approach for named entity recognition in social media data. Dans *Proceedings of the 3rd Workshop on Noisy User-generated Text*, 148–153. Association for Computational Linguistics.

Akbik, A., Blythe, D. et Vollgraf, R. (2018). Contextual string embeddings for sequence labeling. Dans *Proceedings of the 27th International Conference on Computational Linguistics*, 1638–1649. Association for Computational Linguistics.

Amplayo, R. K., Lim, S. et Hwang, S.-w. (2018). Entity commonsense representation for neural abstractive summarization. Dans *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long Papers)*, 697–707. Association for Computational Linguistics.

Appelt, D. E., Hobbs, J. R., Bear, J., Israel, D., Kameyama, M., Kehler, A., Martin, D., Myers, K. et Tyson, M. (1995). SRI International FASTUS systemMUC-6 test results and analysis. Dans *Sixth Message Understanding Conference (MUC-6) : Proceedings of a Conference Held in Columbia, Maryland, November 6-8, 1995*, 237 – 248.

Artetxe, M., Labaka, G. et Agirre, E. (2018). A robust self-learning method for fully unsupervised cross-lingual mappings of word embeddings. Dans *Proceedings of the 56th Annual Meeting of the Association for Computational Lin-*

*guistics (Volume 1 : Long Papers)*, 789–798., Australia. Association for Computational Linguistics.

Athiwaratkun, B. et Wilson, A. (2017). Multimodal word distributions. Dans *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, 1645–1656., Canada. Association for Computational Linguistics.

Auer, S., Bizer, C., Kobilarov, G., Lehmann, J., Cyganiak, R. et Ives, Z. (2007). Dbpedia : A nucleus for a web of open data. Dans *The Semantic Web*, 722–735. Springer Berlin Heidelberg.

Bai, Y., Ding, H., Bian, S., Chen, T., Sun, Y. et Wang, W. (2019). Simgnn : A neural network approach to fast graph similarity computation. 384–392. http://dx.doi.org/10.1145/3289600.3290967

Balahur, A., Hermida, J. M. et Montoyo, A. (2011). Detecting implicit expressions of sentiment in text based on commonsense knowledge. Dans *Proceedings of the 2nd Workshop on Computational Approaches to Subjectivity and Sentiment Analysis (WASSA 2.011)*, 53–60. Association for Computational Linguistics.

Bari, M. S., Joty, S. et Jwalapuram, P. (2020). Zero-resource cross-lingual named entity recognition. Dans *Thirty-Fourth AAAI Conference on Artificial Intelligence*. AAAI.

Bengio, Y., Ducharme, R., Vincent, P. et Janvin, C. (2003). A neural probabilistic language model. *The Journal of Machine Learning Research*, *3*, 1137–1155.

Bengio, Y., Simard, P. et Frasconi, P. (1994). Learning long-term dependencies with gradient descent is difficult. *IEEE Transactions on Neural Networks*, *5*(2), 157–166.

Bhatt, H. S., Sinha, M. et Roy, S. (2016). Cross-domain text classification with multiple domains and disparate label sets. Dans *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, 1641–1650. Association for Computational Linguistics.

Bird, S., Klein, E. et Loper, E. (2009). *Natural Language Processing with Python*. O'Reilly.

Bollacker, K., Evans, C., Paritosh, P., Sturge, T. et Taylor, J. (2008). Freebase : a collaboratively created graph database for structuring human knowledge. Dans *In SIGMOD Conference*, 1247–1250. ACM.

Bosselut, A., Rashkin, H., Sap, M., Malaviya, C., Çelikyilmaz, A. et Choi, Y. (2019). Comet : Commonsense transformers for automatic knowledge graph construction. Dans *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 4762–4779. ACL.

Camacho-Collados, J. et Pilehvar, M. T. (2018). From word to sense embeddings : A survey on vector representations of meaning. *Journal of Artificial Intelligence Research*, *63*(1), 743–788.

Chen, L. et Moschitti, A. (2019). Transfer learning for sequence labeling using source model and target data. *CoRR*, *abs/1902.05309*.

Chiu, J. P. et Nichols, E. (2015a). Named entity recognition with bidirectional LSTM-CNNs. *Transactions of the Association for Computational Linguistics*, *4*(1), 357–370.

Chiu, J. P. C. et Nichols, E. (2015b). Named entity recognition with bidirectional lstm-cnns. *Transactions of the Association for Computational Linguistics*, *4*, 357–370.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K. et Kuksa, P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, *12*, 2493–2537.

Cowie, J. et Lehnert, W. (1996). Information extraction. *ACM*, *39*(1), 80–91.

Devlin, J., Chang, M., Lee, K. et Toutanova, K. (2018). BERT : pre-training of deep bidirectional transformers for language understanding. *CoRR*, *abs/1810.04805*.

Doddington, G., Mitchell, A., Przybocki, M., Ramshaw, L., Strassel, S. et Weischedel, R. (2004). The automatic content extraction (ACE) program – tasks, data, and evaluation. Dans *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Portugal. European Language Resources Association (ELRA).

Dong, X., Chowdhury, S., Qian, L., Guan, Y., Yang, J. et Yu, Q. (2017). Transfer bi-directional lstm rnn for named entity recognition in chinese electronic medical records. Dans *2017 IEEE 19th International Conference on e-Health Networking, Applications and Services (Healthcom)*, 1–4.

Dyer, C., Ballesteros, M., Ling, W., Matthews, A. et Smith, N. A. (2015). Transition-based dependency parsing with stack long short-term memory. volume 1, p. 334 – 343. Récupéré de `http://arxiv.org/abs/1505.08075`

Freund, Y. et Schapire, R. E. (1999). Large margin classification using the perceptron algorithm. *Machine Learning*, *37*(3), 277–296.

Friburger, N. (2002). Reconnaissance automatique des noms propres : application à la classification automatique de textes journalistiques. *(Thèse de doctorat). Université de Tours.*

Gillick, D., Brunk, C., Vinyals, O. et Subramanya, A. (2016). Multilingual language processing from bytes. Dans *Proceedings of NAACL-HLT*, 1296 – 1306. Association for Computational Linguistics.

Goldberg, Y. (2016). A primer on neural network models for natural language processing. *Journal of Artificial Intelligence Research*, *57*(1), 345–420.

Goller, C. et Kuchler, A. (1996). Learning task-dependent distributed representations by backpropagation through structure. Dans *Proceedings of International Conference on Neural Networks (ICNN'96)*, 347–352 vol.1. IEEE.

Google. (2012). *Introducing the Knowledge Graph : things, not strings*. Récupéré de `https://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html`

Goyal, A., Gupta, V. et Kumar, M. (2018). Recent named entity recognition and classification techniques : A systematic review. *Computer Science Review*, *29*(1), 21–43.

Hanisch, D., Fundel, K., Mevissen, H.-T., Zimmer, R. et Fluck, J. (2005). Prominer : rule-based protein and gene entity recognition. *BMC Bioinformatics*, *6*(1), S14 – S14.

Hasan, F., UzZaman, N. et Khan, M. (2007). Comparion of different pos tagging technique (n-gram, hmm and brill's tagger) for bangla. Dans *Elleithy K. (eds) Advances and Innovations in Systems, Computing Sciences and Software Engineering*, 121–126. Springer.

Hochreiter, S. et Schmidhuber, J. (1997). Long short-term memory. *Neural Computation*, *9*(8), 1735–1780.

Hu, A., Dou, Z., Nie, J.-Y. et Wen, J.-R. (2020). Leveraging multi-token entities in document-level named entity recognition. Dans *Thirty-Fourth AAAI Conference on Artificial Intelligence*. AAAI.

Huang, Z., Xu, W. et Yu, K. (2015). Bidirectional LSTM-CRF models for sequence tagging. *CoRR*, *abs/1508.01991*.

Humphreys, K., Gaizauskas, R., Azzam, S., Huyck, C., Mitchell, B., Cunningham, H. et Wilks, Y. (1998). University of Sheffield : Description of the LaSIE-II system as used for MUC-7. Dans *Seventh Message Understanding Conference (MUC-7) : Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998*. Morgan.

Jilek, C., Schröder, M., Novik, R., Schwarz, S., Maus, H. et Dengel, A. (2018). Inflection-tolerant ontology-based named entity recognition for real-time applications. *CoRR*, *abs/1812.02119*.

Joshi, M., Hart, E., Vogel, M. et Ruvini, J.-D. (2015). Distributed word representations improve NER for e-commerce. Dans *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, 160–167., Colorado. Association for Computational Linguistics.

Kenter, T. M. (2017). Text understanding for computers. *(PhD thesis). University of Amsterdam*.

Khan, M. R., Ziyadi, M. et Abdelhady, M. (2020). Mt-bioner : Multi-task learning for biomedical named entity recognition using deep bidirectional transformers. *ArXiv*, *abs/2001.08904*.

Krizhevsky, A., Sutskever, I. et Hinton, G. (2012). Imagenet classification with deep convolutional neural networks. *Neural Information Processing Systems*, *25*. http://dx.doi.org/10.1145/3065386

Krupka, G. R. et Hausman, K. (1998). IsoQuest inc. : Description of the NetOwl$^{\text{tm}}$ extractor system as used for MUC-7. Dans *Seventh Message Understanding Conference (MUC-7) : Proceedings of a Conference Held in Fairfax, Virginia, April 29 - May 1, 1998*, 21 – 28.

Lafferty, J. D., McCallum, A. et Pereira, F. C. N. (2001). Conditional random fields : Probabilistic models for segmenting and labeling sequence data. Dans *Proceedings of the Eighteenth International Conference on Machine Learning*, 282–289. Morgan Kaufmann Publishers Inc.

Lample, G., Ballesteros, M., Subramanian, S., Kawakami, K. et Dyer, C. (2016). Neural architectures for named entity recognition. Dans *Proceedings of NAACL-HLT 2016*, p. 260–270.

Le, T. N. (2019). Traduction automatique pour une paire de langues peu dotée. *(Thèse de doctorat). Université du Québec à Montréal.*

Lee, J., Kim, H., Lee, J. et Yoon, S. (2017). Transfer learning for deep learning on graph-structured data. Dans *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2154–2160. AAAI Press.

Lehmann, J., Isele, R., Jakob, M., Jentzsch, A., Kontokostas, D., Mendes, P. N., Hellmann, S., Morsey, M., van Kleef, P., Auer, S. et Bizer, C. (2015). Dbpedia - a large-scale, multilingual knowledge base extracted from wikipedia. *Semantic Web*, *6*(2), 167–195.

Levy, O. et Goldberg, Y. (2014). Dependency-based word embeddings. Dans *52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014 - Proceedings of the Conference*, volume 2, 302–308.

Li, J., Sun, A., Han, J. et Li, C. (2018). A survey on deep learning for named entity recognition. *CoRR*.

Lin, H., Lu, Y., Han, X., Sun, L., Dong, B. et Jiang, S. (2019). Gazetteer-enhanced attentive neural networks for named entity recognition. Dans *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 6232–6237. Association for Computational Linguistics.

Ling, W., Luís, T., Marujo, L., Astudillo, R. F., Amir, S., Dyer, C., Black, A. W. et Trancoso, I. (2015). Finding function in form : Compositional character models for open vocabulary word representation. Dans *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Liu, P., Fu, J., Dong, Y., Qiu, X. et Cheung, J. C. K. (2018). Multi-task learning over graph structures. *CoRR*, *abs/1811.10211*.

Luo, Y., Xiao, F. et Zhao, H. (2019). Hierarchical contextualized representation for named entity recognition. *CoRR*, *abs/1911.02257*.

Ma, K., Francis, J., Lu, Q., Nyberg, E. et Oltramari, A. (2019). Towards generalizable neuro-symbolic systems for commonsense question answering. 22–32. `http://dx.doi.org/10.18653/v1/D19-6003`

Ma, X. et Hovy, E. (2016a). End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. Dans *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, 1064–1074. Association for Computational Linguistics.

Ma, X. et Hovy, E. H. (2016b). End-to-end sequence labeling via bi-directional lstm-cnns-crf. Dans *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics*, p. 1064 – 1074.

Marcus, M. P., Marcinkiewicz, M. A. et Santorini, B. (1993). Building a large

annotated corpus of english : The penn treebank. *Computational Linguistics*, *19*(2), 313–330.

Meftah, S., Semmar, N. et Sadat, F. (2018). A Neural Network Model for Part-Of-Speech Tagging of Social Media Texts. Dans *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2821 – 2828., Japan. European Language Resources Association (ELRA).

Mikolov, T., Chen, K., Corrado, G. et Dean, J. (2013a). Efficient estimation of word representations in vector space. *Proceedings of Workshop at ICLR*, *2013*.

Mikolov, T., Yih, W.-t. et Zweig, G. (2013b). Linguistic regularities in continuous space word representations. Dans *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, 746–751., Georgia. Association for Computational Linguistics.

Moon, S. et Carbonell, J. (2016). Proactive transfer learning for heterogeneous feature and label spaces. Dans P. Frasconi, N. Landwehr, G. Manco, et J. Vreeken (dir.). *Machine Learning and Knowledge Discovery in Databases*, 706–721. Springer International Publishing.

Nadeau, D. et Sekine, S. (2007). A survey of named entity recognition and classification. *Lingvisticae Investigationes*, *30*(1), 3–26.

Nesterov, Y. (1983). A method for solving a convex programming problem with convergence rate o(1/k2). *Soviet Mathematics Doklady*, *27*(2), 372–367.

Nguyen, D. Q. (2019). An overview of embedding models of entities and relationships for knowledge base completion. *CoRR*, *abs/1703.08098*.

Nouvel, D., Antoine, J.-Y., Friburger, N. et Soulet, A. (2013). Fouille de règles d'annotation pour la reconnaissance d'entités nommées. *Traitement Automatique des Langues*, *54*(2), 13 –41.

Pan, S. J. et Yang, Q. (2010). A survey on transfer learning. *IEEE Transactions on Knowledge and Data Engineering*, *22*(10), 1345–1359.

Pennington, J., Socher, R. et Manning, C. D. (2014). Glove : Global vectors for word representation. Dans *In 12th Proceedings of the EmpiricialMethods in Natural Language Processing (EMNLP 2014)*, p. 1532–1543.

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K. et Zettlemoyer, L. (2018). Deep contextualized word representations. Dans *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long Papers)*, 2227–2237. Association for Computational Linguistics.

Poibeau, T., Acoulon, A., Avaux, C., Beroff-Bénéat, L., Cadeau, A., Calberg, M., Delale, A., Temmerman, L. D., Guenet, A.-L., Huis, D., Jamalpour, M., Krul, A., Marcus, A., Picoli, F. et Plancq, C. (2003). The multilingual named entity recognition framework. Dans *10th Conference of the European Chapter of the Association for Computational Linguistics*, 155 – 158., Hungary. Association for Computational Linguistics.

Quimbaya, A. P., Múnera, A. S., Rivera, R. A. G., Rodríguez, J. C. D., Velandia, O. M. M., Peña, A. A. G. et Labbé, C. (2016). Named entity recognition over electronic health records through a combined dictionary-based approach. *Procedia Computer Science*, *100*(1), 55 – 61.

Ratinov, L. et Roth, D. (2009). Design challenges and misconceptions in named entity recognition. Dans *Proceedings of the Thirteenth Conference on Compu-*

*tational Natural Language Learning*, 147–155. Association for Computational Linguistics.

Rodriguez, J. D., Caldwell, A. et Liu, A. (2018). Transfer learning for entity recognition of novel classes. Dans *Proceedings of the 27th International Conference on Computational Linguistics*, 1974–1985. Association for Computational Linguistics.

Ruder, S., Peters, M. E., Swayamdipta, S. et Wolf, T. (2019). Transfer learning in natural language processing. Dans *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Tutorials*, 15–18. Association for Computational Linguistics.

Sachan, D. S., Xie, P., Sachan, M. et Xing, E. P. (2018). Effective use of bidirectional language modeling for transfer learning in biomedical named entity recognition. Dans *Proceedings of the 3rd Machine Learning for Healthcare Conference*, 383–402. PMLR.

Sap, M., LeBras, R., Allaway, E., Bhagavatula, C., Lourie, N., Rashkin, H., Roof, B., Smith, N. A. et Choi, Y. (2018). ATOMIC : an atlas of machine commonsense for if-then reasoning. *CoRR, abs/1811.00146*.

Schwartz, R., Reichart, R. et Rappoport, A. (2015). Symmetric pattern based word embeddings for improved word similarity prediction. Dans *Proceedings of the Nineteenth Conference on Computational Natural Language Learning*, 258–267., China. Association for Computational Linguistics.

Shaha, M. et Pawar, M. (2018). Transfer learning for image classification. 656–660. `http://dx.doi.org/10.1109/ICECA.2018.8474802`

Simonyan, K. et Zisserman, A. (2015). Very deep convolutional networks for large-scale image recognition. *CoRR, abs/1409.1556*.

Sousa, D. et Couto, F. M. (2020). Biont : Deep learning using multiple biomedical ontologies for relation extraction. Dans J. M. Jose, E. Yilmaz, J. Magalhães, P. Castells, N. Ferro, M. J. Silva, et F. Martins (dir.). *Advances in Information Retrieval*, 367–374. Springer International Publishing.

Speer, R., Chin, J. et Havasi, C. (2017). Conceptnet 5.5 : An open multilingual graph of general knowledge. *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence (AAAI-17)*, 4444–4451.

Straková, J., Straka, M. et Hajic, J. (2019). Neural architectures for nested NER through linearization. Dans *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, 5326–5331. Association for Computational Linguistics.

Suchanek, F., Kasneci, G. M. et Weikum, G. M. (2007). Yago : A Core of Semantic Knowledge. Dans *16th international conference on World Wide Web*, 697 – 697. http://dx.doi.org/10.1145/1242572.1242667

Tjong Kim Sang, E. F. et De Meulder, F. (2003). Introduction to the conll-2003 shared task : Language-independent named entity recognition. Dans *Proceedings of the Seventh Conference on Natural Language Learning at HLT-NAACL 2003*, 142–147. Association for Computational Linguistics.

Trichelair, P., Emami, A., Cheung, J. C. K., Trischler, A., Suleman, K. et Díaz, F. (2018). On the evaluation of common-sense reasoning in natural language understanding. *ArXiv, abs/1811.01778*.

Trinh, T. H. et Le, Q. V. (2018). A simple method for commonsense reasoning. *CoRR, abs/1806.02847*.

Veenstra, J. et Tjong Kim Sang, E. (1999). Representing text chunks. Dans

*Proceedings of the Ninth Conference of the European Chapter of the Association for Computational Linguistics (EACL '99)*, 173–179.

Wang, L., Sun, M., Zhao, W., Shen, K. et Liu, J. (2018). Yuanfudao at SemEval-2018 task 11 : Three-way attention and relational knowledge for commonsense machine comprehension. Dans *Proceedings of The 12th International Workshop on Semantic Evaluation*, 758–762. Association for Computational Linguistics.

Wang, Q., Mao, Z., Wang, B. et Guo, L. (2017). Knowledge graph embedding : A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, *29*(12), 2724–2743.

Wieting, J., Bansal, M., Gimpel, K. et Livescu, K. (2016). Charagram : Embedding words and sentences via character n-grams. Dans *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, p. 1504–1515.

Yadav, V. et Bethard, S. (2018). A survey on recent advances in named entity recognition from deep learning models. Dans *Proceedings of the 27th International Conference on Computational Linguistics*, 2145–2158. Association for Computational Linguistics.

Yadav, V., Sharp, R. et Bethard, S. (2018). Deep affix features improve neural named entity recognizers. Dans *Proceedings of the Seventh Joint Conference on Lexical and Computational Semantics*, 167–172. Association for Computational Linguistics.

Yao, Y. et Huang, Z. (2016). Bi-directional lstm recurrent neural network for chinese word segmentation. Dans *Neural Information Processing*, 345–353. Springer International Publishing.

Yin, W., Kann, K., Yu, M. et Schütze, H. (2017). Comparative study of CNN and RNN for natural language processing. *CoRR*, *abs/1702.01923*.

Zeiler, M. D. (2012). ADADELTA : an adaptive learning rate method. *CoRR*, *abs/1212.5701*.

Zhang, S. et Elhadad, N. (2013). Unsupervised biomedical named entity recognition : Experiments with clinical and biological texts. *Journal of Biomedical Informatics*, *46*(6), 1088 – 1098.

Zhang, S., Liu, X., Liu, J., Gao, J., Duh, K. et Durme, B. V. (2018). Record : Bridging the gap between human and machine commonsense reading comprehension. *CoRR*, *abs/1810.12885*.

Zhao, S., Liu, T., Zhao, S. et Wang, F. (2018). A neural multi-task learning framework to jointly model medical named entity recognition and normalization. Dans *AAAI*.

Zhong, W., Tang, D., Duan, N., Zhou, M., Wang, J. et Yin, J. (2019). Improving question answering by commonsense-based pre-training. Dans *Natural Language Processing and Chinese Computing*, 16–28. Springer International Publishing.

Zhu, Y. et Wang, G. (2019). CAN-NER : Convolutional Attention Network for Chinese Named Entity Recognition. Dans *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, 3384–3393. Association for Computational Linguistics.

Zou, W. Y., Socher, R., Cer, D. et Manning, C. D. (2013). Bilingual word embeddings for phrase-based machine translation. Dans *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, 1393–1398., USA. Association for Computational Linguistics.