

MÉMOIRE DE FIN D'ÉTUDES

Pour l'obtention du diplôme du master de recherche

*BIOINFORMATIQUE ET MODÉLISATION DES SYSTÈMES
COMPLEXES LIÉES À LA SANTÉ*



Construction d'un graphe de connaissances biomédicales : Cas de COVID-19

Réalisé par :
SNOUSSI Youssef

Encadré par :
Pr. ABIK Mounia

Soutenu le 29 Septembre 2021, Devant le jury composé de :

Président	:	Pr. TABII YOUNESS	ENSIAS
Examinateur	:	Pr. EZZAHOUT ABDERRAHMANE	FSR
Encadrant	:	Pr. ABIK MOUNIA	ENSIAS
Co-Encadrant	:	Mr. HAJHOUJ MOHAMMED	ENSIAS

Remerciements

Toute recherche universitaire, pour qu'il puisse être menée à bien, nécessite un encadrement de qualité.

Je tiens donc à remercier le corps professoral de l'École Nationale d'Informatique et d'Analyse de Systèmes qui, durant ces deux années, nous ont guidé pour décrocher ce diplôme, et plus particulièrement **Pr. Mounia Abik** pour son encadrement sans faille, et ses nombreux conseils qui m'ont permis d'avancer.

Je tiens aussi à remercier les membres du jury, **Pr. TABII Youness** et **Pr. EZ-ZAHOUT Abderrahmane** qui ont accepté d'évaluer mon modeste travail.

Je veux également remercier et exprimer ma gratitude au doctorant **Mr. Hajhouj Mohammed** pour ces précieux conseils, et pour son aide durant toutes les étapes de réalisation de ce projet.

Résumé

La construction des graphes de connaissances est une tâche fastidieuse, mais c'est l'un des outils puissants dont nous disposons pour représenter les connaissances et les manipuler. Dans ce travail, nous avons essayé de présenter une lecture rapide de l'état de l'art des différentes étapes de création et d'application des graphes de connaissances, à savoir, la reconnaissance d'entités nommées, l'extraction de relations et l'intégration de graphes de connaissances. Ensuite, nous avons choisi parmi les différentes approches existantes celles les plus performantes pour réaliser les étapes mentionnées. Pour la reconnaissance des entités nommées, nous avons utilisé une variation de BERT, pour l'extraction de relations, nous avons utilisé une variation de BERT qui effectue l'étiquetage des rôles sémantiques et par la suite l'extraction des relations sous forme de verbes. Enfin nous avons utilisé les algorithmes d'intégration des graphes de connaissances pour prévoir de nouveaux liens.

Mots clés : Graph de connaissances, Reconnaissance d'entités nommées, Extraction des Relations, Représentation des connaissances, Covid-19.

Abstract

Knowledge graph construction is a tedious task, yet it is one of the efficient tools we have to represent knowledge and manipulate it. In this work, we have tried to present a quick read in the literature state of the art methods of creating and applying knowledge graphs, specifically Named Entity Recognition, Relation Extraction and Knowledge Graph Embeddings. After that, we construct a "COVID-19 Knowledge Graph" from scientific publications and tried to apply KGE algorithms to infer new links between the entities we already have.

Keywords : Knowledge Graph, NER, RE, KGE, COVID-19.

Table des matières

Remerciements	II
Résumé	IV
Abstract	V
Introduction générale	1
1 Généralités sur le NLP, text mining, et le corpus biomédical	5
1.1 Contexte et fondamentaux	6
1.1.1 Sous tâches de NLP	6
1.1.2 Processus de text mining	7
1.1.3 Approches de text mining	8
1.1.4 Corpus biomédical	9
1.1.5 Conclusion	10
2 État de l'art	11
2.1 Reconnaissance d'entités nommées	12
2.1.1 Approches traditionnelles	12
2.1.2 Techniques d'apprentissage profond	13
2.2 Normalisation des entités nommées	18
2.2.1 Bases de connaissances utilisées pour la liaison des entités nommées biomédicales	19
2.2.2 Génération d'entités candidates	21
2.2.3 Classement des entités candidates	23
2.3 Extraction des relations	23
2.3.1 Méthodes traditionnelles	24
2.3.2 Méthodes basées sur les DNNs	26
2.4 L'intégration des graphes des connaissances	28
2.4.1 Différentes étapes d'intégration des graphes de connaissances	28
2.4.2 Modèles de distance translationnelle	28
2.4.3 Modèles de correspondance sémantique	30
2.4.4 Modèles basé sur les réseaux neuronaux de convolution	33
2.5 Conclusion	33
3 Réalisation et résultats	34
3.1 Préparation des données	35
3.1.1 Format des fichiers	35

Table des matières

3.1.2	Pré-traitement	35
3.2	Reconnaissance et liaison d'entités nommées	35
3.2.1	spaCy et scispaCy	36
3.2.2	Implémentation et résultats	39
3.3	Extraction des relations	40
3.3.1	AllenNLP et AllenNLP-models	40
3.3.2	Réalisation et résultats	41
3.4	Knowledge Graph Embeddings	43
3.4.1	Pykeen	43
3.4.2	Entraînement des modèles	45
3.4.3	Prédiction des nouveaux liens	48
3.5	Conclusion	51
Conclusion et perspectives		52
Bibliographie		54

Table des figures

2.1	Modèles basés sur CNN et RNN pour l'extraction de la représentation au niveau des caractères d'un mot.	14
2.2	Réseau d'approche des phrases basé sur CNN. La couche de convolution extrait les caractéristiques de la phrase entière, en la traitant comme une séquence avec une structure globale.	16
2.3	L'architecture de l'encodeur de contexte basée sur RNN.	16
2.4	Réseaux neuronaux récursifs bidirectionnels pour la NER	17
2.5	Architectures basées sur les Transformers pour la NER	18
2.6	L'idée principale de DIPRE.	24
2.7	Le processus de supervision à distance.	26
2.8	Illustration simple de TransE.	29
2.9	Illustration simple de TransH.	29
2.10	Illustration simple de TransR.	30
2.11	Illustration simple de RESCAL.	31
2.12	Illustration simple de DistMult.	32
2.13	Illustration simple de HolE.	32
3.1	Architecture du pipeline de traitement	37
3.2	Distribution des 30 types les plus et les moins fréquents.	41
3.3	Sous-graphe contenant les maladies et les substances chimiques	42
3.4	Sous-graphe contenant les noeuds ayant les degrés ≥ 100	43
3.5	Sous-graphe contenant les noeuds ayant les degrés entre 30 et 100	44
3.6	Développement du loss et du Hits @ k durant l'entraînement de TransE. .	46
3.7	Développement du loss et du Hits @ k durant l'entraînement de TransH. .	46
3.8	Développement du loss et du Hits @ k durant l'entraînement de TransR. .	47
3.9	Développement du loss et du Hits @ k durant l'entraînement de ComplEx. .	47
3.10	Développement du loss et du Hits @ k durant l'entraînement de ConvE. .	47

Liste des tableaux

1.1	Corpus biomédicaux reliés aux tâches de biomedical text mining	9
2.1	Exemple du motif "such X as Y".	24
3.1	Performances des modèles de langage de scispaCy.	38
3.2	Performances des modèles de reconnaissance d'entités nommées de scispaCy.	39
3.3	Exemples d'entités nommées extraites.	40
3.4	Performances des modèles choisis pour l'apprentissage des connaissances. .	48
3.5	Quelques triplets inférés par le modèle TransE.	49
3.6	Quelques triplets inférés par le modèle TransH.	49
3.7	Quelques triplets inférés par le modèle TransR.	50
3.8	Quelques triplets inférés par le modèle ComplEx.	50
3.9	Quelques triplets inférés par le modèle ConvE.	50

Liste des sigles et acronymes

SARS-COV-2 *Sever Acute Resperatory Syndrom Corona Virus 2*

WHO *World Health Organisation*

AllenAI *Allen Institute for Artificial Intelligence*

NIH *National Institutes of Health*

COVID-19 *Corona Virus Diseas 2019*

NLP *Natrual Language Processing*

NER *Named Entity Recognition*

RE *Relation Extraction*

KGC *Knowledge Graph Completion*

KRL *Knowledge Representation Learning*

KGE *Knowledge Graph Embedding*

POS *Part-Of-Speech*

DNN *Deep Neural Networks*

SDP *Shortest Dependency Path*

ML *Machine Learning*

DS *Distant Supervision*

SVM *Support Vector Machine*

HMM *Hidden Markov Models*

Liste des tableaux

CRF	<i>Conditional Random Fields</i>
CNN	<i>Convolutional Neural Networks</i>
RNN	<i>Recurrent Neural Networks</i>
BiLSTM	<i>Bidirectional Long-Short-Term Memory</i>
GPT	<i>Generative Pretrained Transformer</i>
BERT	<i>Bidirectional Encoder Representations from Transformers</i>
ELMO	<i>Embeddings from Language Models</i>
GloVe	<i>Global Vectors for words Representations</i>
UMLS	<i>Unified Medical Language System</i>
CPT	<i>Current Procedural Terminology</i>
LOINC	<i>Logical Observation Identifiers Names and Codes</i>
Mesh	<i>Medical Subject Headings</i>
GO	<i>Gene Ontology</i>
ICD-10-CM	<i>International Classification of Diseases</i>
NLM	<i>National Library of Medicine</i>
API	<i>Application Program Interface</i>
USP	<i>United States Pharmacopeia</i>
HPO	<i>Human Phenotype Ontology</i>
NIH	<i>National Institute of Health</i>
DIPRE	<i>Dual Iterative Pattern Relation Expansion</i>
KB	<i>Knowledge Base</i>
KG	<i>Knowledge Graph</i>
CUI	<i>Concept Unique Identifier</i>
SRL	<i>Semantic Role Labeling</i>

Liste des tableaux

Introduction générale

Introduction générale

Avec l'émergence de la pandémie de Coronavirus causé par le nouveau virus appelé "sever acute respiratory syndrome 2 (SARS-COV-2)", le monde s'est trouvé dans une situation critique, qui l'a poussé vers des changements radicales sur plusieurs niveaux. Cette pandémie a été premièrement apparue en Chine vers la fin de l'année 2019, ensuite elle s'est propagée dans le monde entier. En 11 Mars 2020, elle a été déclarée pandémie globale par l'organisation mondiale de la santé (WHO).

L'explosion des recherches scientifiques publiées concernant cette maladie était l'une des réactions les plus importantes pour la lutte contre cette pandémie, ces publications ont étudié les aspects de propagation, résultats des patients hospitalisés, les diagnostics et tests nécessaires, et elles ont couvert aussi la santé mentale et sociale[1], le nombre des publications est estimé par 100,000 publications en 2020 [1]. avec un record de 4000 publications dans une seule semaine [2]. Malgré ce nombre immense et important des publications scientifiques, les chercheurs n'avaient pas de chance d'exploiter ces quantités immenses de publications, car tout simplement l'être humain ne peut pas lire et assimiler, et surtout, faire le lien entre toutes les connaissances et informations contenues dans ces publications, ce qui rend la majorité de ces recherches inutiles.

Pour s'adresser à ce problème, et dans le cadre d'une initiative importante, la Maison Blanche, l'institut AllenAI, NIH et de nombreux autres partenaires ont lancé une compétition sur la plateforme des sciences des données Kaggle, où ils ont rassemblé plus de 59,000 articles, dont des études sur les coronavirus remontant aux années 1950. Aujourd'hui, l'ensemble de données contient presque 500,000 publications scientifiques, dont plus de 200,000 textes intégraux en accès libre[3]. Le but principale de cette compétition était d'encourager les chercheurs, surtout les informaticiens, de développer des solutions automatisées pour exploiter ces publications, à travers la réalisation de dix-sept tâches :

- Que savons-nous sur la transmission, l'incubation et la stabilité environnementale ?
- Que savons-nous sur les facteurs de risque du COVID-19 ?
- Que savons-nous sur les vaccins et les traitements ?
- Que savons-nous sur la génétique, l'origine et l'évolution du virus ?
- Qu'est-ce qui a été publié sur les soins médicaux ?
- Que savons-nous sur les interventions non pharmaceutiques ?
- Qu'est-ce qui a été publié sur les considérations éthiques et de sciences sociales ?
- Qu'est-ce qui a été publié sur le partage de l'information et la collaboration intersectorielle ?
- Que savons-nous sur le diagnostic et de la surveillance ?
- Créer des tableaux récapitulatifs qui traitent les facteurs pertinents liés à COVID-19.
- Créer des tableaux récapitulatifs qui traitent la thérapeutique, des interventions et des études cliniques.

Introduction générale

- Créer des tableaux récapitulatifs qui traitent les facteurs de risque liés au COVID-19.
- Créer des tableaux récapitulatifs qui traitent les diagnostics du COVID-19.
- Créer des tableaux récapitulatifs qui traitent les études matérielles liées au COVID-19.
- Créer des tableaux récapitulatifs qui traitent les modèles et les questions ouvertes liés à COVID-19.
- Créer des tableaux récapitulatifs qui traitent les études de population liées à COVID-19.
- Créer des tableaux récapitulatifs qui traitent les descriptions des patients liées à COVID-19.

Les techniques d'intelligence artificielle, et surtout du traitement automatique du langage naturel (NLP) et du text mining, peuvent faciliter la réalisation des tâches mentionnées ci-dessus, l'utilisation de ces techniques peut aider à regrouper ces publications selon les mots clés, ce qui va faciliter la recherche des articles relatifs à un sujet spécifique, elles peuvent également aider à traduire et à résumer des textes volumineux et à réaliser des systèmes de réponses aux questions, mais surtout, les techniques NLP et text mining peuvent être utilisées pour extraire automatiquement des informations du texte, ce qui peut être réalisé en effectuant deux étapes majeures, que nous allons aborder dans les chapitres suivants, la première est la reconnaissance des entités nommées (Named Entity Recognition NER), et la seconde est l'extraction des relations (Relation Extraction RE).

L'objectif de cette recherche est de réaliser une lecture dans la littérature pour découvrir les différentes techniques qui s'adressent à la réalisations de ces tâches, par la suite, nous allons choisir parmi ces techniques les plus convenables pour la construction d'un graphe de connaissances de la maladie COVID-19 à partir des publications scientifiques mentionnées ci-dessus[3], les connaissances extraites peuvent être utilisées par la suite de différentes manières. Les graphes de connaissances peuvent être utiles dans plusieurs situations, comme la recherche, la réponse aux questions directes, mais le cas d'utilisation le plus important et qui nous intéresse le plus est la compléction des graphes de connaissances (Knowledge Graph Completion KGC), c'est en principe la prédiction de nouvelles relations entre les nœuds existants, il y en a plusieurs façons dans la littérature pour s'adresser à ce problème, mais nous concentrerons notre recherche sur l'apprentissage de la représentation des connaissances (Knowledge Representation Learning KRL), qui repose principalement sur l'intégration des graphes de connaissances (Knowledge Graph Embeddings KGE).

Dans la suite de ce rapport, qui sert à résumer mes recherches pendant la durée du projet de fin d'études, nous allons présenter les différentes étapes que nous avons suivi, ainsi que les problèmes que nous avons rencontré pendant la réalisation de ce projet. Le plan général de ce rapport est donc le suivant :

Introduction générale

Le premier chapitre : Dans ce chapitre nous allons présenter le traitement du langage naturel et sa relation avec le traitement du texte biomédicale, nous allons donc présenter les notions de base du NLP, et nous allons mettre en évidence des travaux en relation avec le traitement des textes biomédicales.

Le deuxième chapitre : Présentation d'une lecture de la littérature, où nous allons présenter les différentes approches proposées pour l'extraction des entités nommées, ainsi que l'extraction des relations et finalement les méthodes utilisées pour l'apprentissage de la représentation des connaissances.

Le troisième chapitre : Présentation des différentes méthodes utilisées dans la réalisation de ce projet, ainsi que les raisons justifiant le choix de chaque méthode, nous vous présentons aussi le matériel utilisé dans la réalisation et pour l'obtention des différents résultats.

Dernièrement on va finir avec une conclusion, en résumant toutes les étapes que nous avons effectué, ainsi que les perspectives d'avenir de ce projet. Enfin on va discuter quelques réflexions personnelles sur l'utilisation de l'intelligence artificielle dans le domaine biomédicale.

Chapitre 1

Généralités sur le NLP, text mining,
et le corpus biomédical

La littérature biomédicale est l'une des sources majeures de la connaissance biomédicale actuelle, elle est la méthode standard utilisée par les chercheurs pour partager leurs recherches, sous forme d'articles, brevets ou bien d'autres types de rapports écrits. Toutefois, il est indispensable qu'un groupe de chercheurs travaillant sur un sujet donné ait la connaissance sur les recherches publiées par d'autres chercheurs. Cette tâche nécessite un effort énorme, et peut prendre beaucoup de temps, à cause du nombre immense de publications.

Les méthodes automatiques de l'extraction des informations ont le but d'obtenir des informations utiles depuis des textes de grandes tailles. Le vrai défi est de développer des algorithmes qui peuvent être appliqués aux textes non structurés afin d'obtenir des informations structurées. La littérature biomédicale est particulièrement difficile, le style de présentation se diffère entre les différents types de publication : article, brevet ou bien rapport clinique [4]. Il y a aussi différents termes se référant aux gènes, protéines, procédures et techniques, et dans chaque terme, on peut trouver plusieurs orthographes et abréviations[5]. Ces problèmes ont fait de l'application de traitement du langage naturel dans le domaine biomédical un domaine très fertile grâce aux défis qu'ils présente. Les connaissances trouvées dans la littérature biomédicale peuvent être très utiles, soit pour valider des résultats des nouvelles recherches, ou bien pour formuler de nouvelles questions de recherche, qui peuvent être testées expérimentalement. L'un des premier exemple dans ce sens est le travail mené par l'étude [6] qui a découvert que les huiles de poisson alimentaires pouvaient bénéficier aux patients atteints du syndrome de Raynaud, en reliant les informations présentes dans deux groupes d'articles différents qui ne se citaient pas mutuellement. Cette inférence a été confirmée indépendamment dans d'autres essais clinique [7]. Dans la même étude [6], les auteurs ont montré qu'il y en a des inférences qui ne peuvent pas être découvertes sans la combinaison de plusieurs études.

Dans ce qui suit dans ce chapitre, on va présenter quelques notions de base du NLP, ainsi que quelques travaux reliés au domaine biomédical.

1.1 Contexte et fondamentaux

Lors du développement et l'utilisation des outils du text mining, il est nécessaire de premièrement définir le type d'information à extraire. cette décision va par la suite définir l'ensemble de données à utiliser, les tâches de text mining à explorer ainsi que les outils à utiliser.

1.1.1 Sous tâches de NLP

Le traitement automatique du langage naturel a fait l'objet d'une attention particulière de plusieurs chercheurs depuis son apparence dans les années 50. la différence majeure entre le NLP et le "text mining" est l'objectif de tâches. Alors que les techniques NLP visent à donner du sens au texte, déterminer sa structure ou son sentiment, l'objectif des tâches de "text mining" est d'obtenir des connaissances structurées concrètes à partir du texte. Cependant, les deux domaines se chevauchent et les outils de text mining utilisent généralement les concepts et les tâches du NLP.

La liste suivante présente des concepts de NLP qui sont souvent utilisés en text mining :

Token : Une séquence de caractères qui a un sens, nombre, ou symbole. La tâche de reconnaissance des tokens dans un texte est appelée 'tokenization'. Elle a une importance particulière pour le text mining puisque la plupart des algorithmes considère un token l'élément de base d'un texte.

Part-of-speech (POS) : C'est la catégorie lexicale de chaque token, par exemple, nom, adjective, etc... La catégorie confère la sémantique du token. "Part-of-speech tagging" est une tâche de NLP qui consiste à classer les tokens automatiquement.

Lemma et Stem : La forme de base d'un mot. le 'Lemma' représente la forme canonique du mot, qui correspond à un mot réel. Le 'Stem' ne correspond pas toujours à un mot réel, mais à un fragment du mot qui ne change pas.

Sentence splitting : La tâche d'identifier les limites d'une phrase dans un texte. les méthodes utilisées pour accomplir cette tâche doivent différencier entre un point à la fin d'une phrase, et à la fin d'un acronyme ou une abréviation. Il est toujours recommandé de diviser le texte en des phrases, car chaque phrase représente une idée indépendante. Bien que le contexte du texte entier est ainsi important, l'extraction des connaissances de chaque phrase séparément peut présenter des résultats utiles [5].

Entité : Un segment du texte ayant un rapport avec un domaine spécifique. Une entité peut être composée d'un ou plusieurs tokens. Les entités biomédicales peuvent être des gènes, protéines, substances chimiques, des processus biologiques...

Représentation de mot (Word Embedding) : est une méthode utilisée pour représenter les mots en NLP, en tirant parti de vecteurs uniformes de faible dimension, continus et à valeur réelle pour représenter le langage. L'une des formes précédentes est le one-hot, qui présente certains problèmes tels que la rareté des données, l'absence de sens, les désastres dimensionnels. Pour résoudre ces problèmes, certains chercheurs [8] proposent une nouvelle méthode appelée word2vec pour surmonter ces inconvénients. Dans cette méthode, tous les vecteurs de mots sont distribués, les dimensions du vecteur peuvent être arbitraires, généralement entre 50 et 100 dimensions, et la valeur de l'élément peut être n'importe quelle valeur réelle. Le plus grand avantage de cette approche est que les informations sémantiques et contextuelles des mots peuvent être capturées, et que la similarité des mots peut être calculée par simple addition et soustraction. Par conséquent, word2vec est un composant commun dans les ER basés sur les DNN. En dehors de la méthode word2vec, certains chercheurs ont également conçu d'autres méthodes [9].

Chemin de dépendance le plus court (SDP) est une méthode de débruitage au niveau des mots, dérivée d'un arbre de dépendance grammaticale, qui masque les mots non essentiels influençant la relation entre les entités dans une phrase.

1.1.2 Processus de text mining

Les techniques de text mining se concentrent sur une ou plusieurs tâches, ainsi, il est nécessaire de définir ces tâches adéquatement pour savoir le type d'outils à utiliser pour un problème spécifique. Les tâches que nous allons présenter sont commun pour tous les domaines et sources de texte. malgré que la performance des méthodes utilisées sur

différents domaines peut être différente [5]. L'objectif final commun de toutes ces tâches, comme pour tout le domaine de text mining, est d'extraire des connaissances depuis des textes volumineux, ces connaissances peuvent être utiles pour plusieurs applications qu'on va discuter ultérieurement.

Modélisation du thème : C'est la classification des documents selon leurs thèmes ou sujets. L'objectif de cette tâche est d'organiser un ensemble de documents pour identifier lesquels ont un rapport avec un sujet donné [10]. On peut aussi citer d'autres tâches relatives comme le triage de documents [11], et le clustering des documents.

Reconnaissance des entités nommées (NER) : Consiste à identifier les entités mentionnées dans le texte. Dans la plupart des cas, l'emplacement exacte de chaque entité est requis, donné par l'emplacement de ses premier et dernier caractères. Dans des cas, les entités discontinues peuvent être considérées, et par conséquent, requis des emplacements multiples. La classification des entités selon les propriétés (gène, protéine...) peut être incluse dans cette tâche [12].

Normalisation : Consiste à lier les entités à un identifiant unique appartenant à une base de connaissance qui, sans ambiguïté, représente son concept. Par exemple, une protéine peut être mentionnée par son nom complet, ou bien par un acronyme, dans ce cas, la normalisation doit affecter aux deux occurrences le même identifiant. L'identifiant peut être fourni par une base de connaissance, ou une ontologie extérieures [13]. Parmi les sous-tâches relatives à cette tâche on cite la désambiguisation des entités nommées (Named Entity Disambiguation), la liaison des entités (Entity Linking), et l'harmonisation.

L'extraction des relations (RE) : L'identification des entités qui participent dans une relation décrite dans le texte. La plupart des approches considère que les relations entre les entités dans la même phrase. Les relations biomédicales les plus extraites sont les interactions protéine-protéine et médicament-médicament.

L'extraction des événements : Peut être considéré comme une extension de RE, où les relations extraites doivent être étiquetées, et le rôle de chaque entité doit être spécifié. les événements extraits doivent représenter les mécanismes décrit dans le texte [14].

1.1.3 Approches de text mining

Afin d'accomplir les tâches mentionnées précédemment, les techniques de text mining emploient plusieurs approches. Pour une certaine tâche, on peut utiliser une ou plusieurs approches. Les différentes approches peuvent être aussi utilisées pour accomplir différentes tâches.

Approches classiques : Ceux sont des approches basées sur les statistiques qui peuvent être calculées sur un grand corpus de documents [15]. Parmi les approches les plus populaires, on cite "la fréquence du terme", "la fréquence inverse du document" pour la modélisation des sujets, et la "co-occurrence" pour l'extraction des relations. Ces approches sont apparues avant les algorithmes de l'apprentissage automatique, toutefois, la plupart des approches actuelles ont encore un fond statistique.

Approches basées sur les règles : Consiste à définir un ensemble de règles pour

extraire l'information désirée. Ces règles peuvent être des listes de termes, expressions régulières, ou des constructions de phrases. Due à l'effort manuel requis pour développer ces règles, les techniques de text mining basées sur cette approche ont des applications limitées [5].

Algorithmes d'apprentissage automatique (ML) : Utilisés pour apprendre automatiquement des tâches différentes, Dans le cas du text mining, il est nécessaire de convertir les mots en représentations vectorielles qui est l'entrée attendue par ces algorithmes. Les différentes méthodes de ML sont utilisées dans le text mining : apprentissage supervisé, apprentissage non supervisé...[5].

Supervision à distance (DS) : Une méthode d'apprentissage, qui attribue de façon heuristique des étiquettes aux données suivant les informations fournies par une base de connaissances. Ces annotations sont susceptibles d'être erronées, mais en utilisant des algorithmes de ML adaptés à cette méthode, elle peut fournir des modèles de classification pertinents. La DS est parfois référée comme une supervision faible [5].

1.1.4 Corpus biomédical

Le corpus biomédical est nécessaire pour le développement et l'évaluation des techniques du text mining. Un corpus biomédical consiste d'un ensemble de documents associés à un thème spécifique (maladies, gènes...). Pour des tâches, comme la modélisation de thèmes simple, il est suffisant de savoir les documents en question. Toutefois, la plupart des algorithmes de ML nécessitent un texte annoté pour effectuer l'entraînement des modèles. Le type d'annotation nécessaire pour évaluer une tâche doit être similaire au type des annotations à extraire (les tâches NER nécessite du texte annoté avec des entités pertinentes, quant à la RE nécessite que les relations entre les entités soient annotées). Les annotations doivent être consultées manuellement par les experts du domaine pour fournir des recommandations.

Nom	Référence	Annotations	Type documents
CRAFT	[16]	Entité biomédicale	Articles complets
MedTag	[17]	Entité biomédicale	Résumé PubMed
BC5CDR	[18]	Maladies et composants chimique	Articles PubMed
Genia	[19]	Entité biomédicale et événements	Résumé PubMed
CHEMDNER	[20]	Composants chimiques	Résumé PubMed
JNLPBA	[21]	Entité biomédicale	Résumé PubMed
DDI	[22]	Interactions entre médicaments	Description des médicaments
BioNLP13CG	[23]	Entité biomédicale	Résumé PubMed
MLEE	[24]	Événements biologiques	Résumé PubMed

TAB. 1.1 : Corpus biomédicaux reliés aux tâches de biomedical text mining .

La taille du corpus biomédical annotés est relativement limitée à cause de l'effort manuel requis pour annoter les documents. RE nécessite un effort intensif, pour premièrement identifier les entités mentionnées dans le texte, puis extraire les relations entre eux.

Pour cette raison, le développement d'un corpus annoté pour cette tâche est coûteux. Malgré ça, plusieurs corpus standard ont été créés pour réaliser et évaluer cette tâche, et d'autres tâches de text mining. La table 1.1 présente une liste des corpus des données biomédicales.

1.1.5 Conclusion

Dans ce chapitre, nous avons essayé de présenter les concepts (tâches) de base de NLP qui sont indispensables dans la réalisation de n'importe tâche de text mining, nous avons aussi cité les différents problèmes que le text mining vise à résoudre à travers les différentes tâches qui adopte, et enfin nous avons présenté les défis que le traitement de la littérature biomédicale présente pour le NLP et le text mining. Le chapitre suivant sera consacré à une lecture dans la littérature pour découvrir les différentes méthodes et approches utilisées pour réaliser les tâches reliées à la réalisation de notre projet, à savoir, la reconnaissance des entités nommées, la normalisation et l'extraction des relations, et les représentations de connaissances.

Chapitre 2

État de l'art

Dans le chapitre précédent nous avons vu les notions les plus importantes dans le traitement automatique du langage naturel, ainsi que les tâches du text mining et les approches utilisées par ce dernier pour réaliser ces tâches, nous avons aussi parlé un peu du corpus biomédical, dans ce chapitre nous allons présenter une étude non exhaustive de l'état de l'art, nous allons tout d'abord parler de la reconnaissance des entités nommées et les différentes approches utilisées au cours des années, ensuite nous allons présenter les approches utilisées dans la normalisation des entités nommées, ensuite nous allons parler de l'extraction des relations et les différentes méthodes utilisées, et en fin nous allons présenter l'apprentissage des présentations.

2.1 Reconnaissance d'entités nommées

Une entité nommée est un mot, ou phrase qui identifie clairement un élément parmi un ensemble d'autres éléments ayant des attributs similaires [25]. Parmi les exemples des entités nommées on peut citer "organisation", "personne",... et dans le domaine biomédical on peut citer "maladie", "protéine", "gène"... La reconnaissance des entités nommées est le processus de localisation et de regroupement de ces entités dans un texte. Le NER est une étape de pré-traitement très importante pour plusieurs tâches en aval, comme la récupération des informations, réponse aux questions. En ce qui suit, nous allons présenter les différentes approches proposées pour la réalisation de cette tâche, ces approches sont divisées en deux grandes catégories ; Les approches traditionnelles et les approches basées sur les DNNs.

2.1.1 Approches traditionnelles

Les approches traditionnelles du NER sont classées en trois grandes catégories : les approches basées sur les règles, l'apprentissage non supervisé et l'apprentissage supervisé basé sur les caractéristiques.

Approches basées sur les règles

Les systèmes NER basés sur les règles reposent sur des règles générées manuellement. Les règles peuvent être conçues sur la base de nomenclatures spécifiques à un domaine [26], et les modèles syntaxiques-lexicaux [27]. L'étude [28] a proposé d'utiliser la règle d'inférence de Brill. Ce système génère automatiquement les règles basées sur l'étiqueteur de parties de la parole de Brill. Dans le domaine biomédical, on trouve ProMiner [29], qui s'appuie sur un dictionnaire de synonymes pré-traité pour identifier les mentions de protéines et de gènes potentiels dans des textes biomédicaux. L'étude [30] a proposé une approche basée sur les dictionnaires pour le NER dans les dossiers médicaux électroniques. Les résultats expérimentaux montrent que l'approche améliore le rappel tout en ayant un impact limité sur la précision.

D'autres modèles NER à base des règles bien connus sont : NetOwl, Facile, SAR, Ces systèmes sont basés sur des règles sémantiques et lexiques élaborées manuellement pour reconnaître les entités.

Approches d'apprentissage non supervisées

Une approche typique de l'apprentissage non supervisé est le clustering [12]. Un système NER basé sur le clustering extrait les entités en se basant sur les regroupements basés sur la similarité des contextes. L'idée principale ici est que les ressources lexicales, les motifs lexicaux, et les statistiques calculées sur un corpus large peuvent être utilisées pour inférer les mentions des entités nommées. L'étude [31] a observé que l'utilisation des données non étiquetées réduit la nécessité de supervision à 7 règles simples. Par conséquence, les auteurs ont proposé deux algorithmes non supervisés pour la reconnaissance des entités nommées. D'une manière similaire, KNOWITALL [32] a exploité un ensemble de noms de prédictifs en tant qu'entrée et démarre son processus de reconnaissance à partir d'un petit ensemble de modèles d'extraction génériques.

Approches d'apprentissage supervisées basées sur les caractéristiques

En appliquant l'apprentissage supervisé, la NER est coulée dans une tâche de classification multi-classes ou d'étiquetage de séquences. Les échantillons de données annotées et les caractéristiques sont soigneusement conçues pour représenter chaque exemple de l'entraînement. Les algorithmes d'apprentissage automatique sont ensuite utilisés pour entraîner des modèles à reconnaître les motifs similaires à partir de nouvelles données. L'ingénierie des caractéristiques est très importante pour les systèmes de NER supervisés. Le vecteur de représentation d'une caractéristique est une abstraction sur le texte, où un mot est représenté par une, ou plusieurs, valeurs booléennes ou nominales [12]. Les caractéristiques au niveau du mots (la morphologie, POS...) [33], "lookup features", et les caractéristiques du documents et du corpus sont tous utilisées dans les systèmes de NER supervisés. En se basant sur ces caractéristiques, plusieurs algorithmes d'apprentissage automatique ont été appliqués sur la NER, y compris HMM "Hidden Markov Models" [34], "Decision Trees" [35], "Maximum Entropy Models", SVM "Support Vector Machines" [36], et les CRF "Conditional Random Fields" [37].

2.1.2 Techniques d'apprentissage profond

Dans les années récentes, les modèles de NER basés sur l'apprentissage profond sont devenus dominant. En les comparant aux modèles basés sur les caractéristiques, l'apprentissage profond est montré d'être bénéfique pour la reconnaissance des entités nommées [25].

L'apprentissage profond est une sorte d'apprentissage automatique composée de plusieurs couches de traitement pour apprendre les représentations des données avec multiples niveaux d'abstraction. L'avantage de ces modèles est la capacité d'apprendre les représentations et la composition sémantique rendu possible par la représentation vectorielle et le traitement neuronal.

En ce qui suit nous allons présenter les différentes approches basées sur l'apprentissage profond proposé pour la NER.

Représentation distribuée des entrées

Une option simple pour représenter un mot est la représentation vectorielle à un coup. Dans un espace vectoriel à un coup, deux mots distincts ont des représentations complètement différentes et sont orthogonaux. La représentation distribuée représente les mots dans des vecteurs denses à faible dimension et à valeur réelle, où chaque dimension représente une caractéristique latente. Apprise automatiquement à partir du texte, la représentation distribuée capture les propriétés sémantiques et syntaxiques du mot, qui ne sont pas explicitement présentes dans le texte.

Représentation au niveau des mots

quelques études emploient la représentation au niveau des mots, ces représentations sont en général pré-entraînées sur des collections de texte larges à travers des modèles non supervisés comme le Continuous bag-of-words ou bien le continuous skip-gram [8]. Les représentations les plus utilisées comprises Google Word2vec, Stanford GloVe, Facebook fastText et SENNA.

Bio-NER [38] est un modèle de NER biomédicales basé sur les DNNs où les représentations des mots utilisées sont entraînées sur la base de données PubMed en utilisant le modèle skip-gram. Le dictionnaire contient 205,924 mots avec des vecteurs de dimension 600. D'autre part, [39] utilise Word2vec pour apprendre les représentations des mots anglais à partir le corpus de Gigaword augmenté par les données de BOLT. Une autre étude [40] décrit un modèle qui consiste de deux sous tâches : la segmentation et l'étiquetage. Le modèle peut prendre comme entrée les représentations de SENNA ou bien des représentation initialisés de façon aléatoire.

Représentation au niveau des caractères

A la place de considérer seulement les représentations au niveau des mots comme entrée, plusieurs études [41], [42] introduisent des représentations basées sur les caractères appris par un modèle neuronal de bout en bout. La représentation basée caractère a été trouvée utile pour exploiter les informations au niveau des sous-mots, comme les préfixes et les suffixes. Un autre avantage de cette méthode c'est qu'elle gère d'une manière automatique les mots non présents dans le vocabulaire. Par conséquent, les modèles basés sur les représentations au niveau des caractères peuvent inférer des représentations pour les nouveaux mots. Il existe deux architectures pour extraire les représentations des caractères : les CNNs et les RNNs, la figure 2.1 illustre les deux architectures.

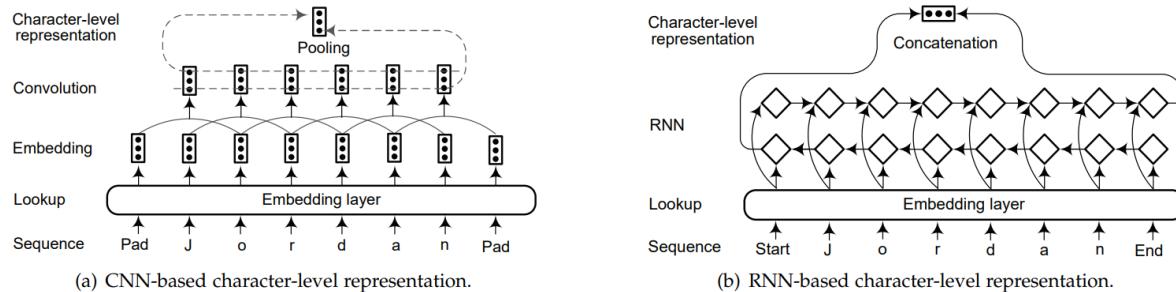


FIG. 2.1 : Modèles basés sur CNN et RNN pour l'extraction de la représentation au niveau des caractères d'un mot.

Représentations hybrides

En plus des représentations au niveau des mots et des caractères, quelques études introduits des informations additionnelles (par exemple les similarités lexicales [43], dépendances linguistiques [44] et les caractéristiques visuelles [45]) aux représentations des mots finales, avant de les alimenter aux couches d'encodage du contexte. En d'autres mots, les représentations basées sur l'apprentissage profond sont combinées avec les approches basées sur les caractéristiques d'une manière hybride. L'ajout des informations additionnelles peut présenter des améliorations aux performances des modèles de NER. L'utilisation des modèles de NER a été lancée par [46], où ils ont proposé une architecture basée sur les réseaux neuronaux temporels appliqués sur des séquences de mots. Avec l'incorporation des connaissances priori commun, le modèle surpassé les performances du modèle de base. Le modèle BiLSTM-CRF [47] utilise quatre types de caractéristiques pour la NER : les caractéristiques d'orthographe, les caractéristiques de contexte, les représentations des mots et les répertoires gazométriques. les résultats expérimentales montre que ces caractéristiques augmentent la précision d'étiquetage. Le modèle BiLSTM-CNN [48] propose un BiLSTM et un CNN au niveau des caractères. En plus des représentations au niveau des mots, le modèle utilise des caractéristiques au niveau des mots (majuscules, lexiques) et les caractéristiques au niveau des caractères.

Architectures des encodeurs de contexte

Dans cette partie nous allons présenter les différentes architectures proposées pour encoder le contexte, à savoir les Réseaux neuronaux convolutifs, les réseaux neuronaux récurrents, les réseaux neuronaux récursifs et les transformateurs profonds.

Réseaux neuronaux convolutifs

L'une des premières études dans ce sens [46] propose une architecture où les mots sont étiquetés en prenant en considération toute la phrase, comme montré dans la figure 2.2. Chaque mot dans la séquence d'entrée est représentée par un vecteur de dimension N lors de l'étape de représentation des entrées. Ensuite, une couche de convolution est utilisée pour la production des caractéristiques locales autour de chaque mot, la taille de la sortie de la couche convolution dépend du nombre de mots dans la phrase. Le vecteur des caractéristiques globales est construit par la combinaison des vecteurs des caractéristiques locales extraits par la couche de convolution. La dimension du vecteur global est fixée, et indépendante de la taille de la phrase. Deux approches sont utilisées pour extraire les caractéristiques globales : le max ou la moyenne des opérations de position dans la phrase. Finalement, ces vecteurs de caractéristiques globales alimentent un décodeur d'étiquetage pour calculer la distribution des scores de toutes les étiquettes possibles des mots de l'entrée du réseau. En suivant la même approche, l'étude [26] propose un modèle pour la reconnaissance des entités nommées biomédicales et l'étude [49] utilise des CNNs pour générer des caractéristiques globales pour les noeuds cachés.

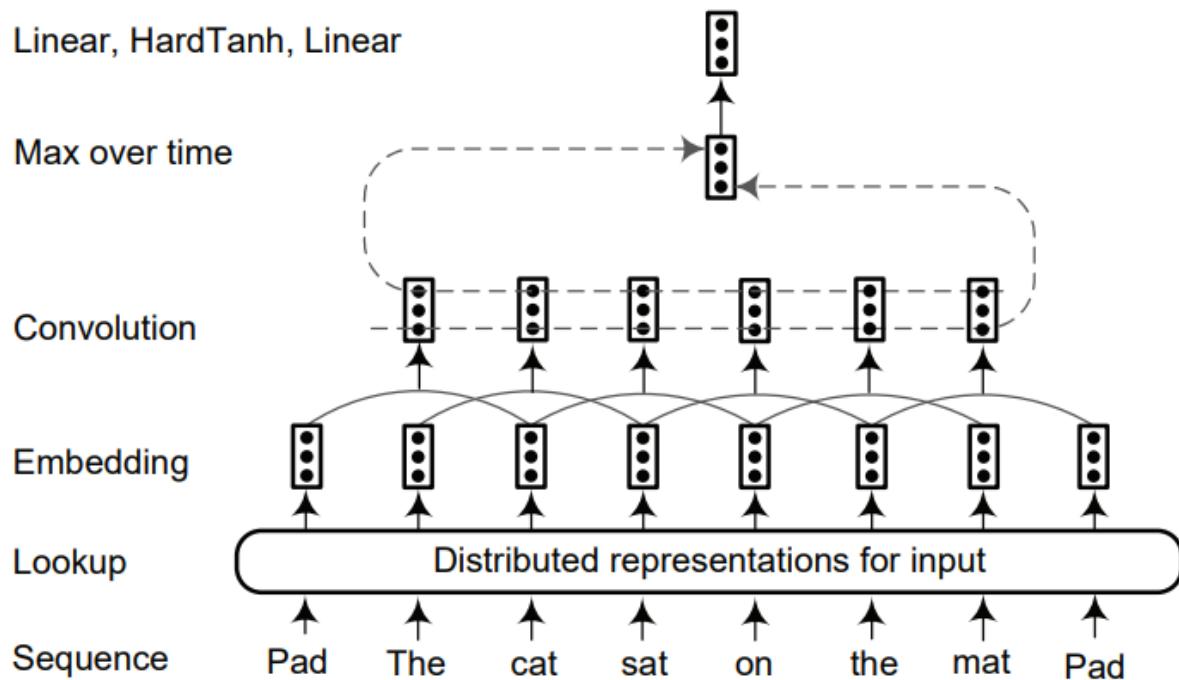


FIG. 2.2 : Réseau d'approche des phrases basé sur CNN. La couche de convolution extrait les caractéristiques de la phrase entière, en la traitant comme une séquence avec une structure globale.

Réseaux neuronaux récurrents

Les réseaux neuronaux récurrents, et ses variations comme les "Gated Recurrent Units" et les BiLSTM, ont démontré des performances remarquables dans la modélisation des données séquentielles. Plus particulièrement, les RNNs bi-directionnels utilisent efficacement les informations passées (par le feed-forward) et les données futur (par le feed-backward) [47]. Ainsi, un token encodé par un RNN bi-directionnel va contenir des preuves de la phrase d'entrée entière. Les RNNs bi-directionnel sont devenus par conséquence la norme pour construire des représentations profondes du texte en fonction du contexte [50]. Une architecture RNN typique des encodeurs basées sur le contexte est présentée dans la figure 2.3.

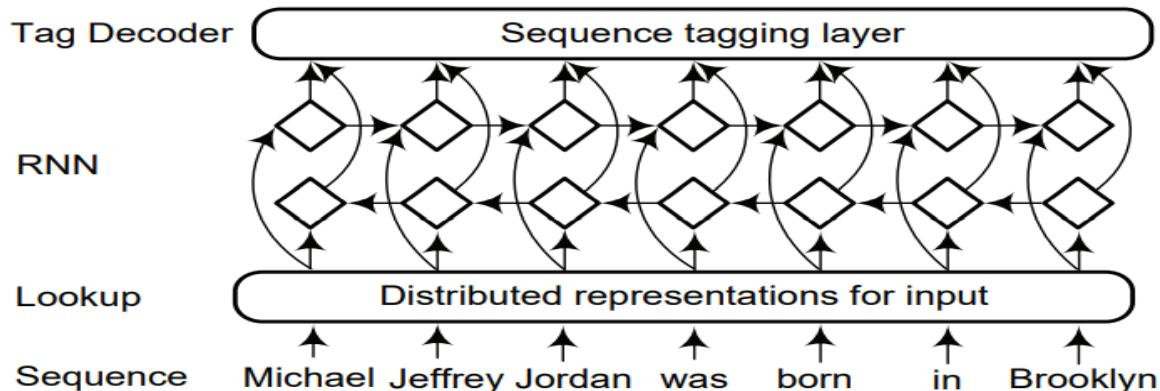


FIG. 2.3 : L'architecture de l'encodeur de contexte basée sur RNN.

Réseaux neuronaux récursifs

Les réseaux neuronaux récursifs sont des modèles adaptatifs non linéaires capables d'apprendre des informations structurées profondes, en parcourant une structure donnée dans un ordre topologique. Les entités nommées sont reliées aux composantes linguistiques. Cependant, les approches d'étiquetage séquentiel typiques ne tiennent pas compte de la structure des phrases. Dans ce but, [51] propose de classer chaque nœud dans une structure de circonscription pour la NER. Ce modèle calcule récursivement les vecteurs d'état caché de chaque nœud et classe chaque nœud en fonction de ces vecteurs cachés. La figure 2.4 montre comment calculer récursivement deux caractéristiques d'état caché pour chaque nœud. La direction ascendante calcule la composition sémantique du sous-arbre de chaque noeud, et la contrepartie descendante propage à ce nœud les structures linguistiques qui contiennent le sous-arbre. Étant donné les vecteurs cachés pour chaque nœud, le réseau calcule une distribution de probabilités des types d'entités plus un type spécial de non-entité.

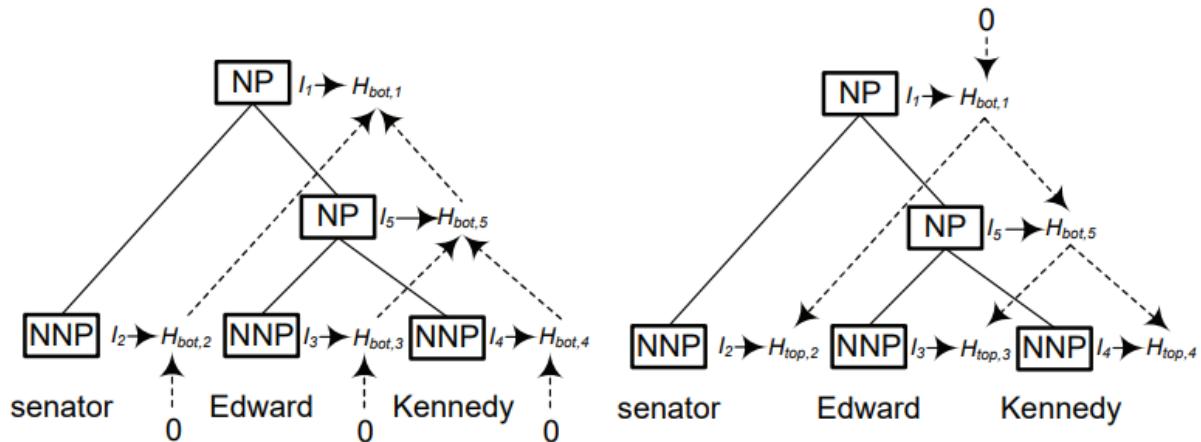


FIG. 2.4 : Réseaux neuronaux récursifs bidirectionnels pour la NER

Modèles neuronaux de langage

Les modèles de langage sont une famille de modèles décrivant la génération de séquences. Étant donnée une séquence de tokens, (t_1, t_2, \dots, t_N) , un modèle de langage prospectif calcule la probabilité de la séquence en modélisant la probabilité du token t_k compte tenu de son historique (t_1, t_2, \dots, t_k) [52] :

$$p(t_1, t_2, \dots, t_N) = \prod_{k=1}^N p(t_k | t_1, t_2, \dots, t_{k-1})$$

Un modèle de langage rétrograde est similaire à un modèle de langage direct, sauf qu'il parcourt la séquence dans l'ordre inverse, en prédisant le token précédent, compte tenu de son contexte futur :

$$p(t_1, t_2, \dots, t_N) = \prod_{k=1}^N p(t_k | t_{k+1}, t_{k+2}, \dots, t_N)$$

Transformateurs profonds

Les modèles neuronaux d'étiquetage des séquences sont généralement basés sur des réseaux complexes convolutifs ou récurrents qui se composent d'encodeurs et de décodeurs. Le "Transformer" proposé par [53] utilise l'auto-attention empilée et des couches ponctuelles et entièrement connectées pour construire les blocs de base du codeur et du décodeur. Les expériences menées sur diverses tâches montrent que les transformateurs sont d'une qualité supérieure tout en nécessitant beaucoup moins de temps d'entraînement. En se basant sur les Transformers, [54] propose Generative Pre-trained Transformer (GPT) pour les tâches de compréhension du langage naturel. GPT a une procédure d'entraînement en deux étapes. D'abord, ils utilisent un objectif de modélisation du langage avec Transformers sur des données non étiquetées pour apprendre les paramètres initiaux. Ensuite, ils adaptent ces paramètres à une tâche cible en utilisant l'objectif supervisé, ce qui entraîne des modifications minimales du modèle pré-entraîné. Contrairement à GPT (une architecture de gauche à droite), Bidirectional Encoder Representations from Transformers, (BERT) [55] est proposé pour pré-entraîner le Transformer bidirectionnel profond en conditionnant conjointement les contextes gauches et droits dans toutes les couches. la figure 2.5 représente les trois architectures de BERT, GPT et ELMo.

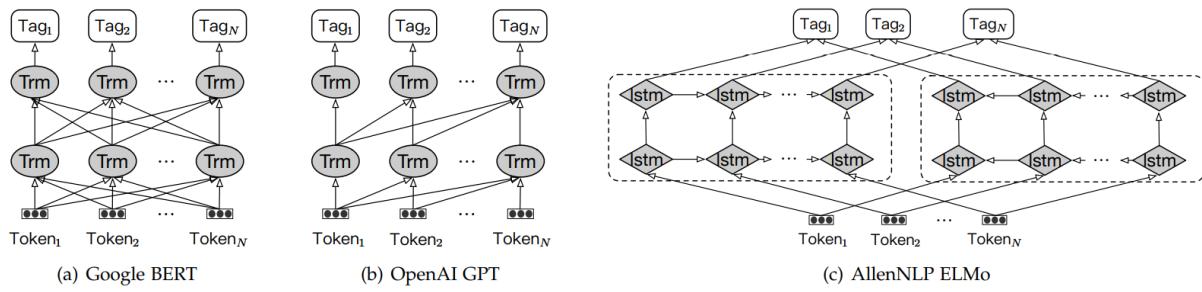


FIG. 2.5 : Architectures basées sur les Transformateurs pour la NER

Ces incorporations de modèles de langue pré-entraînés à l'aide de Transformer sont en train de devenir un nouveau paradigme de la NER. Premièrement, ces incorporations sont contextualisées et peuvent être utilisées pour remplacer les incorporations traditionnelles, telles que Google Word2vec et Stanford GloVe.

2.2 Normalisation des entités nommées

La tâche de liaison des entités est difficile en raison des variations de noms et de l'ambiguïté des entités. Une entité nommée peut avoir plusieurs formes de surface, comme son nom complet, des noms partiels, des alias, des abréviations et des orthographes alternatives. Un système de liaison d'entités doit identifier les entités de correspondance correctes pour les mentions d'entités de diverses formes de surface. D'autre part, une mention d'entité peut éventuellement désigner différentes entités nommées. Par conséquence, le système de liaison des entités doit désambiguier la mention de l'entité dans le contexte textuel et identifier l'entité de correspondance pour chaque mention d'entité.

Une base de connaissances est un composant fondamental pour la tâche de liaison d'entités. Les bases de connaissances fournissent des informations sur les entités du monde

(par exemple, les entités d'Albert Einstein et d'Ulm), leurs catégories sémantiques (par exemple, Albert Einstein a un type de scientifique et Ulm un type de ville), et les relations mutuelles entre les entités (par exemple, Albert Einstein a une relation appelée bornIn avec Ulm).

2.2.1 Bases de connaissances utilisées pour la liaison des entités nommées biomédicales

Dans cette section, nous fournissons une brève introduction à quatre bases de connaissances qui ont été largement exploitées dans le domaine de la liaison d'entités biomédicales.

Unified Medical Language System

L'UMLS [56] est un ensemble de fichiers et de logiciels qui rassemble de nombreux vocabulaires et normes sanitaires et biomédicaux afin de permettre l'interopérabilité entre les systèmes informatiques.

L'UMLS utilise trois sources de connaissances :

- Metathesaurus : Termes et codes issus de nombreux vocabulaires, notamment CPT, ICD-10-CM, LOINC, MeSH, RxNorm et SNOMED CT. Hiérarchies, définitions et autres relations et attributs.
- Semantic Network : Les grandes catégories (types sémantiques) et leurs relations (relations sémantiques)
- SPECIALIST Lexicon and Lexical Tools : Un vaste lexique syntaxique d'anglais biomédical et général et des outils pour normaliser les chaînes de caractères, générer des variantes lexicales et créer des index.

Medical Subject Headings

Le thésaurus MeSH [57] est un vocabulaire contrôlé et organisé de manière hiérarchique produit par la National Library of Medicine. Il est utilisé pour l'indexation, le catalogage et la recherche d'informations biomédicales et liées à la santé. MeSH comprend les vedettes-matières apparaissant dans MEDLINE/PubMed, le NLM Catalog et d'autres bases de données de la NLM.

RxNorm

RxNorm [58] fournit des noms normalisés pour les médicaments cliniques et relie ses noms à de nombreux vocabulaires de médicaments couramment utilisés dans les logiciels de gestion des pharmacies et des interactions médicamenteuses, notamment ceux de First Databank, Micromedex et Gold Standard Drug Database. En fournissant des liens entre ces vocabulaires, RxNorm peut servir de médiateur pour les messages entre les systèmes qui n'utilisent pas le même logiciel et le même vocabulaire.

RxNorm inclut désormais la nomenclature officinale de la United States Pharmacopeia (USP) de la United States Pharmacopeial Convention. L'USP est un ensemble de données cumulatives de tous les ingrédients pharmaceutiques actifs (API).

Gene Ontology

La base de connaissances Gene Ontology (GO) [59] est la plus grande source d'informations sur les gènes et les processus biologiques.

mations au monde sur les fonctions des gènes. Ces connaissances sont à la fois lisibles par l'homme et par la machine, et constituent une base pour l'analyse informatique des expériences de biologie moléculaire et de génétique à grande échelle dans la recherche biomédicale. L'ontologie des gènes (GO) décrit notre connaissance du domaine biologique sous trois aspects :

- Fonction Moléculaire : Activités au niveau moléculaire réalisées par les produits des gènes. Les termes de fonction moléculaire décrivent des activités qui se produisent au niveau moléculaire, telles que la "catalyse" ou le "transport". Les termes de fonction moléculaire GO représentent les activités plutôt que les entités (molécules ou complexes) qui effectuent les actions, et ne précisent pas où, quand ou dans quel contexte l'action a lieu. Les fonctions moléculaires correspondent généralement à des activités qui peuvent être réalisées par des produits génétiques individuels (c'est-à-dire une protéine ou un ARN), mais certaines activités sont réalisées par des complexes moléculaires composés de plusieurs produits génétiques. Des exemples de termes fonctionnels larges sont l'activité catalytique et l'activité de transporteur ; des exemples de termes fonctionnels plus étroits sont l'activité de l'adénylate cyclase ou la liaison des récepteurs de type Toll. Pour éviter toute confusion entre les noms des produits génétiques et leurs fonctions moléculaires, les fonctions moléculaires GO sont souvent accompagnées du mot "activité" (une protéine kinase aurait la fonction moléculaire GO "activité de la protéine kinase").
- Composant Cellulaire : Les emplacements relatifs aux structures cellulaires dans lesquelles un produit génique remplit une fonction, soit des compartiments cellulaires (par exemple, la mitochondrie), soit des complexes macromoléculaires stables dont ils font partie (par exemple, le ribosome). Contrairement aux autres aspects de GO, les classes de composants cellulaires ne font pas référence à des processus mais plutôt à une anatomie cellulaire.
- Processus Biologique : Les processus plus larges, ou "programmes biologiques", accomplis par de multiples activités moléculaires. Des exemples de termes généraux de processus biologiques sont la réparation de l'ADN ou la transduction du signal. Des exemples de termes plus spécifiques sont le processus de biosynthèse de la pyrimidine nucléobase ou le transport transmembranaire du glucose. Notez qu'un processus biologique n'est pas équivalent à une voie. Actuellement, le GO n'essaie pas de représenter la dynamique ou les dépendances qui seraient nécessaires pour décrire complètement une voie.

Human Phenotype Ontology L'ontologie du phénotype humain (HPO) [60] fournit un vocabulaire normalisé des anomalies phénotypiques rencontrées dans les maladies humaines. Chaque terme de l'HPO décrit une anomalie phénotypique, telle que la communication interauriculaire. Le HPO est actuellement développé à partir de la littérature médicale, d'Orphanet, de DECIPHER et d'OMIM. HPO contient actuellement plus de 13 000 termes et plus de 156 000 annotations sur les maladies héréditaires. Le projet HPO et d'autres ont développé des logiciels pour les diagnostics différentiels basés sur le phénotype, les diagnostics génomiques et la recherche translationnelle. Le HPO est un produit phare de l'initiative Monarch, un consortium international soutenu par le NIH,

qui se consacre à l'intégration sémantique de données biomédicales et d'organismes modèles dans le but ultime d'améliorer la recherche biomédicale. Le HPO, en tant que partie intégrante de l'initiative Monarch, est un élément central de l'un des 13 projets moteurs de la feuille de route stratégique de l'Alliance mondiale pour la génomique et la santé (GA4GH).

2.2.2 Génération d'entités candidates

Dans cette approche, pour chaque mention d'entité $m \in M$, le système essaie d'inclure les entités possibles auxquelles la mention d'entité m peut faire référence dans l'ensemble des entités candidates E_m . Les approches de génération d'entités candidates sont principalement basées sur la comparaison de chaînes de caractères entre la forme de surface de la mention d'entité et le nom de l'entité existant dans une base de connaissances. Ce module est aussi important que le module de classement des entités candidates et il est essentiel pour la réussite d'un système de liaison d'entités, selon les expériences menées par [61]. Dans le reste de cette section, nous passons en revue les principales approches qui ont été appliquées pour générer l'ensemble d'entités candidates E_m pour la mention d'entité m .

Techniques basées sur le dictionnaire des noms

Les techniques basées sur les dictionnaires de noms sont les principales approches de la génération d'entités candidates et sont utilisées par de nombreux systèmes de liaison d'entités. Par exemple, la structure de Wikipédia fournit un ensemble de caractéristiques utiles pour générer des entités candidates, comme les pages d'entités, les pages de redirection, les pages de désambiguïsation, les phrases en gras des premiers paragraphes et les liens hypertextes dans les articles de Wikipédia. Ces systèmes de liaison d'entités exploitent différentes combinaisons de ces caractéristiques pour construire un dictionnaire de noms hors ligne D entre divers noms et leurs entités de correspondance possibles, et exploitent ce dictionnaire de noms construit D pour générer des entités candidates. Ce dictionnaire de noms D contient une grande quantité d'informations sur les différents noms des entités nommées, comme les variations de noms, les abréviations, les noms pouvant être confondus, les variations orthographiques, les surnoms, etc.

En utilisant ces caractéristiques, un système de liaison des entités peut construire un dictionnaire de noms D , qui est en principe un mapping $\langle key, value \rangle$, où la colonne key est une liste de noms. Supposons que k est un nom dans la colonne key , et que sa valeur de mapping est $k.value$ dans la colonne $value$, est une en fait un ensemble d'entités nommées auxquelles on pourrait se référer sous le nom k .

En se basant sur le dictionnaire créé, l'approche la plus simple de génération de l'ensemble d'entités candidates E_m pour la mention d'entité $m \in M$ est la correspondance exacte entre le nom k dans la colonne clé et la mention d'entité m . Si un certain k est égal à m , l'ensemble d'entités $k.value$ est ajouté à l'ensemble d'entités candidates E_m .

Extension de la forme de surface à partir du document local

Puisque certaines mentions d'entités sont des acronymes ou une partie de leur nom complet, une catégorie de systèmes de liaison d'entités utilise les techniques d'expansion de formes de surface pour identifier d'autres variations étendues possibles (comme le nom complet) à partir du document associé où la mention de l'entité apparaît. Ils peuvent ensuite exploiter ces formes développées pour générer l'ensemble des entités candidates en utilisant d'autres méthodes, telles que les techniques basées sur le dictionnaire de noms présentées ci-dessus. Nous classons les techniques d'expansion des formes de surface en deux catégories : les méthodes heuristiques et les méthodes d'apprentissage supervisé.

Méthodes heuristiques

Pour la mention de l'entité sous la forme de son acronyme, quelques études [62] l'étend en recherchant le contexte textuel autour de la mention de l'entité par le biais de la correspondance heuristique des motifs. Les modèles les plus courants qu'ils exploitent sont un acronyme entre parenthèses adjacent à l'expansion et une expansion entre parenthèses adjacente à l'acronyme.

Méthodes d'apprentissage supervisé

Les méthodes précédentes basées sur l'heuristique pour l'expansion de la forme de surface ne pouvaient pas identifier la forme expansée pour certains acronymes compliqués, tels que les lettres d'acronymes échangées ou manquantes. L'étude [63] propose un algorithme d'apprentissage supervisé pour trouver les formes développées d'acronymes complexes, ce qui a permis d'améliorer la précision de 15,1% par rapport aux méthodes de développement d'acronymes les plus récentes. Plus précisément, ils ont identifié les expansions candidates possibles à partir du document par le biais de certaines stratégies prédéfinies, notamment les marqueurs de texte et la correspondance de la première lettre (c'est-à-dire que toutes les séquences de mots du document qui commencent par la même première lettre que l'acronyme et qui ne contiennent pas de ponctuation ou plus de deux mots d'arrêt sont extraites comme expansions candidates).

Méthodes basées sur les moteurs de recherche

Quelques systèmes de liaison d'entités nommées [64] tentent d'exploiter l'ensemble des informations du Web pour identifier les entités candidates via les moteurs de recherche du Web (tels que Google). Plus précisément, [65] ont soumis la mention de l'entité ainsi que son contexte court à l'API de Google et n'ont obtenu que des pages Web dans Wikipédia pour les considérer comme des entités candidates. [64] ont interrogé le moteur de recherche Google en utilisant la mention de l'entité et ont identifié les entités candidates dont les pages Wikipédia apparaissent dans les 20 premiers résultats de recherche Google pour la requête. Lehmann et al. [66] ont déclaré que le moteur de recherche Google est très efficace pour identifier certaines correspondances très difficiles entre les formes de surface et les entités. Ils ont effectué la requête en utilisant l'API de Google limitée au site Wikipedia anglais et ont filtré les résultats dont les titres Wikipedia ne sont pas significativement similaires à la requête en termes de dés ou d'acronymes. Enfin, ils ont utilisé les trois premiers résultats comme entités candidates. En outre, le moteur de recherche de Wikipedia est également exploité pour récupérer les entités candidates qui peuvent

renvoyer une liste de pages d'entités Wikipedia pertinentes lorsque vous l'interrogez sur la base de la correspondance des mots clés. [67] ont utilisé cette fonctionnalité pour générer des entités candidates rarement mentionnées en interrogeant ce moteur de recherche à l'aide de la chaîne de caractères de la mention de l'entité.

2.2.3 Classement des entités candidates

Dans la section précédente, nous avons décrit les méthodes permettant de générer l'ensemble d'entités candidates E_m pour chaque mention d'entité m . On note la taille de E_m par $|E_m|$, et on utilise $1 < i < |E_m|$ pour indexer l'entité candidate dans E_m . L'entité candidate avec l'index i dans E_m est notée par e_i . Dans la plupart des cas, la taille de l'ensemble des entités candidates E_m est supérieure à un. Par exemple, [68] ont montré que le nombre moyen d'entités candidates par mention d'entité sur l'ensemble de données TAC-KBP2010 est de 12,9, et ce nombre moyen sur l'ensemble de données TAC-KBP2011 est de 13,1. En outre, ce nombre moyen est de 73 sur l'ensemble de données CoNLL utilisé dans [69]. Par conséquent, le problème restant est de savoir comment incorporer différents types de preuves pour classer les entités candidates dans E_m et choisir l'entité appropriée dans E_m comme entité de correspondance pour la mention d'entité m . Le module de classement des entités candidates est un élément clé du système de liaison d'entités. Nous pouvons diviser ces méthodes de classement des entités candidates en deux catégories :

Méthodes de classement supervisé

Ces approches s'appuient sur des données de formation annotées pour "apprendre" à classer les entités candidates dans E_m . Ces approches comprennent les méthodes de classification binaire, les méthodes d'apprentissage du classement, les méthodes probabilistes et les approches basées sur les graphes.

Méthodes de classement non supervisées

Ces approches sont basées sur des corpus non étiquetés et ne nécessitent pas de corpus annoté manuellement pour entraîner le modèle. Ces approches comprennent les méthodes basées sur le modèle SVM et les méthodes basées sur la recherche d'information.

2.3 Extraction des relations

L'extraction des relations est la tâche de la prévision des relations entre les entités dans une phrase. Par exemple dans la phrase « Obama was born in Honolulu, Hawaii » un classeur de relations est censé de prédire la relation "BornInCity". L'extraction des relations est l'élément clef de la construction des graphes de connaissances, elle est aussi d'une grande importance pour d'autres applications du traitement du langage naturel, comme la recherche structurée, l'analyse des sentiments... Dans cette section on va présenter les différentes approches adoptées pour la réalisation de cette tâche. ces approches peuvent être catégorisées en deux catégories, les approches traditionnelles et les approches basées sur les réseaux de neurones profonds.

2.3.1 Méthodes traditionnelles

Ingénierie manuelle des motifs (Hand-built pattern methods)

Nécessite une intervention des spécialistes de la langue pour construire un ensemble de connaissances des motifs basé sur les mots, des parties de discours (POS), ou la sémantique. Avec ces connaissances linguistiques, et les connaissances du domaine professionnelles, l'extraction des relations peut être réalisé par la correspondance entre le texte et les motifs. S'ils se correspondent, la phrase peut contenir le motif correspondant [70]. La table 2.1 donne une exemple de ce motif.

Pattern	such X as Y
Corpus	... works by such authors as Herrick, Goldsmith, and Shakespeare.
Relation	Hyponym ("author", "Herrick"), Hyponym ("author", "Goldsmith"), Hyponym ("author", "Shakespeare")

TAB. 2.1 : Exemple du motif "such X as Y".

Méthodes semi-supervisées (semi-supervised methods)

Elles sont des méthodes basées sur les motifs, La méthode typique est un algorithme de bootstrapping. Le modèle représentatif de cette méthode est DIPRE (Dual Iterative Pattern Relation Expansion) [71], l'idée est de trouver quelques tuples avec une confiance élevée, l'algorithme de bootstrapping extrait des motifs avec les tuples. Ces motifs peuvent être utilisés ensuite pour extraire de nouveaux triples. Cette méthode est représentée dans la figure 2.6.

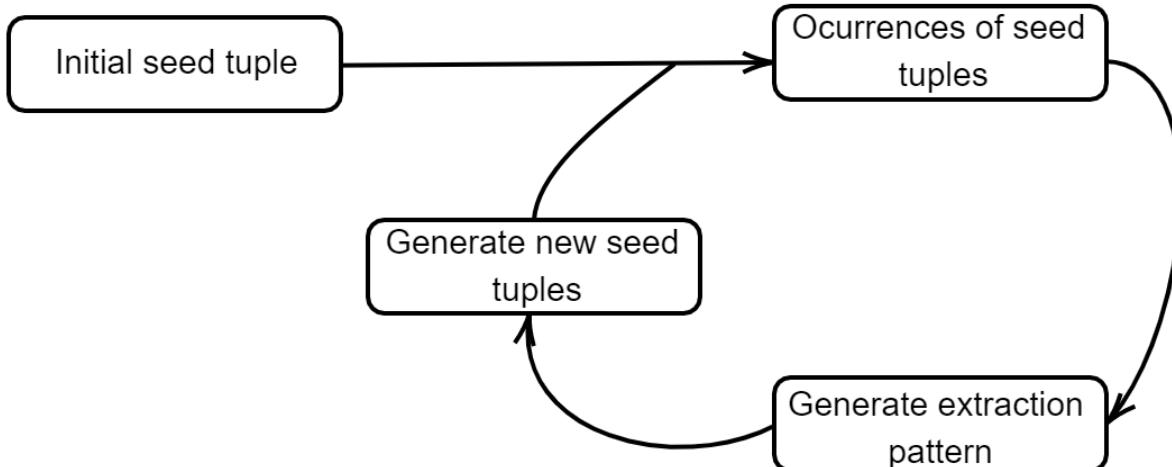


FIG. 2.6 : L'idée principale de DIPRE.

Méthodes non-supervisées (unsupervised methods)

Adopte une approche ascendante basée sur l'hypothèse : Les informations contextuelles de différentes paires d'entités ayant la même relation sémantique sont relativement similaire. Une approche est proposé dans [72], ce processus d'extraction peut être divisé en 3 étapes :

- Extraire un pair d'entités et son contexte
- Regrouper les pairs d'entités selon leurs contextes
- Annoter la relation sémantique de chaque groupe ou bien décrire leurs type de relations

Méthodes supervisées (supervised methods)

Considère l'extraction de relations comme un problème de classification multi-classes. Les approches utilisées sont classées en 2 catégories [73] :

Méthodes basées sur les caractéristiques (feature-based) Chaque instance de relation dans les données étiquetées est utilisée pour entraîner un classificateur alimenté par de nouvelles instances pour la classification. Généralement, ces caractéristiques proviennent d'informations utiles (notamment lexicales, syntaxiques, sémantiques) extraites d'un contexte d'une instance. Sans une sélection des caractéristiques, il est difficile d'améliorer les performances d'une méthode basée sur les caractéristiques.

Méthodes basées sur les noyaux (kernel-based) Nécessitent rarement des étapes de pré-traitement linguistique explicites, mais elles dépendent davantage de la performance de la fonction du noyau conçue. Cette dernière est utilisée pour calculer les similarités entre les représentations des relations, ensuite le SVM est utilisé pour la classification.

Méthodes de supervision à distance (distant supervised methods)

Sont des méthodes basées sur les connaissances ou faiblement supervisée [74], elles sont basées sur l'hypothèse suivantes : "si deux entités participent dans une relation, alors tous les phrases contenant ces entités expriment la même relation". En principe ces méthodes utilisent des bases de connaissances (KB) comme une source de supervision. Quand une phrase et la KB mentionnent le même paire d'entités, la phrase est étiquetée par la relation correspondante. la figure 2.7 montre le processus de cette méthode, la partie supérieure gauche de la figure est la base de connaissances, et la partie inférieure gauche est la source du corpus. Après le processus d'alignement du texte au milieu, le côté droit produit les paquets correspondant à la base de connaissances pour représenter diverses relations, chaque paquet représente une étiquette de relation et ces paquets contiennent plusieurs instances de phrases.

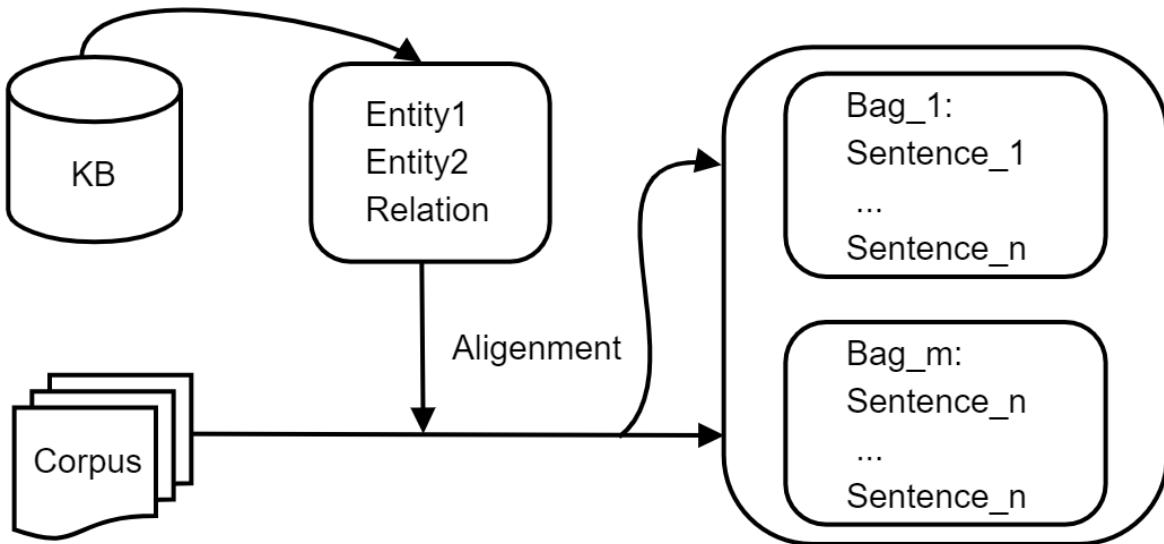


FIG. 2.7 : Le processus de supervision à distance.

2.3.2 Méthodes basées sur les DNNs

Les méthodes basées sur l'apprentissage profond peuvent être divisées en 3 catégories ; CNN, RNN/LSTM, Mixed Structure. On peut généralement différencier entre deux approches utilisées dans les méthodes basées sur les réseaux de neurones profonds, les méthodes orientées structure et les méthodes orientées sémantique. En ce qui suit on va présenter les différentes méthodes et approches existantes.

Méthodes basées sur les CNNs

Les modèles basés sur les CNN sont un modèle généraliste de RE, qui ont atteint de bon résultats.

Modèles orientés structure

Le premier modèle utilisant les CNNs pour l'extraction de relation est proposé par [75]. En utilisant un dictionnaire de synonymes et d'autres caractéristiques lexicales, ce modèle transforme une phrase en une série de vecteurs qui servent comme entrée des CNNs avec une couche de sortie softmax pour calculer une probabilité de classification. Pour améliorer cette méthode, le modèle [76], qui introduit plusieurs couches de filtrage, ainsi que des modules max-pooling aux CNNs. Basée sur ce travail, [77] décrit un CNN dynamique qui utilise des opérateurs max-pooling dynamiques pour choisir des caractéristiques d'après les résultats du CNN. Ces deux modèles ont réalisé des bons résultats.

Modèles orientés sémantique

Pour bien apprendre des représentations de relation plus robustes, [78] propose un modèle à travers les CNNs et SDP. Ce modèle prend le sujet et l'objet de la phrase comme entrée, et supprime les mots qui ne sont pas liés à la discrimination de la relation, après une grande précision avec des échantillons simplement négatifs. Une autre méthode pour améliorer la représentation sémantique, est l'utilisation des mécanismes d'attention. [79], utilise deux niveau de mécanisme d'attention, appelé multi-level attention CNNs, qui permet un

apprentissage de bout en bout à partir de données étiquetées spécifique à une tâche. Ce mécanisme d'attention multiple prend en compte les informations sémantiques au niveau du mot et de la phrase. Ce modèle utilise rarement d'autres informations sémantiques externes.

Méthodes basées sur les RNNs/LSTMs

Le problème des méthodes basées sur les CNNs est que les CNN considèrent rarement les caractéristiques globales et les informations de séquence temporelle, en particulier pour la dépendance à longue distance des paires d'entités.

Modèles orientés structure

Pour apprendre les relations dans un contexte plus long et tenant compte des informations temporelles, l'étude [80] utilise une architecture basée sur les RNNs bidirectionnels pour effectuer cette tâche. Les RNNs combinent l'entrée de chaque état caché et représente les caractéristiques dans le niveau de la phrase. A la fin du modèle, il effectue une opération de max-pooling pour sélectionner quelques caractéristiques de mots déclencheurs pour la prédiction. Bien que l'opération de max-pooling simplifie l'extraction des caractéristiques, l'efficacité de ces caractéristiques reste à discuter. En outre, le modèle RNN présente toujours le problème de l'explosion du gradient.

Modèles orientés sémantique

En se basant sur les problèmes précédent, un nouveau modèle a été proposé, SDP-LSTM [81]. Ce modèle exploite quatre types d'informations : vecteurs de mots, balises POS, relations grammaticales et hypernymes de WordNet, afin de construire quatre canaux permettant à ce modèle de prendre en charge les informations externes. Ensuite, il concatène le résultat des quatre canaux à la couche softmax pour la prédiction.

Méthodes de structure mixte

En addition aux modèles mentionnés ci-dessus, des approches propose de combiner les deux approches.

Modèles orientés structure

Pour intégrer les CNNs et les RNNs, [82] propose deux réseaux neuronales basés sur les CNNs et les LSTM, en joignant l'apprentissage des entités et les motifs de relations, les propriétés sémantiques de l'entité peuvent être reflétées par les mots qui les entourent, ce qui permet de résoudre le problème des mots inconnus dans les entités, et le modèle de relation modélisé par la sous-phrase entre les entités données au lieu de la phrase entière. Avec la sémantique des entités et le modèle de relation, les performances de RE peuvent être améliorées.

Modèles orientés sémantique

Contrairement aux deux travaux précédents, [83] combinent ces deux types de modèles en fonction du SDP. Afin d'améliorer le sens de la directivité de la relation, ce modèle, appelé BRCNN, apprend les caractéristiques de la phrase à partir du SDP dans les deux directions, positive et négative, ce qui est bénéfique pour prédire la direction de la relation.

2.4 L'intégration des graphes des connaissances

L'intégration de graphes des connaissances (Knowledge Graph Embedding) est l'incorporation des composants (les entités et les relations) du graphe de connaissances dans un espace continu de vecteurs, cela sert à simplifier la manipulation tout en gardant la structure du KG. Cette tâche peut aider à réaliser plusieurs tâches en aval, comme la complétion du KG ou bien l'extraction de relations, la classification des entités et la résolution des entités, par conséquent, elle a gagné un attention immense. La plupart des méthodes effectuent cette tâche en se basant sur les faits observés. Dans cette section, on va présenter les différentes techniques existantes pour réaliser cette tâche.

2.4.1 Différentes étapes d'intégration des graphes de connaissances

Soit un KG de n entités et m relations, on suppose que les faits observés dans ce KG sont stockés sous la forme des triples $D^+ = (h, r, t)$. Chaque triple est composé d'une tête $h \in E$, d'une queue $t \in E$ et d'une relation $r \in R$ entre eux.

Une méthode d'intégration typique consiste en général de trois étapes :

- Représentation des entités et des relations
- Définition d'une fonction de scoring
- Apprentissage des représentations des entités et relations

La première étape spécifie la façon dont les entités et les relations seront représentées dans l'espace continu des vecteurs. Les entités sont généralement représentées en utilisant des points déterministes dans l'espace [84]. Les relations sont considérées comme des opérations dans l'espace des vecteurs [85] [84], qui peuvent être représentées comme des vecteur, des matrices ou bien des tenseurs. La fonction de scoring $f_r(h, t)$ est définie pour chaque fait (h, r, t) pour mesurer sa plausibilité. Les faits observés dans le KG tend d'avoir un score plus élevé. En fin pour apprendre ces représentations, la troisième étape résout un problème d'optimisation qui maximise la plausibilité totale des faits observés.

2.4.2 Modèles de distance translationnelle

Les modèles de distance translationnelle exploitent des fonctions de scoring basé sur la distance, ils mesurent la plausibilité d'un fait à partir la distance entre deux entités, généralement après une translation par une relation.

TransE et ses extensions

TransE [85] représente les entités et les relations sous la forme de vecteur dans le même espace, disant R^d . Etant donné un fait (h, r, t) , la relation est interprétée comme un vecteur de translation r de sorte que les représentations des entités h et t soient

connecté par r avec un erreur. La figure 2.8 donne une illustration simple de cette idée. La fonction de scoring de ce modèle est définie comme l'opposé de la distance entre $h + r$ et t

$$f_r(h, t) = -\|h + r - t\|_{1/2}.$$

Cette fonction est prévu d'être grande lorsque (h, r, t) détient. Malgré la simplicité et l'efficacité de ce modèle, il ne peut pas tenir en compte les multi-relations, puisqu'il implique l'unicité de la relation entre deux entités données. Pour s'adresser à ce problème, on doit permettre à chaque entité d'avoir différentes représentations pour chaque relation.

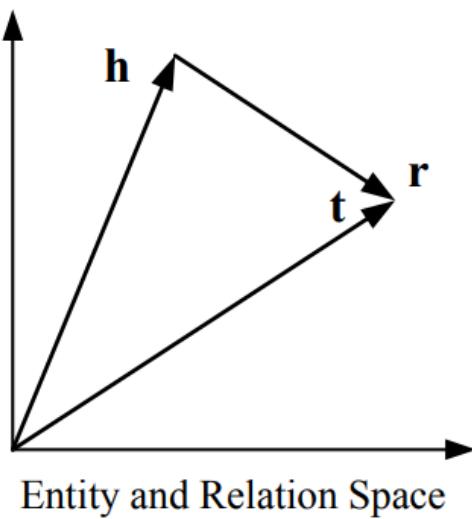


FIG. 2.8 : Illustration simple de TransE.

TransH [84] suit cette idée, et propose des hyperplans spécifique-relation. Comme dans la figure 2.9, TransH, modélise les entités comme des vecteurs, mais chaque relation t comme un vecteur dans un hyperplan avec un vecteur normal w_r .

Étant donné un fait (h, r, t) , les représentations des entités sont premièrement projetées

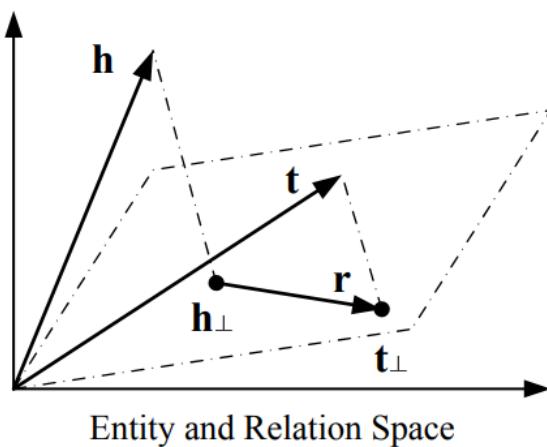


FIG. 2.9 : Illustration simple de TransH.

sur l'hyperplan

$$h_{\perp} = h - w_{\perp}^r h w_r, \quad t_{\perp} = t - w_r^{\perp} t w_r$$

Les projections sont ainsi supposées connecté avec un petit erreur si le fait (h, r, t) détient. La fonction de scoring est définie comme suit :

$$f_r(h, t) = -||h_{\perp} + r - t_{\perp}||_2^2$$

TransR [86] propose que chaque entité possède plusieurs attributs et plusieurs relations. Chaque relation peut être connecté à différents attributs. TransR modélise les entités et les relations dans des différents espaces. La figure 2.10 montre l'intuition derrière cette idée.

Étant donné un fait (h, r, t) , les entités sont premièrement projetées dans l'espace des relations suivant les relations :

$$h_{\perp} = M_r h, \quad t_{\perp} = M_r t$$

Avec $M_r \in R^{kd}$ est la matrice de projection de l'espace des entités vers l'espace des relations de r . La fonction de scoring est donc définie comme :

$$f_r(h, t) = -||h_{\perp} + r - t_{\perp}||_2^2$$

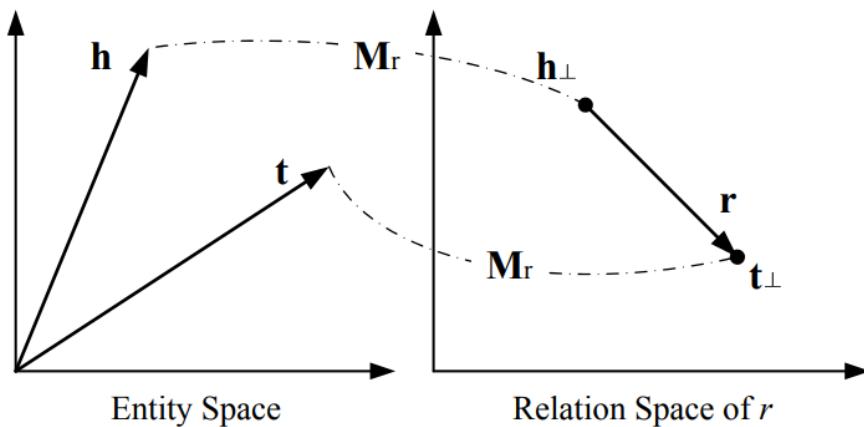


FIG. 2.10 : Illustration simple de TransR.

2.4.3 Modèles de correspondance sémantique

La correspondance sémantique est l'une des tâches de base du traitement automatique du langage naturel. Comme nous avons vu pour les modèles de distance translationnelle utilisent des fonctions de scoring basé sur la distance de translation pour calculer la similarité entre les différentes entités et relations. D'autre part, les modèles de correspondance sémantique utilisent des fonctions de scoring basées sur la similarité. Il y en a plusieurs modèles qui utilisent ce concept, en ce qui suit, on va citer quelques uns.

RESCAL et ses extensions

RESCAL [87] associe chaque entité à un vecteur qui capture sa sémantique latente. Chaque relation est représentée par une matrice qui modélise les interactions par paire entre les facteurs latents. Le score d'un fait (h, r, t) est défini par une fonction bilinéaire

$$f_r(h, t) = h^\top M_r t = \sum_{i=0}^{d-1} \sum_{j=0}^{d-1} [M_r]_{ij} \cdot [h]_i \cdot [t]_j$$

Où $h, t \in R^d$ sont les représentations des entités et $M_r \in R^{dd}$ est une matrice associée à la relation. Ce score capture les interactions par paire entre toutes les composantes de h et t .

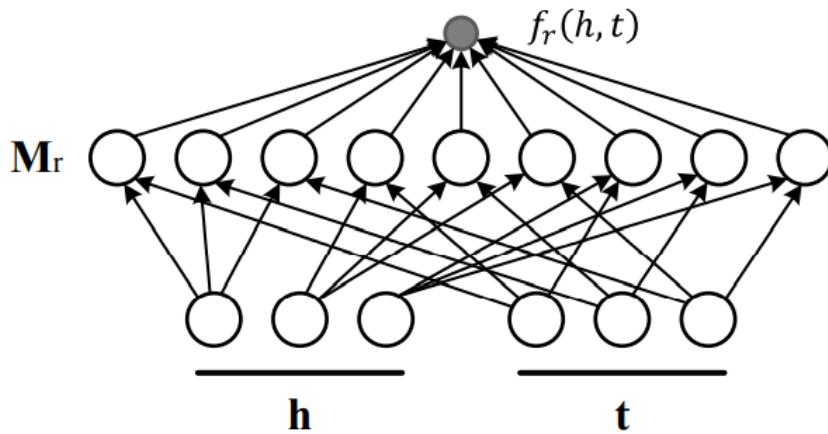


FIG. 2.11 : Illustration simple de RESCAL.

DistMult [88], simplifie RESCAL par la restriction de M_r aux matrices diagonales. Pour chaque relation r , il introduit un vecteur de représentation $r \in R^d$ et nécessite $M_r = \text{diag}(r)$. La fonction de scoring est définie comme suit :

$$f_r(h, t) = h^\top \text{diag}(r) t = \sum_{i=0}^{d-1} [r]_i \cdot [h]_i \cdot [t]_i$$

Ce score capture les interactions par paire entre seulement les composantes de h et t dans la même dimension. La figure 2.12 simplifie l'idée derrière cette idée.

HOLE (Holographic Embeddings) [89], combine la force expressive de RESCAL et la simplicité de DistMult. Il représente les entités et les relations sous la forme des vecteurs dans R^d . Etant donné un fait (h, r, t) , les représentations des entités sont premièrement composées par $h \star t \in R^d$ en utilisant la corrélation circulaire :

$$[h \star t]_i = \sum_{k=0}^{d-1} [h]_k \cdot [t]_{(k+i) \bmod d}$$

Le vecteur oppositionnel est ainsi apparié avec la représentation de la relation pour calculer le score du fait

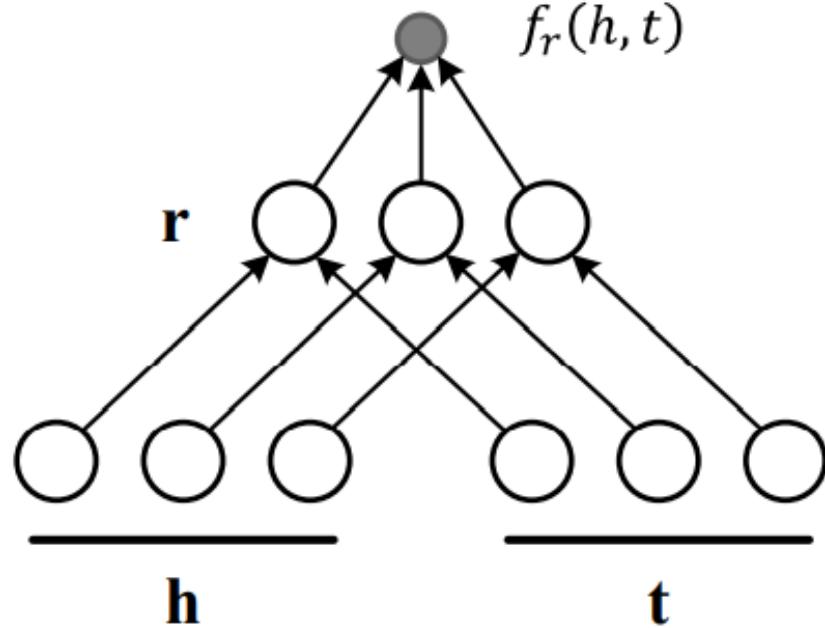


FIG. 2.12 : Illustration simple de DistMult.

$$f_r(h, t) = r^\top (h \star t) = \sum_{i=0}^{d-1} [r]_i \sum_{k=0}^{d-1} [h]_k \cdot [t]_{(k+i) \bmod d}$$

La corrélation circulaire compresse les interactions par pairs 2.13. HolE est capable de modéliser les relations asymétrique comme RESCAL.

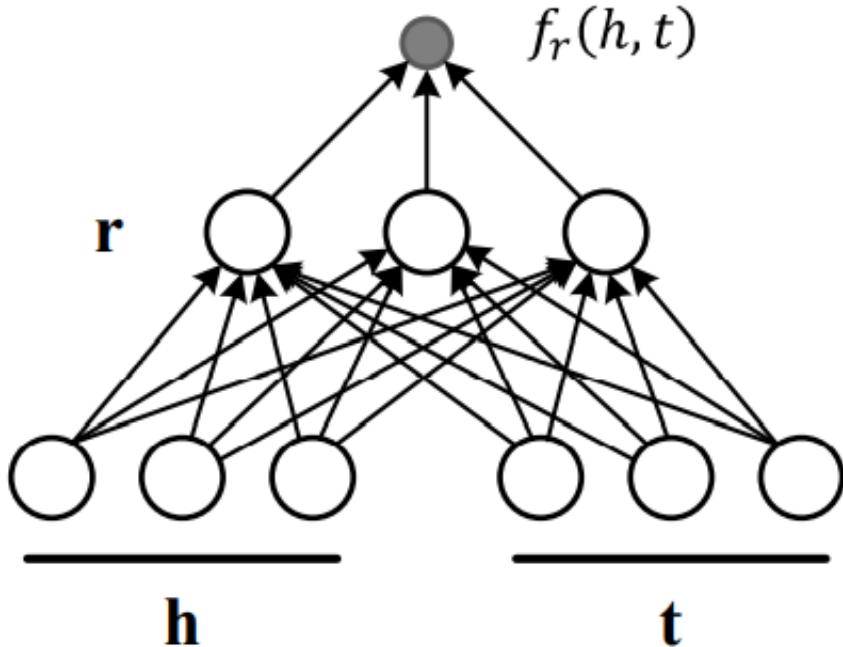


FIG. 2.13 : Illustration simple de HolE.

ComplEx (Complex Embeddings [90], étend DistMult en introduisant des représentations complexes pour une meilleure modélisation des relations asymétriques. Les entités et les relations sont représentées dans un espace complexe, C^d . Le score d'un fait (h, r, t) est défini par

$$f_r(h, t) = \operatorname{Re}(h^\top \operatorname{diag}(r)\bar{t}) = \operatorname{Re}\left(\sum_{i=0}^{d-1} [r]_i \cdot [h]_i \cdot [\bar{t}]_i\right)$$

2.4.4 Modèles basé sur les réseaux neuronaux de convolution

Les réseaux neuronaux constituent une solution prometteuse dans de nombreux domaines et possèdent les capacités d'auto-apprentissage, de stockage associatif et d'optimisation à grande vitesse. Les modèles traditionnels basés sur la distance et la correspondance sémantique ne peuvent pas répondre aux exigences du KGE. Récemment, afin d'obtenir des incorporations d'entités et de relations meilleures et plus efficaces, un modèle de réseau neuronal a également été introduit dans KGE pour propager les informations de voisinage. Ces modèles sont également divisés en deux sous-catégories : les modèles avec informations supplémentaires et les modèles sans informations supplémentaires.

ConvE [91] représente les entités et les relations comme des vecteurs de taille unidimensionnelle. Lors du scoring $< h, r, t >$, il concatène et remodèle h et r en une entrée unique $[h; r]$, de dimensions $d_m d_n$. Cette entrée passe par une couche convulsive avec un ensemble de mn filtres, puis par une couche dense avec d neurones et un ensemble de poids W . La sortie est finalement combinée avec l'encastrement de queue t en utilisant le produit scalaire, ce qui donne le score de fait. La fonction de scoring est définie comme suit :

$$\Psi_r(h, t) = f(\operatorname{vec}(f([\bar{h}; \bar{r}] * w))W)t$$

ConvKB [92] modélise les entités et les relations comme des vecteurs de même taille ; lors de la scoring de $< h, r, t >$, il concatène h et r et t en une matrice $d3$, $[h; r; t]$. Cette entrée subit une convolution par un ensemble de T filtres de forme 13, résultant en une carte de caractéristiques $T3$. La carte de caractéristiques passe par une couche dense avec un neurone et des poids W , ce qui donne le score de faits.

2.5 Conclusion

Dans ce chapitre nous avons essayé de présenter les différentes techniques et approches utilisées pour la réalisation des différentes nécessaires de notre projets, à savoir, la reconnaissance et la liaison d'entités nommées, l'extraction des relations et l'apprentissage des représentations des connaissances. Dans le chapitre suivant on va présenter les différentes techniques choisies pour chaque étape, ainsi que les résultats obtenus pour dans chaque étape.

Chapitre 3

Réalisation et résultats

Dans le chapitre précédent nous avons essayé de présenter une lecture non exhaustive de l'état de l'art des tâches les plus importantes dans la construction des graphes de connaissances, ainsi que l'apprentissage des représentations des connaissances, qui peut être utilisé par la suite pour la prévision des liens. Dans ce chapitre, on va présenter le travail mené au cours de la période de notre projet de fin d'études. Avant de commencer la présentation des outils utilisés, il faut noté que le choix de différentes approches et techniques à utiliser était dépendant aux ressources matériels disponibles puisque la plupart des techniques d'état d'art nécessite des puissances de calculs élevées.

3.1 Préparation des données

3.1.1 Format des fichiers

Nous avons utilisé la base de données CORD-19 [3] comme source des publications scientifiques, comme nous avons déjà mentionné, CORD-19 contient plus que 400 milles publications scientifiques, parmi eux, 200 milles sont en texte intégrale. Les publications sont sous format JSON, et contiennent les informations suivantes :

- Identifiant de la publication.
- Le titre de la publication.
- Les auteurs de la publication.
- Le résumé de la publication.
- Les sections de la publication.
- Le texte de chaque section.

3.1.2 Pré-traitement

nous avons lu les fichiers JSON en utilisant le langage de programmation Python, nous avons tout d'abord récupérer l'identifiant de chaque publication ainsi que son texte complet. Ensuite, nous avons éliminé toutes les publications non anglaises. Les textes complets ont été traités par la suite, en les décomposant en des phrases. Pour chaque phrase nous avons gardé l'identifiant de la publication source, la section et l'emplacement dans la section.

Vue la nature des modèles utilisés dans les prochaines étapes, nous avons décidé de ne pas modifier le texte initial pour tenter d'améliorer l'analyse sémantique.

3.2 Reconnaissance et liaison d'entités nommées

La reconnaissance et la liaison des entités nommées est l'étape la plus importante dans la construction des graphes de connaissances. La qualité des entités extraite joue un

rôle pertinent pour les graphes de connaissance, pour cela nous avons essayé de chercher les méthodes de l'état de l'art, à un coût minimal, l'une de nos meilleures choix était les modèles de langage de *scispacy*.

3.2.1 spaCy et scispacy

spaCy

spaCy [93] est une bibliothèque gratuite, et open-source pour effectuer le traitement du langage naturel avancé dans Python. Elle est conçue spécifiquement pour une utilisation en production et facilite le traitement des textes volumineux, ainsi que le syntaxe est simple. Spacy peut être utilisé pour construire des systèmes d'extraction d'informations ou de compréhension du langage naturel, ou pour pré-traiter du texte pour l'apprentissage profond.

Caractéristiques

spaCy contient plusieurs caractéristiques et utilités qui peuvent être soit des concepts linguistiques, ou bien des fonctionnalités générales d'apprentissage automatique, à savoir :

- Tokenisation
- Étiquetage POS
- Analyse des dépendances
- Lemmatisation
- Détection des limites de phrases
- Reconnaissance des entités nommées
- Liaison des entités
- Similarité
- Classification du texte
- Correspondance basée sur des règles
- Entraînement
- Serialisation

Modèles statistiques

Alors que certaines fonctionnalités de spaCy fonctionnent indépendamment, d'autres nécessitent le chargement de pipelines entraînés, qui permettent à spaCy de prédire les annotations linguistiques - par exemple, si un mot est un verbe ou un nom. Un pipeline entraîné peut consister en plusieurs composants qui utilisent un modèle statistique entraîné sur des données étiquetées. spaCy offre actuellement des pipelines entraînés pour une variété de langues, qui peuvent être installés comme des modules Python individuels.

Les paquets de pipelines peuvent différer en termes de taille, de vitesse, d'utilisation de la mémoire, de précision et des données qu'ils incluent.

Pipeline de traitement

Le pipeline de traitement consiste d'un ou plusieurs composants de pipeline qui s'exécutent en ordre.

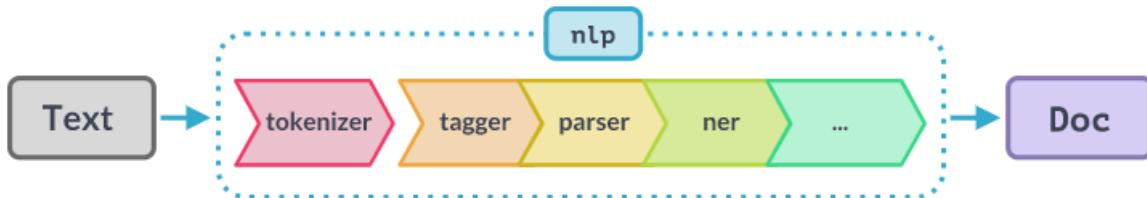


FIG. 3.1 : Architecture du pipeline de traitement

Les composants du pipeline peuvent contenir des modèles statistiques et poids entraînés, ou n'apporter que des modifications basées sur des règles. La bibliothèque fournit plusieurs composants pour différentes tâches du traitement automatique du langage naturel :

- *AttributeRuler* Définir les attributs des tokens en utilisant des règles de correspondance.
- *DependencyParser* Prédire les dépendances syntaxiques.
- *EntityLinker* Désambiguïser les entités nommées par rapport aux nœuds d'une base de connaissances.
- *EntityRecognizer* Prédire des entités nommées, par exemple des personnes ou des produits.
- *EntityRuler* Ajouter des portées d'entités au Doc en utilisant des règles basées sur des tokens ou des correspondances de phrases exactes.
- *Lemmatizer* Déterminer les formes de base des mots.
- *Morphologizer* Prédire les caractéristiques morphologiques et les balises part-of-speech à gros grain.
- *SentenceRecognizer* Prédire les limites des phrases.
- *Sentencizer* Implémentation d'une détection des limites de phrases basée sur des règles qui ne nécessite pas l'analyse des dépendances.
- *Tagger* Prédire les étiquettes de parties du discours.
- *TextCategorizer* Prédire les catégories ou les étiquettes sur l'ensemble du document.
- *Tok2Vec* Appliquer un modèle "token-to-vector" et définir ses sorties.

- *Tokenizer* Segmenter le texte brut et créer des objets Doc à partir des mots.
- *TrainablePipe* Classe dont héritent tous les composants du pipeline pouvant être formés.
- *Transformer* Utiliser un modèle de transformateur et définissez ses sorties.

scispaCy

scispaCy [94] est une extension de spaCy sur les textes scientifiques et propose des modèles de langage prétraîné pour effectuer les différentes tâches du traitement du langage naturel sur des textes scientifiques. scispaCy propose quatre modèles de langage pré-traînés ainsi que quatre modèles spécialisés en reconnaissance d'entité nommées.

Modèles pré-traînés de langage

- *en_core_sci_sm* Une pipeline spaCy complète pour les données biomédicales qui contient un vocabulaire de près 100 milles.
- *en_core_sci_md* Une pipeline spaCy complète pour les données biomédicales qui contient un vocabulaire de près de 360 milles et 50 milles vecteurs de mots.
- *en_core_sci_lg* Une pipeline spaCy complète pour les données biomédicales qui contient un vocabulaire de près de 785 milles et 600 milles vecteurs de mots.
- *en_core_sci_scibert* Une pipeline spaCy complète pour les données biomédicales qui contient un vocabulaire de près de 785 milles et qui utilise le modèle de Transformers *scibert-base*

Les performances de ces modèles sont représentés dans la table 3.2

modèle	UAS	LAS	POS	Mentions (F1)	Web UAS
<i>en_core_sci_sm</i>	89.54	87.62	98.32	68.15	87.62
<i>en_core_sci_md</i>	89.61	87.77	98.56	69.64	88.05
<i>en_core_sci_lg</i>	89.63	87.81	98.56	69.61	88.08
<i>en_core_sci_scibert</i>	92.03	90.25	98.91	67.91	92.21

TAB. 3.1 : Performances des modèles de langage de scispaCy.

Ces performance sont parmi les 3% des performances d'état d'art publiés dans l'analyse des dépendances (Dependency parsing).

Modèle pré-entraînés de reconnaissance d'entités nommées

- *en_ner_craft_md* Modèle de reconnaissance d'entités nommées de spaCy entraîné sur le corpus CRAFT.
- *en_ner_jnppba_md* Modèle de reconnaissance d'entités nommées de spaCy entraîné sur le corpus JNLPBA.

- *en_ner_bc5cdr_md* Modèle de reconnaissance d'entités nommées de spaCy entraîné sur le corpus BC5CDR.
- *en_ner_bionlp13cg_md* Modèle de reconnaissance d'entités nommées de spaCy entraîné sur le corpus BIONLP13CG.

modèle	F1	Types d'entités
<i>en_ner_craft_md</i>	76.11	entité chimiques, GO
<i>en_ner_jnppba_md</i>	71.62	ADN, ARN, protéines..
<i>en_ner_bc5cdr_md</i>	84.49	maladies et composants chimiques
<i>en_ner_bionlp13cg_md</i>	77.75	acides aminés, gènes, organismes..

TAB. 3.2 : Performances des modèles de reconnaissance d'entités nommées de scispaCy.

Liaison d'entités nommées

scispaCy dispose d'un composant de pipeline pour la liaison des entités nommées avec les ontologies extérieures, il supporte également toutes les ontologies que nous avons vu dans la section 2.2.1. Malheureusement ce composant n'est pas supporté par les modèles de NER de scispaCy.

3.2.2 Implémentation et résultats

Reconnaissance d'entités nommées

pour la réalisation de la reconnaissance des entités nommées nous avons choisi d'utiliser le modèle basé sur SCIBERT-BASE. L'entrée utilisée est une phrase extraite depuis la base de données de CORD-19, comme nous avons expliqué dans la section 3.1.

Vu au temps nécessaire pour le calcul, et aux ressources matériels disponibles, nous avons seulement pu traité 2867148 phrases de 6.4 millions phrases, dont nous avons pu extraire 76003 entités nommées.

Pour chaque entité, nous avons gardé un nombre d'informations :

- nom de l'entité
- le CUI de l'entité
- la section de l'entité
- le début de l'entité dans la section
- la fin de l'entité
- le score de liaison avec UMLS

Liaison des entités nommées

Un composant de liaison d'entités nommées a été utilisé avec le modèle de l'étape précédente pour récupérer les CUIs (Concept Unique Identifier) depuis l'UMLS. Le modèle a

pu récupérer 29069 concepts uniques. Ces identifiants ont été utilisés pour récupérer le type sémantique de chaque entité, ainsi que son nom canonique. Pour cela, nous avons utilisé UMLS REST API. Il faut mentionner que UMLS REST API nécessite une licence d'utilisation.

Les entités extraites étaient de 126 types sémantiques différents, à savoir ; 'Disease or Syndrome', 'Organic Chemical', 'Organism', 'Organism Function', 'Therapeutic or Preventive Procedure', 'Gene or Genome', 'Virus', 'Body Part, Organ, or Organ Component', 'Molecular Function', 'Amino Acid, Peptide, or Protein', 'Diagnostic Procedure', 'Pathologic Function', 'Pharmacologic Substance', 'Genetic Function', 'Biologically Active Substance', 'Nucleic Acid, Nucleoside, or Nucleotide', 'Organ or Tissue Function', 'Cell or Molecular Dysfunction', 'Molecular Biology Research Technique', 'Cell Component', 'Cell Function', 'Substance', 'Clinical Attribute', 'Antibiotic', 'Experimental Model of Disease', 'Chemical Viewed Functionally', 'Chemical', 'Amino Acid Sequence', 'Vitamin', 'Clinical Drug'.

CUI	Type sémantique	nom
C5203670	Disease or Syndrome	COVID-19
C0032594	Organic Chemical	Polysaccharides
C030987	Pharmacologic Substance	PREVENT (product)
C0030106	Chemical	Ozone

TAB. 3.3 : Exemples d'entités nommées extraites.

La table 3.3 montre quelques entités résultantes. La distribution des types sémantiques des entités extraites est données dans la figure 3.2, où on vous présente les 30 types les plus et les moins fréquents.

Comme on remarque dans la figure 3.2, les entités nommées les plus fréquents sont les gènes, les composants chimiques organiques, les acides aminés et les protéines et les maladies. Ceux sont en fait les entités les plus importantes dans notre recherche.

3.3 Extraction des relations

Pour l'extraction des relations, nous avons choisi aussi d'utiliser un modèle d'étiquetage des rôles sémantiques pour extraire les prédictats des entités nommées déjà extraites. Pour la réalisation de cette étape nous avons utilisé la bibliothèque AllenNLP.

3.3.1 AllenNLP et AllenNPL-models

AllenAI [95] est un projet open-source maintenu par l'institute AllenAI, il est construite sur pyTorch pour le développement des modèles d'état de l'art dans plusieurs tâches de NLP.

AllenNLP-models Une collection de modèles d'état de l'art, contient des composants pour l'application de AllenNLP dans les différentes tâches de NLP. Elle fournit aussi des

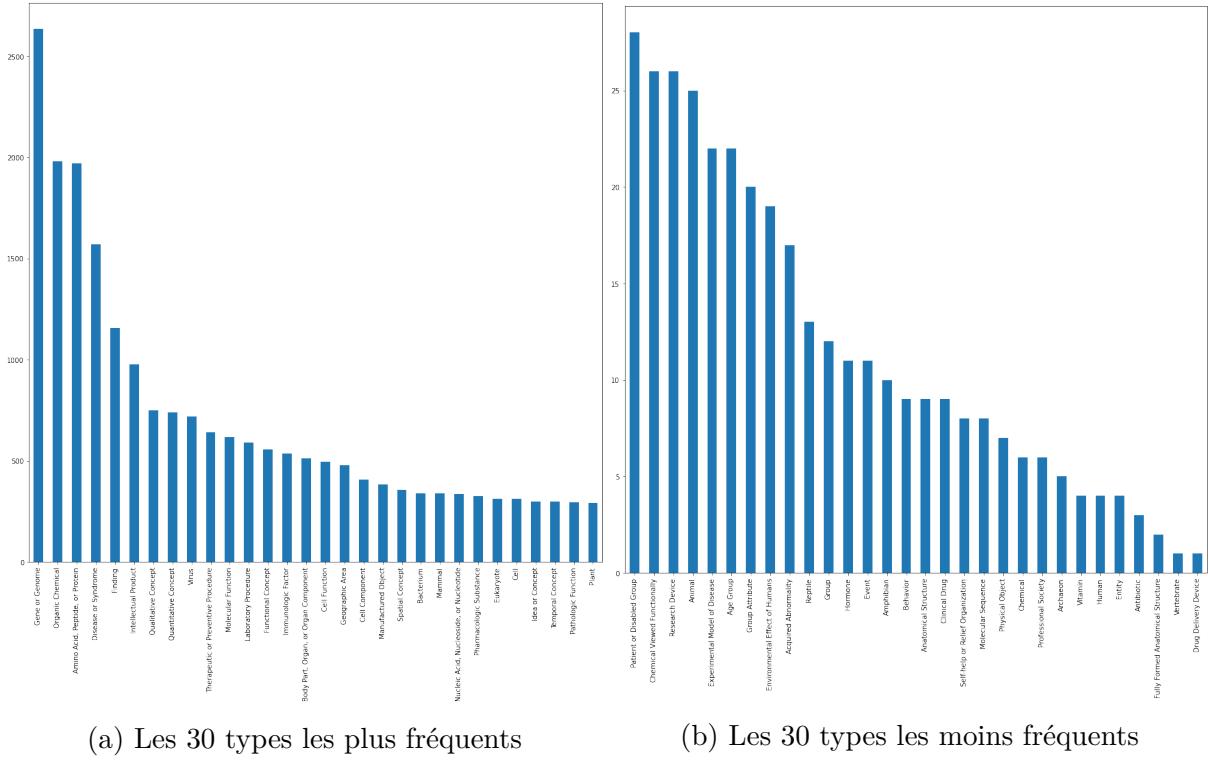


FIG. 3.2 : Distribution des 30 types les plus et les moins fréquents.

méthodes faciles et simples pour télécharger et utiliser les modèles pré-traînés. parmi les modèles inclus dans cette bibliothèque on cite :

- *structured-prediction-srl-bert* [96] Un modèle basé sur BERT avec quelques modifications (pas de paramètres supplémentaires à part une couche de classification linéaire).
- *structured-prediction-biaffine-parser* Un modèle neuronal pour l'analyse syntaxique des dépendances utilisant des classificateurs biaffins au-dessus d'un LSTM bidirectionnel.
- *rc-transformer-qa* Un modèle de compréhension de la lecture calqué sur le modèle proposé par [55], avec des améliorations empruntées au modèle SQuAD du projet transformers.

3.3.2 Réalisation et résultats

nous avons choisi d'utiliser le modèle *structured-prediction-srl-bert* [96] puisqu'il a atteint de bonnes performances pour le SRL ($F1 = 86.49$).

Depuis les 2867148 phrases traitées, nous avons pu extraire 614012 triples contenant les entités que nous avons déjà extrait. Après un traitement rapide sur les triplets obtenus (lemmatisation des relations et élimination des triplets dupliqués) il nous a resté 413064 relations.

Les relations nécessitent encore du travail, puisque nous avons utilisé un modèle d'extraction d'informations ouvertes, il est possible d'avoir une seule relation qui s'exprime

Chapitre 3. Réalisation et résultats

en plusieurs formes. Pour chaque triplet, nous avons gardé les informations de la phrase, ainsi que les informations des entités.

La figure 3.3 montre le sous-graphe contenant les maladies et substances chimiques du graphe principal.

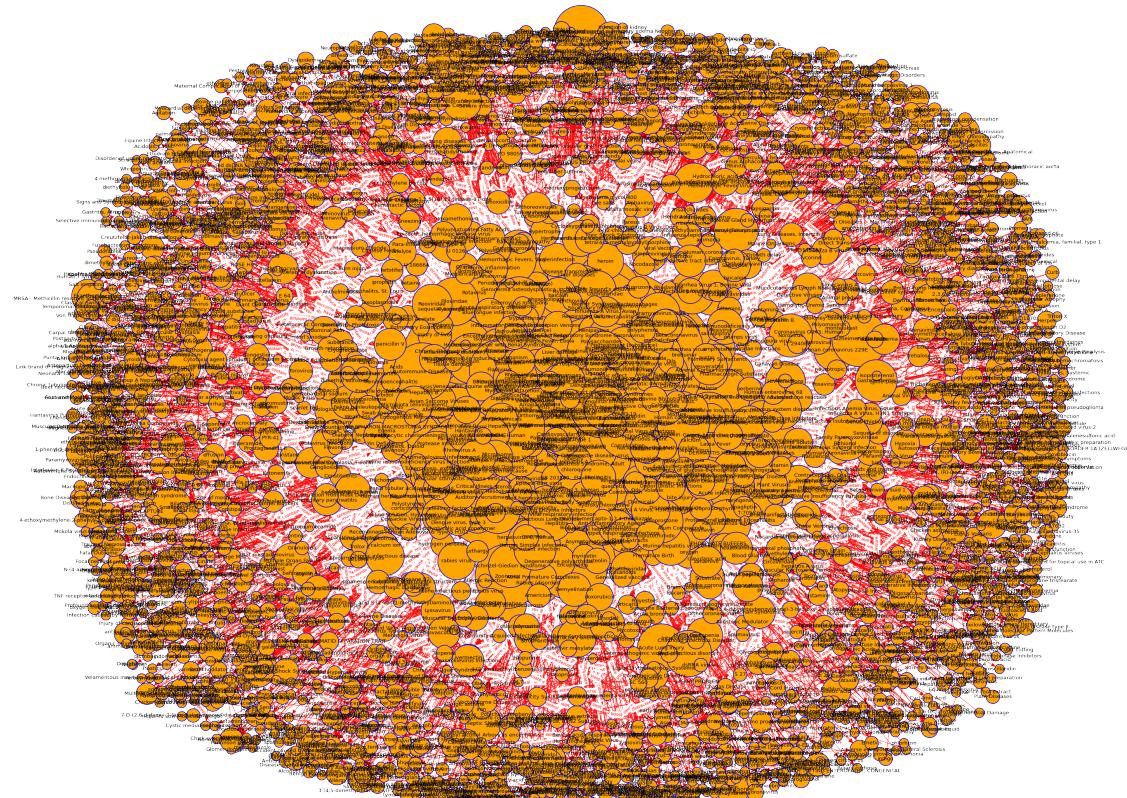


FIG. 3.3 : Sous-graphe contenant les maladies et les substances chimiques

Pour avoir une vision plus claire sur notre graphe, nous avons construit des sous-graphes contenant les noeuds ayant les plus grands degrés. La figure 3.4 montre les noeuds ayant les plus grands degrés de notre graphe. Ces degrés sont en général supérieurs ou égaux à 100. Ces noeuds ont des degrés entre 122 et 229, le noeud le plus connecté est "Gene expression".

La figure 3.5 montre les noeuds ayant un degré moyen, qui varie entre 100 et 30. Ceux sont les noeuds les plus connectés entre eux, ils représentent le cœur de notre graphe de connaissances puisqu'ils contiennent la majorité des relations dans notre graphe.

Le reste des noeuds ont un degré plus ou moins faible. Ce qu'on remarque ici est que la majorité des noeuds ont une connectivité faible (entre 1 et 10).

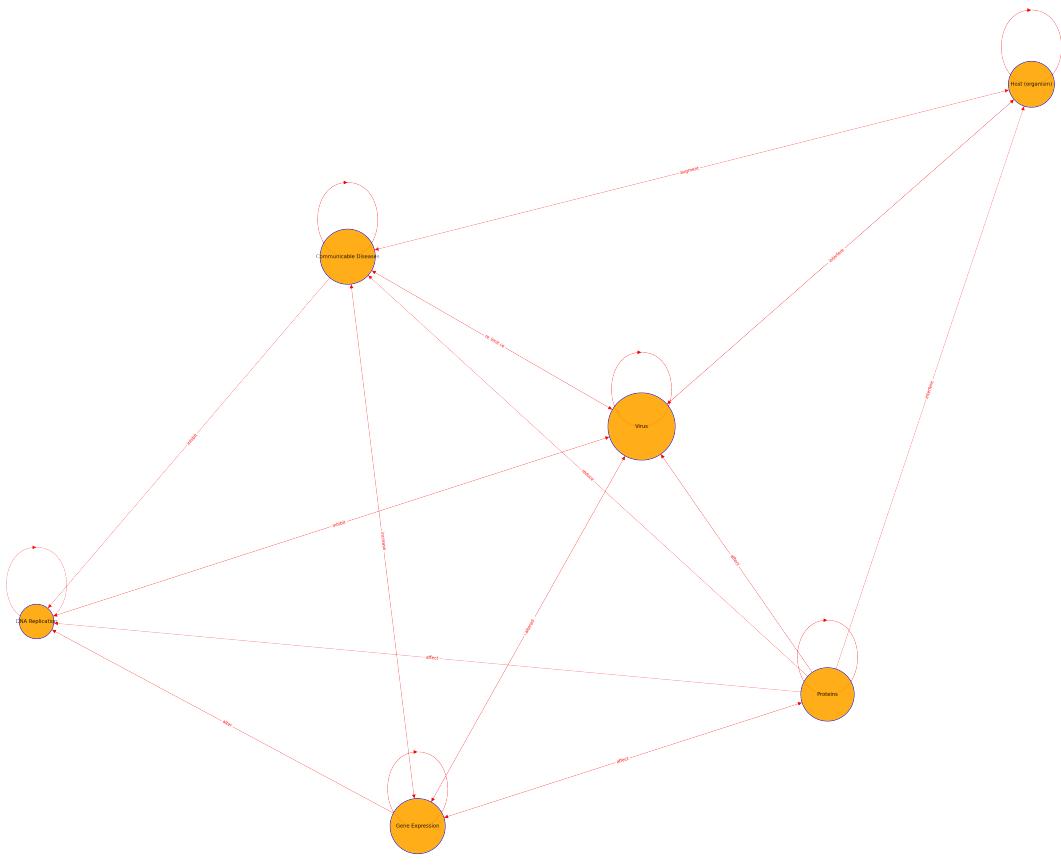


FIG. 3.4 : Sous-graphe contenant les noeuds ayant les degrés ≥ 100

3.4 Knowledge Graph Embeddings

Pour entraîner les représentations des connaissances, plusieurs bibliothèques Python ont été développées, malheureusement, la plupart de ces bibliothèques sont encore dans les versions bêta, ou bien ne supporte pas toutes les fonctionnalités des modèles des KGE, par exemple la prédiction de nouveaux liens nécessite des triplets construits manuellement. Dans ce qui suit, on va présenter la bibliothèque utilisée pour entraîner les modèles choisis.

3.4.1 Pykeen

PyKEEN ou bien Python KnowlEdge EmbeddiNgs [97], est une librairie Python désignée à l’entraînement et l’évaluation des modèles d’incorporation des graphes de connaissances. Elle contient une variétés des modèles d’apprentissage de présentations, ainsi qu’elle fournit des outils simples pour l’entraînement, l’évaluation et l’optimisation des modèles.

Modèles disponibles

Pykeen implémente plusieurs modèles d’embeddings, à savoir :

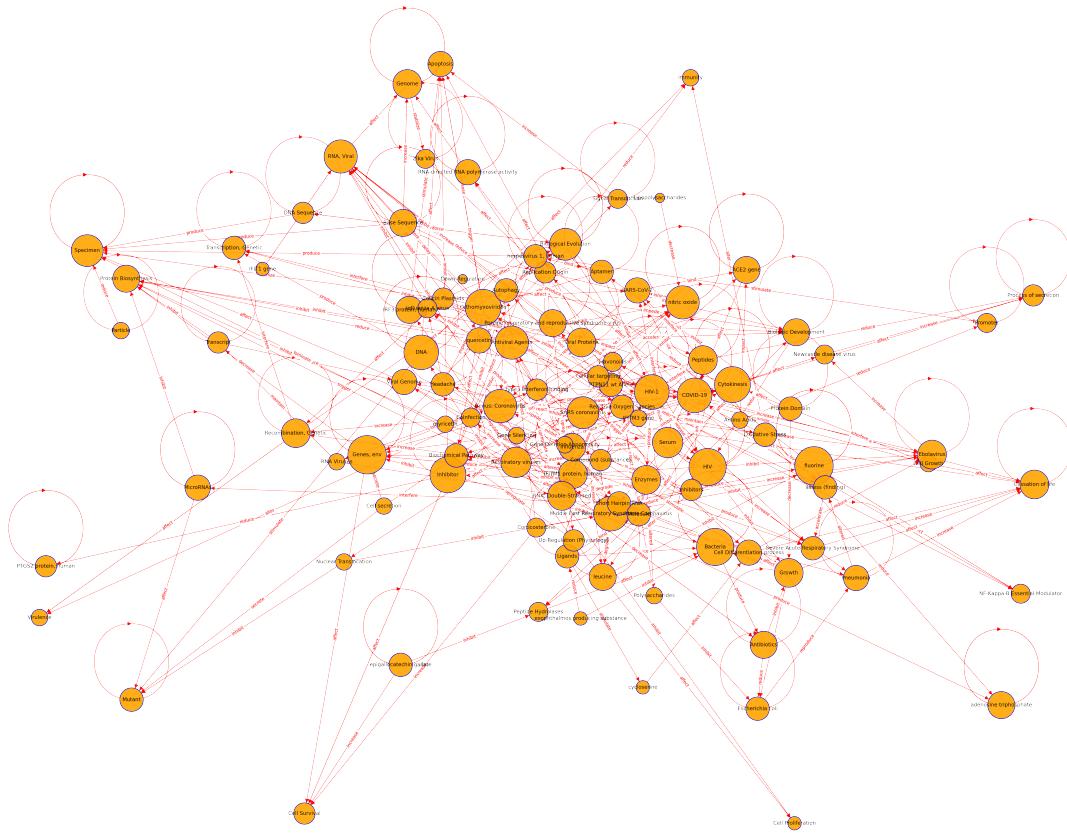


FIG. 3.5 : Sous-graphe contenant les noeuds ayant les degrés entre 30 et 100

- ComplEx une implémentation du modèle [90].
- DistMult une implémentation du modèle [88].
- HolE une implémentation du modèle [89].
- RESCAL une implémentation du modèle [87].
- TransE une implémentation du modèle [85].
- TransH une implémentation du modèle [84].
- TransR une implémentation du modèle [86].

Ceux sont tous des implémentations des modèles que nous avons vu dans la section 2.4. et bien d'autres modèles que nous avons pas couvert dans ce rapport.

Métriques d'évaluation

Pykeen fournit deux méthodes d'évaluations, la première utilise les métriques de la librairie *scikit-learn*, et la deuxième basée sur les ranks.

L'évaluation basé sur les ranks, calcule les métriques suivants :

- Mean Rank (MR) calcule la moyenne arithmétique de tous les rangs individuels. Il est donné comme suit :

$$\text{score} = \frac{1}{|\mathcal{I}|} \sum_{r \in \mathcal{I}} r$$

avec un intervalle de $[1, +\infty]$; où plus proche de 0 est mieux.

- Adjusted Mean Rank (AMR) proposé par [98] et définit par :

$$\text{score} = \frac{MR}{\mathbb{E}[MR]} = \frac{2 \sum_{i=1}^n r_i}{\sum_{i=1}^n (|\mathcal{S}_i| + 1)}$$

Les dérivations de $\mathbb{E}[MR]$ et \mathcal{S}_i sont contenues dans le manuscrit original.

avec un intervalle de $[0, 2]$; où plus proche de 0 est mieux.

- Adjusted Mean Rank Index (AMRI) il est aussi proposé par [98] pour rendre le AMR plus intuitif

$$\text{score} = 1 - \frac{MR - 1}{\mathbb{E}[MR - 1]} = \frac{2 \sum_{i=1}^n (r_i - 1)}{\sum_{i=1}^n (|\mathcal{S}_i|)}$$

avec un intervalle de $[-1, 1]$; où plus proche à 1 est mieux.

- Mean Reciprocal Rank (MRR) est la moyenne arithmétique des rangs réciproques, et donc l'inverse de la moyenne harmonique des rangs. Elle est définie comme suit :

$$\text{score} = \frac{1}{|\mathcal{I}|} \sum_{r \in \mathcal{I}} r^{-1}$$

avec un intervalle de $[0, 1]$; où plus proche à 1 est mieux.

- Hits @ K décrit la fraction des entités vraies qui apparaissent dans les premières entités de la liste de classement triée. Il est donné comme suit :

$$\text{score}_k = \frac{1}{|\mathcal{I}|} \sum_{r \in \mathcal{I}} \mathbb{I}[r \leq k]$$

avec un intervalle de $[0, 1]$ où plus proche à 1 est mieux.

3.4.2 Entraînement des modèles

nous avons choisi de travailler sur quatre modèles, TransE, TransH, TransR et ComplEx et ConvE, pour chacun d'eux, nous avons utilisé les mêmes données pour l'entraînement, le test et la validation, le nombre d'epochs est 400 avec la possibilité de "early stopping" basée sur les Hits @ k.

TransE

L'entraînement est arrêté après 6 évaluations à l'epoch 60.

Le meilleur résultat Hits @ K=0.2247915087187263 à l'epoch 40. La figure 3.6 montre le développement du loss et du Hits @ K tout au long de l'entraînement.

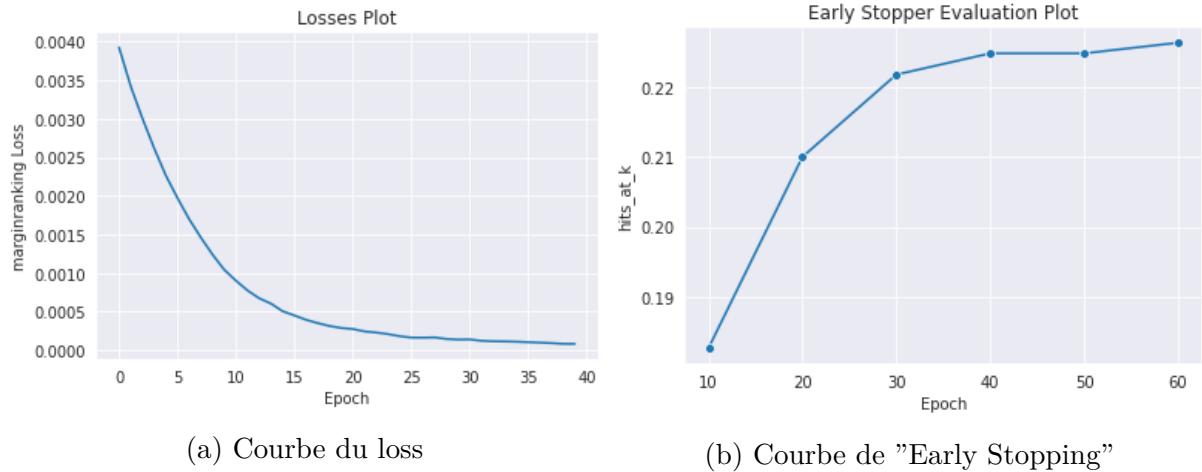


FIG. 3.6 : Développement du loss et du Hits @ k durant l’entraînement de TransE.

TransH

L’entraînement est arrêté après 6 évaluations à l’epoch 60.

Le meilleur résultat Hits @ K=0.1516300227445034 à l’epoch 40. La figure 3.7 montre le développement du loss et du Hits @ K tout au long de l’entraînement.

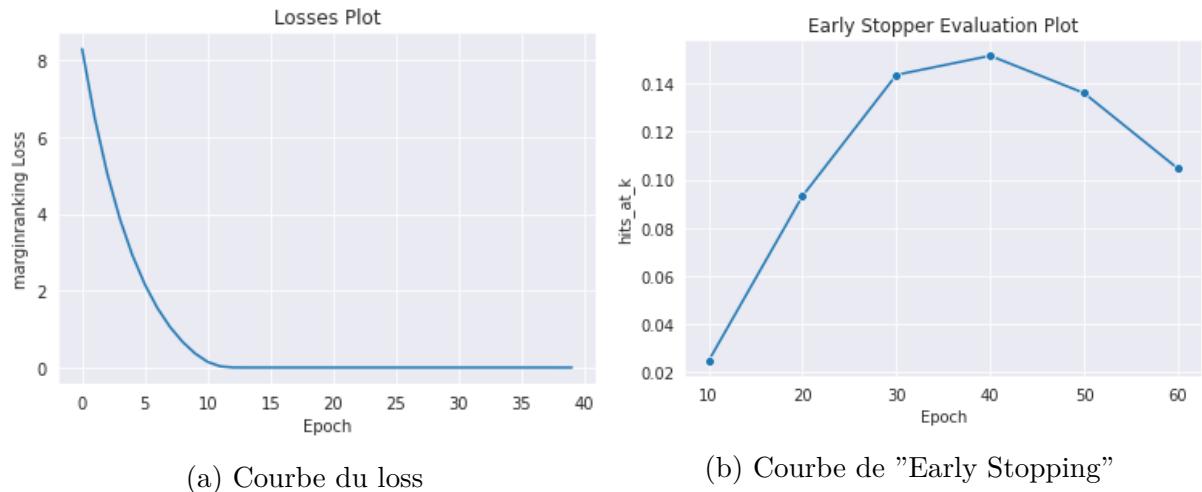


FIG. 3.7 : Développement du loss et du Hits @ k durant l’entraînement de TransH.

TransR

L’entraînement est arrêté après 26 évaluations à l’epoch 260.

Le meilleur résultat Hits @ K=0.13229719484457922 à l’epoch 240. La figure 3.8 montre le développement du loss et du Hits @ K tout au long de l’entraînement.

ComplEx

L’entraînement est arrêté après 23 évaluations à l’epoch 230.

Le meilleur résultat Hits @ K=0.04624715693707354 à l’epoch 210. La figure 3.9 montre le développement du loss et du Hits @ K tout au long de l’entraînement.

ConvE

L’entraînement est arrêté après 6 évaluations à l’epoch 60.

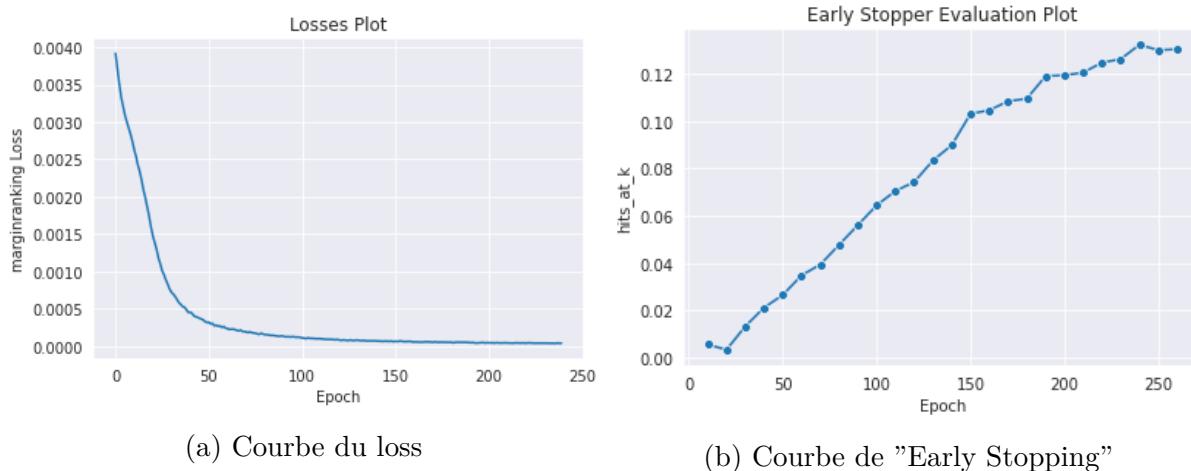


FIG. 3.8 : Développement du loss et du Hits @ k durant l’entraînement de TransR.

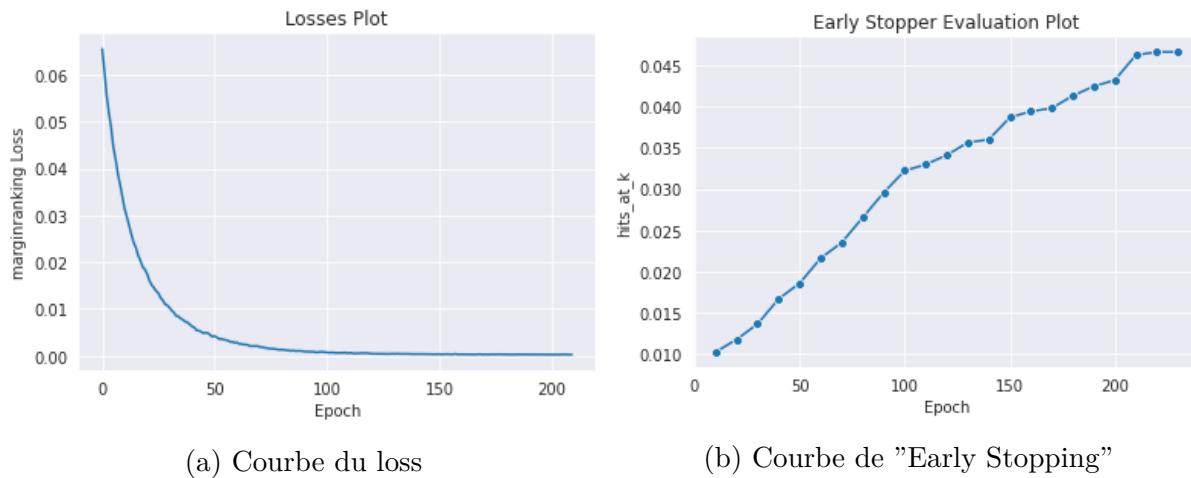


FIG. 3.9 : Développement du loss et du Hits @ k durant l’entraînement de ComplEx.

Le meilleur résultat Hits @ K=0.28203184230477635 à l’epoch 40. La figure 3.10 montre le développement du loss et du Hits @ K tout au long de l’entraînement.

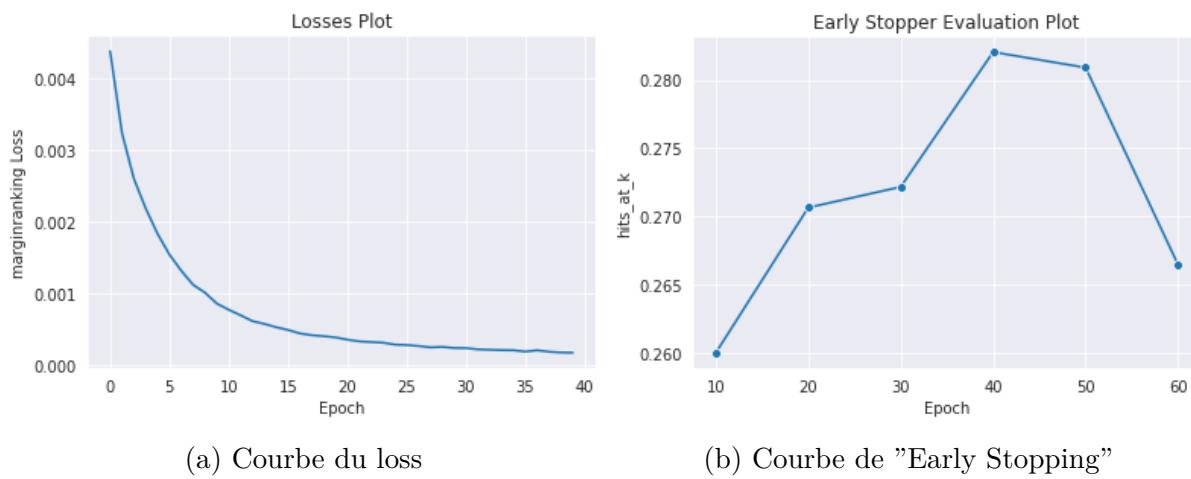


FIG. 3.10 : Développement du loss et du Hits @ k durant l’entraînement de ConvE.

Évaluation des modèles

Pour chaque modèle, nous avons calculé le MR, MRR, AMRI et les Hits @ K. Les résultats de chaque modèle sont reporté dans la table 3.4.

Modèle	MR	MRR	AMRI	Hits@10
TransE	381.40	0.1851	0.7464	0.2440
TransH	607.84	0.0707	0.5955	0.1538
TransR	1063.33	0.0728	0.2920	0.1239
ComplEx	1301.14	0.0322	0.1335	0.0476
ConvE	178.59	0.2542	0.8816	0.4265

TAB. 3.4 : Performances des modèles choisis pour l'apprentissage des connaissances.

Pour les modèles basés sur les translations on remarque que **TransE** a les meilleurs résultats au niveau des métriques d'évaluation, mais en général, ses performance reste inférieur au modèle qui a présenté les meilleurs résultats dans les expériences que nous avons mené, le **ConvE** a vraiment montré de bons résultats qui peuvent être améliorer avec des modifications au niveau des hyper-paramètres.

3.4.3 Prédiction des nouveaux liens

La librairie PyKEEN propose quatre fonctionnalités pour s'adresser à ce problème, presque la seule librairie qui implémente ces fonctionnalités.

- Prédiction des objets pour un sujet et relation spécifiques.
- Prédiction des sujets pour une relation et un objet spécifiques.
- Prédiction des relations pour un sujet et un objet spécifiques.
- Prédiction des nouveaux liens entre les entités sans spécifier ni entités ni relations, mais il faut mentionner que cette tâche est gourmande en terme de ressource et prend une longue période (selon le nombre des entités et des relations dans le graphe).

Pour ce travail, nous avons choisi d'essayer les résultats de la prédiction des nouveaux liens dans notre graphe de connaissances. nous avons choisi de prévoir les cents plus possibles liens. En ce qui suit on va présenter les différents triplets retournés par chaque modèle.

TransE

Parmi les cent triplets inférés, aucun triplet ne contient une relation précise, tous les triples inférés ayant une relation inconnue. Mais nous avons remarqué que malgré les relations ne sont pas connues, TransE a réussi d'inférer qu'il y a une relation entre des entités qui n'étaient pas connecté dans notre graphe original. Par exemple, le modèle a

Chapitre 3. Réalisation et résultats

inféré une relation entre le 'C0020336' qui est l'identifiant de 'Hydroxychloroquine' et le 'C1615607' qui est l'identifiant de 'Influenza A Virus, H1N1 Subtype'. La table 3.5 présente quelques résultats du modèle.

Sujet	Relation	Objet	Score
diabetes insipidus	Inconnue	infectious lung disorder	-3.3193471
Ebola virus	Inconnue	HIV Fusion Inhibitors	-3.3193471
nicotine	Inconnue	Acute gastroenteritis	-3.3193471
Hydroxychloroquine	Inconnue	Influenza A Virus, H1N1 Subtype	-3.3193471
Leukotrienes	Inconnue	Human respiratory syncytial virus	-3.3193471
Malaria, Falciparum	Inconnue	Avian coronavirus	-3.3193471

TAB. 3.5 : Quelques triplets inférés par le modèle TransE.

TransH

De même pour TransH, il n'était pas capable d'inférer les types de relations. Mais cette fois la plupart des triplets inférés par TransH ont été composés de maladies et composants chimiques. La table 3.6 montre quelques exemples des triplets inférés.

Sujet	Relation	Objet	Score
Colorado tick fever virus	Inconnue	iYellow Fever	-3.3304694
Friend Murine Leukemia Virus	Inconnue	Mouse Pox Virus	-3.3360753
Food Allergy	Inconnue	Lysergic Acid Diethylamide	-3.3451164
Colorado tick	Inconnue	fever virus xylene	-3.3489368
Epoxy Compounds	Inconnue	Ischemia	-3.358066
dexamethasone	Inconnue	Ischemia	-3.3592014
Japanese Encephalitis	stand	diltiazem	-3.3720665

TAB. 3.6 : Quelques triplets inférés par le modèle TransH.

Il faut mentionner que après quelques essais, nous avons été capable d'inférer quelques types de relations par exemple : ('Japanese Encephalitis', 'stand', 'diltiazem', -3.3720665), ('Edema', 'stand', 'fructose', -3.3833625) mais ces relations reste non compatible sémantiquement avec le contexte du triplet.

TransR

Par contre les deux modèles précédents, TransR était capable de spécifier les types de quelques relations, ce qui montre que son approche de représentation des relations dans un espace de relations est plus efficace. La table 3.7 montre quelques triplets inférés par ce modèle.

Sujet	Relation	Objet	Score
cisplatin	encourage	phlorofucofuroeckol A	-0.00029573764

Chapitre 3. Réalisation et résultats

Bovine Anaplasmosis	respond	Intolerance to drug	-0.00057641254
amantadine	inactivate	lopinavir / Ritonavir	-0.0014307986
cisplatin	companied	Swab specimen	-0.0017551235
Fatty Liver	host	daclatasvir	-0.0022448506
emetine	motivate	Asymptomatic Infections	-0.0026042976
Tick-Borne Encephalitis	arouse	Cardiac fibrosis	-0.0021618982

TAB. 3.7 : Quelques triplets inférés par le modèle TransR.

ComplEx

ComplEx était capable d’inférer plus de types de relations que TransH, ce qui montre que la notation complexe adoptée par ce modèle est encore plus efficace que celle de TransR. La table 3.8 montre quelques exemples des triplets inférés par le modèle ComplEx.

Sujet	Relation	Objet	Score
Chediak-Higashi Syndrome	surmount	povidone	214.77748
Antibiotics, Aminoglycoside	quantify	Acute hepatitis	209.60573
eicosapentaenoic acid	surmount	Bronchitis, Chronic	206.2193
Arteriosclerosis	arouse	Ascovirus	205.52986
arsenic	optimize	Acute Tubulointerstitial Nephritis	199.36053
Asthma	require	Aarskog syndrome	196.80554
Antioxidants	combat	Extrude	196.39613
Catecholamines	keep	Oncornaviruses	196.2293
acetylcholine	generate	Pulmonary Surfactants	195.55043

TAB. 3.8 : Quelques triplets inférés par le modèle ComplEx.

ConvE

Étant le meilleur parmi les modèles testés dans ce travail au niveau des performances, ConvE était capable aussi d’inférer plus de types de relations que tous les autres modèles. La table 3.9 montre quelques résultats avec les scores de chaque triplet.

Sujet	Relation	Objet	Score
Flavones	defend	Feline Immunodeficiency Virus	7.483093
Mycotoxins	afford	arbidol	7.472434
Flavones	react	Feline Immunodeficiency Virus	7.4006796
Nephritis	elicit	Neu-Laxova syndrome	7.326537
isoproterenol	halt	Cardiac Hypertrophy	7.291639
Poxviridae	precede	SARS-CoV-2	7.2900133
Oral Poliovirus Vaccine	transmit	Porcine epidemic diarrhea virus	7.2835937

TAB. 3.9 : Quelques triplets inférés par le modèle ConvE.

3.5 Conclusion

Dans ce chapitre nous avons vu une simple implémentation pour construire un graphe de connaissances depuis les publications scientifiques partagées dans la base de données CORD19, les publications ont été récupérées et traités, ensuite nous avons segmenté les textes en des phrases. Chaque phrase a été traitée séparément, au premier nous avons extrait les entités de chaque phrase, puis ces entités ont été reliées par des identifiants uniques à partir de l'ontologie UMLS. Pour l'extraction des relations nous avons utilisé un modèle de correspondance sémantique pour extraire les relations (sous forme de verbes) entre les entités extraites. Malheureusement, les entités avait besoin d'un autre traitement, surtout la classification qui n'était pas couvert dans ce travail. Enfin nous avons appliqué des modèles d'apprentissage de représentation pour apprendre les différentes relations entre nos entités, et nous avons utilisé ces modèles par la suite pour inférer d'autres liens manquants. Les modèles avaient la capacité de capturer l'existence des relations entre des entités spécifiques avec une grande certitude, mais ils ont toujours des difficulté au niveau du type de la relation.

Conclusion et perspectives

Tout au long de la préparation de ce projet, la patience et la persévérance étaient les deux clés de la continuation de ce dernier, nous avons essayé de mettre en pratique les connaissances acquises durant ces deux années du cycle du master, afin de construire un graphe de connaissances de la pandémie COVID-19.

Ce projet de fin d'étude consiste en une étude de différentes étapes nécessaires à la construction des graphes de connaissances, à savoir la reconnaissance et la liaison des entités nommées et l'extraction des relations, ainsi qu'une méthode d'application des graphes de connaissances dans la prédiction des nouveaux liens entre les entités existantes. Nous étions intéressés par la création d'un graphe de connaissances biomédicales pour la repositionnement des médicaments pour les maladies, et surtout le COVID-19 qui a mis le monde en une situation critique. Les différentes approches et techniques utilisées durant ce projet ont été choisies en prenant en considérations les contraintes reliées au matériels (puissance de calcul et temps d'exécution). Le travail en entier a été réalisé en utilisant le langage de programmation Python, car c'est le plus utilisé dans le domaine. Pour la reconnaissance des entités nommées, nous avons utilisé une approche basée sur les Transformers en utilisant une variation BERT, ce dernier est connu par ses performances d'état d'art dans plusieurs tâches du traitement automatique du langage naturel, ensuite pour désambiguïser les différentes entités extraites nous avons utilisé le "linker" disponible dans la librairie ScispaCy pour lier les entités extraites avec les concepts de l'UMLS, ensuite ces entités ont été classés selon leurs types sémantiques en utilisant le REST API de l'UMLS.

Pour l'extraction des relations nous avons utilisé une approche d'étiquetage des rôles sémantiques en utilisant encore une fois un variant de BERT, ce dernier considère le verbe entre deux mentions d'entité comme une relation entre ces deux entités. Malgré la simplicité de cette méthode, elle est efficace pour les phrases simples (sujet, verbe, objet), mais trouve des problèmes au niveaux des phrases complexes.

Enfin, après l'extraction des entités et les relations entre elles, nous avons essayé d'utiliser les algorithmes d'apprentissage des représentations des connaissances pour prévoir des nouvelles relations entre les entités existantes, les résultats obtenus étaient intéressants, puisque ces algorithmes avaient la capacité de prévoir des nouvelles relations entre des entités qui sont susceptibles d'être liées en réalité, par exemple des maladies et des médicaments.

Comme perspectives de ce travail, il sera intéressant de travailler sur les relations, il est plus probable de trouver une seule relation qui s'exprime en plusieurs verbes, ce qui rend la tâche d'extraction des relations plus délicate, il est aussi intéressant d'élargir ce travail pour comprendre les différentes maladies et ses symptômes, traitements et causes,

Conclusion et perspectives

cela doit rendre l'apprentissage des représentations des connaissances plus efficace et utile.

Enfin il est important de mentionner que l'utilisation de l'intelligence artificielle dans le domaine biomédical peut être très utile et bénéficiaire, mais il ne faut pas oublier que l'IA est confrontée à de nombreux défis éthiques à savoir :

- Protéger l'autonomie : les humains devraient avoir le contrôle et le dernier mot sur toutes les décisions en matière de santé - elles ne devraient pas être prises entièrement par des machines, et les médecins devraient pouvoir les annuler à tout moment. L'IA ne devrait pas être utilisée pour guider les soins médicaux d'une personne sans son consentement, et ses données devraient être protégées.
- Promouvoir la sécurité humaine : Les développeurs doivent surveiller en permanence tous les outils d'IA pour s'assurer qu'ils fonctionnent comme ils sont censés le faire et ne causent pas de dommages.
- Assurer la transparence : Les développeurs doivent publier des informations sur la conception des outils d'IA. On reproche régulièrement à ces systèmes d'être des "boîtes noires", et il est trop difficile pour les chercheurs et les médecins de savoir comment ils prennent leurs décisions. L'OMS souhaite une transparence suffisante pour qu'ils puissent être pleinement audités et compris par les utilisateurs et les régulateurs.
- Favoriser la responsabilisation : Lorsque quelque chose ne va pas avec une technologie d'IA - par exemple si une décision prise par un outil entraîne un préjudice pour le patient - il devrait y avoir des mécanismes déterminant qui est responsable (comme les fabricants et les utilisateurs cliniques).
- Assurer l'équité : Cela signifie qu'il faut s'assurer que les outils sont disponibles dans plusieurs langues et qu'ils sont formés à partir de divers ensembles de données. Au cours des dernières années, un examen minutieux des algorithmes de santé courants a révélé que certains d'entre eux comportaient des préjugés raciaux.
- Promouvoir une IA durable : Les développeurs devraient pouvoir mettre régulièrement à jour leurs outils, et les institutions devraient disposer de moyens d'ajustement si un outil semble inefficace. Les institutions ou les entreprises ne devraient également introduire que des outils qui peuvent être réparés, même dans les systèmes de santé disposant de peu de ressources.

Bibliographie

- [1] Holly Else. How a torrent of COVID science changed research publishing — in seven charts. *Nature*, 588(7839) :553–553, December 2020.
- [2] Jeffrey Brainard. Scientists are drowning in COVID-19 papers. can new tools keep them afloat ? *Science*, May 2020.
- [3] Lucy Lu Wang, Kyle Lo, Yoganand Chandrasekhar, Russell Reas, Jiangjiang Yang, Doug Burdick, Darrin Eide, Kathryn Funk, Yannis Katsis, Rodney Kinney, Yunyao Li, Ziyang Liu, William Merrill, Paul Mooney, Dewey Murdick, Devvret Rishi, Jerry Sheehan, Zhihong Shen, Brandon Stilson, Alex Wade, Kuansan Wang, Nancy Xin Ru Wang, Chris Wilhelm, Boya Xie, Douglas Raymond, Daniel S. Weld, Oren Etzioni, and Sebastian Kohlmeier. Cord-19: The covid-19 open research dataset, 2020.
- [4] Carol Friedman, Pauline Kra, and Andrey Rzhetsky. Two biomedical sublanguages : a description based on the theories of zellig harris. *Journal of Biomedical Informatics*, 35(4) :222–235, 2002. Sublanguage - Zellig Harris Memorial.
- [5] Andre Lamurias and Francisco M. Couto. Text mining for bioinformatics using biomedical literature. In Shoba Ranganathan, Michael Gribkov, Kenta Nakai, and Christian Schönbach, editors, *Encyclopedia of Bioinformatics and Computational Biology*, pages 602–611. Academic Press, Oxford, 2019.
- [6] D. R. Swanson. Medical literature as a potential source of new knowledge. *Bull Med Libr Assoc*, 78(1) :29–37, Jan 1990.
- [7] Ralph A. DiGiacomo, Joel M. Kremer, and Dhiraj M. Shah. Fish-oil dietary supplementation in patients with raynaud's phenomenon : A double-blind, controlled, prospective study. *The American Journal of Medicine*, 86(2) :158–164, January 1989.
- [8] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space, 2013.
- [9] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove : Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.
- [10] David M. Blei. Probabilistic topic models. *Communications of the ACM*, 55(4) :77–84, April 2012.

Bibliographie

- [11] George Buchanan and Fernando Loizides. Investigating document triage on paper and electronic media. In *Research and Advanced Technology for Digital Libraries*, pages 416–427. Springer Berlin Heidelberg, 2007.
- [12] David Nadeau and Satoshi Sekine. A survey of named entity recognition and classification. *Lingvisticae Investigationes. International Journal of Linguistics and Language Resources*, 30(1) :3–26, August 2007.
- [13] Yoshimasa Tsuruoka, John McNaught, and Sophia Ananiadou. Normalizing biomedical terms by minimizing ambiguity and variability. *BMC Bioinformatics*, 9(S3), April 2008.
- [14] Sophia Ananiadou, Sampo Pyysalo, Jun’ichi Tsujii, and Douglas B. Kell. Event extraction for systems biology by text mining the literature. *Trends in Biotechnology*, 28(7) :381–390, July 2010.
- [15] Christopher D. Manning and Hinrich D. Schütze. *Foundations of statistical natural language processing*. MIT, 2008.
- [16] K. Verspoor, K. B. Cohen, A. Lanfranchi, C. Warner, H. L. Johnson, C. Roeder, J. D. Choi, C. Funk, Y. Malenkiy, M. Eckert, N. Xue, W. A. Baumgartner, M. Bada, M. Palmer, and L. E. Hunter. A corpus of full-text journal articles is a robust evaluation tool for revealing differences in performance of biomedical natural language processing tools. *BMC Bioinformatics*, 13:207, Aug 2012.
- [17] L. H. Smith, L. Tanabe, T. Rindflesch, and W. J. Wilbur. Medtag : A collection of biomedical annotations. In *Proceedings of the ACL-ISMB Workshop on Linking Biological Literature, Ontologies and Databases : Mining Biological Semantics*, ISMB ’05, page 32–37, USA, 2005. Association for Computational Linguistics.
- [18] J. Li, Y. Sun, R. J. Johnson, D. Sciaky, C. H. Wei, R. Leaman, A. P. Davis, C. J. Mattingly, T. C. Wiegers, and Z. Lu. BioCreative V CDR task corpus : a resource for chemical disease relation extraction. *Database (Oxford)*, 2016, 2016.
- [19] J.-D. Kim, T. Ohta, Y. Tateisi, and J. Tsujii. GENIA corpus—a semantically annotated corpus for bio-textmining. *Bioinformatics*, 19(suppl₁) : i180 – –i182, 072003.
- [20] Martin Krallinger, Obdulia Rabal, Florian Leitner, Miguel Vazquez, David Salgado, Zhiyong Lu, Robert Leaman, Yanan Lu, Donghong Ji, Daniel M Lowe, Roger A Sayle, Riza Theresa Batista-Navarro, Rafal Rak, Torsten Huber, Tim Rocktäschel, Sérgio Matos, David Campos, Buzhou Tang, Hua Xu, Tsendsuren Munkhdalai, Keun Ho Ryu, SV Ramanan, Senthil Nathan, Slavko Žitnik, Marko Bajec, Lutz Weber, Matthias Irmer, Saber A Akhondi, Jan A Kors, Shuo Xu, Xin An, Utpal Kumar Sikdar, Asif Ekbal, Masaharu Yoshioka, Thaer M Dieb, Miji Choi, Karin Verspoor, Madian Khabsa, C Lee Giles, Hongfang Liu, Komandur Elayavilli Ravikumar, Andre Lamurias, Francisco M Couto, Hong-Jie Dai, Richard Tzong-Han Tsai, Caglar Ata, Tolga Can, Anabel Usié, Rui Alves, Isabel Segura-Bedmar, Paloma Martínez, Julen Oyarzabal, and Alfonso Valencia. The CHEMDNER corpus of chemicals and drugs and its annotation principles. *Journal of Cheminformatics*, 7(S1), January 2015.

Bibliographie

- [21] Ming-Siang Huang, Po-Ting Lai, Pei-Yen Lin, Yu-Ting You, Richard Tzong-Han Tsai, and Wen-Lian Hsu. Biomedical named entity recognition and linking datasets : survey and our recent development. *Briefings in Bioinformatics*, 21(6) :2219–2238, Jun 2020.
- [22] María Herrero-Zazo, Isabel Segura-Bedmar, Paloma Martínez, and Thierry Declerck. The ddi corpus : An annotated corpus with pharmacological substances and drug–drug interactions. *Journal of Biomedical Informatics*, 46(5) :914–920, 2013.
- [23] Sampo Pyysalo, Tomoko Ohta, Rafal Rak, Andrew Rowley, Hong-Woo Chun, Sung-Jae Jung, Sung-Pil Choi, Jun’ichi Tsujii, and Sophia Ananiadou. Overview of the cancer genetics and pathway curation tasks of bionlp shared task 2013. *BMC bioinformatics*, 16(10) :1–19, 2015.
- [24] Sampo Pyysalo, Tomoko Ohta, Makoto Miwa, Han-Cheol Cho, Jun’ichi Tsujii, and Sophia Ananiadou. Event extraction across multiple levels of biological organization. *Bioinformatics*, 28(18) :i575–i581, 09 2012.
- [25] Jing Li, Aixin Sun, Jianglei Han, and Chenliang Li. A survey on deep learning for named entity recognition, 2020.
- [26] S. Sekine and Chikashi Nobata. Definition, dictionaries and tagger for extended named entity hierarchy. In *LREC*, 2004.
- [27] Shaodian Zhang and Noémie Elhadad. Unsupervised biomedical named entity recognition : Experiments with clinical and biological texts. *Journal of Biomedical Informatics*, 46(6) :1088–1098, December 2013.
- [28] Ji-Hwan Kim and Philip C. Woodland. A rule-based named entity recognition system for speech input. In *Proc. 6th International Conference on Spoken Language Processing (ICSLP 2000)*, pages vol. 1, 528–531, 2000.
- [29] Daniel Hanisch, Katrin Fundel, Heinz-Theodor Mevissen, Ralf Zimmer, and Juliane Fluck. ProMiner : rule-based protein and gene entity recognition. *BMC Bioinformatics*, 6(S1), May 2005.
- [30] Alexandra Pomares Quimbaya, Alejandro Sierra Múnera, Rafael Andrés González Rivera, Julián Camilo Daza Rodríguez, Oscar Mauricio Muñoz Velandia, Angel Alberto García Peña, and Cyril Labbé. Named entity recognition over electronic health records through a combined dictionary-based approach. *Procedia Computer Science*, 100:55–61, 2016.
- [31] Michael Collins and Yoram Singer. Unsupervised models for named entity classification. In *1999 Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*, 1999.
- [32] Oren Etzioni, Michael Cafarella, Doug Downey, Ana-Maria Popescu, Tal Shaked, Stephen Soderland, Daniel S. Weld, and Alexander Yates. Unsupervised named-entity extraction from the web : An experimental study. *Artificial Intelligence*, 165(1) :91–134, June 2005.
- [33] GuoDong Zhou and Jian Su. Named entity recognition using an hmm-based chunk tagger. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, ACL ’02*, page 473–480, USA, 2002. Association for Computational Linguistics.

Bibliographie

- [34] Sean R Eddy. Hidden markov models. *Current Opinion in Structural Biology*, 6(3) :361–365, June 1996.
- [35] J. R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1) :81–106, March 1986.
- [36] M.A. Hearst, S.T. Dumais, E. Osuna, J. Platt, and B. Scholkopf. Support vector machines. *IEEE Intelligent Systems and their Applications*, 13(4) :18–28, 1998.
- [37] J. Lafferty, A. McCallum, and Fernando Pereira. Conditional random fields : Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001.
- [38] M.S. Salleh, S.A. Asmai, H. Basiron, and S. Ahmad. Named entity recognition using fuzzy c-means clustering method for malay textual data analysis. *Journal of Telecommunication, Electronic and Computer Engineering (JTEC)*, 10(2-7) :121–126, Jul. 2018.
- [39] Thien Huu Nguyen, Avirup Sil, Georgiana Dinu, and Radu Florian. Toward mention detection robustness with recurrent neural networks, 2016.
- [40] Feifei Zhai, Saloni Potdar, Bing Xiang, and Bowen Zhou. Neural models for sequence chunking. *Proceedings of the AAAI Conference on Artificial Intelligence*, 31(1), Feb. 2017.
- [41] Onur Kuru, Ozan Arkan Can, and Deniz Yuret. CharNER : Character-level named entity recognition. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics : Technical Papers*, pages 911–921, Osaka, Japan, December 2016. The COLING 2016 Organizing Committee.
- [42] Quan Tran, Andrew MacKinlay, and Antonio Jimeno Yepes. Named entity recognition with stack residual LSTM and trainable bias decoding. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 566–575, Taipei, Taiwan, November 2017. Asian Federation of Natural Language Processing.
- [43] Abbas Ghaddar and Philippe Langlais. Robust lexical features for improved neural network named-entity recognition, 2018.
- [44] Zhanming Jie and Wei Lu. Dependency-guided lstm-crf for named entity recognition, 2019.
- [45] Di Lu, Leonardo Neves, Vitor Carvalho, Ning Zhang, and Heng Ji. Visual attention model for name tagging in multimodal social media. In *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers), pages 1990–1999. Association for Computational Linguistics (ACL), 2018. Funding Information : This work was partially supported by the U.S. DARPA AIDA Program No. FA8750-18-2-0014 and U.S. ARL NS-CTA No. W911NF-09-2-0053. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for Government purposes notwithstanding any

Bibliographie

copyright notation here on. Publisher Copyright : © 2018 Association for Computational Linguistics ; 56th Annual Meeting of the Association for Computational Linguistics, ACL 2018 ; Conference date : 15-07-2018 Through 20-07-2018.

- [46] Ronan Collobert, Jason Weston, Leon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. Natural language processing (almost) from scratch, 2011.
- [47] Zhiheng Huang, Wei Xu, and Kai Yu. Bidirectional lstm-crf models for sequence tagging. 2015.
- [48] Jason P. C. Chiu and Eric Nichols. Named entity recognition with bidirectional lstm-cnns, 2016.
- [49] Y. Wu, M. Jiang, J. Lei, and H. Xu. Named Entity Recognition in Chinese Clinical Text Using Deep Neural Network. *Stud Health Technol Inform*, 216:624–628, 2015.
- [50] Xuezhe Ma and Eduard Hovy. End-to-end sequence labeling via bi-directional lstm-cnns-crf, 2016.
- [51] Peng-Hsuan Li, Ruo-Ping Dong, Yu-Siang Wang, Ju-Chieh Chou, and Wei-Yun Ma. Leveraging linguistic structures for named entity recognition with bidirectional recursive neural networks. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 2664–2669, Copenhagen, Denmark, September 2017. Association for Computational Linguistics.
- [52] Matthew E. Peters, Waleed Ammar, Chandra Bhagavatula, and Russell Power. Semi-supervised sequence tagging with bidirectional language models, 2017.
- [53] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need, 2017.
- [54] Alec Radford and Karthik Narasimhan. Improving language understanding by generative pre-training. 2018.
- [55] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT : Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics.
- [56] Olivier Bodenreider. The Unified Medical Language System (UMLS) : integrating biomedical terminology. *Nucleic Acids Research*, 32(suppl_1) : D267 – – D270, 012004.
- [57] Medical subject headings. <https://www.ncbi.nlm.nih.gov/mesh/meshhome.html>.
- [58] Rxnorm. <https://www.ncbi.nlm.nih.gov/research/umls/rxnorm/index.html>.
- [59] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and

Bibliographie

- G. Sherlock. Gene ontology : tool for the unification of biology. The Gene Ontology Consortium. *Nat Genet*, 25(1) :25–29, May 2000.
- [60] P. N. Robinson, S. Köhler, S. Bauer, D. Seelow, D. Horn, and S. Mundlos. The Human Phenotype Ontology : a tool for annotating and analyzing human hereditary disease. *Am J Hum Genet*, 83(5) :610–615, Nov 2008.
- [61] Ben Hachey, Will Radford, Joel Nothman, Matthew Honnibal, and James R. Curran. Evaluating entity linking with wikipedia. *Artificial Intelligence*, 194:130–150, January 2013.
- [62] Sayali Kulkarni, Amit Singh, Ganesh Ramakrishnan, and Soumen Chakrabarti. Collective annotation of wikipedia entities in web text. In *Proceedings of the 15th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD ’09, page 457–466, New York, NY, USA, 2009. Association for Computing Machinery.
- [63] W. Zhang, Y.C. Sim, J. Su, and C.L. Tan. Entity linking with effective acronym expansion, instance selection and topic modeling, 2011.
- [64] Mark Dredze, Paul McNamee, Delip Rao, Adam Gerber, and Tim Finin. Entity disambiguation for knowledge base population. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 277–285, Beijing, China, August 2010. Coling 2010 Organizing Committee.
- [65] Xianpei Han and Jun Zhao. Nlpr_kbp in tac 2009 kbp track : A two-stage method to entity linking. *Theory and Applications of Categories*, 2009.
- [66] Sean Monahan, John Lehmann, Timothy Nyberg, Jesse Plymale, and A. Jung. Cross-lingual cross-document coreference with entity linking. *Theory and Applications of Categories*, 2011.
- [67] Wei Zhang, Jian Su, Chew Lim Tan, and Wen Ting Wang. Entity linking leveraging automatically generated annotation. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 1290–1298, Beijing, China, August 2010. Coling 2010 Organizing Committee.
- [68] Heng Ji, Ralph Grishman, and Hoa Dang. Overview of the TAC2011 knowledge base population track. In *TAC 2011 Proceedings Papers*, 2011.
- [69] Johannes Hoffart, Mohamed Amir Yosef, Ilaria Bordino, Hagen Fürstenau, Manfred Pinkal, Marc Spaniol, Bilyana Taneva, Stefan Thater, and Gerhard Weikum. Robust disambiguation of named entities in text. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 782–792, Edinburgh, Scotland, UK., July 2011. Association for Computational Linguistics.
- [70] Marti A. Hearst. Automatic acquisition of hyponyms from large text corpora. In *Proceedings of the 14th Conference on Computational Linguistics - Volume 2*, COLING ’92, page 539–545, USA, 1992. Association for Computational Linguistics.
- [71] Sergey Brin. Extracting patterns and relations from the world wide web. In *Lecture Notes in Computer Science*, pages 172–183. Springer Berlin Heidelberg, 1999.

- [72] Takaaki Hasegawa, Satoshi Sekine, and Ralph Grishman. Discovering relations among named entities from large corpora. In *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)*, pages 415–422, Barcelona, Spain, July 2004.
- [73] Sachin Pawar, Girish K. Palshikar, and Pushpak Bhattacharyya. Relation extraction : A survey, 2017.
- [74] Mike Mintz, Steven Bills, Rion Snow, and Dan Jurafsky. Distant supervision for relation extraction without labeled data. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP : Volume 2 - Volume 2*, ACL '09, page 1003–1011, USA, 2009. Association for Computational Linguistics.
- [75] ChunYang Liu, WenBo Sun, WenHan Chao, and WanXiang Che. Convolution neural network for relation extraction. In *Advanced Data Mining and Applications*, pages 231–242. Springer Berlin Heidelberg, 2013.
- [76] Yoon Kim. Convolutional neural networks for sentence classification, 2014.
- [77] Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. A convolutional neural network for modelling sentences, 2014.
- [78] Kun Xu, Yansong Feng, Songfang Huang, and Dongyan Zhao. Semantic relation classification via convolutional neural networks with simple negative sampling. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 536–540, Lisbon, Portugal, September 2015. Association for Computational Linguistics.
- [79] Linlin Wang, Zhu Cao, Gerard de Melo, and Zhiyuan Liu. Relation classification via multi-level attention CNNs. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1298–1307, Berlin, Germany, August 2016. Association for Computational Linguistics.
- [80] Dongxu Zhang and Dong Wang. Relation classification via recurrent neural network, 2015.
- [81] Yan Xu, Lili Mou, Ge Li, Yunchuan Chen, Hao Peng, and Zhi Jin. Classifying relations via long short term memory networks along shortest dependency paths. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1785–1794, Lisbon, Portugal, September 2015. Association for Computational Linguistics.
- [82] Suncong Zheng, Jiaming Xu, Peng Zhou, Hongyun Bao, Zhenyu Qi, and Bo Xu. A neural network framework for relation extraction : Learning entity semantic and relation pattern. *Knowledge-Based Systems*, 114:12–23, 2016.
- [83] Rui Cai, Xiaodong Zhang, and Houfeng Wang. Bidirectional recurrent convolutional neural network for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 756–765, Berlin, Germany, August 2016. Association for Computational Linguistics.

Bibliographie

- [84] Zhen Wang, Jianwen Zhang, Jianlin Feng, and Zheng Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, page 1112–1119. AAAI Press, 2014.
- [85] Antoine Bordes, Nicolas Usunier, Alberto García-Durán, J. Weston, and Oksana Yakhnenko. Translating embeddings for modeling multi-relational data. In *NIPS*, 2013.
- [86] Yankai Lin, Zhiyuan Liu, Maosong Sun, Yang Liu, and Xuan Zhu. Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, page 2181–2187. AAAI Press, 2015.
- [87] Maximilian Nickel, Volker Tresp, and Hans-Peter Kriegel. A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, ICML'11, page 809–816, Madison, WI, USA, 2011. Omnipress.
- [88] Bishan Yang, Wen tau Yih, Xiaodong He, Jianfeng Gao, and Li Deng. Embedding entities and relations for learning and inference in knowledge bases, 2015.
- [89] Maximilian Nickel, Lorenzo Rosasco, and Tomaso Poggio. Holographic embeddings of knowledge graphs, 2015.
- [90] Théo Trouillon, Johannes Welbl, Sebastian Riedel, Éric Gaussier, and Guillaume Bouchard. Complex embeddings for simple link prediction, 2016.
- [91] Tim Dettmers, Pasquale Minervini, Pontus Stenetorp, and S. Riedel. Convolutional 2d knowledge graph embeddings. In *AAAI*, 2018.
- [92] Dai Quoc Nguyen, Tu Dinh Nguyen, Dat Quoc Nguyen, and Dinh Phung. A novel embedding model for knowledge base completion based on convolutional neural network. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 2 (Short Papers)*, 2018.
- [93] Matthew Honnibal, Ines Montani, Sofie Van Landeghem, and Adriane Boyd. spaCy : Industrial-strength Natural Language Processing in Python, 2020.
- [94] Mark Neumann, Daniel King, Iz Beltagy, and Waleed Ammar. ScispaCy : Fast and Robust Models for Biomedical Natural Language Processing. In *Proceedings of the 18th BioNLP Workshop and Shared Task*, pages 319–327, Florence, Italy, August 2019. Association for Computational Linguistics.
- [95] Matt Gardner, Joel Grus, Mark Neumann, Oyvind Tafjord, Pradeep Dasigi, Nelson F. Liu, Matthew Peters, Michael Schmitz, and Luke S. Zettlemoyer. AllenNLP : A deep semantic natural language processing platform. 2017.
- [96] Peng Shi and Jimmy Lin. Simple bert models for relation extraction and semantic role labeling, 2019.

Bibliographie

- [97] Mehdi Ali, Max Berrendorf, Charles Tapley Hoyt, Laurent Vermue, Sahand Sharifzadeh, Volker Tresp, and Jens Lehmann. Pykeen 1.0: A python library for training and evaluating knowledge graph embeddings, 2020.
- [98] Max Berrendorf, Evgeniy Faerman, Laurent Vermue, and Volker Tresp. On the ambiguity of rank-based evaluation of entity alignment or link prediction methods, 2021.