Collège Doctoral
Languedoc-Roussillon

THÈSE

Pour obtenir le grade de
Docteur

Délivrée par **Université de Montpellier**

Préparée au sein de l'école doctorale École doctorale I2S
Et de l'unité de recherche LIRMM UMR 5506

Spécialité : **informatique**

Présenté par : **Massinissa ATMANI**

Extraction de relations sémantiques dans un contexte de veille et d'intelligence d'entreprise

Soutenue le 19/04/2022 devant le jury composé de

M. Patrice BELLOT, PU, Aix-Marseille Université	Rapporteur
M. Didier SCHWAB, MCF HDR, Université Grenoble Alpes	Rapporteur
Mme. Nathalie AUSSENAC-Gilles, DR CNRS, IRIT	Examinatrice
Mme. Marianne HUCHARD, PU, U. de Montpellier	Présidente
Mme. Anne LAURENT, PU, U. de Montpellier	Examinatrice
M. Mathieu LAFOURCADE, MCF HDR, U. de Montpellier	Directeur

UNIVERSITÉ MONTPELLIER

Résumé

Dans cette thèse, nous abordons la question de l'extraction de relations sémantiques depuis des textes avec une attention particulière portée aux bases de connaissances afin de réduire la quantité de données nécessaires pour l'entraînement d'un modèle neuronal basé sur l'apprentissage profond. Ainsi, nous essayons de concevoir un modèle d'extraction de relations en prenant en compte les aspects suivants : 1) sobriété - entraîner efficacement le modèle avec peu de données d'apprentissage en volume, 2) adaptabilité - pouvoir adapter le modèle à différents domaines, alors que la plupart des approches à base d'apprentissage automatique ne sont que difficilement adaptables, et 3) explicabilité - pouvoir fournir automatiquement des éléments d'explication des résultats produits par le modèle.

De ce fait, nous proposons de normaliser des triplets ayant une représentation prédicat-arguments à l'aide de ressources externes (bases de connaissances) de façon à obtenir des performances comparables aux approches basées sur l'apprentissage automatique sans reposer sur un quelconque corpus textuel d'apprentissage. Les inférences de notre approche sont interprétables car le prédicat du triplet est censé capturer le déclencheur (un segment textuel) de la relation présente dans le texte. Pour cela, nous proposons un modèle d'OpenIE (extraction ouverte d'informations) pour extraire les triplets (prédicat-arguments), nommé *UD2OIE*, qui exploite seulement l'analyse syntaxique afin de respecter la contrainte d'adaptation au domaine autant que possible. Nous montrons également que notre système surpasse l'état de l'art actuel sur différents domaines de spécialités et que les approches actuelles basées sur les modèles de langage (*BERT*) ne respectent que peu la contrainte d'adaptation au domaine. Nous abordons également le problème de l'hypothèse distributionnelle, sur laquelle le modèle de langage repose, qui n'est pas suffisante pour capturer les propriétés relationnelles et sémantiques entre les termes. En effet, les approches actuelles différenciant mal la similarité sémantique des autres relations, nous proposons le modèle *Jeux2BERT* tirant profit de bases de connaissances afin que les représentations vectorielles des mots puissent modé-

liser à la fois leurs propriétés distributionnelles et relationnelles. Nous illustrons aussi l'apport de notre méthodologie relativement à l'état de l'art. Finalement, nous présentons un modèle hybride d'extraction de relations *Jeux2BERT+OIE* qui exploite le modèle *Jeux2BERT* pour normaliser les triplets obtenus par le modèle *UD2OIE*. Nous appliquons notre modèle d'extraction de relations sémantiques au domaine de veille technologique et d'intelligence d'entreprise.

Mots clés : *Traitement automatique du langage*, *Extraction de relations sémantiques*, *Apprentissage automatique*, *Base de connaissances*.

Remerciements

Avant tout, une gratitude particulière va à **Pendaries Fabien** mon manager au sein de l'entreprise Mantu, qui a toujours été là pour m'éclairer et me montrer comment bien faire les choses. Nos discussions de longue date et votre véritable charité vont beaucoup nous manquer !

Un merci spécial au responsable du laboratoire R et D de Amaris/Mantu, **Renaud PAWLACK** qui m'a donné l'opportunité en or de faire cette merveilleuse thèse.

Le **Research Developement Lab d'Amaris** est un lieu de travail formidable, et je remercie mes collègues chercheurs pour leur bonne humeur au quotidien et pour m'avoir donné un coup de main dès qu'ils le pouvaient.

Je suis reconnaissant à l'entreprise **Amaris/Mantu** de m'avoir motivé à mener des recherches au sein de son laboratoire R et D et également de m'avoir fourni des installations d'infrastructure et de nombreuses autres ressources nécessaires à mes recherches.

Avec un immense plaisir et un profond sentiment de gratitude, je souhaite exprimer mes sincères remerciements à mon superviseur **Mathieu LAFOURCADE**, MCF HDR à l'université de Montpellier, sans sa motivation et ses encouragements continus, cette recherche ne serait pas complétée avec succès.

Je tiens à remercier les rapporteurs ainsi que les examinateurs pour leur disponibilité.

Je souhaite exprimer ma profonde gratitude à mes parents pour tous les sacrifices qu'ils ont consentis au cours de mes recherches et pour me fournir un soutien moral et des encouragements chaque fois que nécessaire.

Place : Antibes

Date : 17/02/2022 **Massinissa ATMANI**

Table des matières

Résumé i
Remerciements iii
Liste des figures viii
Liste des tables x
Liste des termes et des abréviations xiii

1 Introduction générale **1**
1.1 Contexte RD 1
1.2 Motivation 1
1.3 Défis 2
1.4 Définition du problème 3
1.5 Questions de recherche 4
1.6 Contributions 4

2 État de l'art sur Extraction de Relations **8**
2.1 Approches à base de règles 8
2.2 Approches semi-supervisées 9
2.2.1 Bootsraping 9
2.2.2 Supervision distant 13
2.3 Approches supervisées 15
2.3.1 Approches à base de noyau 17
2.3.2 Approches à base de réseaux de neurones 21
2.3.2.1 Approches à base de pipeline 21
2.3.2.2 Approches à base d'apprentissage conjoint 24
2.3.2.3 Approches à base d'apprentissage conjoint avec le paradigme multi-tâches 27
2.4 Approches Few-Shot 28
2.5 Approches non-supervisées 31

2.6 Résumé 37

3 Open Information Extraction 38
3.1 Introduction 38
3.2 État de L'art 39
3.2.1 Première génération 40
3.2.2 Systèmes à base de réseaux de neurones 43
3.2.3 OpenIE multilingue 45
3.3 Hypothèse 46
3.4 Réseaux de neurones récurrents 50
3.5 Champ aléatoire conditionnel 52
3.6 Méthodologie 53
3.6.1 Définition du Problème 53
3.6.2 Approche 53
3.7 Modèle et architecture 54
3.7.1 Représentation des entrées 55
3.7.2 Extraction de la structure Prédicat-Argument 55
3.7.3 Étiquetage des arguments 59
3.8 Expérimentations 59
3.8.1 Données d'entraînements 61
3.8.2 Hyperparameters 61
3.8.3 Stratégie d'évaluation 61
3.8.4 Référentiel d'évaluation 62
3.8.5 Étude d'ablation 64
3.8.6 Références 64
3.8.7 Performance du temps d'exécution 65
3.9 Résultats et Analyse 65
3.9.1 Résultats de performance monolingue 65
3.9.2 Résultats de performance cross-lingues 67
3.9.3 Résultats de l'étude d'ablation 67
3.9.4 Adaptation au domaine 68
3.9.5 Analyse d'exécution 69
3.9.6 Analyse des erreurs. 70
3.10 Conclusion 71

4 Jeux2BERT 73
4.1 Définition du problème et organisation de ce chapitre 73
4.2 Vecteur de mots 74
4.2.1 Introduction 74
4.2.2 Sac de mots 75
4.2.3 Word2Vec 76
4.3 Vecteur de mots avec représentation contextualisée 79
4.4 Base de connaissances et Réseaux lexico-sémantiques 82
4.5 État de L'art de l'affinement de vecteurs de mots à l'aide de bases de connaissances 83
4.6 Méthode et Modèle proposé 87
4.6.1 Corpus d'apprentissage 88
4.6.2 Méthodologie 94
4.7 Expérimentations 105
4.7.1 Données d'apprentissage et d'évaluation 105
4.7.2 Modèle et Hyper-paramètres 106
4.7.3 Systèmes de référence 107
4.7.4 Étude d'ablation 107
4.7.5 Évaluations 108
4.7.6 Résultats et Discussions 109
4.8 Conclusion 116

5 Extraction de Relations via OpenIE et Vecteurs de Mots 117
5.1 Introduction 117
5.2 Redéfinition du problème 118
5.3 Méthode proposée 121
5.3.1 Prétraitement 122
5.3.2 Inférences 126
5.3.3 Normalisation 133
5.4 Expérimentations et Analyse 139
5.4.1 Données d'évaluation et protocole 139
5.4.2 Résultats 142
5.4.3 Analyse des erreurs 144
5.5 Conclusion 148

6 Conclusion générale **149**
6.1 Principales conclusions 149
6.2 Apport des connaissances à l'extraction de relations 150
6.3 Résumé de nos contributions au domaine 151
6.4 Quelques perspectives 151
6.5 Pour finir 152
Références 152

Appendices

Annexe A Liste des architectures **174**

Annexe B Sélection des relations de JDM pour l'apprentissage **179**

Annexe C Lien vers les ressources **199**

Table des figures

1 Exemple d'une information d'acquisition. 2
2 Représentation de la relation sémantique d'acquisition. 2
3 Illustration d'un modèle de réseau de neurones convolutionel. . . . 22
4 Schéma d'un réseau de neurones récurrents à une unité reliant l'entrée et la sortie du réseau. 51
5 Schéma d'un réseau de neurones récurrents à mémoire court-terme et long terme 51
6 Objectif de la vectorisation dense. 56
7 Extraction de la structure prédicat-arguments. 58
8 Étiqueteur des arguments. 60
9 Modèle CBOW : Illustration adaptée de l'article de Mikolov (Mikolov et al. 2013). 77
10 Modèle Skip-Gram. Illustration adaptée de l'article de Mikolov (Mikolov et al. 2013). 77
11 Architecture du Word2Vec. 78
12 Illustration du réglage fin de KG-BERT pour prédire la plausibilité d'un triplet. Illustration par Yao, Mao and Luo (2019). 85
13 Illustration du réglage fin de KG-BERT pour prédire la relation entre deux entités. Illustration par Yao, Mao and Luo (2019). . . . 85
14 Architecture de la méthode d'apprentissage multi-tâche proposée par Kim et al. (2020) 88
15 Illustration du chemin existant dans le graphe entre le couple d'entité d'évaluation. 93
16 Architecture de la méthode d'apprentissage multi-tâche de *Jeux2BERT*. 105

17 Exemple de reconnaissance d'entités nommées dans la plate-forme de démonstration de Stanza. Les termes *'Altran'* et *'Capgemini'* sont identifiés comme étant des organisations. Techniquement le résultat fourni prend la forme d'une table associant des segments des textes (chaînes de caractères et positions) à leurs types. 120
18 Approche générale pour l'extraction de relations sémantiques. . . . 122
19 Fonctionnement simplifié de notre système d'extraction de relations sémantiques. 123
20 Fonctionnement de notre système d'extraction de relations sémantiques. L'exemple inclut la résolution de coréférence dans la chaîne de traitement. 124
21 Exemple de phrases contenant un faux signal de relations sémantiques. 125
22 Algorithme de l'inférence avec appositions. 126
23 Exemple d'une phrase liée à l'inférence avec apposition. 127
24 Exemple de co-référence dans la plateforme de démonstration AllenNLP. 127
25 Algorithme d'inférence avec les coréférences. 128
26 Exemple de phrase liée à une résolution coréférence. 128
27 Algorithme d'inférence avec les synonymes. 130
28 Exemple de phrases liée à l'inférence par synonymes 130
29 Phrase comprenant des tuples d'OpenIE imbriqués 131
30 Algorithme de détection du contexte. 132
31 Exemple de normalisation 137
32 Extraction de la structure prédicat-arguments. 175
33 Étiqueteur des arguments. 176
34 Architecture de la méthode d'apprentissage multi-tâche de *Jeux2BERT*. 177
35 Figure illustrant le fonctionnement de notre système d'extraction de relations sémantiques. L'exemple inclut la résolution de coréférence dans la chaîne de traitement. 178

Liste des tableaux

2.1 Récapitulatif des approches de l'état de l'art pour l'extraction de relations 37
3.1 Exemples d'extraction d'information ouvert(OpenIE). 39
3.2 Exemples de triplets d'Open Information Extraction. 49
3.3 Exemple de la représentation de la sortie du modèle. 53
3.4 Données d'entraînements. 61
3.5 hyper-paramètres. 62
3.6 Référentiels d'évaluation. 63
3.7 Exemples d'extraction de UD2OIE pour chaque langue. 66
3.8 Résultats de l'évaluation monolingue des systèmes d'OpenIE sur le référentiel standard CARB (anglais). 67
3.9 Résultats de l'évaluation multilingue des systèmes d'OpenIE sur les différents référentiels. 68
3.10 Résultats de l'étude d'impact de l'architecture. 68
3.11 Résultats de l'évaluation des systèmes de référence sur le référentiel français de Wikipédia. 69
3.12 Temps d'exécution des systèmes OpenIE. 70
3.13 Analyse et types des erreurs les plus récurrentes. 72
4.1 Exemples de prédictions d'un modèle entraîné à partir de triplets issus de JeuxDeMots. 86
4.2 Exemple d'absence de triplet de support pour le triplet d'évaluation. 89
4.3 Exemple de présence de triplet de support pour le triplet d'évaluation. 90
4.4 Premier exemple de fuite de donnée. 90
4.5 Deuxième type de fuite de donnée. 91
4.6 Troisième type de fuite de donnée. 91
4.7 Quatrième type de fuite de donnée. 92

4.8 Illustration de la flexibilité de la stratégie de filtrage. 92
4.9 Description des données d'apprentissage et d'évaluation. 93
4.10 Description finale des données d'apprentissage et d'évaluation. . . 94
4.11 Composition des données d'apprentissage et d'évaluation. 105
4.12 Description des données d'apprentissage pour chaque tâche. 106
4.13 Liste des hyper-paramètres du modèle Jeux2BERT. 107
4.14 Résultats de l'évaluation de la tâche de prédiction de lien. 109
4.15 Résultats de l'évaluation de la tâche de prédiction de relations. . . . 109
4.16 Résultats de l'évaluation de la tâche de prédiction de liens avec classement. 110
4.17 Résultats de l'étude d'ablation. 110
4.18 Exemple des triplets résultant de la requête ***<?, r_sentiment, compassion>*** **.** 111
4.19 Exemple des triplets résultant de la requête ***<?, r_lieu, armoire>*** **.** . . 112
4.20 Exemple des triplets résultant de la requête ***< tablier>vêtement, r_lieu, ?>*** **.** 113
4.21 Exemple des triplets résultant de la requête ***<?, r_syn, couvrir>*** **.** . . 113
4.22 Exemple des triplets résultant de la requête ***<?, r_associated, choc anaphylactique>*** **.** 114
4.23 Exemple des triplets résultant de la requête ***<requin nourrice, r_has_part, ?>*** **.** 115
4.24 Exemple des triplets résultant de la requête ***<?, r_lieu, liquide céphalo rachidien>.*** 115
5.1 Tableau des relations pour le domaine de la structure d'entreprise. . 119
5.2 Tableau des relations pour le domaine d'activité d'entreprise. 119
5.3 Tableau des relations et leurs mots clés. 135
5.4 Tableau des types des relations et leurs descriptions. 140
5.5 Tableau des types des relations et leurs exemples. 140
5.6 Résultats de l'évaluation. 142
5.7 Résultats de l'évaluation des variantes du modèle de réseaux prototypiques (Snell et al. 2017) 143
5.8 Résultats de l'évaluation de Jeux2BERT+OIE. 144
5.9 Analyse et types des erreurs les plus récurrentes. 146

5.10 Analyse et types des erreurs les plus récurrentes. 147
5.11 Exemple de phrases relationnelles pour lesquels le modèle *Jeux2BERT* n'arrive pas à les projeter vers leurs types de relations correspondants. 148
B.1 Définition et description des relations de JeuxDeMots. 181

Préambule

Cette thèse de doctorat a été menée dans le cadre d'une thèse CIFRE entre la société AMARIS et le laboratoire de recherche LIRMM sous la supervision de M. Mathieu LAFOURCADE, MC et rattaché à l'équipe TEXTE.

Chapitre 1

Introduction générale

1.1 Contexte RD

Le numérique a déclenché le déclin des médias conventionnels et a donné naissance à de nouvelles formes de médias grâce à l'utilisation d'Internet. Avec l'essor du *big data* et la disponibilité de large volume de données variées, l'analyse de donnée Web devient trop complexe dans un contexte de veille et par conséquent l'utilisation d'outils d'aide à la décision devient incontournable.
Afin de pouvoir analyser rapidement le contexte industriel, les analystes ont besoin de :

— Surveiller et de traiter un grand volume de données en provenance de sources multiples.
— Détecter rapidement des relations telles que des opérations d'achat, de rapprochement ou de partenariat entre entreprises.
— Trouver des informations utiles à un business telles que la montée en puissance de certaines technologies.

Dans ce contexte, l'entreprise MANTU souhaite développer de nouvelles solutions de veille technique et d'intelligence technologique afin de favoriser le développement de son activité et de sa capacité d'innovation. L'objectif de la thèse est d'extraire les connaissances, les structurer, et enfin les mettre à disposition des analystes.

1.2 Motivation

Notre travail consiste à apporter des méthodes pour chercher, représenter et comprendre les informations et connaissances cibles telles qu'elles se prêtent à un

traitement automatisé par ordinateur.

L'enjeu porte sur la mise en place d'algorithmes et d'approches capables de comprendre le contenu d'un ensemble de texte et d'en tirer les valeurs informatives les plus importantes telles que les noms d'entités, leur type (e.g., personne, organisation, etc.) et les relations sémantiques qui lient ces entités entre elles.

> *La nouvelle de l'acquisition d'Apple de l'application de reconnaissance musicale Shazam a fait des vagues non seulement dans le monde de la musique, mais aussi en Europe.*

FIGURE 1 Exemple d'une information d'acquisition.

La phrase sur-mentionnée 1 exprime une relation d'acquisition entre deux entreprises, et qui est représentée par la figure ci-dessous 2.



FIGURE 2 Représentation de la relation sémantique d'acquisition.

L'analyse sémantique du langage naturel vise à comprendre la structure et le sens le plus complet d'un énoncé, en essayant de comprendre « *Qui a fait quoi, à qui et quand ?* » . Une approche naïve pour comprendre la sémantique qui se cache derrière un énoncé consiste à convertir sa représentation textuelle simple en une représentation structurée, qui détermine explicitement la nature d'un événement ou d'un fait entre entités, et détermine également le rôle de chaque entité dans la relation comme *Agent*, *Patient* et *Thème*. C'est dans ce domaine de recherche que cette thèse s'inscrit.

1.3 Défis

Dans la tâche d'extraction de relations, l'objectif est de détecter automatiquement les relations sémantiques reliant deux ou plusieurs entités du monde, les relations cibles sont prédéfinies à l'avance et consistent en un vocabulaire de relations relativement petit. Les systèmes traditionnels d'extraction de relations s'appuient sur des techniques basées sur des approches à base de règles ou sur des approches supervisées pour extraire les relations d'intérêt, nécessitant de spécifier manuellement à l'avance les relations d'intérêt et leurs règles correspondantes ou d'avoir

un corpus annoté. Par conséquent, de telles approches ne s'adaptent pas souvent à des corpus hétérogènes et issus de domaines différents, en plus de l'effort exigé pour représenter toutes les relations et leurs règles associées.

La plupart des systèmes d'extraction de relations sont monolingues et beaucoup d'entre eux se concentrent sur la langue anglaise, d'où une pénurie de systèmes d'extraction de relations pour les systèmes multilingues. La conception de systèmes d'extraction de relations à partir de zéro nécessite la création manuelle de règles centrées sur la linguistique, ce qui rend une telle tâche très laborieuse.

1.4 Définition du problème

Pour pallier les limites de l'extraction de relations susmentionnées, Yates et al. (2007*b*) propose le paradigme d'*Open Information Extraction* (OpenIE) et le présente comme *« un nouveau paradigme d'extraction qui, à partir de textes facilite la découverte de relations d'une manière indépendante et s'adapte facilement à la diversité et à la taille du corpus Web ».*

L'extraction d'informations ouvertes (OpenIE) est indépendante du type d'événement ou de fait liant les entités, mais cherche plutôt à extraire la structure de la relation dans une représentation d'arguments et de prédicat. Le prédicat représente le déclencheur de relation et indique que deux entités sont liées, mais ne représente pas explicitement le type de la relation.

Dans cette thèse, nous choisissons de 1) nous concentrer d'abord sur l'OpenIE. De plus, 2) nous abordons le problème de l'exploitation des connaissances externes présentes dans les bases de connaissance afin de passer de l'OpenIE à la tâche traditionnelle d'extraction de relations. Pour cela, nous nous intéressons aussi à l'enrichissement des vecteurs de mots à l'aide de base de connaissances. Une fois les deux axes de recherche bien définies, 3) nous abordons le problème de comment gérer la transition d'une représentation en surface des triplets vers une représentation suivant un schéma relationnel bien défini avec un ensemble de relations cibles.

1.5 Questions de recherche

Ainsi, nous proposons dans cette thèse de répondre aux questions de recherche suivantes :

— Comment concevoir un système d'extraction de relations avec une supervision humaine aussi minimale que possible ?
— Comment concevoir un système d'extraction de relations bilingues et satisfaisant le critère d'adaptation au domaine ?
— Comment exploiter les bases de connaissances pour l'extraction de relations ?

1.6 Contributions

Dans cette thèse, nous présentons les contributions suivantes :

— **Corpus :**
La première contribution de cette thèse est la création de corpus d'apprentissage ainsi qu'un référentiel d'évaluation pour la langue française pour la tâche de l'OpenIE, en plus de la ré-annotation de corpus déjà existant en langue anglaise. Dans l'OpenIE, la plupart des corpus existants sont en langue anglaise et obtenus automatiquement, ce qui fait que ces corpus sont erronés et contiennent beaucoup de bruit. Pour annoter les corpus, nous suivons les recommandations d'annotation de Lechelle et al. (2019).

— **UD2OIE :**
Une autre contribution de cette thèse, est la conception d'un système OpenIE multilingues exploitant le formalisme *Universal Dependency*.
Les travaux récents (Cui et al. 2018*a*, Stanovsky et al. 2018*a*, Kolluru, Aggarwal, Rathore, Mausam and Chakrabarti 2020) basés sur l'apprentissage profond préconisent des systèmes neuronaux de bout en bout exploitant uniquement la sémantique lexicale (vecteur de mots) et ignorent d'autres représentations telles que les représentations syntaxiques et sémantiques en raison du fait que les erreurs de l'analyseur linguistique sont propagées au modèle et ne sont pas récupérables.
En raison de cela, ces approches ne sont pas adaptables au domaine et

violent le principe même de l'OpenIE. Un autre reproche concernant l'état de l'art actuel est l'abondance des systèmes pour la langue anglaise et la carence pour d'autres langues autres que l'anglais, tandis que les systèmes OpenIE multilingues sont quasi rare.
Ainsi, nous proposons un système basé sur l'apprentissage profond n'exploitant seulement l'analyse syntaxique et ignorant la sémantique lexicale (la représentation du sens des mots) et qui obtient les meilleurs résultats dans les référentiels d'évaluation pour la langue anglaise, française, espagnole et portugaise.

— **Jeux2BERT :**
Finalement, nous présentons une approche pour améliorer la représentation vectorielle des mots en exploitant la base de connaissances *JeuxdeMots*.
Plus précisément, nous nous concentrons sur les vecteurs de mots avec des représentations contextualisées qui ont reçu pas mal d'attention ces dernières années. Les représentations contextualisées de mots tel que ELMO (Peters et al. 2018), GPT (Radford et al. 2018) et BERT (Devlin et al. 2019) sont obtenues en pré-entraînant des modèles de langue sur de grandes quantités de texte brut à l'aide d'objectifs auto-supervisés.
Cependant, de telles approches ne permettent pas de tirer le meilleur parti des connaissances exprimées dans ces textes puisqu'elles s'expriment plus ou moins implicitement en plus de la complexité de la langue.
De plus, de tels corpus ne sont pas complets et augmenter ces approches avec de telles connaissances avec la même procédure ne peut être qu'une tâche coûteuse et fastidieuse.
Comme ces approches suivent le principe

> You shall know a world by the company it keeps, ce dernier que est la raison du succès de ces approches fait qu'elles souffrent d'un inconvénient majeur en faisant l'amalgame entre la similarité sémantique et d'autres types de relations sémantiques.
> En effet, ces approches sont toujours plus ou moins naïves en supposant que tous les mots apparaissant dans des contextes presque identiques sont sémantiquement similaires et de ce fait leur assignent des vecteurs de distribution similaires.
> Pour cette fin, nous exploitons les réseaux lexico-sémantiques pour

affiner ces représentations et démêler les propriétés relationnelles des mots à l'aide d'une approche multi-tâches.

Jeux2BERT+OIE :
Nous proposons une nouvelle approche pour l'extraction de relations sémantiques. Le système d'extraction de relations proposé combine les systèmes présentés dans cette thèse : *UD2OIE* et *Jeux2BERT* pour pallier les limites des systèmes d'extraction de relations sémantiques. La nouveauté de notre système réside en une approche en deux temps qui suit les niveaux stylistiques du langage. En effet le système *UD2OIE* se concentre sur la représentation syntaxique afin d'obtenir la représentation prédicat-arguments et *Jeux2BERT* qui dérive des vecteurs de mots modélisant les relations lexicales qui existent entre les mots et modélisant aussi quelques relations sémantiques.
La sémantique qui est associée à un niveau supérieur du langage est obtenu en combinant deux autres niveaux du langage et qui lui sont inférieur dans l'ordre des niveaux stylistiques du langage : la syntaxe et le lexique.
Ainsi, notre système d'extraction de relations repose sur les vecteurs de mots *Jeux2BERT* pour normaliser les triplets prédicat-argument extrait par *UD2OIE* vers un ensemble de relations cibles prédéfinies à l'avance.
Cet ensemble de relations sémantiques est défini en même temps que l'ensemble de règles qui permettent de passer du paradigme d'extraction d'information ouvert *OpenIE* vers le paradigme d'extraction fermé (ensemble de types de relations fini) *Extraction de relations sémantiques*.

Publications :
Nous listons les publications que nous avons rédigées au cours de cette thèse :

— *Open Information Extraction : Approche Supervisée et Syntaxique pour le Français (Supervised Syntactic Approach for French Open Information Extraction), Actes de la 28e Conférence sur le Traite-*

ment Automatique des Langues Naturelles. Volume 1 : conférence principale,
2021, Lille, France, ATALA

— *Universal Dependencies for Multilingual Open Information Extraction, 3rd Conference on Language, Data and Knowledge (LDK 2021),Open Access Series in Informatics (OASIcs),*
Schloss Dagstuhl – Leibniz-Zentrum für Informatik,Dagstuhl, Germany

Chapitre 2

État de l'art sur Extraction de Relations

La tâche d'extraction de relations consiste à capturer les relations sémantiques d'intérêt entre deux entités. Plus formellement, étant donné un ensemble de relations prédéfinies $R = \{R_1, R_2, ..., R_n\}$ et un ensemble de paires d'entités $E = \{(E_1, E_2), ..., (E_k, E_n)\}$, l'objectif est de détecter les relations existantes entre entités $R_k = \{(E_i, E_j), (E_m, E_n, E_p)\}$ exprimées dans une phrase donnée.
Dans ce chapitre nous passons en revue les principales approches pour la tâche d'extraction de relations, telles que les méthodes à base de règles, les méthodes supervisées, les méthodes semi-supervisées et les méthodes non-supervisées.

2.1 Approches à base de règles

Les systèmes à base de règles reposent sur un ensemble de règles rédigées à la main par des experts du domaine. Cet ensemble de règles est souvent basé sur des patrons lexico-syntaxiques qui exploitent différents types d'informations linguistiques tels que l'étiquetage grammatical (*POS*) et l'analyse de dépendance syntaxique, certains patrons peuvent aussi exploiter la notion de cooccurrences entre une paire de mots.
Le système à base de règles extrait les relations en faisant correspondre les patrons au texte à partir duquel les relations doivent être extraites. S'il y a plusieurs correspondances possibles de patrons, le patron ayant la plus longue phrase relationnelle est choisi. Si le motif correspond à plusieurs séquences adjacentes, le système fusionne ces motifs en une seule phrase relationnelle.
Les premiers travaux qui mettent en lumière ce type d'approches, Hearst (1992) proposent d'exploiter certains patrons lexico-syntaxiques pour détecter les relations de type *is-a*. Par exemple, des motifs tels que « NP_y tel que NP_x » ou «

NP_x et autres NP_y » indiquent souvent des relations d'hyperonymie de la forme $xis - ay$.
Compte tenu de l'ambiguïté de certains patrons qui peuvent correspondre à plusieurs relations distinctes, les patrons linguistiques sont souvent augmentés avec des contraintes sur les arguments du patron (Girju et al. 2006, Fabre and Bourigault 2006).

Cependant, les systèmes à base de règles ont souvent des performances à faible rappel à cause de la variété présente dans les langues. Aussi, la tâche de création de telles règles est souvent laborieuse et nécessite un effort humain considérable.
Certains travaux proposent d'augmenter les *Patrons de Hearst* pour augmenter les performances de rappel, Jacques and Aussenac-Gilles (2006) présentent de nouveaux variants de patrons pour chaque *patron de Hearst* en remplaçant certains mots des patrons par d'autres mots ayant une signification similaire. Par exemple, remplacer « *tels que* » dans le motif « NP tels que NP s » par « *comme* » pour obtenir un nouveau motif « NP comme NP s »

2.2 Approches semi-supervisées

Après les approches à base de règles, les approches semi-supervisées telles que le *bootsraping* (Brin 1998, Agichtein and Gravano 2000, Pantel and Pennacchiotti 2006, Etzioni et al. 2004, 2005) et l'apprentissage distant *(distant learning)* (Mintz et al. 2009, Hoffmann et al. 2011, Surdeanu et al. 2012, Yao et al. 2013) ont fait l'objet d'une attention particulière.
Les approches semi-supervisées consistent à pallier le manque de données étiquetées et d'exploiter l'abondance de données non étiquetées, en s'appuyant sur des techniques automatisées pour étiqueter les données ou en utilisant des méthodes faiblement supervisées.

2.2.1 Bootsraping

Les systèmes d'extraction de relations basés sur le bootstraping traitent efficacement de grandes quantités de données à partir du Web.
L'approche consiste à commencer avec un ensemble de règles élaborées à la main et d'en trouver automatiquement de nouvelles à partir des données de texte non étiquetées, grâce à un processus itératif (bootstrapping).

Alternativement, nous pouvons commencer avec un ensemble de départ, décrivant des entités avec une relation spécifique. De manière générique, l'idée du paradigme des approches à base de bootstraping consiste à :

— Collecter un ensemble de départ pour chaque relation (soit manuellement ou à l'aide des règles de base).
— Extraire les segments de textes qui correspondent aux instances de l'ensemble de départ.
— Exploiter les relations extraites à l'étape précédente pour peupler l'ensemble de départ.
— Inférer les patrons à partir du nouvel ensemble de départ (en exploitant la notion de co-occurrences, préfixe ou suffixe des segments de textes obtenue à l'étape précédente).
— Extraire les segments de textes qui correspondent aux patrons pour peupler l'ensemble de départ et affiner les patrons.
— Réitérer les deux étapes précédentes jusqu'à un critère d'arrêt est satisfait.

(Brin 1998) ont conçu le système *DIPRE (Dual Iterative Pattern Relation Extraction)*, un système d'extraction de relations s'appuyant sur un ensemble initial d'exemples de départ pour extraire d'autres exemples de départ, y compris les deux entités en utilisant la correspondance de patrons. Ensuite, le système exploite le préfixe, le milieu et le suffixe de la relation extraite pour dériver les patrons (règles). Sur la base de ces patrons, le système *DIPRE* extrait d'autres relations pour remplir l'ensemble d'exemples et affiner les modèles. Le processus est répété jusqu'à ce qu'un critère d'arrêt soit atteint (plus aucun patron est produit).
Le système *DIPRE* vise à trouver des modèles de citation de livres sur le Web, et arrive à extraire des relations auteur-titre dans 24 millions de pages Web à partir d'un ensemble initial de 5 livres.

Le système *Snowball* développé par (Agichtein and Gravano 2000), est similaire à *DIPRE* mais utilise un système de correspondance plus flexible en évitant les motifs longs comme le fait le système *DIPRE*. Il utilise également un score de confiance pour représenter la pertinence des patrons.
La principale amélioration du système *SnowBall* par apport au système *DIPRE*, était l'inclusion de balises d'entité nommées (*PER, ORG, LOC,* etc.) dans les

patrons. La présence d'entités nommées dans les patrons les a rendus plus significatifs et a réduit le nombre de faux positifs extraits par ceux-ci.
Dans le cas des patrons de *DIPRE*, nous nous attendons à ce que les chaînes de préfixe, de suffixe et du milieu des patrons correspondent exactement. Cette condition entrave la couverture des patrons.
Les patrons de *SnowBall* sont conçus pour surmonter ce problème afin que les variations mineures dans le texte (par exemple, les fautes d'orthographe, les articles supplémentaires) ne provoquent pas de décalage. Au lieu d'avoir des patrons de chaîne, le système *SnowBall* représente le contexte des entités (c'est-à-dire le préfixe, le suffixe et la droite) en utilisant des vecteurs de mots dans le modèle d'espace vectoriel. Plus le produit scalaire entre deux vecteurs de mots de contexte est élevé, plus la similarité des contextes est élevée.
Autre aspect important de *SnowBall* est le rejet de tous les patrons qui ne sont pas assez précis, c'est-à-dire les patrons qui sont plus susceptibles d'extraire des faux positifs. Une façon d'éliminer de tels patrons consiste à filtrer tous les patrons non pris en charge par un nombre minimum d'exemples de départ.
Le système *SnowBall* extrait la relation *Organisation-Siège* des pages Web et inclut l'utilisation de balises d'entités nommées basées sur l'annotation d'étiqueteur de catégories grammaticales à partir de l'*Alembic Workbench* (Day et al. 1997).
Les approches de bootstraping pour explorer les patrons génériques est également proposée. Le système *Espresso* (Pantel and Pennacchiotti 2006) est un système généraliste faiblement supervisé, conçu pour extraire diverses relations sémantiques des textes. Pour le couple d'entités x et y qui est un des exemples d'une relation sémantique, le système va capturer toutes les phrases mentionnant les entités x et y et les généraliser en un nouvel ensemble de phrases en remplaçant les noms propres et noms communs de la phrase avec un token spécial (*TR*). Ce n'est qu'est après cette étape que le système va extraire les sous-chaines de la phrase qui lient le couple (x, y) et inférer les patrons de la relations.

KnowItAll est un système non supervisé et indépendant du domaine qui extrait des faits du Web (Etzioni et al. 2004, 2005), qui a la particularité d'utiliser une nouvelle forme de *bootsraping* qui ne nécessite aucune phrase d'entraînement étiquetée manuellement. Ce dernier système commence par un ensemble de patrons d'extractions génériques indépendants du domaine à partir duquel il induit un ensemble d'instances de départ.

L'extraction d'informations semi-supervisée ne nécessite pas de données d'entraînement étiquetées manuellement mais juste un ensemble d'exemples ; ensuite, les systèmes d'extraction semi-supervisés peuvent découvrir de manière récursive de nouveaux attributs, instances et relations automatiquement sans intervention humaine.

La principale limite des approches à base de *bootstraping* est la propagation d'erreurs qui s'accumule graduellement à chaque itération. Par exemple, les erreurs qui sont dues à la reconnaissance d'entités nommées, où l'extraction de noms propres est incomplète, entraînant un choix d'ensemble de patron initial incorrect pour la prochaine étape de *bootstraping*.
Les patrons obtenus par des approches semi-supervisées souffrent aussi de dérive sémantique *(semantic drift)* (Curran et al. 2007, McIntosh and Curran 2009) , où les concepts obtenus dérivent des concepts initiaux voulus. Cela se produit lorsque les sens des mots ne sont pas pris en compte et, par conséquent, chaque itération entraîne un écart par rapport au sens original.
Par exemple, si l'entité *Apple* est reconnue comme *Fruit* et non *Entreprise*, le système pourrait ajouter des phrases qui mentionnent un type de fruit.

Carlson et al. (2009) ont suggéré que ce problème puisse être traité en couplant l'apprentissage semi-supervisé avec les catégories des entités et leurs relations.
Carlson et al. (2009) décrivent l'utilisation de l'algorithme CBL *(Coupled Bootstrap Learner)*, qui prend en entrée une ontologie pour appliquer des contraintes de types sur les entités de la relation et un vaste corpus de pages Web.

Motivé par le fait que ces systèmes renvoient un nombre important d'erreurs ou des instances de relation de faible pertinence, une méthode basée sur des graphes pour classer les instances de relation renvoyées par le système d'extraction à base de *bootstraping* est proposé dans (Li et al. 2011). Le classement des exemples extraits est effectué en fonction de leur pertinence par rapport à l'ensemble d'exemples initiaux et pour filtrer les instances non pertinentes.
Pour ce faire, ils construisent deux graphs complets G_c et G_e respectivement pour l'ensemble contextes et d'entités (le noeud consiste en un couple d'entités), et un graphe bipartie G_b pour les contextes et entités. Plus un couple d'entités est lié à d'autre couples d'entités et d'autres patrons avec des score de classement plus

élevé, plus le couple d'entités et son contexte aura un score de classement plus elevé.

Les graphes sont aussi utilisés pour stocker des informations sur les modèles, les mots trouvés et les liens vers les entités qu'ils ont aidé à trouver.
Ces données sont ensuite utilisées pour calculer la popularité et la productivité des mots candidats. Cette approche permet d'améliorer la précision du bootstraping et de trouver des informations de haute qualité en utilisant seulement quelques instances de l'ensemble de départ.

2.2.2 Supervision distant

Outre le *bootstraping*, une autre approche plus récente faiblement supervisée et nécessitant une base de connaissances, appelée supervision distant, a été appliquée pour l'extraction de relations.
Selon Mintz et al. (2009), si deux entités participent à une relation, toute phrase qui inclut ces entités exprime également cette relation particulière.
Mintz et al. (2009) utilisent des caractéristiques extraites de différentes phrases contenant des paires d'entités de la base de connaissances *Freebase* (Bollacker et al. 2008) pour construire un vecteur de caractéristiques riche. De telles caractéristiques sont basées sur des informations lexicales, syntaxiques et d'entités nommées (*NER*).
Dans Hoffmann et al. (2011), le système *MultiR* pour l'apprentissage multi-instances (considèrent plusieurs types de relations pour un couple d'entités) est présenté, qui utilise également une approche de supervision à distance à l'aide de *Freebase* et s'applique aux caractéristiques proposées dans (Mintz et al. 2009).

Comme les approches à base de *bootstraping*, les approches à base de supervision distante souffrent également de propagation d'erreurs (erreurs dans les exemples annotés) accumulées au fur et à mesure de chaque itération. Une autre limite est l'ambiguïté de certains exemples de relations présents dans l'ensemble de départ. Par exemple pour la relation *Lieu_de_naissance*, l'ensemble de départ $\{(\textit{Barack Obama, Hawaï})\}$ peut générer des patrons erronés à partir du segment de texte : *Le président Barack Obama est arrivé à Hawaï jeudi pour passer des vacances en famille*.

Un autre problème qui a été abordé est l'ambiguïté linguistique (Yao et al. 2011, 2012). La plupart des méthodes regroupent des patrons peu profonds ou syntaxiques de mentions de relations, mais ne considèrent qu'un seul sens possible par modèle. Cependant, cette hypothèse est souvent violée dans la réalité. Yao et al. (2011) utilisent des modèles probabilistes génératifs, où les contraintes de type d'entité au sein d'une relation et les caractéristiques sur le chemin d'arbre de dépendance entre les mentions d'entité sont exploitées.
Une autre hypothèse souvent supposée dans cette approche est qu'une paire d'entités ne peut être associée qu'une seule relation. Cependant, si nous examinons les exemples suivants – *« Elon Musk a fondé Tesla »* et *« Elon Musk est le PDG de Tesla »* – nous pouvons voir que cette hypothèse est plutôt restrictive.

Hoffmann et al. (2011) ont identifié ce problème avec les systèmes d'extraction de relations précédents et tentent de résoudre le problème en utilisant l'approche *Multi-Instance Multi-label* (MIML). Ils utilisent l'apprentissage à distance avec l'apprentissage multi-instances en considérant le chevauchement de relations (où deux mêmes instances peuvent être dans deux relations différentes).
Le système résultant, *MultiR* atteint une précision compétitive ou supérieure sur toutes les plages de rappel.

Surdeanu et al. (2012) abordent le même problème. Ils identifient deux problèmes principaux de l'apprentissage à distance :
— Certains exemples d'apprentissage obtenus grâce à cette heuristique ne sont pas valides (ils en rapportent 31 %).
— Un même couple d'entités peut avoir plusieurs relations.

Par conséquent, ils essaient d'améliorer l'apprentissage à distance en prenant en compte les paramètres multi-instances et multi-étiquettes et en utilisant un cadre bayésien (ils appellent leur système *MIML-RE*) qui peut capturer les dépendances entre les labels et apprendre en présence de labels incorrectes et incomplets.
Dans les paramètres d'origine de « supervision à distance », nous sommes limités par le schéma imposé par la base de données qui est utilisée pour la correspondance des relations vers les patrons. Yao et al. (2013) suggèrent plusieurs façons de le surmonter. Ils suggèrent d'utiliser des textes bruts en plus de la supervision à distance, ainsi les relations dans le texte et les bases de données structurées préexistantes peuvent être utilisées ensemble. Yao et al. (2013) abordent ce problème

en utilisant la factorisation matricielle et le filtrage collaboratif.

2.3 Approches supervisées

Les deux principales approches supervisées d'extraction de relations dans la littérature sont les approches à base de pipeline et approches à base de prédiction conjointe (*Joint Prediction*).
Les méthodes supervisées à base de pipeline approchent la tâche d'extraction de relations en entraînant un classificateur binaire pour déterminer s'il existe une relation spécifique entre deux entités. Ils attendent aussi en entrée le couple d'entités nommées sujet et objet (s, o) souvent obtenus à l'aide d'un modèle de reconnaissance d'entités nommé (NER) ou qui peut être renseigné manuellement.

Ces classificateurs prennent en entrée des caractéristiques du texte obtenues à l'aide d'autres modules de TALN, les caractéristiques les plus couramment utilisées par les systèmes d'extraction de relation sont énumérées ci-dessous :

— Types d'entités (Personne, Organisation, Localisation etc.).
— L'étiquetage grammatical (*POS*) de chaque mot.
— L'étiquetage grammatical (*POS*) de la séquence de mots entre les deux entités cibles.
— Le nombre de mots séparant les deux entités.
— Le mot tête de segment (la phrase relationnelle qui décrit la relation).
— Les mots du segment (la phrase relationnelle qui décrit la relation).
— Une fenêtre de K mots à gauche de l'entité tête et de leur étiquetage grammatical (*POS*).
— Une fenêtre de K mots à gauche de l'entité queue et de leur étiquetage grammaticale (*POS*).
— Informations syntaxiques (relation de dépendance grammaticale) de chaque mot.
— Informations syntaxiques (relation de dépendance grammaticale) de la séquence de mots entre les deux entités.

Les premières approches supervisées étaient basées sur l'apprentissage machine et représentaient les entrées en tant que vecteur de caractéristiques.

Ces entrées telles que les caractéristiques sémantiques et syntaxiques sont transmises au modèle de classification de relations.

Kambhatla (2004) a entraîné un classificateur d'entropie maximale avec 49 classes : deux pour chaque sous-type de relation (*ACE 2003* a 24 sous-types de relation et chaque sous-type donne lieu à 2 classes en considérant l'ordre des arguments de relation) et une classe *NONE* pour le cas où les deux mentions d'entités nommées ne sont pas liées. Kambhatla (2004) ont aussi décrit diverses caractéristiques lexicales, syntaxiques et sémantiques pour extraire les caractéristiques finales (entrée du modèle).

S'appuyant sur les travaux de Kambhatla (2004), Zhou et al. (2005) a exploré quelques caractéristiques supplémentaires (caractéristiques basées sur des ressources sémantiques et caractéristiques de segmentation de phrase) pour améliorer davantage les performances des systèmes d'extraction de relations.
Zhou et al. (2005) ont utilisé un classificateur *SVM* utilisant ces fonctionnalités et ont obtenu de meilleures performances que le système de (Kambhatla 2004). Comme le modèle *SVM* (*Support Vector Machines*) est un classificateur binaire, pour obtenir une classification multi-classe, une stratégie contre les autres est utilisée (la stratégie divise une classification multi-classes en un problème de classification binaire par classe).
Ils ont également analysé les résultats pour trouver les contributions de divers types de caractéristiques. Les caractéristiques de segmentation basées sur des phrases (*chunking*) ont été observées comme contribuant le plus à l'augmentation de la performance. Les caractéristiques syntaxiques basées sur l'arbre de dépendance et l'arbre d'analyse syntaxique en constituent n'ont que légèrement contribué. L'une des principales raisons à cela est que la plupart des relations dans le référentiel d'évaluation *ACE* (Doddington et al. 2004) sont des relations à courte distance (locale) et que des caractéristiques simples telles que des mots et des fragments de phrase (*chunking*) suffisent à identifier de telles relations.

L'un des problèmes les plus récurrents qui se posent dans les méthodes supervisées d'extraction de relations est celui du déséquilibre de classe.
Cela se produit parce que le nombre d'instances négatives (c'est-à-dire les paires d'entités n'ayant aucune relation significative) dépasse largement le nombre d'ins-

tances positives (c'est-à-dire les paires d'entités ayant l'un des types de relation prédéfinis). Ce déséquilibre de classe se traduit par une précision plus élevée et un rappel plus faible car les modèles de classifications ont tendance à surproduire la classe *NONE*. Kambhatla (2006) a présenté une nouvelle solution à ce problème basé sur le vote parmi un comité de classificateurs qui augmente considérablement le rappel dans de telles situations.
Cependant, la majeure limite de ces approches est la sélection des caractéristiques les plus pertinentes et les plus optimales. Aussi, les performances globales des méthodes basées sur les caractéristiques dépendent largement de l'efficacité et du design des caractéristiques conçues.

2.3.1 Approches à base de noyau

Pour cela, les approches à base de noyau ont été proposées pour tirer le meilleur parti d'une représentation de caractéristiques plus riche. Dans les méthodes à base de noyau, les fonctions du noyau sont conçues pour calculer les similitudes entre les représentations de deux instances de relation et le modèle SVM est utilisé pour la classification.
Les approches à base de noyau au lieu de représenter les caractéristiques par le moyen de vecteurs, exploitent au maximum les entrées en prenant en compte leur structure comme le noyau d'arbre qui prend en entrée l'arbre syntaxique ou sémantique de la phrase.
Divers systèmes d'extraction de relations basés sur le noyau proposent différentes représentations pour les instances de relations telles que les séquences, les arbres d'analyse syntaxique, etc.
La plupart des techniques mesurent la similitude entre deux représentations (par exemple les arbres) en termes de nombre de sous-représentations partagées (sous-arbres) entre elles.

Motivés par le noyau de sous-séquence de chaînes (Lodhi et al. 2002), (Bunescu and Mooney 2005) ont proposé un noyau de séquence pour la tâche d'extraction de relations. La façon la plus simple de construire une séquence pour représenter une instance de relation est de simplement considérer la séquence de mots allant de la première entité jusqu'à la seconde entité dans la phrase.
Plutôt que d'avoir chaque élément de séquence sous la forme d'un mot singleton,

les auteurs ont proposé de généraliser chaque mot à un vecteur de caractéristiques. Chaque instance de relation est alors représentée comme une séquence de vecteurs de caractéristiques, un vecteur de caractéristiques pour chaque mot.
Les caractéristiques proviennent des domaines suivants : l'ensemble des mots de la phrase, l'ensemble des catégories grammaticales, l'ensemble des catégories grammaticales génériques et l'ensemble des types des entités nommées.
Ils exploitent le modèle de classification SVM avec le noyau de relation (relation Kernel) en le définissant comme la somme de 4 sous-noyaux (chaque noyau est associé à un des domaines de caractéristiques présentées), et dont chacun un capture un type spécifique de patrons et est basé sur le noyau de sous-séquence généralisé.
L'arbre d'analyse syntaxique en constituants définit la syntaxe de la phrase en termes de constituants tels que les phrases nominales (NP), les phrases verbales (VP), les phrases prépositionnelles (PP), les balises de catégories grammaticales (NN, VB, IN, $etc.$) en tant que non-terminaux et les mots comme des feuilles. La syntaxe est généralement régie par la grammaire sans contexte (CFG).

Collins and Miller (1998), Miller et al. (2000) ont proposé des modèles d'analyse syntaxique statistique pour extraire des relations du texte où ils considéraient les arbres d'analyse augmentés d'informations sur les entités et les relations.
Collins and Duffy (2001) ont proposé *Convolution Parse Tree Kernel* (KT) pour calculer la similarité entre deux arbres syntaxiques quelconques. Il calcule le nombre de sous-arbres communs partagés par deux arbres d'analyse syntaxique.
Ici, un sous-arbre est défini comme tout sous-graphe d'un arbre qui satisfait les deux conditions :

1. Il doit inclure plus d'un nœud.
2. Des productions entières (règles de la grammaire hors contexte) doivent être incluses à chaque nœud, il doit contenir des noeuds non-terminaux (catégories grammaticales).

Le noyau est conçu de telle sorte que chaque sous-arbre possible devienne une dimension dans l'espace projeté. L'image d'un arbre syntaxique T dans l'espace transformé est $h(T) = [h_1(T), h_2(T), ..., h_n(T)]$, où $h_i(T)$ désigne le nombre d'occurrences du i^{eme} sous-arbre dans l'arbre T et n désigne le nombre de tous les sous-arbres possibles (taille du vocabulaire des sous-arbres).

Pour deux arbres $T1$, $T2$ la valeur du noyau K_T est simplement le produit scalaire de leurs images dans l'espace transformé, c'est-à-dire $K_T(T1, T2) = h(T1) \times h(T2)$.

Zhang et al. (2006) ont décrit cinq cas pour construire une représentation arborescente pour une instance de relation donnée. Ces représentations couvrent l'instance de relation constituant le couple d'entités nommées :

1. L'arbre complet minimum *(Minimum Complete Tree)* : C'est le sous-arbre complet formé par le plus petit ancêtre commun des deux entités nommées.
2. L'arbre à chemin fermé *(Path-enclosed Tree)* : c'est le plus petit sous-arbre comprenant les deux entités. Il peut également être décrit comme le sous-arbre entouré par le chemin le plus court reliant deux entités dans l'arbre d'analyse de la phrase.
3. L'arbre de chemin contextuel *(Contextual-sensitive Path Tree)* : c'est la version étendue de l'arbre à chemin fermé où un mot à gauche de la première entité et un mot à droite de la deuxième entité sont inclus dans l'arbre.
4. L'arbre du chemin clos aplati *(Flattened Path-enclosed Tree)* : il s'agit de la version modifiée de l'arbre à chemin fermé où les nœuds non terminaux et qui ne sont pas des étiquettes grammaticales qui ont un seul arc d'entrée et de sortie sont contournés.
5. L'arbre de chemin contextuel aplati *(Flattened Contextual Path Tree)* : il s'agit de la version modifiée de l'arbre de chemin contextuel où les nœuds non terminaux et qui ne sont pas des étiquettes grammaticales qui ont un seul arc d'entrée et de sortie sont contournés.

Dans leur analyse expérimentale, Zhang et al. (2006) ont trouvé que l'arbre à chemin fermé *(Path-Enclosed Tree)* est le plus performant lorsqu'il est utilisé pour le calcul du noyau K_T .
Zhou et al. (2007) ont étendu ce travail en déterminant automatiquement une étendue d'arbre dynamique sensible au contexte pour l'extraction de relations en étendant l'arbre à chemin fermé *(Path-Enclosed Tree)* pour inclure les informations de contextes nécessaires.

Une autre approche pour déterminer dynamiquement la portée de l'arbre a été proposée par Qian et al. (2007). Ils ont utilisé les informations sur les dépendances en constituants pour conserver les nœuds et leurs enfants principaux le long du chemin reliant les deux entités nommées et ont supprimé les autres informations bruitées de l'arbre d'analyse syntaxique.
Une autre extension du noyau de l'arbre syntaxique a été proposée par Qian et al. (2008) où l'arbre d'analyse est complété par des caractéristiques d'entité telles que le type d'entité, le sous-type et le niveau des entités nommées. Ils ont obtenu des performances supérieures à l'état de l'art sur l'ensemble de données de référence *ACE 2004* (le meilleur score de F-mesure pour les sept principaux types de relations).

Outre les arbres d'analyse syntaxique en constituants, les arbres d'analyse peu profond et les arbres d'analyse syntaxique de dépendance ont aussi été proposé.
Dans l'analyse syntaxique de dépendance, les mots de la phrase deviennent les nœuds de l'arbre et les relations de dépendance entre eux deviennent les arcs. Chaque mot, à l'exception de la racine de l'arbre, a exactement un parent dans l'arbre.
Zelenko et al. (2003) utilisent un arbre d'analyse peu profond (l'extraction terminologique) pour construire la représentation des caractéristiques, et génèrent des instances d'apprentissage en considérant si le plus petit sous-arbre couvre les entités associées. S'inspirant de ce dernier, Culotta and Sorensen (2004) se démarquent en utilisant un arbre d'analyse syntaxique de dépendance au lieu d'un arbre d'analyse peu profond pour augmenter les performances au détriment d'un temps d'exécution plus long.
Contrairement à leurs prédécesseurs, Bunescu and Mooney (2005) ont mis en lumière le chemin le plus court entre les entités dans l'analyse syntaxique de dépendance, considérant que le chemin le plus court capture suffisamment d'informations pour extraire les relations pertinentes.

Culotta et al. (2006) proposent l'intégration d'un apprentissage automatique supervisé qui apprend des patrons relationnels et contextuels pour l'extraction d'une relation familière à partir de textes biographiques. Pour cela, il est proposé un modèle d'extraction de relations à l'aide de CRF *(Conditional Random Fields)*, qui vérifie si les entités trouvées dans les textes biographiques sont liées au thème

de la page, à partir d'un ensemble de relations préalablement connues à l'avance. Harabagiu et al. (2005) ont proposé d'augmenter les noyaux d'arbre de dépendance syntaxique avec les informations sémantiques obtenues à partir d'analyseurs sémantiques peu profonds utilisant *Prop-Bank* (Kingsbury and Palmer 2002) et *FrameNet* (Baker et al. 1998).

2.3.2 Approches à base de réseaux de neurones

Des travaux récents sur l'extraction de relations ont montré une amélioration des performances en appliquant des réseaux de neurones.
Contrairement aux approches mentionnées précédemment, les réseaux de neurones ont l'avantage d'apprendre les représentations des caractéristiques automatiquement.

2.3.2.1 Approches à base de pipeline

L'une des premières approches exploitant les réseaux de neurones a utilisé les réseaux de neurones convolutionnels (*CNN*). Les réseaux de neurones convolutionnels sont appliqués en TALN en représentant la phrase par une matrice dont chaque ligne exprime la représentation vectorielle du mot correspondant.
Ensuite, une matrice carrée de dimension relativement petite (entre 3×3 et 5×5), appelé filtre, est multipliée avec la sous-matrice de la matrice des entrées en utilisant le principe de fenêtre glissante. Une étape additionnelle de *pooling* est appliquée sur les matrices-filtre obtenues pour compresser l'information.
Les sorties des couches de convolution sont transmises à d'autre couches de convolutions avant d'être envoyées comme entrées pour la couche de classification.
La figure 3 montre un exemple de réseau de neurones convolutionel appliqué à l'extraction de relations sémantiques.

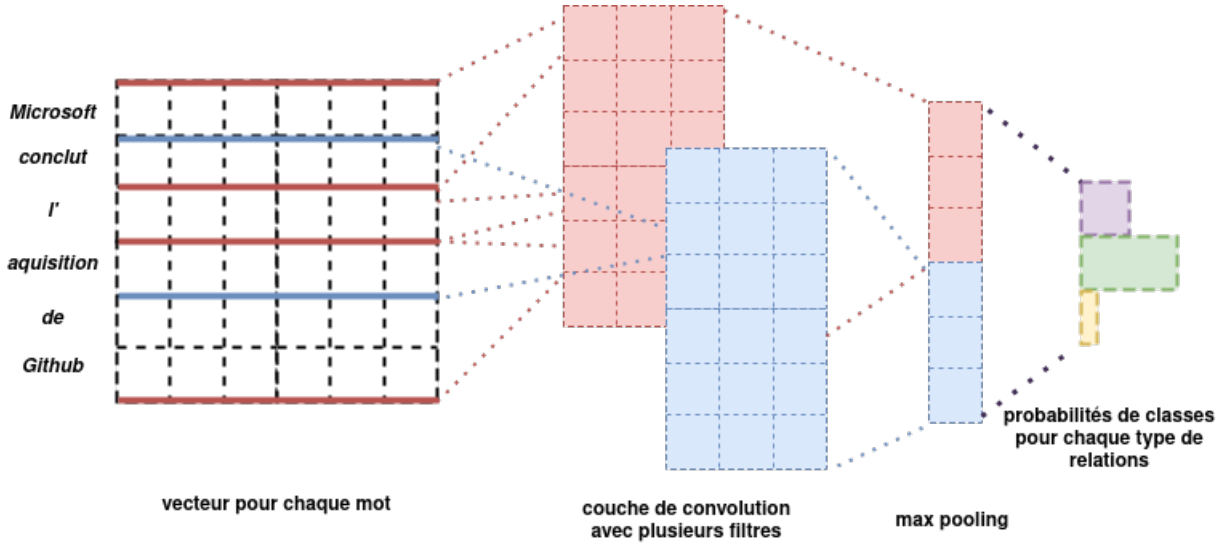


FIGURE 3 Illustration d'un modèle de réseau de neurones convolutionel.

D'un autre côté, Socher et al. (2012) ont proposé des réseaux de neurones récursifs (ne pas confondre avec les réseaux de neurones récurrents *RNN*) pour exploiter la structure arborescente de l'analyse syntaxique en constituans où chaque vecteur correspond à une représentation compositionnelle et est obtenue en fusionnant récursivement les représentations vectorielles des enfants du nœud (correspond à un mot si le nœud est une feuille, sinon correspond à un syntagme, constituant syntaxique ou sémantique de la phrase) considéré.
Les réseaux de neurones récurrents *(Recurrent Neural Network)* ont aussi été proposés avec une variante (*one to many*) spéciale à cause de la spécificité de la tâche d'extraction de relations. Contrairement aux réseaux de neurones récurrents qui calculent une représentation à chaque itération correspondant à un mot de la phrase, une seule représentation vectorielle est calculée après avoir traité tous les mots de la phrase.

Aussi, des réseaux de neurones ont été proposés pour exploiter l'arbre d'analyse syntaxique partiellement ou en totalité (Xu et al. 2016, Hong et al. 2020, Miwa and Bansal 2016). Récemment, les transformeurs (Vaswani et al. 2017) et modèles de langage préentraînés (Devlin et al. 2019) ont également été exploités pour l'extraction de relation (Du et al. 2018, Verga et al. 2018, Wang et al. 2018, Soares et al. 2019) et ont atteint un nouvel état de l'art.
Les approches supervisées sont dépendantes du domaine car elles nécessitent d'avoir l'ensemble du type de relation à l'avance et il est assez difficile d'ajouter de nouvelles relations (nécessité d'entraîner un nouveau modèle de classifica-

tion). Elles nécessitent également une grande quantité de données étiquetées pour pouvoir entraîner le classificateur de relations.

On aimerait aussi attirer l'attention sur le fait que l'extraction de relations sémantiques peut être approché d'une manière totalement différente qu'une approche de classification binaire et qui est aussi la tendance dans les récents travaux d'extraction de relations sémantiques (Zhang et al. 2017, Bekoulis et al. 2018, Sun et al. 2019, Fu et al. 2019, Wang and Lu 2020).
Toutes les techniques d'extraction de relations présentées précédemment supposent que les connaissances sur les limites et les types des entités nommées sont connues à l'avance. Si une telle connaissance n'est pas disponible, afin d'utiliser ces techniques d'extraction de relations, il faut d'abord appliquer une technique d'extraction d'entité nommée. Une fois que les mentions d'entités et leurs types d'entités sont identifiés, les techniques d'extraction de relations peuvent être appliquées. La limite la plus récurrente reprochée au modèle à base de pipeline est la propagation d'erreurs. En effet si le modèle d'entité nommée (*NER*) alimente le modèle d'extraction de relations avec un couple d'entités erronés alors cela aura un impact négatif sur l'inférence. Il est aussi à noter que ces approches ne modélisent pas l'interdépendance entre le modèle de reconnaissance d'entités nommée (*NER*) et le modèle d'extraction de relations.

Pour cela, les approches d'apprentissage à base de pipeline (Roth and Yih 2002, 2004, 2007, Kate and Mooney 2010) tentent de lier les deux modèles à l'aide de diverses dépendances, y compris à travers les contraintes résolues par le biais de représentation d'entier. Ils modélisent l'interdépendance entre le modèle de reconnaissance d'entités nommé (*NER*) et le modèle d'extraction de relations sous forme de contrainte (par exemple les couples d'entités sujet et objet pour la relation *PlaceOfBirth* doivent respectivement être de type *Personne* et de type *Lieu*).

Roth and Yih (2004) ont proposé un modèle, qui apprend d'abord des classificateurs locaux indépendants pour l'extraction d'entités et l'extraction de relations. Lors de l'inférence de la relation, étant donnée une phrase, une décision globale est produite de telle sorte que les contraintes spécifiques au domaine soient satisfaites.

Le choix d'une décision globale cohérente avec les contraintes est résolu en utilisant une approche de programmation linéaire en nombres entiers (*Integer Linear Programming*) . L'approche en programmation linéaire en nombre entier minimise la somme de la fonction de coût d'affectation (coûts liés au classificateur local d'extraction d'entités) et de la fonction de coût de contrainte. La fonction de coût d'affectation est conçue de telle manière que si la prédiction la plus probable d'un classificateur local est choisie, alors le coût le moins élevé est encouru. La fonction de coût de contrainte est conçue pour imposer un coût (pénalisation) si violation des contraintes entre les entités liées et les relations.
Roth and Yih (2007) ont détaillé encore plus l'inférence globale pour l'extraction d'entités et de relations en explorant d'autres techniques d'inférence globale que la programmation linéaire en nombres entiers comme l'algorithme de *Viterbi*.

Roth and Yih (2002) ont proposé une approche à base de modèles graphiques où les classificateurs indépendants locaux sont appris pour l'identification des entités et des relations. Les dépendances entre les entités et les relations sont codées a sein d'un réseau de croyances bayésien qui est un graphe acyclique orienté bipartite. Les entités sont représentées comme des nœuds dans une couche dans un graphe bipartite tandis que les relations sont représentées comme des nœuds dans l'autre couche. Chaque nœud d'instance de relation R_{ij} a deux arcs entrants à partir de ses nœuds d'instance d'entité argument E_i , E_j .
Étant donné le vecteur caractéristique X de la phrase, les classificateurs d'entités locales et de relations sont utilisés pour calculer respectivement $P_r(E_i|X)$ et $P_r(R_{ij}|X)$. Les contraintes sont codées par les probabilités conditionnelles $P_r(R_{ij}|E_i, E_j)$, qui peuvent être fixées manuellement ou estimées à partir des entités et relations étiquetées dans le corpus.
Une tout autre approche prônée par Bekoulis et al. (2018) est de considérer chaque paire de mot de la phrase (S, O) comme un couple candidat et ensuite d'appliquer le système d'inférence pour déterminer le type de relation existant entre le couple d'entités si une telle relation existe (ce cas géré avec une classe *NoRel*) .

2.3.2.2 Approches à base d'apprentissage conjoint

La majorité des travaux récents considèrent le problème d'extraction de relations de bout-en-bout (prédiction jointe), c'est-à-dire qu'ils extraient les entités en

même temps que les relations éventuelles qui pourraient les lier avec les entités en cours de traitement. Ainsi, les couples d'entités sujet et objet (S, O) ne sont pas fournis en entrée au modèle mais sont cherchés pendant l'extraction.
Cette procédure de prédiction commune peut aider à améliorer toutes les étapes, afin que chaque composant soit en mesure de compenser les faiblesses de l'autre et donc d'améliorer les performances des deux. Ce type d'approches est actuellement le plus performants sur les référentiels d'évaluation

Parmi les récentes approches d'extraction de relations à base d'apprentissage conjoint (prédiction jointe), deux types d'approches se distinguent : approches à base de prédiction structurée et approche à base d'apprentissage multi-tâches.
Les approches de prédiction structurée regroupent les deux tâches de reconnaissance d'entité et d'extraction de relation dans un cadre de travail unifié (framework), bien qu'il puisse être formulé de diverses manières.
Une ligne de recherche pour l'extraction conjointe d'entités et de relations a été proposée par Kate and Mooney (2010), ils ont utilisé un graphe appelé *card-pyramid*. Le graphe est appelé ainsi car il encode les dépendances mutuelles entre les entités et les relations dans une structure de graphe qui ressemble à une pyramide construite à l'aide de cartes à jouer.
Il s'agit d'un graphe arborescent qui a une racine au niveau le plus élevé, des nœuds internes aux niveaux intermédiaires et des feuilles au niveau le plus bas. Chaque entité de la phrase correspond à une feuille et s'il y a n feuilles alors le graphe possède n niveaux. Chaque niveau l contient un nœud de moins que le nombre de nœuds du niveau (l - 1). Le nœud à la position i dans le niveau l est parent des nœuds aux positions i et (i + 1) dans le niveau (l - 1).
Chaque nœud dans les couches supérieures (c'est-à-dire les couches à l'exception de la couche la plus basse) correspond à une relation possible entre les nœuds les plus à gauche et les plus à droite en dessous dans la couche la plus basse. L'objectif est d'étiqueter conjointement les nœuds dans le graphe carte-pyramide.
Étant donnée les entités dans une phrase, la grammaire de la *carte-pyramide* et les classificateurs locaux d'entités nommées et d'extraction de relations, l'algorithme d'analyse de la *carte-pyramide* essaye de trouver l'étiquetage le plus probable de tous ses nœuds qui correspond aux types d'entités et des relations.

Li and Ji (2014) proposent un système basé sur l'action qui identifie de nou-

velles entités ainsi que des liens vers des entités précédentes, tout en permettant l'extraction simultanée de mentions d'entités et de relations, qui intègre également le problème de la détection des limites pour les mentions d'entités.
Les approches précédentes modélisaient des classificateurs locaux indépendants pour identifier des entités nommées et des relations. Même si la décision globale optimale est prise ultérieurement, l'interaction entre l'extraction d'entités et les modules d'extraction de relations est interdite pendant l'apprentissage. Par conséquent, les auteurs ont proposé de reformuler ce problème comme un problème de prédiction structurée. Ils essaient de prédire la structure de sortie ($y \in Y$) pour une phrase donnée ($x \in X$), où cette structure peut être considérée comme une modélisation graphique mentionnant les entités comme des nœuds et les relations comme des arcs dirigés avec des types de relation comme étiquettes.

Une autre approche pour l'extraction conjointe d'entités et de relations a été proposée par Miwa and Sasaki (2014) qui utilise une structure de tableau. Ce tableau représente les structures d'entités et de relations dans une phrase.
Si le nombre de mots dans une phrase est n, alors le tableau est une matrice $n \times n$ triangulaire inférieure où la $i^{ème}$ cellule diagonale représente le type d'entité du $i^{ème}$ mot. Toute cellule (i,j) représente le type de relation (le cas échéant) entre la mention d'entité tête au $i^{ème}$ mot et une autre mention d'entité en-tête au $ji^{ème}$ mot. Avec cette représentation de tableau, le problème d'extraction conjointe est réduit à un problème de remplissage de tableau où des étiquettes sont attribuées aux cellules de la table.

Zhang et al. (2017) et Wang and Lu (2020) adoptent une approche de remplissage de table proposée dans (Miwa and Sasaki 2014). Katiyar and Cardie (2016) et Zheng et al. (2017) emploient une approche basée sur l'étiquetage de séquence. Sun et al. (2019) et Fu et al. (2019) proposent des approches basées sur les graphes pour prédire conjointement les types d'entités et de relations.
Takanobu et al. (2019) ont utilisé l'apprentissage par renforcement hiérarchique pour extraire des entités et des relations d'une manière hiérarchique.
Li et al. (2019) réduisent la tâche d'extraction de relation en un problème de système de question-réponse avec plusieurs-tours.
Toutes ces approches ont besoin de s'attaquer à un problème d'optimisation globale et d'effectuer un décodage conjoint au moment de l'inférence, en utilisant

une recherche de faisceau ou un apprentissage par renforcement.

2.3.2.3 Approches à base d'apprentissage conjoint avec le paradigme multi-tâches

Les modèles des approches multi-tâches construisent essentiellement deux modèles distincts pour la reconnaissance d'entité et l'extraction de relations et optimisent conjointement les deux modèles grâce au partage de paramètres.
Miwa and Bansal (2016) proposent d'utiliser un modèle basé sur l'étiquetage de séquence pour la prédiction d'entité et un autre modèle *LSTM* exploitant l'arbre d'analyse syntaxique (*Tree-LSTM*) pour l'extraction de relations. Les deux modèles partagent une couche *LSTM* pour la représentation contextualisée des vecteurs de mots et trouvent que le partage des paramètres améliore (légèrement) les performances des deux modèles.
L'approche de Bekoulis et al. (2018) est similaire mais ils modélisent la classification des relations comme un problème de sélection de têtes multi-étiquettes.
Giorgi et al. (2019) adoptent une architecture similaire mais en l'étendant avec le modèle de langage *BERT* (Devlin et al. 2019).
Enfin, Sanh et al. (2019) s'appuient sur Bekoulis et al. (2018) pour explorer un cadre multi-tâche plus large en intégrant la tâche de résolution de coréférence (*CR*) et un autre corpus pour la tâche de reconnaissance d'entité (*NER*). Ils utilisent le modèle de vecteur *ELMo* (Peters et al. 2018) en tant que représentation contextualisée des vecteurs de mots.

Notez que les approches multi-tâches effectuent toujours un décodage en pipeline : les entités sont d'abord extraites et le modèle d'extraction de relations est appliqué sur les entités prédites. La majorité des approches qui extraient conjointement des entités et des relations font état d'une amélioration significative par rapport à l'approche de base du pipeline.
L'extraction conjointe améliore non seulement les performances de l'extraction de relations, mais s'avère également efficace pour l'extraction d'entités. Parce que, contrairement aux méthodes de pipeline, le modèle conjoint facilite l'utilisation des informations sur les relations pour l'extraction des entités.

2.4 Approches Few-Shot

Ces dernières années, des modèles suivant l'approche *few-shot* ont été appliqués avec succès à une variété de tâches de TALN.
Han et al. (2018) ont introduit un cadre d'apprentissage à la *few-shot* pour la classification des relations, et depuis lors, plusieurs modèles ont dépassé les performances humaines dans cette tâche, donnant l'impression que la classification des relations à la *few-shot* est résolue.
L'apprentissage suivant le paradigme *few-shot* prend en compte la non-abondance de données d'apprentissage étiquetées et se base sur un nombre limité d'échantillons d'apprentissage pour construire le modèle d'apprentissage.
L'apprentissage par *few-shot* est différent de l'apprentissage supervisé standard. L'objectif de l'apprentissage par *few-shot* n'est pas de laisser le modèle reconnaître les classes (images ou relations) dans l'ensemble d'apprentissage, puis de les généraliser à l'ensemble de test. Au lieu de cela, le but est d'apprendre les différences existantes entre les classes et qui peuvent faciliter ultérieurement l'apprentissage de la tâche en question.
Ainsi, dans le jargon de *few-shot*,un ensemble support joue un rôle similaire à l'ensemble d'apprentissage mais l'ensemble support est relativement petit pour respecter la contrainte de limite de données d'apprentissage. La configuration du paradigme *few-shot* peut être réduite à la configuration K-way N-shot, où K-way signifie que l'ensemble support contient K classes et N-shot signifie que chaque classe contient N exemples dans l'ensemble support.

Les auteurs (Han et al. 2018) ont fourni un ensemble de données et un cadre d'évaluation pour mesurer la performance sur l'apprentissage *few-shot* à la classification de relations. Ils ont également adapté plusieurs systèmes de classification *few-shot* à la classification des relations. L'ensemble de données était composé de 700 exemples d'instances de Wikipédia pour chacune des 100 types de relations. Parmi les 100 types de relations, 64 ont été désignés pour l'apprentissage, 16 pour la validation, et 20 ont été retenus pour les tests.
L'évaluation était structurée comme suit : pour chaque instance, N relations ont été choisies à partir de l'ensemble englobant toutes les relations et, pour chaque relation, K exemples support ont été échantillonnés. Ces exemples étaient les seules informations fournies au modèle concernant ces relations. De plus, Q exemples

de requêtes (exemple d'évaluation) ont été sélectionnés pour chacune des N relations, et la tâche du modèle était de décider le bon libellé pour chaque requête parmi N réponses possibles (classification à N classes).

En 2018, Sung et al. (2018) ont proposé un réseau de relations pour un apprentissage *few-shot*. Le réseau de relations apprend une représentation vectorielle et une métrique de distance non linéaire profonde pour comparer la requête et les éléments support.
Des travaux antérieurs ont présenté un modèle *few-shot* basé sur les réseaux *CNN* et ont utilisé une approche de prototype simple. Dans cette approche, chaque exemple a été codé en tant que vecteur, et la représentation du prototype pour chaque relation a été dérivée en tant que moyenne sur tous les vecteurs des exemples existants. La représentation d'un exemple de requête a ensuite été comparée aux représentations prototypiques des relations candidates, et la plus similaire (par produit interne) a été choisie comme étiquette (label).
Aussi, un modèle de réseau prototypique plus simple et plus efficace a été proposé par Snell et al. (2017). Le réseau prototypique (Snell et al. 2017) est un modèle de classification *few-shot* basé sur l'hypothèse que pour chaque classe il existe un prototype. Le modèle essaie de trouver les prototypes des classes à partir des instances support, et compare la distance entre l'instance de requête et chaque prototype sous une certaine métrique de distance. Le réseau prototype apprend une fonction u pour intégrer les instances de chaque classe, et calcule chaque prototype en faisant la moyenne sur tous les vecteurs de sortie d'instances dans l'ensemble support S qui sont étiquetés avec la classe correspondante.

Améliorant le réseau de correspondance (*matching*), Ye and Ling (2019) proposent un réseau d'appariement et d'agrégation à plusieurs niveaux *(Multi-Level Matching and Aggregation Network)* pour la classification des relations à plusieurs plans. Différent des réseaux prototypiques, qui représentent des ensembles support sans dépendance aux instances de requête, leur système *(MLMAN)* encode chaque instance de requête et chaque ensemble support de manière interactive en considérant leurs informations de correspondance au niveau local et au niveau de l'instance. Au niveau local, les représentations de contexte local d'une instance de requête et d'un ensemble support sont légèrement mises en correspondance les unes avec les autres en suivant le cadre de correspondance de phrases (Chen et al.

2016).

Baldini Soares et al. (2019) abordent aussi le problème de préparer un modèle d'IA pour extraire des relations entre entités textuelles, sans lui donner d'étiquettes spécifiques (non supervisées) en définissant la tâche/ le problème pour que le modèle comprenne la problématique adressée.
Considérez les deux phrases (exemple extrait de (Baldini Soares et al. 2019)) de relation ci-dessous :

— **R1** : *[BLANK], inspired by Cale's earlier cover, recorded one of the most acclaimed versions of "[BLANK]".*
— **R2** :*[BLANK]'s rendition of "[BLANK]" has been called "one of the great songs" by Time, and is included on Rolling Stone's list of "The 500 Greatest Songs of All Time".*

Ici, une déclaration de relation fait référence à une phrase dans laquelle deux entités ont été identifiées pour l'extraction/classification de la relation.
Mathématiquement, nous pouvons représenter un énoncé de relation comme suit :

$$r^i = (x^i, s_1^i, s_2^i) \tag{2.1}$$

Ici, x est la phrase séparée en mots (tokenisée), $s1$ et $s2$ étant les étendues des deux entités au sein de cette phrase. Alors que les deux déclarations de relation $r1$ et $r2$ ci-dessus consistent en deux phrases différentes, elles contiennent toutes deux la même paire d'entités, qui a été remplacée par le symbole *« [BLANK] »*.
Maintenant, l'intuition est que si $r1$ et $r2$ contiennent la même paire d'entités ($s1$ et $s2$), elles devraient contenir la même relation $s1 - s2$.
Par conséquent, la tâche de pré-apprentissage pour le modèle d'IA est celle, étant donné tout couple ($r1$, $r2$), de les intégrer de telle sorte que leur produit interne soit élevé lorsque $r1$ et $r2$ contiennent tous deux la même paire d'entités ($s1$ et $s2$), et faible lorsque leur entité les paires sont différentes.
L'estimation par contraste de bruit est implémentée ici pour ce processus d'apprentissage, car il n'est pas possible de comparer explicitement chaque paire $r1$ et $r2$ pendant l'apprentissage. Le choix du symbole *« [BLANK] »* réside dans le fait que les entités dans la déclaration de relation sont intentionnellement masquées avec le symbole *« [BLANK] »* avec une certaine probabilité, de sorte que

pendant le pré-apprentissage, le modèle ne peut pas simplement compter sur les noms d'entités eux-mêmes pour apprendre les relations (si c'est le cas, le modèle mémorisera simplement, n'apprendra rien d'utile), mais devra également prendre en compte leur contexte (les mots environnants).
Cette méthode d'apprentissage basée sur la représentation de relation sans aucune supervision (*matching the blanks*) exploite des données obtenues à partir d'un graphe de connaissances ou d'annotateurs humains.
Une fois que le modèle a été pré-entraîné de cette manière, la représentation de sortie de n'importe quel x peut ensuite être utilisée pour n'importe quelle tâche en aval. Dans le cas de classification des relations, c'est-à-dire étant donnée deux entités quelconques dans une phrase, pour classer la relation entre elles (par exemple, cause-effet, entité-emplacement, etc.), il suffit d'empiler simplement un classificateur linéaire par-dessus (la représentation des états cachés en sortie) et d'entraîner ce classificateur sur des déclarations de relation étiquetées.

2.5 Approches non-supervisées

Les systèmes non supervisées d'extraction de relations capturent automatiquement les relations inconnues a priori sans aucune forme de supervision (ensemble de règles ou ensemble de données), et sont donc sans domaine et évolutifs pour de grands corpus. À cette fin, l'extraction de relation non supervisée s'appuie sur l'hypothèse de la relation latente : la co-occurrence fréquente de paires d'entités dans des motifs similaires est exploitée pour extraire les relations entre les paires d'entités (Turney 2008, Rosenfeld and Feldman 2007, Hasegawa et al. 2004, He et al. 2006).

Les méthodes élémentaires calculent une matrice entre les patrons et les paires d'entité, où les lignes représentent les paires d'entités et les colonnes représentent les patrons et les valeurs des cellules expriment la fréquence des paires d'entités dans ces patrons. La similitude de deux paires d'entités est dérivée à l'aide de la distribution de paires d'entités à travers les patrons observés, et permet de regrouper des paires d'entités ayant des patrons similaires, et éventuellement de représenter une relation.
La technique de clustering permet de représenter chaque relation avec un cluster distinct qui regroupe les paires d'entités liées par cette relation, et fournit un en-

semble de patrons discriminants pour la relation considérée, qui peuvent en outre être exploités par les extracteurs de relations à base de règles.
Une approche alternative au clustering est de mesurer la similarité lexicale entre les phrases relationnelles des patrons. Les performances des systèmes d'extraction de relations non supervisés dépendent beaucoup de la qualité du clustering.

Le système FASTUS *(Finite State Automaton Text Understanding)* (Hasegawa et al. 2004) fonctionne comme un automate à états finis non déterministe en cascade : il sépare le traitement en étapes, et chaque niveau correspond à un genre de traitement linguistique. Les auteurs pensent qu'une décomposition de l'extraction de relations selon les niveaux du langage naturel est essentielle à l'approche.
L'approche d'extraction de relation présentée dans (Hasegawa et al. 2004) est basée sur un corpus défini en regroupant des paires d'entité nommée (*NE*) selon sa similarité de contexte (similarité de cosinus) et l'utilisation du système de reconnaissance d'entité nommée (*NER*) pour l'identification des entités nommées.
Les auteurs ont supposé que les paires d'entités nommées ((*NE*)) qui apparaissent dans un contexte similaire peuvent être regroupées dans un même cluster et que chacune de ces paires est une instance de la même relation.
L'approche peut être décrite de la manière suivante :

1. Les entités nommées dans les corpus de texte sont étiquetées.
2. Les paires d'entités nommées co-occurrentes sont formées et leurs contextes sont enregistrés.
3. Les similitudes de contexte entre les paires identifiées à l'étape 2 sont calculées.
4. En utilisant les valeurs de similarité calculées à l'étape précédente, les paires d'entités nommées sont regroupées.
5. Comme chacun de ces clusters représente une relation, une étiquette est automatiquement attribuée (en utilisant les mots les plus fréquent dans les contextes de toutes les paires d'entités nommées du cluster.) à chaque cluster décrivant le type de relation qu'il représente.

Chen et al. (2005) suivent la même approche que Hasegawa et al. (2004) mais en proposant une méthode non supervisée de sélection de caractéristiques pour

supprimer les mots non informatifs et qui contiennent du bruit exploité lors du calcul de similarité.
Yan et al. (2009) considèrent des paires de Concepts au lieu de paires d'entités nommées en utilisant la structure Wikipedia.
Pour un article Wikipédia, son titre devient le concept principal et il est associé à d'autres concepts secondaires liant l'article actuel à d'autres articles Wikipédia. Ils ont proposé une approche de clustering en deux étapes consistant à regrouper les paires de concepts avec le même type de relation.
Dans la première étape, les paires de concepts sont regroupées en utilisant la similitude des patrons linguistiques profonds reliant les deux concepts.
Ces patrons linguistiques sont dérivés des arbres syntaxiques de dépendance des phrases contenants les phrases cibles.
Une fois que ces clusters sont dérivés dans la première étape, afin d'améliorer la couverture, la seconde étape de clustering est effectuée sur les paires de concepts non clustérisés restantes. Cette étape de clustering utilise les centroïdes de clusters créés lors de la première étape et est basée sur des patrons de surface simples appris à partir d'un corpus Web plus important.

Lin and Pantel (2001) ont proposé un algorithme DIRT *(Discovery of Inference Rules)* basé sur l'induction de types de relations en généralisant les chemins de dépendance, et en conjecturant une hypothèse de similarité distributionnelle.
Plutôt que d'appliquer cette hypothèse pour découvrir des mots similaires, ils l'exploitent afin de découvrir des chemins de dépendance similaires qui tendent à relier le même ensemble de mots.

Poon and Domingos (2009) ont proposé une approche similaire pour USP *(Unsupervised Semantic Parsing)*, qui regroupe de manière récursive des fragments d'arbres de dépendances syntaxiques de telle sorte que diverses variations syntaxiques véhiculant la même signification soient regroupées.
Suivant la même ligne de recherche, Lewis and Steedman (2013) regroupent des expressions véhiculant le même sens et l'ont étendue au cas bilingue pour l'anglais et le français. Ils ont appris les groupes d'expressions sémantiquement similaires en anglais et en français représentant certaines relations. Plutôt que d'utiliser des corpus parallèles bilingues, les auteurs ont exploité les alignements entre entités nommées dans deux langues.

La tâche d'extraction de relation est généralement une tâche fastidieuse car elle implique largement l'humain dans la boucle, de la conception des règles d'extraction à l'étiquetage des données d'entraînement. Pour cela, l'extraction d'information ouverte a été proposée et entre également dans la catégorie de l'extraction de relations non supervisée.
Plusieurs travaux ont essayé d'exploiter l'extraction d'information ouvert en tant que tâche intermédiaire et de l'adapter à une tâche bien spécifique.
Ce genre d'approches soulève cependant un défi majeur : comment déduire le type de la relation à partir du prédicat (phrase relationnelle) du triplet ?
Aussi, la correspondance entre le type de la relation et le prédicat n'est pas évidente, dans la mesure ou une relation donnée peut être exprimée par plusieurs prédicat et qu'un même prédicat peut correspondre à plusieurs types de relations.
Finalement, les prédicats trop spécifiques ou implicites ne fournissent pas un signal assez fort et les informations adéquates pour les applications en aval.

Soderland et al. (2010) ont proposé un système de correspondance entre les extractions d'OpenIE et un ensemble de relations cibles du domaine du football américain (*NFL*). Pour commencer, ils conçoivent un système de reconnaissance de classe à l'aide de dictionnaires ou d'un système de reconnaissance d'entités. La conception d'un tel système permet de vérifier la satisfaction des contraintes de type posées sur les arguments d'une relation donnée.
Une fois les tuples d'OpenIE augmentés par les classes du domaine, un algorithme de couverture par ensemble basé sur l'apprentissage machine est exploité pour faire correspondre ces entités aux relations prédéfinies.
Pour cela, à chaque instance du corpus d'entraînement est associée une règle aussi spécifique que possible (pour permettre une généralisation des règles avec seulement quelques exemples). Ces règles sont exprimées en posant des contraintes de types sur les arguments et prédicats du tuple d'OpenIE.
L'algorithme de couverture par ensemble considère les contraintes de types comme les éléments de l'ensemble. Chaque règle et sa représentation générique (qui est aussi une règle) est appliquée aux instances du corpus d'entraînement, les relations extraites à l'aide de ces règles sont considérées correctes si elles correspondent à une annotation positive, sinon elles sont considérées erronées.
Ce qui permet d'associer un score de confiance pour chaque règle, qui permet à

son tour à l'algorithme de garder les k-règles les plus pertinentes.
A chaque itération et pour chaque règle, l'algorithme infère une règle plus générique en supprimant une ou plusieurs contraintes de la règle.
Les règles inférées sont ensuite ajoutées à l'ensemble de règles afin d'amorcer une nouvelle itération de l'algorithme jusqu' à ce qu'il n'a aucune amélioration de l'ensemble de règles.

En se rendant compte dans leurs précédents travaux (Soderland et al. 2010) qu'une approche exploitant l'apprentissage machine ne permet d'avoir un système d'extraction que lorsqu'il y a une abondance de données d'entraînement même lorsque elles sont augmentées avec l'apprentissage actif, Soderland et al. (2013) choisissent de créer manuellement les règles de correspondance de la forme de surface des textes vers les relations cibles . S'appuyant sur le fait que la forme de surface des textes correspondant à la relation cible suit une distribution de *Zipf* (il y a peu de formes de surface à haute fréquence utilisées pour exprimer une relation telle que "achat de", et une longue queue d'autres formes de surface avec une fréquence décroissante). Soderland et al. (2013) ont proposé une approche à base de règles avec une couverture des formes de surface des textes permettant d'avoir un système d'extraction avec une très grande précision.
Soderland et al. (2013) proposent un système d'extraction de relations adapté aux spécifications de la compétition *TAC-KBP* (*TAC Knowledge Base Population*).
Comme (Soderland et al. 2010), ils utilisent un système d'identification d'entité nommée et un marqueur de mots-clés pour identifier les entités de tuple et leur type, sur lesquels seront définies les contraintes sémantiques ou lexicales. Après avoir défini les règles pour chaque type de relation, l'extraction est déclenchée si un tuple respecte toutes les contraintes de la règle.
Le tableau 2.1 ci-dessous liste les principales approches et systèmes d'extraction de relations sémantiques.

<table>
<tr><td rowspan="2">Approches à base de règles</td><td colspan="2">*Part-Whole Patterns* Girju et al. 2006</td></tr>
<tr><td colspan="2">*Proximité Nom/Verbe* (Fabre and Bourigault 2006)</td></tr>
<tr><td rowspan="5">Approches semi-supervisées</td><td rowspan="5">Bootsraping</td><td>*DIPRE* (Brin 1998)</td></tr>
<tr><td>*Snowball* (Agichtein and Gravano 2000)</td></tr>
<tr><td>*Espresso* (Pantel and Pennacchiotti 2006)</td></tr>
<tr><td>*KnowItAll* (Etzioni et al. 2004)</td></tr>
<tr><td>*Coupled Bootstraped Learner* (Carlson et al. 2009)</td></tr>
</table>

	Apprentissage distant	*MultiR* (Hoffmann et al. 2011)
		Multi-Instance Multi-Label (Hoffmann et al. 2011)
		MIML-RE (Surdeanu et al. 2012)
Approches supervisées	Apprentissage automatique	Maximum Entropy (Kambhatla 2004)
		SVM (Zhou et al. 2005)
		Voting Entropy (Kambhatla 2006)
	Apprentissage à base de noyaux	*String Kernel* (Lodhi et al. 2002)
		String Kernel (Bunescu and Mooney 2005)
		Syntax-based Kernel (Collins and Miller 1998)
		Syntax-based Kernel (Miller et al. 2000)
		Convolution Parse Tree Kernel (Collins and Duffy 2001)
		Path-Enclosed Tree (Zhang et al. 2006)
		Context Sensitive Path-Enclosed Tree (Zhou et al. 2007)
		Convolution Tree Kernel Expanded with Entity Features (Qian et al. 2007)
		constituent dependencies for Tree Kernel (Qian et al. 2008)
		Shallow Kernel-based Tree (Zelenko et al. 2003)
		CRF (Culotta et al. 2006)
		Shallow semantics for Kernel Tree (Harabagiu et al. 2005)
	Apprentissage à base de pipeline	*Reccursive Neural Networks* (Socher et al. 2012)
		Graphicals Networks (Roth and Yih 2002)
		Integer Linear Programming (Roth and Yih 2004)
		Integer Linear Programming (Roth and Yih 2007)
		Self Attention (Du et al. 2018)
		Transformers (Verga et al. 2018)
		Adversarial Learning (Wang et al. 2018)
		Matching the Blanks (Soares et al. 2019)
	Apprentissage conjoint	*Card Pyramid* (Kate and Mooney 2010)
		Joint Entity and Relation Modeling with Table Representation (Miwa and Sasaki 2014)
		Joint Entity and Relation Modeling with Table Representation (Zhang et al. 2017)
		Joint Entity and Relation Modeling with Table Representation (Wang and Lu 2020)
		GraphRel (Fu et al. 2019)
		Graph Convolution Networks (Sun et al. 2019)
		Reinforcement Learning (Takanobu et al. 2019)
		Multi-Turn Question Answering (Li et al. 2019)

	Apprentissage conjoint multi tâches	*Tree-LSTM* (Miwa and Bansal 2016)
		Hierarchical Decomposition (Bekoulis et al. 2018)
		Hierarchical Decomposition with BERT (Giorgi et al. 2019)
		Hierarchical Multi-task (Sanh et al. 2019)
Approches Few-Shot	*Relation Classification* Han et al. 2018	
	Relation Network (Sung et al. 2018)	
	Prototypical Network (Snell et al. 2017)	
	Multi-Level Matching and Aggregation Network (Ye and Ling 2019)	
	Matching the Blanks (Baldini Soares et al. 2019)	
Approches non-supervisées	Clustering	*FASTUS* (Hasegawa et al. 2004)
		supervised feature selection (Chen et al. 2005)
		Wikipedia's Concepts Wise Extraction (Yan et al. 2009)
		Discovery of Inference Rules (Lin and Pantel 2001)
		Unsupervised Semantic Parsing (Poon and Domingos 2009)
		Unsupervised cross-lingual Semantic Parsing (Lewis and Steedman 2013)
	OpenIE	*Covering Algorithm for Relation Mapping* (Soderland et al. 2010)
		Rules-based Relation Mapping (Soderland et al. 2013)

TABLE 2.1 Récapitulatif des approches de l'état de l'art pour l'extraction de relations

2.6 Résumé

Dans ce chapitre, nous avons présenté les principales approches pour l'extraction de relations sémantiques. Comme nous l'avons mentionnée, les approches supervisées nécessitent de vastes corpus annoté et les approches à base de règles ont souvent des performances limités par rapport aux autres approches. Pour cela, nous choisissons de nous concentrer sur l'approche non-supervisé via le paradigme *OpenIE*, nous incluons aussi les approches *Few Shot* dans notre évaluation pour la tâchés d'extraction de relations sémantiques.

Chapitre 3

Open Information Extraction

Dans ce chapitre, nous commençons par définir le paradigme d'*Open Information Extraction*, puis nous présentons notre méthode proposée pour concevoir un système d'extraction n-aire, multilingues et adaptable au domaine.
La section 3.1 présente la problématique que nous traitons dans ce chapitre, nous la section 3.2 présente ensuite l'état de l'art pour l'*Open Information Extraction*. Ensuite l'hypothèse sur lequel se base notre approche et notre méthode dans la section 3.3n et la section 3.6. Après, la section 3.7 présente notre modèle et son architecture, et les sections 3.8 et 3.9 illustrent les expérimentations et les résultats de l'évaluation. La section 3.10 résume la conclusion de ce chapitre.

3.1 Introduction

Dans la tâche d'extraction de relations, le but est de détecter automatiquement les relations sémantiques liant deux ou plusieurs mentions linguistiques, les relations cibles sont prédéfinies à l'avance et sont constituées d'un ensemble relativement petit et spécifique à un domaine bien précis.
Les systèmes traditionnels d'extraction de relations reposent sur des techniques à base de règles (Lafourcade and Ramadier 2016) ou des techniques supervisées pour extraire les relations d'intérêt, nécessitant de spécifier manuellement à l'avance les relations d'intérêt et leurs règles correspondantes ou ayant un corpus d'apprentissage correspondant à ces relations.
Par conséquent, de telles approches ne s'adaptent pas bien à de grands corpus hétérogènes et inter-domaines, en raison de l'effort exigeant pour capturer toutes les relations et leurs règles.

L'*Open Information Extraction* (OpenIE) cherche à extraire des faits et des

événements affirmés par une phrase, à travers une représentation prédicat-argument. Yates et al. (2007*b*) le présente comme *« un nouveau paradigme d'extraction qui facilite la découverte indépendante du domaine des relations extraites du texte et s'adapte facilement à la diversité et à la taille du corpus Web »*.

Phrase	Bennett a confirmé lorsqu'il s'est adressé au conseil municipal ce soir que le bureau du procureur des États-Unis avait demandé des informations au canton.
Extractions	(**A0 :**Bennett **; P :**a confirmé **; A1 :**le bureau du procureur des États-Unis avait demandé des informations au canton.) (**A0 :**le bureau du procureur des États-Unis **; P :**avait demandé des informations **; A1 :**au canton) (**A0 :**il **; P :**s'est adressé **; A1 :**au conseil municipal **; A2 :**ce soir) (**A0 :**Bennett **; P :**s'est adressé **; A1 :**au conseil municipal **; A2 :**ce soir)

TABLE 3.1 Exemples d'extraction d'information ouvert(OpenIE).

3.2 État de L'art

Mesquita et al. (2013) a classé l'OpenIE en trois approches majeures, selon le type de caractéristiques qu'ils exploitent : exploitation superficielle de l'OpenIE, modèles basés sur les parties du discours, OpenIE via l'analyse des dépendances, exploitant tout ou partie du graphe de dépendance, et l'OpenIE via l'analyse sémantique, exploitant l'étiquetage de rôle sémantique (*Semantic Role Labelling* SRL) ou la représentation de signification abstraite (*Abstract Meaning Representation* AMR).
Niklaus et al. (2018) ont classé dans leur travaux les systèmes de l'OpenIE en quatre catégories de systèmes et d'approches basés sur : l'apprentissage, les règles, les clauses et les approches capturant les relations entre les propositions.
Les systèmes basés sur des clauses reposent sur la paraphrase et la connaissance de la grammaire pour transformer des phrases plus complexes en un ensemble de clauses indépendantes.
Les systèmes capturant les relations d'inter-proposition prennent en compte le contexte des relations extraites et font la distinction entre les faits affirmés et les faits hypothétiques.

3.2.1 Première génération

Les premiers systèmes OpenIE n'exploitaient qu'une analyse syntaxique basique telle que l'étiquetage grammatical et l'extraction terminologique (*chunking*) (Yates et al. 2007*a*,*b*, Banko and Etzioni 2008, Fader et al. 2011*a*).
Yates et al. (2007*b*) a conçu un extracteur auto-supervisé *(TextRunner)* composé d'un pipeline à trois étapes. Sur la base d'heuristiques prédéfinies, de parties de discours (catégories grammaticales) et d'extractions terminologique, ils dérivent un ensemble d'apprentissage avec des exemples positifs et négatifs. Ensuite, ils induisent un classificateur de Bayes naïf à partir de l'ensemble d'apprentissage. Enfin, les extractions sont filtrées selon des contraintes basées sur la redondance.
Banko and Etzioni (2008) présentent le système *O-CRF* basé sur un modèle de champ aléatoire conditionnel (CRF). Les auteurs montrent que de nombreuses relations peuvent être catégorisées à l'aide d'un ensemble compact de patrons lexicaux-syntaxiques.

Wu and Weld (2010) a suivi une approche distante ou faiblement supervisée pour développer le système *WOE* (*Wikipedia-based Open Extractor*), en dérivant un ensemble d'apprentissage à partir de Wikipédia en faisant correspondre le contenue de l'Info-box de Wikipédia avec les patrons adéquats.
Contrairement au système *TextRunner*, qui n'utilise que des fonctionnalités superficielles dans le processus d'extraction, *WOE* peut apprendre deux types d'extraction. WOE_{parse} dérivé à partir de patrons exploitant l'analyse syntaxique de dépendance ; et WOE_{pos} dérivé à partir de caractéristiques peu profondes comme les balises d'étiquettes grammaticales.
Un extracteur basé sur un champ aléatoire conditionnel (*Conditional Random Field*) (Lafferty et al. 2001) est appris pour prédire une relation pour chaque paire candidate d'arguments (Noms Propre).

Fader et al. (2011*b*) ont mis en lumière les régularités des phrases relationnelles, ils ont constaté qu'un petit ensemble de phrases relationnelles augmentées de contraintes représentent 85% des relations binaires extraites.
Fader et al. (2011*b*) montrent aussi que les sorties des systèmes OpenIE (tels que TextRunner et WOE) sont limitées et proposent trois axes d'amélioration :

1. **Extractions incohérentes** : aucune interprétation significative des phrases de

relations extraites ne peut être faite. De telles extractions sont le résultat d'une prise de décision mot par mot concernant l'inclusion ou non d'un mot dans une phrase de relation.

2. **Extractions non informatives** : ces extractions omettent des informations critiques, et sont généralement causées par une mauvaise gestion des phrases de relation qui sont exprimées par les constructions de verbes légers (LVC). Les constructions de verbes légers sont des expressions à plusieurs mots composées d'un verbe et d'un nom ; le nom apportant un contenu sémantique,par exemple (*est l'auteur de*).
   À partir de la phrase *"John a fait une promesse à Alice"*, le système *TextRunner* fait une extraction non informative *(John ; a fait ; une promesse)* alors que l'extraction correcte est *(John ; a fait une promesse à ; Alice)*.
3. **Extractions trop spécifiques** : *TextRunner* peut extraire des phrases de relation très spécifiques qui ne sont pas utiles. Par exemple. *(L'administration Obama ; n'offre que des objectifs modestes de réduction des gaz à effet de serre ; lors de la conférence)*.

Pour traiter ces problèmes, les auteurs ont implémenté des contraintes syntaxiques et lexicales sur les phrases de relation à extraire dans le système d'OpenIE *ReVerb*. Par conséquent, ils extraient des relations basées sur des phrases de relation satisfaisant des contraintes syntaxiques et lexicales. Les contraintes syntaxiques permettent de rejeter les extractions incohérentes et non informatives, tandis que les contraintes lexicales permettent d'éviter d'avoir des extractions trop spécifiques. Finalement, le système *ReVerb* suit une approche « les relations d'abord » plutôt que l'approche « les arguments d'abord » de *TextRunner*, car elle identifie d'abord une phrase de relation valide, puis en extrait les arguments.

Nakashole et al. (2012) abordent ce problème sous un autre angle. Ils essaient de rechercher des patrons exprimant diverses relations et les organisent ensuite en hiérarchies. Ils explorent les relations binaires entre les entités et utilisent l'exploration d'items fréquent (Agrawal et al. 1993, Srikant and Agrawal 1996) pour identifier les patrons les plus fréquents. Leur travail aboutit à une ressource appelée *PATTY* qui contient 350569 synsets de patrons et relations et atteint une précision de $84,7\%$.

Contrairement à *ReVerb* (Fader et al. 2011*b*) qui contraint les patrons aux verbes ou aux phrases verbales se terminant par des prépositions, *PATTY* peut apprendre des motifs arbitraires. Les auteurs emploient des modèles dits syntaxiques-ontologiques-lexicaux (modèles SOL). Ces modèles constituent une séquence de mots, de balises de catégories grammaticales, de caractères génériques et de types ontologiques.
Par exemple, le motif (patron) SOL **« Company 's [adj] acquisition * Company »** correspondrait aux segments de texte : *"Google's biggest acquisition to date is Motorola Mobility"* et *"Google's largest acquisition at the time, DoubleClick"* .
Leur approche est basée sur la collecte de chemins de dépendance à partir des phrases où deux entités nommées sont étiquetées (*YAGO2* (Hoffart et al. 2011) est utilisé comme base de connaissance de toutes les entités nommées *NE*). Ensuite, le motif textuel est extrait en trouvant les chemins les plus courts reliant deux entités. Tous ces motifs sont transformés en SOL (abstraction d'un motif textuel). La technique des item-sets fréquents est utilisée pour cela : tous les motifs textuels sont décomposés en n-grammes (n mots consécutifs).
Un modèle SOL contient uniquement les n-grammes qui apparaissent fréquemment dans le corpus et les séquences de mots restantes sont remplacées par des caractères génériques.
L'ensemble de support du modèle est décrit comme l'ensemble de paires d'entités qui apparaissent à la place des espaces réservés d'entité dans toutes les chaînes du corpus qui correspondent au patron. Les motifs (patrons) sont connectés dans un synset (ils sont donc considérés comme synonymes) si leurs ensembles de support coïncident. Le chevauchement des ensembles de support est également utilisé pour identifier les relations de subsomption entre deux synsets.

Des systèmes plus avancés ont considérablement amélioré les performances en exploitant un traitement linguistique plus élaboré.
Del Corro and Gemulla (2013) ont utilisé l'arbre d'analyse syntaxique des dépendances pour décomposer des phrases complexes en un ensemble de clauses indépendantes, où chaque type de clause peut exprimer une extraction avec une structure prédicat-arguments prédéfinie.

L'étiquetage de rôles sémantiques (*Semantic Role Labelling* (SRL)) consiste à détecter les arguments sémantiques associés à un verbe dans une phrase et leurs

rôles, tel que thème, agent et artefact.
La tâche d'étiquetage de rôle sémantique est quelque peu similaire à la tâche d'OpenIE, et en raison de la disponibilité de corpus pour cette tâche, Christensen et al. (2010) ont utilisé un analyseur d'étiquetage de rôle sémantique pour dériver leur système *SRLIE*.
Plusieurs systèmes d'OpenIE extraient des relations exprimées par des verbes et ignorent les relations nominales. Yahya et al. (2014) ont proposé *RENOUN* pour extraire les relations nominales. Pal and Mausam (2016) ont conçu un système OpenIE adapté aux relations exprimées par un Gentilé (*demonyms*) et des noms composés relationnels.
OPENIE4 a été conçu de la fusion des systèmes *SRLIE* (Christensen et al. 2010) et *RelNoun* (Pal and Mausam 2016). Ils ont augmenté *OpenIE4* avec un système d'OpenIE adapté aux relations numériques ainsi qu'un système analysant les conjonctions de coordination afin de concevoir *OpenIE5*.
Gotti and Langlais (2016) ont proposé un système d'OpenIE pour le français en adaptant le système *Reverb* (Fader et al. 2011*a*) afin qu'il extraie des faits simples à partir de Wikipédia français.

3.2.2 Systèmes à base de réseaux de neurones

Plus récemment, des méthodes neuronales ont été utilisées pour la tâche d'OpenIE, en mettant l'accent sur la dérivation automatique de corpus suffisamment large pour entraîner des approches neuronales d'extraction pour l'OpenIE.
Stanovsky and Dagan (2016) a dérivé le premier corpus pour l'OpenIE (*OIE2016*) à partir d'un corpus *QA-SRL (Question Answer - Semantic Role Labeling)*.
Stanovsky et al. (2018*b*) a fusionné le corpus susmentionné avec un corpus de l'OpenIE dérivé du corpus de Question relatif à la représentation de signification abstraite (*Abstract Meaning Representation*), qui englobe un vaste éventail de structures prédicat-argument.
La combinaison des corpus susmentionnés permet d'obtenir un ensemble de données suffisamment vaste et diverse pour entraîner un extracteur précis.
Cui et al. (2018*b*) a dérivé un corpus pour l'OpenIE neuronal en utilisant le bootstraping sur les extractions d'*OpenIE4* (Christensen et al. 2010).
Zhan and Zhao (2019) ont amorcé un corpus pour l'extraction d'informations ouvertes neuronales à partir d'extractions d'*OpenIE4* (Mausam 2016), le corpus

obtenu est fusionné avec le corpus amorcé par Cui et al. (2018*a*). Contrairement à Cui et al. (2018*a*), ils ne filtrent pas les extractions en utilisant le score de confiance, mais ils injectent à la place le score de confiance de l'extraction dans la fonction de perte de leur modèle.
Les ensembles de données obtenus sont suffisamment volumineux pour entraîner des modèles d'apprentissage en profondeur, mais au détriment d'être erronés et de contenir du bruit. Par conséquent, Kolluru, Aggarwal, Rathore, Mausam and Chakrabarti (2020) ont proposé une approche de *Score et Filtre* pour réduire la redondance et le bruit dans ces ensembles de données issus du bootstraping, et également appliquent le bootstrapping sur plusieurs systèmes OpenIE.

Stanovsky et al. (2018*b*) a abordé l'OpenIE comme un problème d'étiquetage de séquence (Sequence Labelling) avec le format d'étiquetage *BIO*
(Begin, Inside, Outside), en utilisant un *Bi-LSTM* avec la fonction *Softmax* pour chaque mot de la phrase. Cui et al. (2018*b*) a formulé l'OpenIE comme un problème de génération de relations, avec une architecture encodeur-décodeur utilisant un mécanisme d'attention. S'inspirant des travaux récents du SRL Ouchi et al. (2018), Zhan and Zhao (2019) a formulé l'OpenIE comme un problème de *Span Sélection*, aussi ils ignorent complètement les relations de dépendance grammaticale en expliquant qu'elles dégradent les performances globales dans leur cas.

Kolluru, Aggarwal, Rathore, Mausam and Chakrabarti (2020) ont utilisé un encodeur *BERT* et un décodeur itératif pour stocker les relations déjà extraites et modéliser leurs interdépendances.
Kolluru, Adlakha, Aggarwal, Mausam and Chakrabarti (2020) ont abordé la tâche d'OpenIE comme une tâche itérative d'étiquetage de grille *2D* (2D Grid Labelling) utilisant un encodeur *BERT*, une telle approche aide à modéliser les dépendances entre les extractions tout en étant beaucoup plus rapide que Kolluru, Aggarwal, Rathore, Mausam and Chakrabarti (2020).
Ils ont également enrichi leur modèle avec un analyseur de coordination pour mieux gérer les structures de coordination complexes, après avoir constaté que les systèmes d'OpenIE actuels échouent à analyser des structures contenant plusieurs entités séparées par de conjonction de coordination.

3.2.3 OpenIE multilingue

La plupart des systèmes OpenIE se concentrent sur la langue anglaise (à quelques exceptions près comme l'espagnol), d'où une pénurie de système OpenIE pour d'autres langues.
La conception de systèmes OpenIE à partir de zéro inclut la création manuelle de règles centrées sur la linguistique, ce qui rend une telle tâche très laborieuse.
Une approche alternative pour alléger le fardeau de la conception de systèmes OpenIE consiste à exploiter les systèmes OpenIE disponibles en anglais pour dériver de tels systèmes dans une autre langue en utilisant le paradigme de transfert. Falke et al. (2016) a montré qu'une telle approche de transfert est efficace en couvrant une grande partie de l'ensemble de règles avec seulement quelques réglages fins, 89% de l'ensemble de règles sont couverts avec seulement 10% de l'effort de la conception du système anglais.

La plupart des systèmes OpenIE pour les langues autres que l'anglais sont des approches ad-hoc à base de règles, avec des performances assez limitées. Parmi ces approches, deux systèmes se distinguent : *ArgOIE* et *PredPatt*.
Gamallo and Garcia (2015) présentent *ArgOIE* qui prend comme entrée l'analyse syntaxique de dépendances au format *CoNLL-X*, identifie les structures d'arguments dans l'analyse des dépendances et extrait un ensemble de propositions basiques de chaque structure d'argument. *ArgOIE* traite l'OpenIE dans les trois langues : anglais, espagnol et portugais.
Similaire à *ArgOIE*, *PredPatt* White et al. (2016) prend lui aussi en entrée l'analyse syntaxique de dépendances au format *Universal Dependency* (Nivre et al. 2016) et retourne un ensemble de structures prédicat-arguments en appliquant des patterns syntaxiques, il peut en principe traiter toutes les langues gérées par *Universal Dependency*.
White et al. (2016) appliquent un ensemble de règles interprétables, extensibles et non lexicalisés sur les dépendances universelles pour extraire les prédicats de relation et leurs arguments correspondants.
PredPatt est évolutif dans toutes les langues, en utilisant des modèles indépendants de la langue ainsi que des modèles de dépendance universels (le même formalisme d'arbre de dépendance pour toutes les langues).

Ro et al. (2020) ont proposé *Multi2OIE*, un modèle d'étiquetage de séquence

pour OpenIE, qui prédit d'abord tous les prédicats de relation en utilisant *BERT*, puis prédit les arguments sujet et objet associés à chaque relation en utilisant des blocs d'attention multi-têtes.
Plus précisément, ils utilisent la version multilingue de *BERT* afin de traiter l'OpenIE dans toutes les langues gérées par *BERT-Multilingue*.
Leur approche à l'avantage de pouvoir s'adapter aux différentes langues sans aucune langue pivot, puisque leur modèle est entraîné seulement sur un corpus en anglais. Ils ont évalué leur approche sur des corpus : en anglais, en espagnol et en portugais, en traduisant le référentiel d'évaluation standard de l'anglais vers l'espagnol et le portugais.

3.3 Hypothèse

Pour imiter l'intelligence humaine, les systèmes à base de règles s'appuient sur un ensemble de règles conçues par l'homme pour représenter le raisonnement du système, le raisonnement est déclenché par inférence : en appliquant l'ensemble de règles sur les connaissances stockées, et peut aboutir à une nouvelle connaissance qui peut être exploité dans une autre inférence.
Les connaissances stockées sur lesquelles l'inférence est appliquée doivent être représentées de manière formelle. Par conséquent, la conception de tels systèmes nécessite un expert du domaine pour représenter efficacement les connaissances et répertorier tous les ensembles de règles pertinents.
Pourtant, la plupart des connaissances du monde ne peuvent pas être exprimées dans une représentation formelle, et l'énumération de toutes les règles n'est pas faisable et impose un fardeau énorme aux experts du domaine.

En revanche, les systèmes basés sur l'apprentissage acquièrent l'ensemble de règles implicitement par l'expérience sans supervision humaine directe.
L'apprentissage automatique classique nécessitait toujours une ingénierie des caractéristiques (*features*) conçue à la main. La qualité de la représentation de ces caractéristiques avait un impact important sur les performances du système et il n'est pas simple pour de nombreuses tâches de les concevoir.
L'apprentissage par représentation résout ce problème en apprenant lui-même ces représentations à partir des données brutes tout en rassemblant les règles de tâches spécifiques. Cette approche permet d'apprendre de bien meilleures représenta-

tions, et réduit considérablement la supervision humaine. Dans un cadre d'apprentissage supervisé, l'apprentissage par représentation vise à apprendre une fonction qui rapproche la correspondance entre les données d'entrée et les données cibles, tout en cherchant à obtenir une représentation qui facilite la réalisation de cet objectif.
Apprendre une bonne représentation est un pas vers la généralisation, car il vise à démêler les facteurs de variation pertinents qui expliquent les données observées et la fonction de mapping tout en écartant les facteurs de variation non pertinents qui ne sont pas liés aux données observées et n'impactent pas la fonction de mapping.

Dans l'apprentissage des représentations, les facteurs de variation sont importants car ils fournissent des explications utiles ou des causes inférées des données observées, ils peuvent également être considérés comme des concepts ou des abstractions qui nous aident à donner un sens riche aux données Goodfellow et al. (2016). Pour éclairer un peu plus sur les facteurs de variation, nous citons l'explication de Desjardins et al. (2012) :
« Considérez, par exemple, la tâche de reconnaissance des expressions faciales. Deux images d'individus différents avec la même expression faciale peuvent donner des images bien séparées dans l'espace des pixels.
En revanche, deux images des mêmes individus présentant des expressions différentes peuvent très bien être positionnées très proches l'une de l'autre dans l'espace des pixels.
Dans ce scénario simplifié, il y a deux facteurs en jeu : (1) l'identité de l'individu et (2) l'expression du visage.
L'un de ces facteurs, l'identité, n'est pas pertinente pour la tâche de reconnaissance des expressions faciales et pourtant des deux facteurs, elle pourrait bien dominer la représentation de l'image dans l'espace des pixels. »
En reconnaissance vocale, les facteurs de variation peuvent être : l'âge et le sexe du locuteur (Goodfellow et al. 2016).
Les facteurs de variation ne sont pas facilement démêlés par l'apprentissage des représentations car ils sont extrêmement couplés dans les données et entraînent des interactions complexes de plusieurs facteurs.
De plus, chaque facteur de variation influence chaque échantillon de données (Goodfellow et al. 2016).

Pour résumer, les modèles d'apprentissage en profondeur devraient apprendre des représentations qui expliquent les causes sous-jacentes des données tout en étant invariables aux facteurs de variation qui ne sont pas informatifs pour la tâche en cours et n'influencent pas la relation entre les données d'entrée et les données cibles.

Un autre problème lié à la généralisation est l'adaptation du domaine, les représentations apprises doivent être invariantes au changement de distribution dû au changement de domaine, ainsi la représentation apprise doit être suffisamment abstraite pour capter les facteurs de variation apparaissant dans plus d'un cadre ou domaine.
Pour pallier cette limitation majeure, l'apprentissage profond (*deep learning*) construit des concepts plus abstraits et complexes qui permettent de comprendre les données, et ainsi de démêler efficacement les facteurs de variation.
Ces concepts de haut niveau et hiérarchiques sont obtenus et appris à partir de représentations ou de concepts très simples, où la représentation de la couche courante est directement dérivée de la représentation de la couche précedente.
Ainsi les réseaux de neurones profonds auront tendance à obtenir des concepts de haut niveau, d'où le nom d'apprentissage profond.
Il est impossible de saisir tous les facteurs de variation corrélés aux données cibles tout en rejetant ceux qui ne sont pas pertinents, car le nombre de facteurs de variation possibles dans les données est souvent énorme.
En apprentissage supervisé, l'impact et l'utilité des facteurs de variation sont évalués à travers la vérité de terrain (corpus d'entraînement) des données cibles.

D'un point de vue linguistique, nous essaierons de mettre en lumière les principaux facteurs de variation de la tâche de l'OpenIE en analysant les exemples du tableau 3.2.

Dans les relations correspondant aux phrases (1) et (2), les mots *Some statements* et *John* partagent le même rôle syntaxique dans la relation (sujet), bien qu'étant sémantiquement différents (ils ont un sens différent) .
Les phrases (3) et (4) ont des structures syntaxiques différentes mais expriment la même relation, ces deux relations sont sémantiquement similaires, malgré des prédicats différents (dans la phrase (3) le sujet et l'objet sont liés par une apposi-

Id	Phrase	Sujet	Prédicat	Objet
1	Some statements speak about themselves.	Some statements	speak about	themselves
2	John speaks about himself.	John	speaks about	himself
3	Vegetarian, Leon took a pizza.	Leon	[be]	Vegetarian
3	Vegetarian, Leon took a pizza.	Leon	took	a pizza
4	Leon is vegetarian.	Leon	is	vegetarian
5	Leon ordered a pizza.	Leon	ordered	a pizza
6	United States President Obama.	Obama	[be] President [of]	United States
7	New York Banker Association.	-	-	-

TABLE 3.2 Exemples de triplets d'Open Information Extraction.

tion, alors qu'ils sont liés par une copule dans les phrases (4)).
Les phrases (3) et (5) expriment également des relations sémantiquement similaires, bien qu'étant liées par deux prédicats différents (mais sémantiquement proches).
Les phrases (6) et (7) sont syntaxiquement similaires, mais dans la phrase (7) nous ne pouvons pas avoir la relation (Association ; [be] Banker [of] ; New York), puisqu'il s'agit d'un nom d'organisation.

Dans l'OpenIE, les relations peuvent être exprimées par des prédicats verbaux ou des prédicats nominaux.
Nous pouvons voir que dans la relation prédicat basée sur le verbe, la sémantique lexicale et la signification des mots ne sont pas un facteur clé de variation pour la structure prédicat-argument d'une relation, et c'est toujours la structure syntaxique et la règle de grammaire qui détermineront la structure prédicat-argument. Cependant, les relations nominales ne sont pas toujours syntaxiques, et dans certains cas, la sémantique lexicale et la signification des mots sont nécessaires, ces exceptions s'appliquent principalement aux Gentilé (Demonyms) (Pal et al. 2016).
Ainsi, nous proposons une approche exploitant seulement l'analyse syntaxique et ignorant la sémantique lexicale pour apprendre à raisonner à partir de la grammaire, réduire la quantité de données nécessaires et améliorer à la fois la généralisation et la robustesse du modèle.

3.4 Réseaux de neurones récurrents

Un réseau de neurones récurrents est un réseau de neurones artificiels présentant des connexions récurrentes.
Un réseau de neurones récurrents est constitué d'unités (neurones) interconnectées interagissant non-linéairement et pour lequel il existe au moins un cycle dans la structure. Les unités sont reliées par des arcs (synapses) qui possèdent un poids. La sortie d'un neurone est une combinaison non linéaire de ces entrées.
Les réseaux de neurones récurrents sont adaptés pour des données d'entrée de taille variable. Ils conviennent en particulier pour l'analyse de séries temporelles. Ils sont utilisés en reconnaissance automatique de la parole ou de l'écriture manuscrite - plus en général en reconnaissance de formes - ou encore en traduction automatique.
« Dépliés », ils sont comparables à des réseaux de neurones classiques avec des contraintes d'égalités entre les poids du réseau (voir schéma 4).
Les techniques d'entraînement du réseau sont les mêmes que pour les réseaux classiques (rétropropagation du gradient).
Néanmoins, les réseaux de neurones récurrents se heurtent au problème de disparition du gradient pour apprendre à mémoriser des évènements passés.
Des architectures particulières répondent à ce dernier problème, parmi lesquelles nous pouvons citer les réseaux *Long short-term memory* (LSTM) (Hochreiter and Schmidhuber 1997).
On peut étudier les comportements des réseaux de neurones récurrents avec la théorie des bifurcations, mais la complexité de cette étude augmente très rapidement avec le nombre de neurones.

Le réseau *LSTM*, ou réseau de neurones récurrents à mémoire court-terme et long terme, est l'architecture de réseau de neurones récurrents la plus utilisée en pratique, qui permet de répondre au problème de disparition de gradient (voir schéma 5).
L'idée associée au *LSTM* est que chaque unité computationnelle est liée non seulement à un état caché h mais également à un état c de la cellule qui joue le rôle de mémoire.

Le passage de c_{t-1} à c_t se fait par transfert à gain constant et égal à 1.

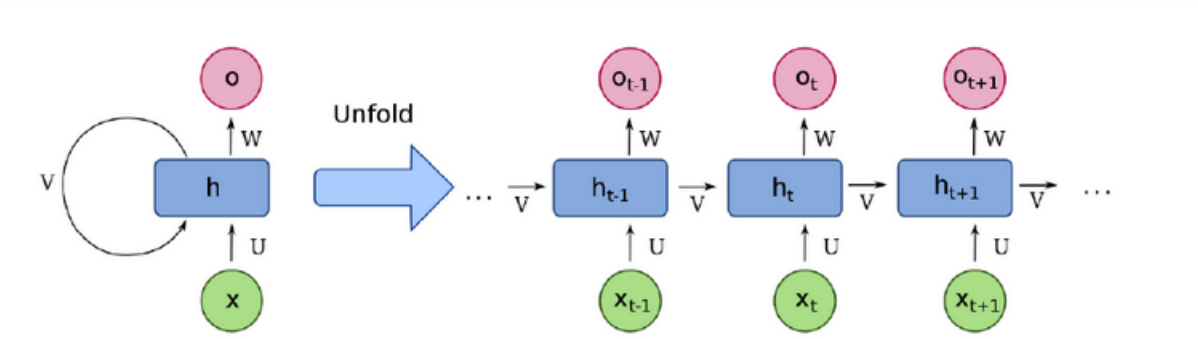


FIGURE 4 Schéma d'un réseau de neurones récurrents à une unité reliant l'entrée et la sortie du réseau.

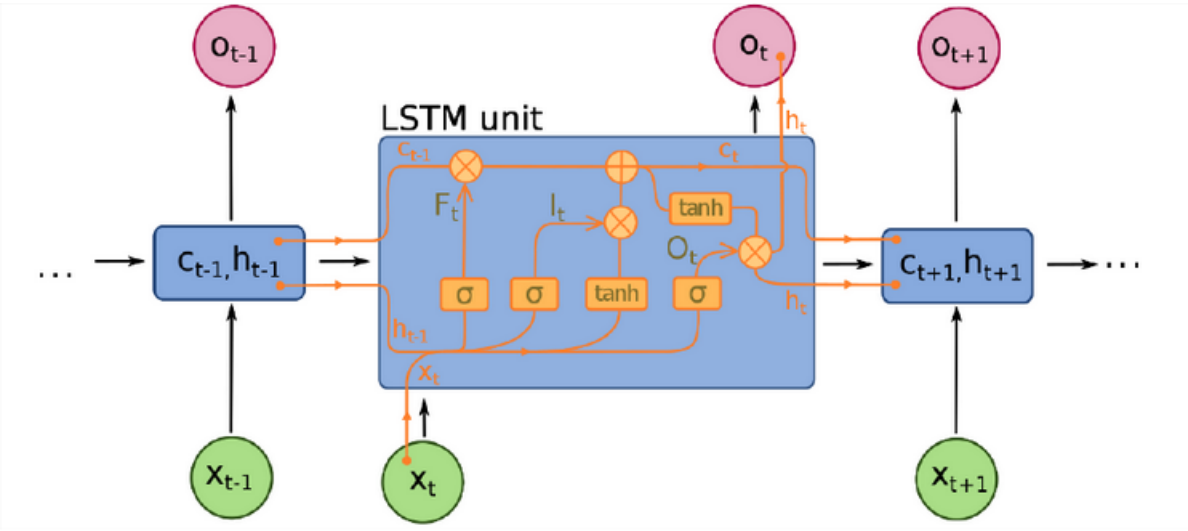


FIGURE 5 Schéma d'un réseau de neurones récurrents à mémoire court-terme et long terme

De cette façon les erreurs se propagent aux pas antérieurs (jusqu'à 1000 étapes dans le passé) sans phénomène de disparition de gradient.
L'état de la cellule peut être modifié à travers une porte qui autorise ou bloque la mise à jour (porte des entrées).
De même une porte contrôle si l'état de cellule est communiqué en sortie de l'unité *LSTM* (porte des sorties).
La version la plus répandue des réseaux de neurones récurrents, *LSTM* utilise aussi une porte permettant la remise à zéro de l'état de la cellule (forget gate).

3.5 Champ aléatoire conditionnel

Étant donné la séquence d'entrée du décodeur $H = \{h_i\}_{i=1}^{n}$ et une séquence d'étiquettes $Y = \{y_i\}_{i=1}^{n}$, le décodeur calcule le score de décodage $S(H, Y)$.

$$S(H, Y) = \sum_{i=1}^{n-1} A_{y_i, y_{i+1}} + \sum_{i=1}^{n} H_{i, y_i} \tag{3.1}$$

H est une matrice d'émission $n \times k$, où n est la longueur de la séquence,
k le nombre d'étiquettes distinctes et H_{ij} est le score de j-ème tag à la position i de la séquence.
A est une matrice de transition $k \times k$, où A_{ij} représente le score de transition du i-ème tag vers le j-ème tag.
Puis $p(Y|H)$ est calculée, une probabilité conditionnelle sur toutes les séquences d'étiquettes possibles Y en utilisant la fonction Softmax, où Y_H représente les séquences d'étiquettes possibles pour H.

$$p(Y|H) = \frac{\mathrm{e}^{S(H,Y)}}{\sum_{Y' \in Y_H} \mathrm{e}^{S(H,Y')}} \tag{3.2}$$

Lors du décodage, nous recherchons la séquence ayant le score maximum y^*, en utilisant l'algorithme de Viterbi (Forney 1973).

$$y^* = argmax_{Y \in Y_H} S(H, Y) \tag{3.3}$$

3.6 Méthodologie

Nous présentons en détail notre approche générale pour l'extraction d'information ouvert dans cette section.
Tout d'abord, nous donnons la formulation de la tâche et l'aperçu de notre approche neuronale pour l'OpenIE dans la section 3.6.1 et la section 3.6.2.

3.6.1 Définition du Problème

Étant donné une phrase $S = (w_1, w_2, ..., w_n)$, nous dérivons d'abord l'arbre syntaxique des dépendances pour obtenir l'étiquetage grammatical et les relations de dépendance syntaxique.
Nous transmettons ces plongements au modèle pour produire une balise de séquence $T = (y_1, y_2, ..., y_n)$, avec l'ensemble de balises $Y = \{A0, P, A1, A2, O\}$. La séquence produite représente le tuple ($A0$: sujet, P : prédicat, $A1$: objet ...) au format modèle ***BIES*** (***Begin, Inside, End, Single***).

Phrase	Brady tente dappeler le Shérif .
Séquence d'étiquettes	$A0_S\ P_B\ P_I\ P_E\ A1_B\ A1_E\ O$
Représentation de la sortie	Brady$_{A0_S}$ tente$_{P_B}$ d'$_{P_I}$ appeler$_{P_E}$ le$_{A1_B}$ Shérif$_{A1_E}$.$_O$
Relation	($A0$:Brady, P :tente d'appeler, $A1$:le Shérif)

TABLE 3.3 Exemple de la représentation de la sortie du modèle.

3.6.2 Approche

S'inspirant de Stanovsky et al. (2018*b*), nous abordons la tâche d'OpenIE comme un problème d'étiquetage de séquence (***Sequence Labelling***) avec le format d'étiquetage ***BIES*** (***Begin, Inside, End, Single***).
A partir d'une phrase donnée $S = (w_1, w_2, ..., w_n)$, l'étiquetage de séquence vise à assigner à chaque mot de la phrase l'étiquette la plus probable, donnant lieu à une séquence de labels $T = (y_1, y_2, ..., y_n)$.
On extrait une relation à la fois, en considérant à chaque itération un mot de la phrase comme potentiel prédicat de la relation, à partir duquel nous déduisons un masque binaire $M = (m_1, m_2, ..., m_n)$.

Notre modèle proposé se compose de deux modules faiblement couplés, le premier module gère l'inférence de prédicat et d'arguments, et alimente les bornes du prédicat inférées au second module, qui classe les arguments extraits.
De plus, nos deux modules ne sont pas optimisés conjointement et chacun est entraîné séparément, l'interaction entre eux se produit uniquement lorsque le premier alimente les bornes du prédicat au second.
Nous conjecturons que le module de prédicat ne doit pas être pénalisé lorsque les prédictions du module d'argument sont erronées, nous pensons également que la phase d'inférence d'argument est plus difficile que l'inférence de prédicat.

Le premier module est un modèle multi-tâches qui extrait la relation basée sur le prédicat, puis cherche à trouver tous les arguments étant donné la relation de prédicat extraite. Le deuxième module prend en entrée le prédicat et les arguments extraits, puis attribue l'étiquette la plus probable à chaque argument identifié tel que sujet, objet, argument temporel ou argument de localisation.
La raison d'une telle approche découle des tendances récentes dans l'analyse des dépendances syntaxiques neuronales (Dozat and Manning 2016), qui consiste à trouver la structure de dépendance syntaxique non étiquetée (topologie de l'arbre syntaxique), et seulement ensuite attribuer une étiquette pour chaque arc prédit de l'arbre.
Plus précisément, pour chaque paire de mots leur modèle calcule la probabilité d'existence d'un arc reliant ces deux mots ainsi qu'une étiquette de fonction syntaxique pour chaque arc de l'arbre syntaxique.
Contrairement à leur approche, nous ne calculons que la probabilité entre un mot et la plage de mots représentant la phrase du prédicat extraite à l'étape précédente. L'inférence de l'étiquette liant les deux syntagmes correspondant au type d'argument est gérée par le module d'étiquetage.
Dans notre configuration, les arcs prédits indiquent les arguments extraits de la relation de prédicat, ces arguments extraits seront affinés et étiquetés à l'étape suivante.

3.7 Modèle et architecture

Nous présentons en détail notre méthode proposée dans cette section.
Tout d'abord, nous présentons la représentation d'entrée dans la section 3.7.1,

notre architecture de modèle pour OpenIE dans est présentée dans les sections 3.7.2 et 3.7.3.

3.7.1 Représentation des entrées

Nos deux sous-modules exploitent les mêmes entrées, à l'exception du modèle d'arguments qui attend les bornes inférées par le modèle de prédicat, et utilise un masque différent pour modéliser la topologie de l'arbre syntaxique.
Nous utilisons la bibliothèque Stanza (Qi et al. 2020) pour obtenir l'étiquetage grammatical (*POS*) et l'arbre d'analyse syntaxique (relations de dépendances), avec la représentation Universal Dependency (Nivre et al. 2016).
Les vecteurs de l'étiquetage grammatical et des relations syntaxiques sont obtenus en utilisant l'encodage *One-Hot Encoding* (encodage 1 parmi n) où chaque catégorie est assignée à un vecteur différent.

La représentation des données d'entrée (textuelles) dans un espace vectoriel est un élément essentiel de tout système de traitement automatique de texte. Il permet de représenter les données de manière mathématiques et rend possible l'exploitation des données en aval.
La représentation par défaut des mots dans un modèle d'apprentissage machine est l'encodage *One-Hot Encoding*, qui consiste à représenter chaque mot par un vecteur d'une taille correspondant au vocabulaire entier considéré pour chaque variable textuelle manipulée par le modèle.
Le terme *One-Hot* indique qu'une seule valeur du vecteur contient la valeur 1 tandis que les autres contiennent des 0. Cet encodage permet d'assurer que les vecteurs soient uniques et consistants pour l'implémentation du modèle.
La figure 6 illustre un exemple de vectorisation dense d'un vecteur encodé selon le format *One-Hot*.

3.7.2 Extraction de la structure Prédicat-Argument

Notre premier sous-module extrait la représentation prédicat-argument tout en ignorant l'étiquette ou le type des arguments. Par conséquent, le sous-module est optimisé vis-à-vis de deux tâches : l'extraction de relation de prédicat et l'identification des arguments, La dernière tâche dépendant de la sortie de la tâche précé-

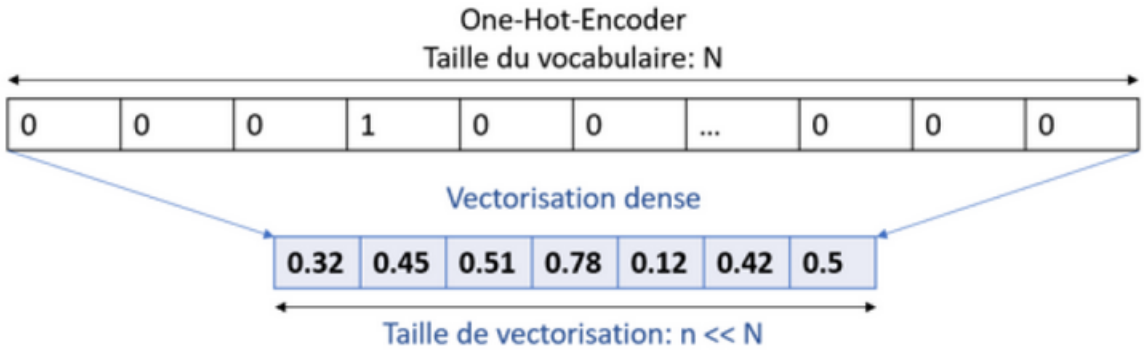


FIGURE 6 Objectif de la vectorisation dense.

dente.
Les entrées pour le sous-module sont la concaténation des trois vecteurs de caractéristiques :

— E_{pos} représente le plongement vectoriel de l'étiquetage grammatical
— E_{dep} représente le plongement vectoriel de la relation syntaxique
— E_{masque} représente le masque de prédicat binaire.

Puisque nous extrayons une relation à la fois, E_{masque} est un simple vecteur binaire pour indiquer quel mot de la phrase est le prédicat candidat.
Le sous-module partage une couche *BiLSTM* pour les deux tâches et exploite une couche de champs aléatoires conditionnels (*CRF*) (Lafferty et al. 2001) pour chaque tâche.
Étant donnée une instance d'entrée (S, M) avec S une phrase et M un vecteur binaire (0 et 1), pour chaque mot $w_i \in S$ nous calculons un vecteur de caractéristiques :

$$x_i = E_{pos}(w_i) \oplus E_{dep}(w_i) \oplus E_{masque}(w_i) \tag{3.4}$$

Le vecteur de caractéristiques (équation 3.4) est transmis au *BiLSTM*, qui calcule une représentation conceptualisée bidirectionnelle de chaque mot de la phrase (le contexte précédent (*forward*) et le contexte suivant (*backward*) de chaque mot). Après, la moyenne des représentations conceptualisées du contexte précédant et suivant du *BiLSTM* est calculée pour chaque mot et est transmise à une couche

dense.

$$v_i^{\rightarrow}, v_i^{\leftarrow} = Bi - LSTM(x_i) \tag{3.5}$$

$$u_i = AVG(v_i^{\rightarrow}, v_i^{\leftarrow}) \tag{3.6}$$

$$h_i = Wu_i + b \tag{3.7}$$

Ensuite, la représentation est transmise au décodeur de chaque tâche.
La sortie de l'encodeur (équation 3.7) est d'abord transmise à l'extracteur de prédicat, qui identifie le prédicat.
Après avoir extrait le prédicat (équation 3.3), la phrase de prédicat est transmise à l'extracteur d'arguments en tant que vecteur binaire qui indique les bornes du prédicat extrait.
Enfin, la sortie de l'encodeur (équation 3.7) est concaténée avec la sortie du décodeur *CRF* du prédicat (équation 3.3) et est envoyée au décodeur *CRF* de l'extracteur d'arguments.
La nouvelle représentation est donnée par l'équation suivante :

$$h_i(Argument) = h_i \oplus y_i(Predicat) \tag{3.8}$$

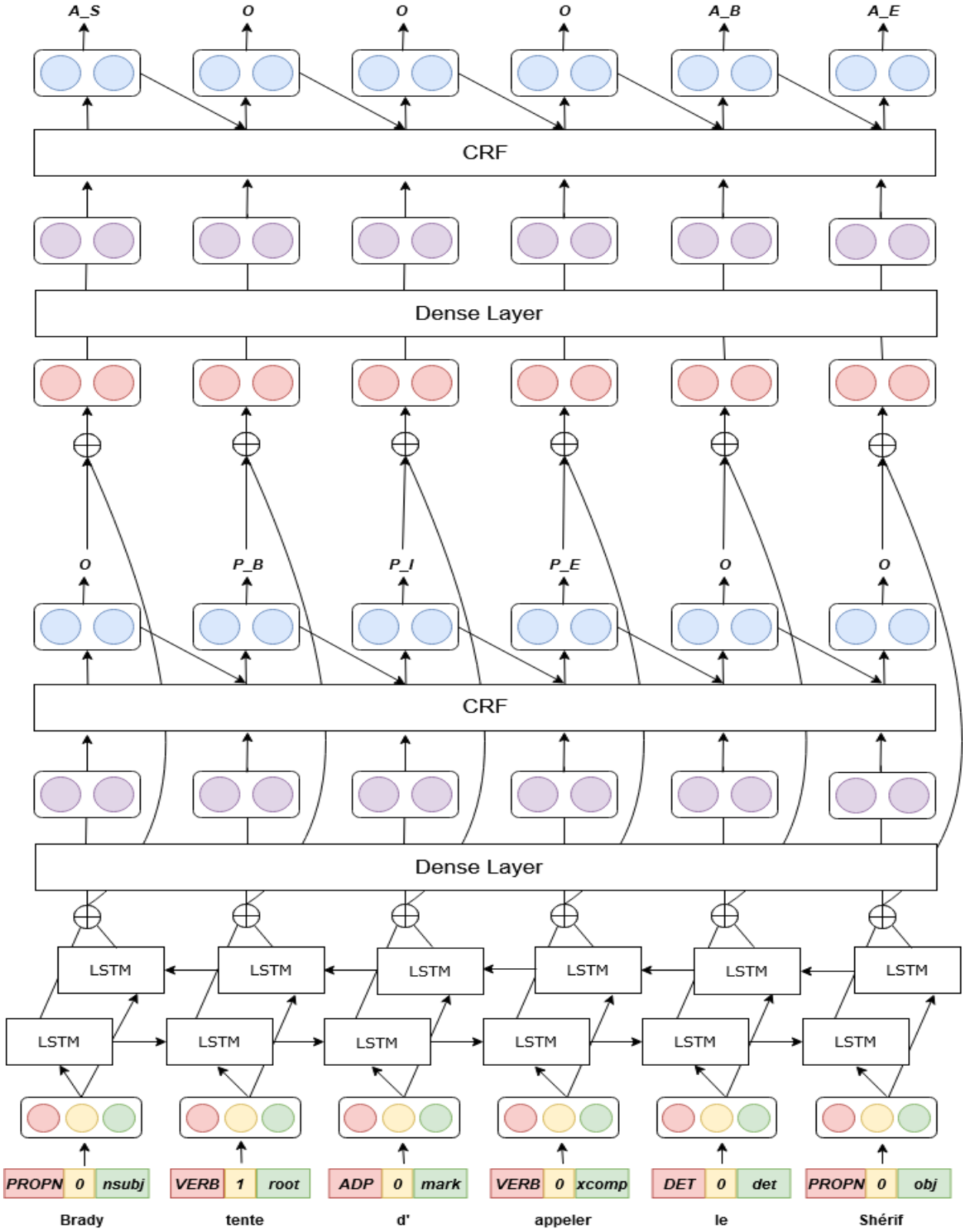
Les deux tâches sont optimisées conjointement et nous maximisons la vraisemblance logarithmique de la séquence d'étiquettes correcte de chaque tâche sur l'ensemble d'apprentissage $\{(H_j, Y_j)\}$, en minimisant la fonction de coût : la vraisemblance logarithmique négative (*Negative Log Likelihood*) (NLL) (Yao, Zhu, Jiang and Yu 2019), et qui est exprimé par l'équation suivante :

$$NLL = -\sum_j \log p(Y|H) \tag{3.9}$$

La fonction de coût du module est simplement la somme de la fonction de coût de chaque tâche :

$$NLL = -\sum_j \log p(Y|H)_{predicat} - \sum_j \log p(Y|H)_{argument} \tag{3.10}$$

FIGURE 7 Extraction de la structure prédicat-arguments.



3.7.3 Étiquetage des arguments

Après l'extraction de la structure prédicat-arguments, le premier sous-module transmet le prédicat et les arguments extraits au deuxième module.
En plus de l'incorporation des étiquettes grammaticales et des relations de dépendance syntaxique, il prend comme entrée un autre vecteur, qui indique pour chaque mot de la phrase s'il : fait partie de la phrase du prédicat, est un argument du prédicat extrait ou aucun des deux.
L'entrée du modèle est le vecteur de caractéristiques défini par l'équation 3.11, et consiste en la concaténation des vecteurs suivants :

— E_{pos} représente le plongement vectoriel de l'étiquetage grammatical
— E_{dep} représente le plongement vectoriel de la relation syntaxique
— E_{masque} représente le vecteur inféré dans la première étape qui représente le prédicat et les arguments extraits.

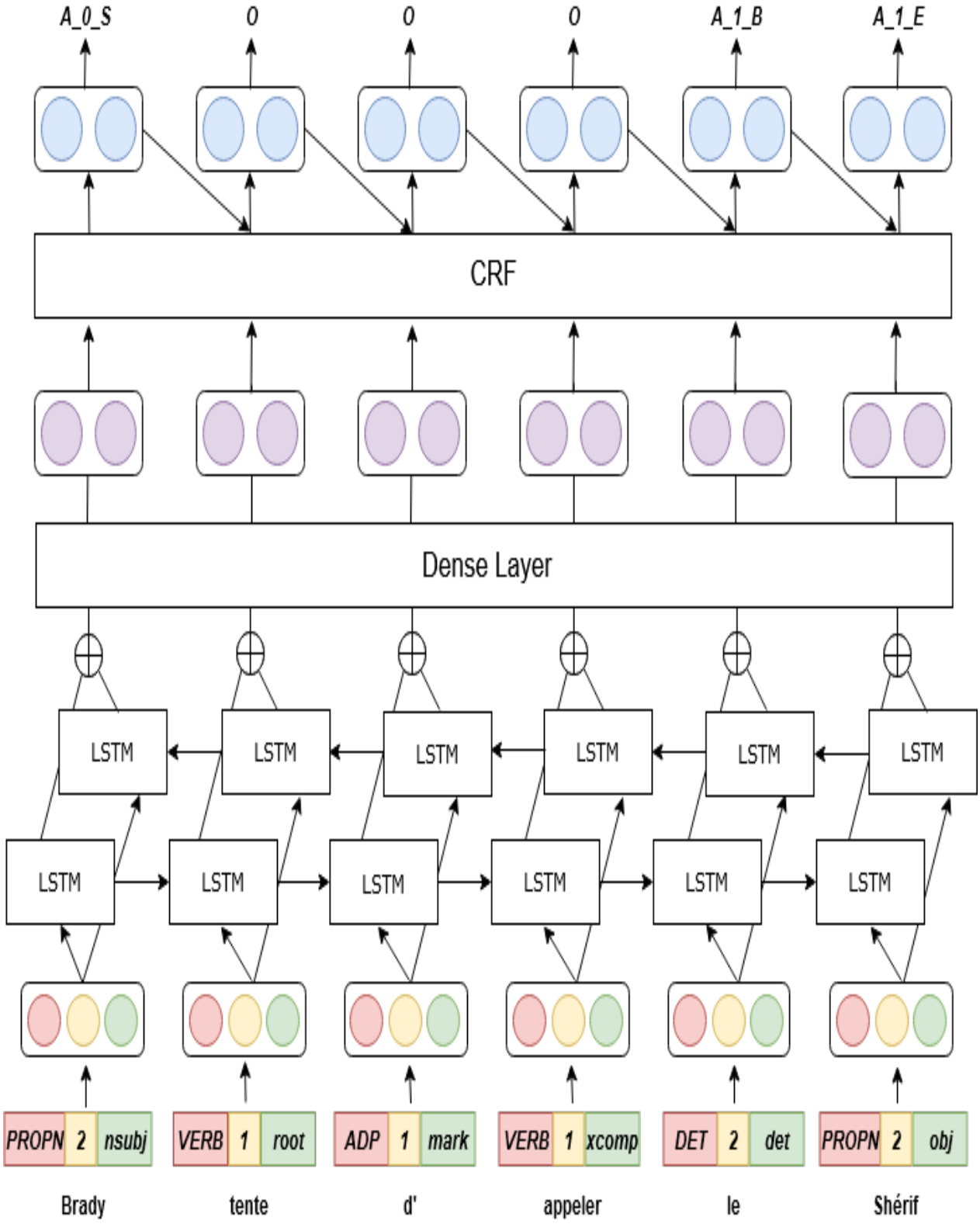
$$x_i = E_{pos}(w_i) \oplus E_{dep}(w_i) \oplus E_{pr-arg}(w_i) \tag{3.11}$$

Le module exploite la même architecture que le premier module qui consiste en un décodeur *CRF* empilé sur une couche *BiLSTM* et cherche à attribuer l'étiquette la plus probable aux arguments extraits lors de la première étape.
Comme l'extracteur d'arguments de prédicat, le modèle est optimisé pendant l'entraînement en minimisant la *Negative Log Likelihood* (équation 3.10) .

3.8 Expérimentations

Dans cette section, les jeux de données d'apprentissage et les hyperparamètres sont respectivement présentés dans les sections 3.8.1 et 3.8.2, puis les sections 3.8.3 et 3.8.4 décrivent la stratégie d'évaluation et le référentiel d'évaluation.
Nous présentons l'étude d'ablation et les systèmes de référence dans les sections 3.8.5 et 3.8.6.
Nous concluons l'étude expérimentale par une analyse des performances du temps d'exécution dans la section 3.8.7.

FIGURE 8 Étiqueteur des arguments.



3.8.1 Données d'entraînements

Contrairement aux travaux précédents, nous choisissons des ensembles de données d'entraînement annotés manuellement utilisés dans (Del Corro and Gemulla 2013) comme données d'entraînement. Étant donné que ces ensembles de données contiennent des relations binaires, nous les annotons à nouveau pour convertir les relations binaires en relations n-aires.
L'annotation suit la directive de (Lechelle et al. 2019), à l'exception de la résolution d'Anaphore. Le tableau 3.4 décrit les jeux de données après ré-annotation.

TABLE 3.4 Données d'entraînements.

Corpus d'apprentissage	#Phrases	#Relations
Reverb	500	1 551
New York Times	200	642
Wikipedia	200	568

3.8.2 Hyperparameters

Le tableau 3.5 ci-dessous, reprend les hyperparamètres de notre modèle, qui sont les mêmes pour les deux sous-modules. Nous avons entraîné notre modèle à l'aide de l'optimiseur *Adam*.

3.8.3 Stratégie d'évaluation

Nous utilisons la stratégie d'évaluation standard *CARB* Bhardwaj et al. (2019) pour évaluer notre système et les systèmes de références.
Cette stratégie ne compare pas les extractions en utilisant la correspondance lexicale des deux extractions, mais va plutôt la correspondance lexicale entre les arguments séparément et les prédicats séparément.
Suivant Kolluru, Adlakha, Aggarwal, Mausam and Chakrabarti (2020), nous rapportons également les résultats de la fonction de notation *CARB(1-1)* qui pénalise la division incorrecte des structures de coordination. Nous rapportons le score F1 et l'AUC (Area Under the Curve).
Notre modèle et les fonctions de références sont évalués en exploitant le code et les données utilisées par Kolluru, Adlakha, Aggarwal, Mausam and Chakrabarti (2020) dans leur travail.

TABLE 3.5 hyper-paramètres.

hyper-paramètres du modèle	
Taille de l'état caché du LSTM	128
Dropout de l'état récurrent du LSTM	0.3
Dropout de l'entrée du LSTM	0.3
Dropout de la sortie du LSTM	0.3
Dropout du vecteur	0.1
Regularization L2	0.001
Taille du vecteur	20
Taille du batch	5
Taux d'apprentissage	0.001
Nombre des Hyper-Paramètres	
Extraction de la structure Prédicat-Argument	590 553
Étiquetage des arguments	592 911
Model complet	1 183 464

3.8.4 Référentiel d'évaluation

Au vu de l'absence de référentiel d'évaluation pour le Français, nous annotons aussi un référentiel d'évaluation en prenant des phrases issues d'articles de journaux du domaine de la finance, et qui ont été décrites dans (Jabbari et al. 2020). Nous avons décidé de choisir un domaine différent de celui des données d'apprentissage qui couvre principalement des phrases issues du Web.
Pour annoter le corpus, nous suivons les recommandations d'annotation de (Lechelle et al. 2019), qui ont aussi été suivis par Bhardwaj et al. (2019) pour construire *CARB* : le référentiel d'évaluation pour l'anglais.

Cependant, nous avons observé quelques annotations et extractions complexes ou ambiguës :

— **Conjonction de coordination** : Lechelle et al. (2019) préconisent de séparer les conjonctions de coordination dans les arguments pour générer plusieurs relations, sauf que la conjonction de coordination avec le connecteur *et* peut faire l'objet de deux interprétations, l'une cumulative (parfois dite collective) et l'autre distributive.
Dans le cas d'une conjonction de coordination cumulative, nous avons trouvé

des difficultés à trouver la meilleure annotation de la relation. Nous avons décidé de ne pas séparer la conjonction de coordination dans l'argument.
Par exemple, dans la phrase : *Plus tard , Han Sui et Ma Teng ont partagé une relation difficile l'un avec l'autre.* nous avons l'extraction suivante : ($A0$: Han Sui et Ma Teng, P : ont partagé une relation difficile, $A1$: l'un avec l'autre).

— **Coréférence** : Contrairement aux recommandations d'annotation, nous avons opté pour la non-résolution des Coréférences dans les extractions.

— **Apposition** : nous avons aussi décidé d'inclure l'extraction introduite par l'apposition alors que cette extraction peut être considérée comme redondante.
Par exemple, dans la phrase : *La livraison de l'A321 s'est déroulée en présence de le pdg de la compagnie nationale Iranienne, Farhad Parvaresh*, nous avons les deux extractions :
($A0$:La livraison de l'A321, P :s'est déroulée en présence de, $A1$:le pdg de la compagnie nationale Iranienne)
et ($A0$:La livraison de l'A321, P :s'est déroulée en présence de, $A1$:Farhad Parvaresh).

Le référentiel d'évaluation final tel indiqué dans le tableau 3.6 se compose de 506 phrases et $1\ 783$ relations. Afin d'évaluer les systèmes OpenIE multilingues sur l'espagnol et le portugais, Ro et al. (2020) ont dérivé les référentiels d'évaluation *Re-OIE2016_Sp* et *Re-OIE2016_Pt* en traduisant le référentiel d'évaluation anglais *Re-OIE2016*. Nous utilisons ces deux référentiels d'évaluation pour évaluer les différents systèmes sur l'espagnol et le portugais. Nous utilisons le référentiel d'évaluation standard *CARB* pour évaluer les systèmes OpenIE en anglais.

TABLE 3.6 Référentiels d'évaluation.

Corpus d'évaluation	#Phrases	#Relations
CARB	641	2 715
Re-OIE2016_Sp	595	1 508
Re-OIE2016_Pt	595	1 508
Finance_French	506	1 783

3.8.5 Étude d'ablation

Nous appliquons une étude d'ablation pour étudier l'impact de notre nouvelle architecture, qui vise à séparer l'identification et l'étiquetage des arguments.
Par conséquent, nous considérons un système de référence légèrement similaire à l'architecture utilisée par Zhan and Zhao (2019).
Notre architecture proposée introduit une étape auxiliaire pour identifier les arguments du prédicat extrait avant d'étiqueter ces arguments extraits, tandis que Zhan and Zhao (2019) identifient et étiquettent les arguments du prédicat extrait simultanément.

3.8.6 Références

Nous nous référons à notre modèle comme *UD2OIE*, tandis que nous nous référons au système de référence défini dans la section de l'étude d'ablation comme *UD2OIE (-Arg Identification)*.
Pour l'évaluation en anglais sur le référentiel *CARB*, nous choisissons des approches basées sur les règles, l'étiquetage de séquences neuronales et la génération de relations neuronales.
Nous choisissons *ClauseIE* (Del Corro and Gemulla 2013), *OpenIE4* (Christensen et al. 2010) et *OpenIE5* (Christensen et al. 2010) comme systèmes de référence basés sur les règles.
En ce qui concerne les systèmes de référence basées sur l'étiquetage des séquences, nous choisissons *RnnOIE* (Stanovsky et al. 2018*b*), *SpanOIE*(Zhan and Zhao 2019) et *OpenIE6* (Kolluru, Adlakha, Aggarwal, Mausam and Chakrabarti 2020).
Et les systèmes de référence retenus pour les approches de génération de relations sont *NeuralOIE* (Cui et al. 2018*b*) et *IMOJIE* (Kolluru, Aggarwal, Rathore, Mausam and Chakrabarti 2020).
Enfin, pour l'évaluation multilingue, nous choisissons les deux approches basées sur les règles : *PredPatt* (White et al. 2016) et *ArgOIE* (Gamallo and Garcia 2015), tandis que le seul système de référence à base de réseau de neurones disponible est *Multi2OIE* (Ro et al. 2020).

3.8.7 Performance du temps d'exécution

Puisque les systèmes OpenIE doivent s'adapter à la diversité et à la taille du corpus Web, nous rapportons également le temps d'inférence de notre modèle sur un lot de 3 200 phrases (8 477 relations) (Stanovsky et al. 2018*b*), qui a également été utilisé dans (Kolluru, Adlakha, Aggarwal, Mausam and Chakrabarti 2020) pour rapporter la vitesse des différents systèmes. Contrairement à Kolluru, Adlakha, Aggarwal, Mausam and Chakrabarti (2020) qui rapportait les performances de vitesse des systèmes de référence basés sur les réseaux de neurones à l'aide d'un GPU V100, nous rapportons la vitesse de notre modèle en utilisant 4 cœurs de processeur Intel Core i5-8300H. Les performances de vitesse des systèmes non neuronaux ont été rapportées dans (Kolluru, Adlakha, Aggarwal, Mausam and Chakrabarti 2020) en utilisant 4 cœurs de processeur Intel Xeon.
Nous rapportons la vitesse de notre modèle avec et sans le temps d'exécution de l'analyseur de dépendance.

3.9 Résultats et Analyse

Cette section présente les résultats clés des expérimentations dans les sections 3.9.1 et 3.9.2. L'étude d'ablation et les résultats d'adaptation au domaine sont discutés dans les sections 3.9.3 et 3.9.4.
Enfin, l'analyse du temps d'exécution est rapportée dans les sections 3.9.5, et 3.9.6 fournit une analyse d'erreur du modèle.
Le tableau 3.7 montre des exemples d'extraction multilingues de notre modèle.

3.9.1 Résultats de performance monolingue

Les résultats de performance pour chaque système sur le référentiel anglais *CARB* avec les métriques présentées sont rapportés dans le tableau 3.8.
Les résultats de l'évaluation montrent que notre méthode proposée surpasse de loin les autres systèmes de référence.
De ces résultats, nous concluons que pour cette approche la qualité des données d'apprentissage est plus importante que la quantité. En effet, contrairement aux approches d'apprentissage profond traditionnelles qui nécessitent de vastes corpus d'apprentissage obtenus automatiquement (ce qui introduit du bruit et des annotations erronées), notre approche n'a besoin que d'un corpus d'apprentissage annoté manuellement mais relativement petit du fait qu'elle n'utilise que l'analyse

Phrase	Returning home, Ballard delivers her report, which her superiors refuse to believe.
Anglais	(**A0 :**Ballard **; P :**Returning **; A1 :**home) (**A0 :**Ballard **; P :**delivers **; A1 :**her report) (**A0 :**her superiors **; P :**refuse to believe **; A1 :**her report)
Phrase	De retour chez elle, Ballard livre son rapport, que ses supérieurs refusent de croire.
Français	(**A0 :**Ballard **; P :**De retour chez **; A1 :**elle) (**A0 :**Ballard **; P :**livre **; A1 :**son rapport) (**A0 :**ses supérieurs **; P :**refusent de croire **; A1 :**son rapport)
Phrase	Al volver a casa, Ballard entrega su informe, que sus superiores se niegan a creer.
Espagnol	(**A0 :**Ballard **; P :**volver a **; A1 :**casa) (**A0 :**Ballard **; P :**entrega **; A1 :**su informe) (**A0 :**sus superiores **; P :**niegan a creer **; A1 :**su informe)
Phrase	Voltando para casa, Ballard entrega seu relatório, que seus superiores se recusam a acreditar.
Portugais	(**A0 :**Ballard **; P :**Voltando para **; A1 :**casa) (**A0 :**Ballard **; P :**entrega **; A1 :**seu relatório) (**A0 :**seus superiores **; P :**se recusam a acreditar **; A1 :**seu relatório)

TABLE 3.7 Exemples d'extraction de UD2OIE pour chaque langue.

syntaxique qui restitue des représentations assez abstraites.

Système	CARB		CARB(1-1)	
	F1	AUC	F1	AUC
ClauseIE	45.0	22.0	40.2	17.7
OpenIE4	51.5	29.1	40.4	19.7
OpenIE5	46.7	24.5	41.2	19.6
SpanOIE	48.5	-	37.9	-
NeuralOIE	51.6	32.8	38.7	19.8
RnnOIE	49.0	26.0	39.5	18.3
IMOJIE	53.5	33.3	41.4	22.2
OpenIE6	52.7	33.7	46.4	26.8
UD2OIE	**58.2**	**39.0**	**49.9**	**29.7**

TABLE 3.8 Résultats de l'évaluation monolingue des systèmes d'OpenIE sur le référentiel standard CARB (anglais).

3.9.2 Résultats de performance cross-lingues

Les résultats de performance multilingue pour chaque système sur les quatre référentiels d'évaluation selon la stratégie d'évaluation *CARB* sont rapportés dans le tableau 3.9.
Les résultats de l'évaluation montrent que notre méthode surpasse tous les systèmes OpenIE multilingues dans tous les référentiels d'évaluation.
Les résultats de l'évaluation démontrent l'impact d'exploiter la représentation *Universal Dependency* pour l'analyse syntaxique qui associe une représentation uniforme pour l'analyse syntaxique dans les quatre langues. Notre système arrive à s'adapter à d'autres langues bien qu'il n'ait été entraîné que sur des corpus d'apprentissage en anglais.

3.9.3 Résultats de l'étude d'ablation

Les résultats de l'ablation résumés dans le tableau 3.10, montrent que notre architecture apporte un gain de performance dans tous les cas-tests.
L'architecture proposée cible les performances de rappel. Elles les améliore les

Système	Anglais		Français		Espagnol		Portugais	
	F1	AUC	F1	AUC	F1	AUC	F1	AUC
ArgOIE Gamallo and Garcia (2015)	36.4	24.4	-	-	39.4	28.3	38.3	26.4
PredPatt White et al. (2016)	44.6	34.6	42.0	34.7	44.3	39.8	42.9	38.0
Multi2OIE Ro et al. (2020)	52.1	31.5	43.2	24.5	61.5	43.2	61.2	42.1
UD2OIE	**58.2**	**39.0**	**67.3**	**49.6**	**68.1**	**51.9**	**68.0**	**51.6**

TABLE 3.9 Résultats de l'évaluation multilingue des systèmes d'OpenIE sur les différents référentiels.

performances de rappel tout en entraînant une baisse des performances de la précision. Nous attribuons cela au fait que la recherche de tous les arguments pertinents avant de les étiqueter à l'étape suivante est moins complexe et aboutit à un nombre plus important de relations prédicat-argument.
Par conséquent, la performance de rappel augmente à mesure que le nombre de relations prédicat-argument augmente. Cependant, des relations prédicat-argument plus erronées seront propagées au module classificateur, qui ne cherche qu'à étiqueter les arguments extraits et ne peut pas rejeter les erronés, ce qui entraîne une baisse des performances de la précision.

Système	Anglais		Français		Espagnol		Portugais	
	F1	AUC	F1	AUC	F1	AUC	F1	AUC
UD2OIE	**58.2**	**39.0**	**67.3**	**49.6**	**68.1**	**51.9**	**68.0**	**51.6**
UD2OIE (-Arg Identification)	57.0	35.6	64.5	44.9	64.3	43.7	64.9	45.4
Système	Anglais		Français		Espagnol		Portugais	
	PRE	REC	PRE	REC	PRE	REC	PRE	REC
UD2OIE	61.4	**55.3**	72.7	**62.7**	**72.6**	**64.1**	72.8	**63.8**
UD2OIE (-Arg Identification)	**63.2**	52.0	**76.1**	56.0	71.0	58.7	**76.7**	56.2

TABLE 3.10 Résultats de l'étude d'impact de l'architecture.

3.9.4 Adaptation au domaine

Tout en surpassant de loin tous les systèmes à base de règles sur les référentiels d'évaluation en anglais, en espagnol et en portugais, *Multi2OIE* (Ro et al. 2020) ne surpasse que légèrement *PredPatt* (White et al. 2016) sur le référentiel d'éva-

luation en français.
Pour enquêter sur la source de cet écueil, nous construisons un deuxième référentiel d'évaluation français tiré du domaine Wikipédia.
Pour ce faire, nous traduisons l'ensemble de données d'entraînement Wikipédia en anglais décrites dans le tableau3.4 vers le français, et l'annotons manuellement en suivant les mêmes directives (Lechelle et al. 2019).

Système	Français	
	F1	AUC
PredPatt White et al. (2016)	37.6	30.4
Multi2OIE Ro et al. (2020)	**53.6**	**32.9**

TABLE 3.11 Résultats de l'évaluation des systèmes de référence sur le référentiel français de Wikipédia.

Les résultats du tableau 3.11 montrent que *Multi2OIE* surpasse largement PredPatt par rapport au référentiel d'évaluation de Wikipédia en français.
Nous supposons que Multi2OIE, qui est basé sur *BERT*, n'obtient de bonnes performances sur le référentiel d'évaluation Wikipédia que parce que *BERT* a été pré-entraîné sur les données Wikipédia.
De plus, à cause de *BERT*, *Multi2OIE* est instable et échoue face à des échantillons hors domaine tels que des textes financiers.

Comme indiqué par Li et al. (2020), malgré leur capacité à extraire des représentations indépendantes de la langue dans leur version multilingue, les modèles de langage tels que *BERT* ne capturent que les caractéristiques spécifiques au domaine et n'extraient pas des caractéristiques invariantes au domaine.
Par conséquent, les approches basées sur *BERT* telles que *Multi2OIE* ne sont pas adaptées à plusieurs domaines, ce qui viole le principe du paradigme OpenIE.

3.9.5 Analyse d'exécution

Le tableau 3.12 montre que notre modèle peut traiter environ 20 phrases par seconde, bien qu'il soit exécuté sur CPU et non sur GPU.
Cela montre également que notre modèle peut traiter 141 phrases par seconde si

nous excluons le temps d'exécution de l'analyse des dépendances.
Bien que les résultats rapportés ne soient pas justes en raison de l'écart de performances entre le CPU et le GPU, le tableau 3.12 montre que notre modèle atteint des résultats comparables avec l'approche basée sur les règles (il utilise un analyseur sémantique, et le temps d'exécution de cet analyseur est inclut) sur le CPU.

Système	Temps d'exécution Phrases/Secondes
ClauseIE	4.0
OpenIE4	20.1
OpenIE5	3.1
SpanOIE	19.4
NeuralOIE	11.5
RnnOIE	**149.0**
IMOJIE	2.6
OpenIE6	31.7
UD2OIE	20.1
UD2OIE (W/o Stanza)	141.2

TABLE 3.12 Temps d'exécution des systèmes OpenIE.

3.9.6 Analyse des erreurs.

Comme prévu, la principale source d'erreurs sont les erreurs de propagation de l'analyseur. Nous constatons que notre système échoue à traiter les constructions linguistiques complexes.
Le dernier exemple dans le tableau 3.13 montre un exemple de *Gapping*, un type d'ellipse, où notre système échoue à extraire les relations correspondantes.
La bibliothèque *Stanza* que nous avons utilisée considère le *Gapping* comme une simple clause de conjonction et alimente notre modèle avec un arbre syntaxique incorrect.
Une autre source importante d'erreur était le champ d'argument n-aire, où la relation n-aire était extraite en tant que relation binaire, avec l'argument n-aire manquant ou se trouvant dans le champ objet.
Le premier exemple dans le tableau 3.13 montre un exemple dû à l'ambiguïté de l'attachement de préposition, où *the Battle of Jamal* est extrait dans le cadre du

champ d'objet *Ali's army*.
De plus, notre système échoue plus souvent à extraire des relations basées sur des prédicats nominaux, comme indiqué dans le tableau 3.13.
Enfin, le dernier exemple dans le tableau 3.13 montre une construction spécifique au langage spécifique au français (*agentive indirect object* (exprimé par **iobj :agent** dans l'arbre syntaxique *UD*) où l'agent initial (le pronom *lui* dans l'exemple) a été rétrogradé et est devenu un objet indirect.
Puisque notre système a été entraîné sur des corpus d'entraînement en anglais, il échouera naturellement face à ces constructions spécifiques à la langue française.

3.10 Conclusion

Dans ce chapitre, nous avons proposé une approche pour l'OpenIE multilingue, tout en introduisant un nouveau référentiel d'évaluation pour le français. Nous avons montré que notre approche s'adapte à d'autres langues sans données d'apprentissage de la langue cible. Nous avons introduit un modèle simple mais efficace, qui surpasse les approches standard en deux étapes (extraire le prédicat et les arguments puis les types d'arguments). Les résultats de l'expérience suggèrent que les approches actuelles basées sur le modèle *BERT* ne sont pas compatibles avec plusieurs domaines et gèrent mal l'adaptation au domaine.

Type d'erreur	Exemple
Arguments n-aire	And he was in Ali's army in the Battle of Jamal. **Extraction :** (**A0 :**he **; P :**was in **; A1 :**Ali's army in the Battle of Jamal) **Référence :** (**A0 :**he **; P :**was in **; A1 :**Ali's army **;** **A2 :**in the Battle of Jamal)
Relations nominales	FBI Director Clive Anderson is the same kind of avuncular superior as Chief Brandon. (**A0 :**Clive Anderson ; **P :**[be] Director [of] ; **A1 :**FBI)
Constructions linguistiques complexes ou ambiguës	A cafeteria is also located on the sixth floor, a chapel on the 14th floor, and a study hall on the 15th floor. **Extraction :** (**A0 :**A cafeteria **; P :**is also located on **; A1 :**the sixth floor) **Extraction :** (**A0 :**A cafeteria **; P :**is also located on **;** **A1 :**a chapel on the 14th floor) **Extraction :** (**A0 :**A cafeteria **; P :**is also located on **;** **A1 :**a study hall on the 15th floor) **Référence :** (**A0 :**A cafeteria **; P :**is also located on **; A1 :**the sixth floor) **Référence :** (**A0 :**a chapel **; P :**is also located on **; A1 :**the 14th floor) **Référence :** (**A0 :**a study hall **; P :**is also located on **; A1 :**the 15th floor)
Constructions linguistiques propres au français	Google et Facebook en embuscade face à Apple, seul Google lui tient un peu tête. Google and Facebook in ambush against Apple, only Google is standing up to it a bit. **Extraction :** (**A0 :**Google **; P :**tient un peu tête **;**) **Référence :** (**A0 :**Google **; P :**tient un peu tête **; A1 :**lui)

TABLE 3.13 Analyse et types des erreurs les plus récurrentes.

Chapitre 4

Jeux2BERT

Dans ce chapitre, nous présentons la méthodologie suivie pour la conception du système *Jeux2BERT*, qui est un modèle de langue (*Flaubert*) augmenté par le réseau lexico-sémantique *JeuxDeMots*. Ainsi, ce modèle capture les propriétés distributionnelles et relationnelles des mots, mais aussi arrive à discriminer les différentes propriétés relationnelles entre mots ou syntagmes.
La conception de ce système est obligatoire dans la mesure où il intervient aussi dans notre modèle d'extraction de relations sémantiques. Spécifiquement, le modèle *Jeux2BERT* est censé normaliser les extractions d'*OpenIE* vers un des types de l'ensemble de relations sémantiques.

4.1 Définition du problème et organisation de ce chapitre

La question des représentations vectorielles des mots est un point crucial dans les modèles de langage. De façon classique, ces représentions sont construites par le calcul de cooccurrences entre termes à partir de corpus de très grande taille. Cependant, les liens sémantiques entre termes ne sont que partiellement contenus dans les textes, ces derniers contenant de grandes quantités d'informations implicites, qu'il est au mieux particulièrement difficile d'inférer depuis des cooccurrences. Plus précisément, la problématique de l'amélioration des représentations vectorielles de mots peut être abordée par l'exploitation de ressources externes telles que les dictionnaires, bases de connaissances et réseaux lexico-sémantiques. Cette approche qui semble bien naturelle car c'est effectivement ce que font les locuteurs humains, n'est généralement pas adoptée soit à cause du manque de bases de connaissances satisfaisantes soit de par la difficulté d'inclure correctement ces connaissances dans un modèle de langage.

La section 4.2 présente la définition des vecteurs de mots et les différentes approches pour générer les représentations vectorielles des mots. L'évolution des vecteurs des mots est présentée en détails jusqu'à l'état de l'art actuel des vecteurs de mots.
Les bases de connaissances et les réseaux lexico-sémantiques sont abordés dans la section 4.4 afin de faciliter la compréhension de la problématique et mieux situer les points d'amélioration possibles. Ensuite, le réseau lexico-sémantique pour le français *JeuxDeMots* est présenté.
La section 4.5 enchaîne sur la description des inconvénients des représentations vectorielles des mots et présente en détail les approches exploitant des ressources externes telles que les bases de connaissance pour pallier les limites des représentations vectorielles des mots. Nous exposons aussi les inconvénients des approches proposés actuellement afin de préparer le terrain pour la méthode que nous proposons.
Enfin, nous livrons les détails de notre approche dans la section 4.6 ainsi que les résultats des expérimentations.

4.2 Vecteur de mots

La représentation des données d'entrée textuelles dans un espace vectoriel est un élément essentiel de tout système de traitement automatique de texte. Il permet de représenter les données de manière mathématique et rend possible l'exploitation des données en aval, comme pour des applications prédictives.

4.2.1 Introduction

La représentation par défaut des mots dans un modèle d'apprentissage machine est l'encodage *One-Hot Encoding*, qui consiste à représenter chaque mot par un vecteur, d'une taille correspondant au vocabulaire entier considéré pour chaque variable textuelle manipulée par le modèle. Le terme *One-Hot* indique qu'une seule valeur du vecteur contient la valeur 1 tandis que les autres contiennent des 0. Cet encodage permet d'assurer que les vecteurs soient uniques et consistants pour l'implémentation du modèle.
Cette représentation ne capture pas d'information sémantique du mot mis à part

sa distinction par rapport à une autre. Cet encodage n'a pour but que d'associer un identifiant unique à un mot donné et de lier chaque mot à un vecteur suivant les propriétés de cet encodage.

Avec cette représentation, il est impossible d'exprimer une similarité plus intense entre *« professeur »* et *« encadrant »*, qu'entre *« professeur »* et *« manager »*. De plus, la dimension correspond à la taille du vocabulaire de la variable textuelle, qui peut prendre plus de $30\ 000$ valeurs. La manipulation de tels vecteurs est extrêmement coûteuse en temps de calcul et surtout en mémoire vive.

Grâce à d'autres méthodes de vectorisation, les valeurs des variables textuelles sont associées à des vecteurs de faible dimension (par rapport à la taille de leur vocabulaire d'entrée), et les fonctions de similarité, et en particulier la similarité cosinus, permet d'identifier les termes "sémantiquement" proches.
Nous remarquons que bien que souvent employé la "similarité sémantique" est davantage une similarité de ressemblance de contextes qu'une similarité en termes de signification. L'hypothèse fondatrice est que la sémantique émerge des contextes d'apparition des termes.
Parmi les méthodes de vectorisation de mots, deux approches sont principalement utilisées :

— Le modèle sac de mots (*bag-of-words*).
— Le modèle word2Vec.

4.2.2 Sac de mots

Le modèle considére le texte comme un simple ensemble de mots sans prendre en compte leur ordre d'apparition dans la phrase ni leurs relations syntaxiques et sémantiques. Si l'on a n documents composés de m mots différents, nous assignons à chaque mot un numéro compris entre 1 et m.
Chaque document peut ensuite être converti en un vecteur $(x_1, x_2, ..., x_m)$ ou x_i est égal à 0 si le terme i est absent du document.
Si le terme est présent, nous pouvons donner pour valeur de x_i le nombre d'occurrences du mot dans le texte (*term frequency*) ou des mesures plus fines comme le *TF-IDF*.

Ce modèle est de loin le plus utilisé en traitement automatique de la langue, tant pour la recherche de documents que pour les systèmes de recommandation.
Son incapacité à tenir compte de la syntaxe de la phrase présente cependant des limites. En effet les phrases « *John aime le sucre mais déteste le sel* » et « *John déteste le sucre mais aime le sel* », bien que de sens complètement opposés se retrouveront vectorisées exactement de la même manière.
Il sera donc impossible par la suite pour un algorithme d'apprentissage machine de les distinguer (tant pour l'apprentissage que pour les prédictions).
Ce modèle gère en outre mal la synonymie. Si nous pouvons regrouper (avec le stemming ou la lemmatisation) un mot au singulier avec le même mot au pluriel dans un même index, nous ne pourrons pas saisir dans la vectorisation le fait que « *John takes the subway* » et « *John rides the metro* » sont des phrases sémantiquement presque identiques (« *take* » aura un index différent de « *ride* » et « *subway* » un index différent de « *metro* »).

4.2.3 Word2Vec

Le *Word2Vec* est un algorithme de représentation vectorielle de la donnée basée sur l'intuition suivante : « les mots sémantiquement proches apparaissent souvent dans le même contexte » (*you shall know a word by the company it keeps*).
Il s'agit d'un modèle d'apprentissage profond qui capture les sens sémantiques d'un mots en considérant pour chaque mot son voisinage dans une fenêtre de taille donnée(soit un groupe de mots avec lesquels il apparaît).
Ces modèles ont été développés par une équipe de recherche chez *Google* sous la direction de Tomas Mikolov. Cette approche s'appuie sur des réseaux de neurones artificiels à deux couches entraînées pour reconstruire le contexte linguistique des mots.
Ces modèles sont entraînés sur des corpus très volumineux. Le principe de base est de tenter de prédire un mot à partir de son contexte ou vice-versa. Un sous-produit de cet apprentissage constitue les plongements lexicaux (Mikolov et al. 2013).
Deux approches sont possibles pour entraîner les représentations vectorielles des mots :

— C-BOW : le modèle est entraîné à prédire le mot particulier à partir de son

voisinage (voir schéma 9).

— Skip-gram : le modèle est entraîné à prédire les mots composant le voisinage à partir du mot particulier (voir schéma 10).

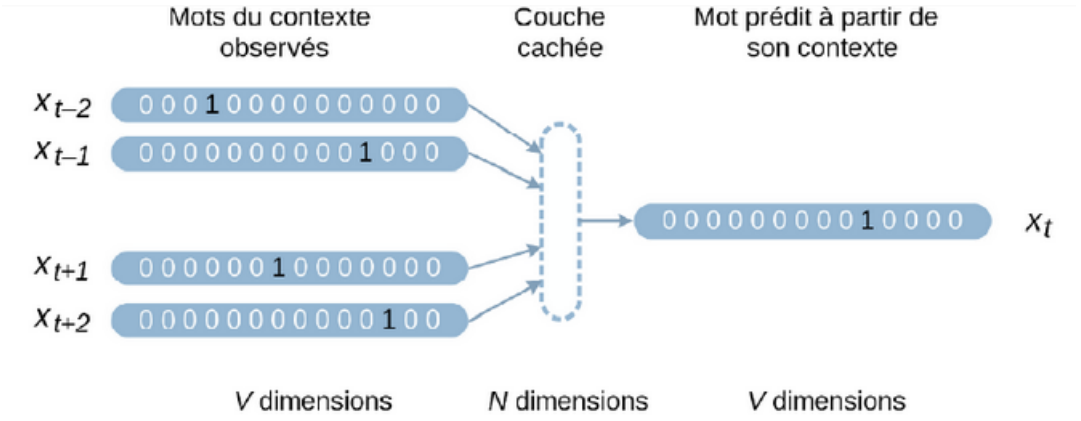


FIGURE 9 Modèle CBOW : Illustration adaptée de l'article de Mikolov (Mikolov et al. 2013).

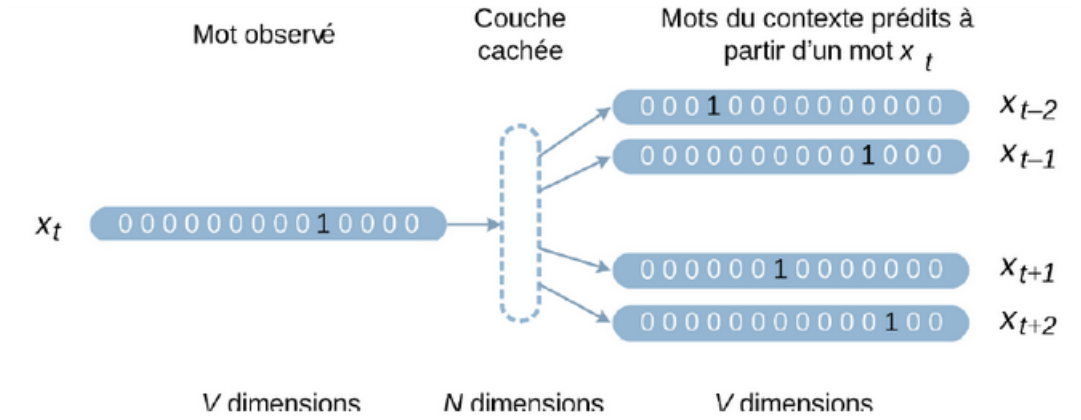


FIGURE 10 Modèle Skip-Gram. Illustration adaptée de l'article de Mikolov (Mikolov et al. 2013).

La figure 11 illustre la différence entre l'architecture *Skip-Gram* et *CBOW* du modèle *Word2Vec*. Dans les deux cas, nous définissons un réseau de neurones artificiels qui a une couche d'entrée X, et une couche de sortie Y. Ces vecteurs sont binaires et de taille V. Une couche cachée à N dimensions est intercalée entre les deux, N est un hyperparamètre d'apprentissage.
L'entraînement du réseau produit une matrice W de taille $V \times N$, connectée à gauche aux entrées et à droite à la couche cachée. Au terme de l'apprentissage, la matrice W contient les plongements de tous les mots du vocabulaire (Mikolov

et al. 2013).

Les vecteurs X et Y sont construits avec un encodage « *One-Hot* ». Ils possèdent un unique 1 dans leur i^{eme} dimension et des zéros partout ailleurs.
La dimension i non nulle correspond au mot i du vocabulaire.
Pour donner suite à la multiplication matricielle, la i^{eme} ligne de W encode donc le plongement du i^{eme} mot.
La propagation consiste à mettre à jour les différentes couches du réseau.
La couche cachée est une application linéaire des entrées X et de la couche W.
La couche de sortie Y est calculée grâce à la fonction softmax.

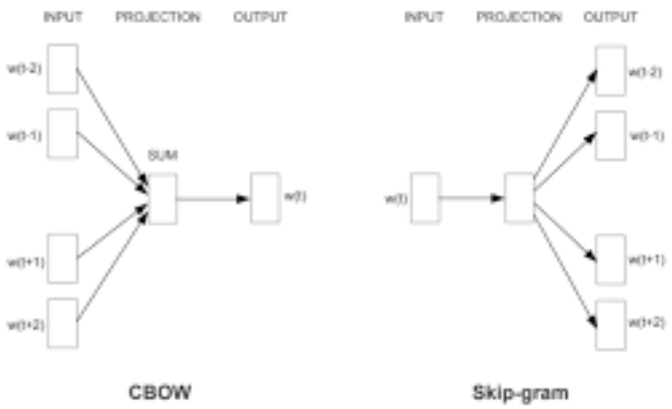


FIGURE 11 Architecture du Word2Vec.

Au terme de l'entraînement, les paramètres entraînés par le modèle (poids et biais) sont utilisés pour constituer les vecteurs. Le modèle *Word2Vec* est entraîné jusqu'à l'état d'apprentissage par cœur (*overfitting*).
L'évaluation quantitative du modèle consiste à se rapprocher le plus possible des probabilités aléatoires de prédiction du voisinage d'un mot à partir d'un autre.
L'évaluation qualitative est réalisée en général, au cas par cas avec un ensemble de mots dont nous observons les mots les plus proches et les similarités associées.

De plus, il a été validé de manière expérimentale que le modèle *Word2Vec* supporte l'opération de concaténation : la vectorisation d'un groupe nominal est la somme des vecteurs de chacun des termes le composant (Mikolov et al. 2013).
Les modèles *Word2Vec* plus récents associent à chaque mot une représentation vectorielle (*word embedding*) de dimensionnalité réduite.

Ces représentations permettent de capturer les relations sémantiques entre les mots, ainsi le vecteur du mot « *subway* » sera proche de vecteur du mot « *metro* » et loin de vecteur du mot « *banana* ».
On obtient ainsi des représentations très fines au niveau des mots, permettant des calculs intéressants, mais les choses se compliquent lorsque nous voulons obtenir une représentation de la phrase.

Si nous souhaitons éviter l'écueil du modèle *bag-of-words* qui ne prend pas en compte l'ordre des mots, nous pouvons concaténer les vecteurs mais nous sommes vite limités par la taille du texte.
Au-delà de la phrase courte, la dimension sera trop élevée. Les seules solutions véritablement pratiques pour passer au niveau de la phrase sont l'addition et la moyenne des vecteurs, mais l'on reste alors toujours incapable de saisir les relations syntaxiques entre les mots.

4.3 Vecteur de mots avec représentation contextualisée

Les approches susmentionnées (Turian et al. 2010, Mikolov et al. 2013, Pennington et al. 2014) ignorent le contexte dans leur représentation des mots.
Wieting et al. (2016), Bojanowski et al. (2017) ont abordé une certaine limite des représentations des mots en considérant leur morphologie (lexème, flexion, dérivation), alors que Neelakantan et al. (2014) ont considéré la représentation du mot pour chacun des sens qu'il porte.

Peters et al. (2018) prennent également en compte la morphologie des mots à travers la représentation au niveau des caractères et transporte implicitement les informations multi-sens d'un mot.
Pour pallier la problématique du contexte, Peters et al. (2018) introduisent la représentation de mot contextualisée en profondeur, où chaque mot se voit attribuer une représentation au niveau de la phrase via une fonction avec la phrase entière en entrée. La représentation des mots est dérivée d'un *LSTM* bidirectionnel d'un modèle de langage, pré-entraîné sur un large corpus de texte. Les mots sont obtenus en apprenant une combinaison linéaire de toutes les couches *LSTM* empilées les unes au-dessus des autres, une telle architecture surpasse une architecture si-

milaire mais en ne considérant que la couche *LSTM* supérieure.
L'évaluation montre que les couches inférieures du modèle de langage bi-directionnel réussissent bien à capturer les aspects des mots de bas niveau tels que les catégories grammaticales, tandis que les couches supérieures capturent les propriétés des mots de haut niveau telles que le contexte et la sémantique.

Des travaux récents sur la vision par ordinateur ont fait une percée majeure dans l'apprentissage par transfert inductif, où des tâches telles que la détection d'objets, la classification et la segmentation utilisent les connaissances transférées à partir d'un modèle déjà pré-entraîné comme *ImageNet*.
Les modèles d'apprentissage en profondeur atteignent des performances de pointe sur de nombreuses tâches de TALN, mais au prix d'une énorme quantité de données nécessaires pour former des modèles à partir de zéro.
Le transfert inductif n'a été considéré que pour affiner les plongements de mots pré-entraînés (Mikolov et al. 2013), où seule la première couche est considérée par le transfert.
Les approches récentes (McCann et al. 2017, Peters et al. 2018) fusionnent différents plongements avec l'entrée à différentes couches, mais traitent ces plongements comme des paramètres fixes (qui limitent leur utilité) et entraînent le modèle de la tâche principale à partir de zéro.
La représentation des plongements, dérivée de corpus non étiquetés, a montré un succès significatif dans l'apprentissage par transfert, mais seules les informations au niveau des mots sont prises en compte et aucune information sémantique n'est capturée. Pour donner suite à la représentation des plongements, des travaux récents ont considéré les plongements de phrases pour coder une représentation vectorielle appropriée.

Le paradigme de pré-apprentissage non supervisé permet de trouver un bon paramètre d'initialisation pour l'apprentissage supervisé. Des recherches ultérieures ont montré que l'étape de pré-entraînement permet de régulariser le modèle et d'améliorer la généralisation des réseaux de neurones profonds.
Le réglage fin (fine-tuning) de la modélisation linguistique a été considéré comme inefficace en raison du manque de connaissances sur la façon dont le modèle affiné est entraîné.

Mou et al. (2016) a dérivé un modèle linguistique affiné mais basé sur des données spécifiques à un domaine et une énorme quantité de données.
Le processus de réglage fin consiste à trouver un équilibre entre la généralisation excessive
(sur-apprentissage) sur un petit ensemble de données et l'oubli catastrophique des connaissances déjà apprises lorsqu'il est réglé de manière agressive.
De nouvelles stratégies devaient être envisagées pour affiner l'apprentissage par transfert du modèle de traitement automatique du langage (TALN), car le modèle de TALN d'apprentissage en profondeur est plus superficiel et assez différent des modèles de vision par ordinateur.

Howard and Ruder (2018) gèrent ces problèmes et dérivent un apprentissage par transfert inductif robuste à partir de n'importe quelle tâche de TALN.
Analogue au réglage fin *Image-Net* (modèle de classification d'images), la même architecture *LSTM* à 3 couches optimisée avec des modèles d'ingénierie lourds de battement de décrochage.
La nouveauté de leur approche réside dans une nouvelle méthode de transfert inductif de réglage fin a la Computer Vision, avec : réglage fin discriminant ; taux d'apprentissage triangulaire incliné ; gel progressif pour éviter les oublis catastrophiques lors des réglages fins.

Les principales approches permettant d'exploiter les connaissances capturées par les modèles pré-entraînés dans une tâche en aval sont : les approches basées sur les caractéristiques (*feature based*) et les approches basées sur le réglage fin (*fine-tuning based*) .
L'approche basée sur les caractéristiques transfère la représentation pré-entraînée en tant que vecteur à l'architecture spécifique à la tâche en tant que caractéristiques supplémentaires, tandis que l'approche de réglage fin telle que *OpenAI GPT* (Radford et al. 2018) induit un ajustement minimal et ajoute simplement une couche de tâche spécifique au modèle pré-entraîné et ensuite règle conjointement les paramètres de la couche de tâche spécifique et les paramètres du modèle de langage pendant la phase de réglage fin.
Les deux approches considèrent la même fonction objective et le même modèle de langage unidirectionnel.

Ces approches ne tirent pas le meilleur parti de la représentation pré-entraînée (en particulier celle de réglage fin), étant donné qu'elles utilisent un modèle de langage unidirectionnel, où chaque phrase est scannée dans une seule direction, donc le jeton (mot) actuel ne peut s'occuper que du jeton précédent dans la couche d'auto-attention (self attention) du transformateur.
Cette limite entrave les performances de la tâche au niveau de la phrase et détériore considérablement les tâches au niveau du jeton pendant le réglage fin, où le contexte des deux côtés du mot pivot (courant) est important.
Devlin et al. (2019) présentent la représentation d'encodeur bidirectionnel à partir de transformateurs, pour aborder les limites susmentionnées en considérant un nouvel objectif de pré-apprentissage : le modèle de langage masqué.
Le modèle de langage masqué masque au hasard une entrée de jeton et vise à prédire le jeton masqué en ne considérant que le contexte de l'entrée.
Contrairement au modèle de langage unidirectionnel, le modèle de langage masqué permet de bénéficier à la fois du contexte droit et gauche, permettant de dériver des transformateurs bidirectionnels profonds.
De plus, une tâche de « prédiction de la phrase suivante » pré-entraîne conjointement les représentations des paires de textes.

4.4 Base de connaissances et Réseaux lexico-sémantiques

Les bases de connaissances (*Knowledge Base*) fournissent une riche source de connaissances, qui peuvent être utiles pour plusieurs cas d'utilisation. De plus, elels incluent souvent des informations complémentaires à celles trouvées dans les textes bruts et peuvent coder des connaissances factuelles difficiles à apprendre, comme des préférences de sélection, soit en raison de mentions peu fréquentes des connaissances de sens commun ou de dépendances à long portée.
Excepté l'approche manuelle pour peupler les bases de connaissances, les approches contributives ont l'avantage d'être moins coûteuses en termes de temps et d'argent. Elles permettent aussi d'avoir une base de connaissance à jour puisque de nouveaux faits et connaissances émergent, et que le champ lexical d'une langue n'est pas figé mais bien évolutif.

Les GWAP (*Games With A Purpose*), ou *"jeux avec finalité d'acquisition de ressources"* est une des approches contributives pour peupler une base de connaissance. Le réseau léxico-sémantique *JeuxDeMots* a été construit en suivant l'ap-

proche *GWAP*. Il a aussi été augmenté de *contre-jeux* pour corriger et affiner les connaissances acquises. À ce jour, *JeuxDeMots* englobe plus de 377 millions de relations entre plus de 5,3 millions de termes (mots, syntagmes, noms et phrases).

4.5 État de L'art de l'affinement de vecteurs de mots à l'aide de bases de connaissances

Malgré l'avancée des approches neuronales dans la représentation vectorielle des mots, leurs limites sont assez flagrantes. En effet elles ne peuvent pas saisir les connaissances de ce monde et ni le champ lexical d'une langue d'une manière exhaustive.
Ces approches font toujours l'amalgame entre la similarité sémantique et d'autre types de relations sémantiques. En effet, ces approches sont toujours plus ou moins naïves en supposant que tous les mots apparaissant dans des contextes presque identiques sont sémantiquement similaires et de ce fait leur assigne des vecteurs de distribution similaires.
Aussi, elles ne tiennent pas compte des informations précieuses contenues dans les ressources de lexique sémantique telles que *WordNet* Miller (1995), *FrameNet* et la base de données *Paraphrase*. Pour ces raisons, plusieurs travaux ont essayé d'améliorer les représentations vectorielles des mots en utilisant des ressources externes tel que les bases de connaissances ou les dictionnaires.

Fried and Duh (2014) proposent une méthode d'entraînement des représentations vectorielles des mot, *Alternating Direction Method of Multipliers* (ADMM), en considérant les propriétés usuelles obtenues à travers les approches traditionnelles, mais aussi les propriétés relationnelles issues de ressources externes. Cette approche a l'avantage d'introduire une certaine flexibilité lors de l'optimisation des deux objectifs : la représentation distribuée à partir de textes bruts et la représentation relationnelle à partir de la base de connaissances (*WordNet*) .

De manière similaire à (Fried and Duh 2014), Yu and Dredze (2014) proposent aussi une approche pour optimiser conjointement les représentations relatives aux propriétés distributionnelles et propriétés relationnelles. Cependant ils n'exploitent que la relation de synonymie à partir des ressources sémantiques : *WordNet* (Miller 1995) et *Paraphrase Database* (Ganitkevitch et al. 2013).

Contrairement aux approches de bout-en-bout susmentionnées, Faruqui et al. (2015) proposent une approche en post-traitement.
L'approche proposée : "*retrofitting*", est une approche d'apprentissage basée sur les graphes qui consiste à s'assurer que les vecteurs de représentation des mots soient conjointement similaires aux vecteurs des mots similaires et au vecteur du mot avant l'affinement. Ils mesurent la similarité à travers la distance euclidienne. Contrairement aux approches précédentes, cette approche a l'avantage d'être assez rapide et prends environ 5 secondes pour un graphe de $100\ 000$ vecteurs d'une dimension de 300.

Yao, Mao and Luo (2019) suivent le même principe d'affinement de vecteur de représentation des mots. Ils réentraînent des modèles de langage déjà pré-entraînés (*BERT*) pour faire face à la non-complétude des graphes de connaissances.
Leur Framework considère en entrée le triplet de la relation comme une séquence de texte et consiste à prédire la plausibilité du triplet à travers une tâche de classification binaire, un autre moyen d'approcher la complétion de graphe de connaissances. Pour améliorer les performances du modèle, ils ajoutent aussi la tâche de prédiction de relation qui consiste à prédire la relation qui pourrait lier deux entités. Ils évaluent les différents systèmes en utilisant les métriques de classement de pertinence (*relevance rank*) et taux de réussite (*hit ratio*).
Les figures 12 et 13 illustrent l'architecture du modèle proposé par Yao, Mao and Luo (2019).

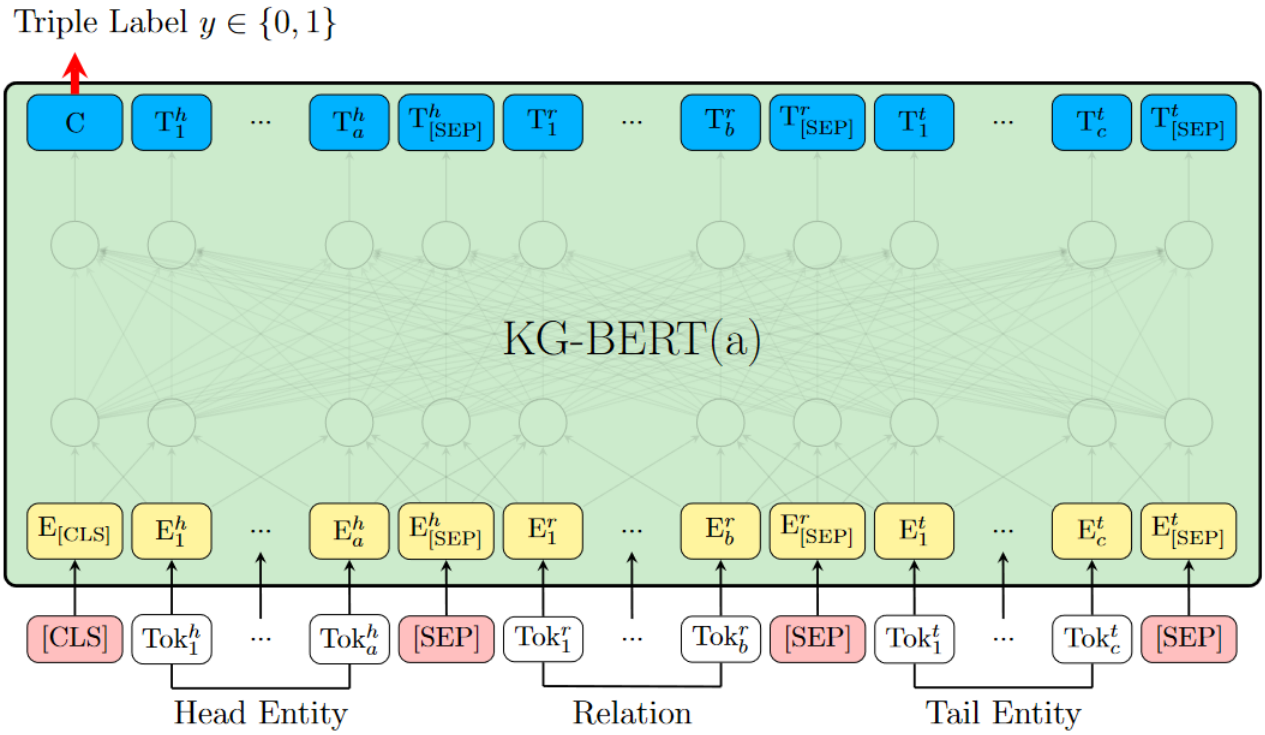


FIGURE 12 Illustration du réglage fin de KG-BERT pour prédire la plausibilité d'un triplet. Illustration par Yao, Mao and Luo (2019).

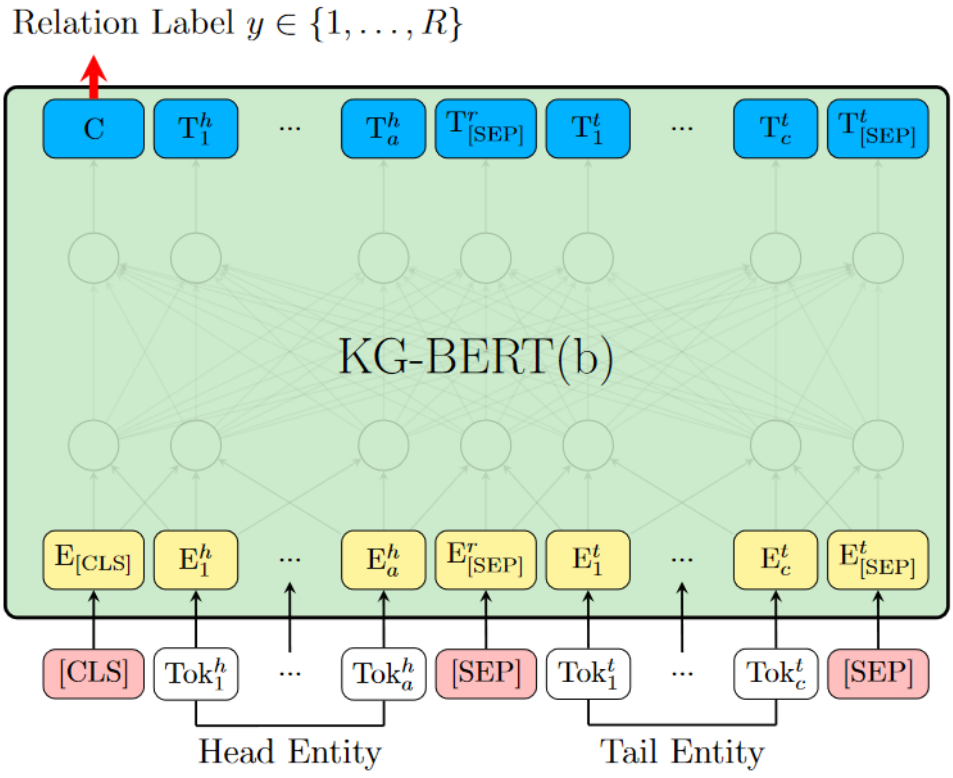


FIGURE 13 Illustration du réglage fin de KG-BERT pour prédire la relation entre deux entités. Illustration par Yao, Mao and Luo (2019).

Requête	Eléments proposés	Elément correct	Classement de pertinence	Taux de réussite @5
<virus du nil occidental ag> r_isa <antigène>	**<virus du nil occidental ag>** <rg11 (rg1) antigène> <phadiatop pneumallergènes ag> <antigène scianna> <poulet (déjections) ac>	<virus du nil occidental ag>	1	1
<contrasté> r_anto **<appareillé>**	<saisissant> <allongé> <écaillé> <déchanté> <sermonné> <élogieux> <révélateur> **<appareillé>** <tabou>	<appareillé>	8	0
<virus respiratoire syncytial a arn> r_isa <analyte>	<fièvre jaune virus ac 2nd prélèvement> <chromosome humains de la paire 10> <antagonistes des récepteurs alpha-2 adrénergiques> **<virus respiratoire syncytial a arn>** <récepteur orphelin bêta apparenté aux récepteurs des rétinoïdes>	<virus respiratoire syncytial a arn>	4	1

TABLE 4.1 Exemples de prédictions d'un modèle entraîné à partir de triplets issus de JeuxDeMots.

Le tableau 4.1 illustre un exemple des deux métriques susmentionnées. La première métrique consiste à trouver le rang de l'élément le plus pertinent dans une liste de propositions ou de recommandations pour une requête donnée.
La deuxième métrique, Taux de réussite@k signifie que seulement les premières k réponses ou propositions sont considérés pour l'évaluation. Ainsi, Taux de réussite@5 signifie que seulement les cinq premières propositions sont considérées pour trouver l'élément le plus pertinent.

Kim et al. (2020) indiquent que les approches susmentionnées basées sur le modèle *BERT* ont des performances limitées par apport à l'état de l'art dans la métrique de Taux de réussite@k. Ils expliquent cela par l'apprentissage insuffisant des informations relationnelles dans les graphes de connaissances et la difficulté à choisir la bonne réponse parmi des candidats lexicaux similaires.
Ils proposent une méthode d'apprentissage multi-tâches efficace pour surmonter les limites des travaux antérieurs. En combinant les tâches de prédiction de relation et de classement de pertinence avec la prédiction de lien cible, le modèle proposé peut apprendre plus de propriétés relationnelles dans les bases de connaissances et fonctionner correctement même lorsqu'une similitude lexicale des candidats se produit.
La figure 14 illustre l'architecture du modèle proposé par Kim et al. (2020).

4.6 Méthode et Modèle proposé

Dans nos travaux de recherche, nous suivons l'approche de Kim et al. (2020), nous traitons d'abord les entités, les relations et les triplets comme des séquences textuelles et considérons la complétion du graphe de connaissances comme un problème de classification de séquences.
Nous affinons ensuite le modèle de langage *BERT* sur ces séquences pour prédire la plausibilité d'un triple ou d'une relation. Pour cela, nous considérons les trois tâches résumées dans (Kim et al. 2020) et qui sont : prédiction de la plausibilité d'un triplet, prédiction de la relation la plus plausible pour deux entités ainsi que le classement de pertinence.

Nos contributions dans la tâche de complétion de graphe de connaissances en utilisant les modèles de langage pré-entraînés se résument comme suit :

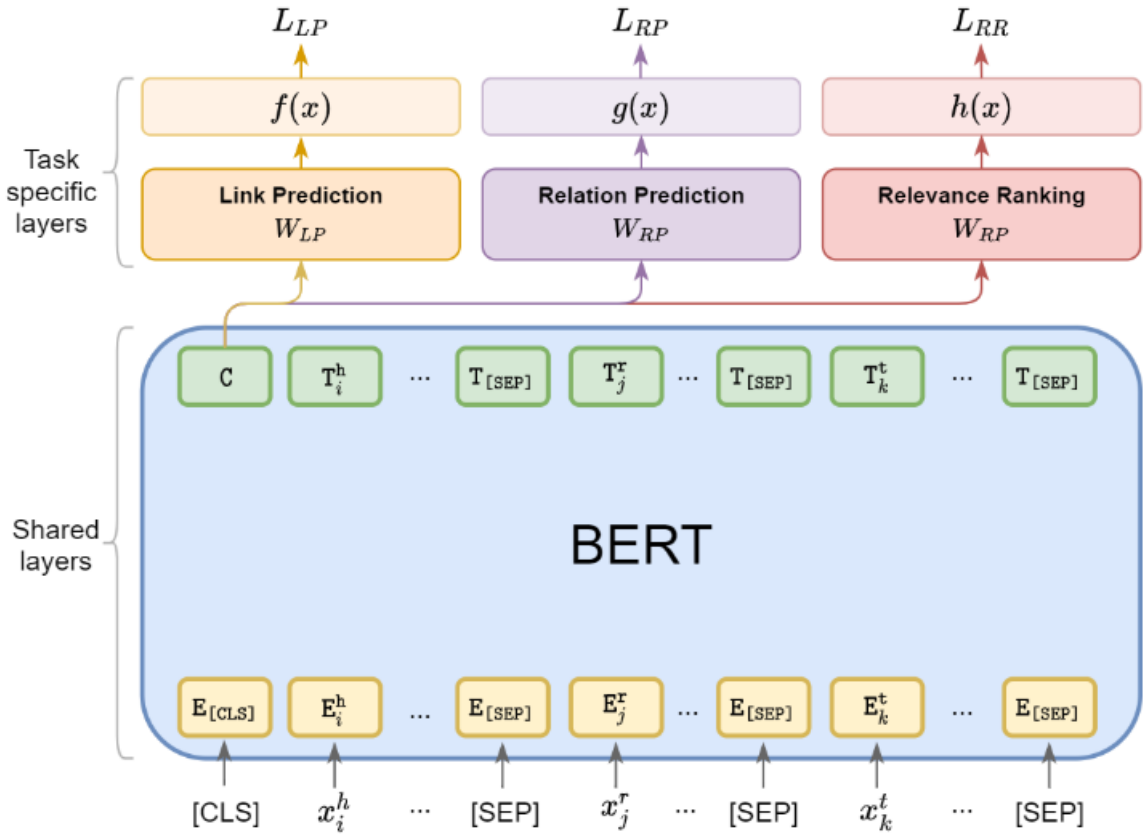


FIGURE 14 Architecture de la méthode d'apprentissage multi-tâche proposée par Kim et al. (2020)

— Nous proposons d'améliorer la tâche de classement de pertinence pour atténuer les limites engendrées par la façon dont la tâche de classement de pertinence est exploitée par Kim et al. (2020).
— En plus des trois tâches susmentionnées, nous introduisons aussi la tâche de rétrofitting pour affiner l'espace vectoriel des représentations en utilisant des informations relationnelles à partir de lexiques sémantiques.
— Nous exploitons aussi le réseau sémantique-lexical *JeuxDeMots* pour dériver un corpus en français pour la tâche de complétion de graphe.

4.6.1 Corpus d'apprentissage

Comme indiqué précédemment, nous exploitons le réseau sémantique-lexical *JeuxDeMots* pour affiner les représentations contextuelles du modèle de langage *BERT*. Nous commençons par filtrer les relations sémantiques-lexicales que nous

exploitons, parmi les 159 relations nous ne retenons que 132 relations.
Le tableau B.1 liste ces relations ainsi que leurs descriptions.

Lors de l'échantillonnage du graphe de connaissances en données d'apprentissage et données d'évaluation, nous prenons deux aspects en compte :

— Les triplets d'évaluation doivent être inférables à partir des triplets d'apprentissage.
— Éviter la fuite des triplets d'apprentissage dans l'ensemble de triplets d'évaluation.

Ces deux aspects d'échantillonnage peuvent sembler contradictoires, mais sont nécessaires pour la construction d'un corpus propre. La première proposition est due à l'échantillonnage aléatoire où des triplets d'évaluation n'ont pas un ensemble de triplets qu'ils pourraient exploiter lors de l'inférence dans le corpus d'apprentissage. Pour cela, certains triplets sont dénués de sens ou imprédictibles même pour les humains. La tableau 4.2 illustre un exemple où il est demandé de prédire la plausibilité du triplet <*"chat"; r_associated; "tigre"*> en exploitant les triplets d'apprentissage.
On peut voir que le corpus ne contient aucun triplet dans lequel apparaît le terme *"tigre"*.

	Tête	**Prédicat**	**Queue**
Évaluation	chat	r_associated	? tigre
Apprentissage	chat	r_isa	félin
Apprentissage	chat	r_associated	félin
Apprentissage	félin	r_associated	jaguar

TABLE 4.2 Exemple d'absence de triplet de support pour le triplet d'évaluation.

D'un autre côté, le tableau 4.3 illustre le même exemple où il est demandé de prédire la plausibilité du triplet <*"chat"; r_associated; "tigre"*>, mais avec un ensemble de triplets d'apprentissage enrichi.
Ce nouveau corpus enrichi contient des triplets d'apprentissage dans lesquels apparaissent les deux entités du triplet <*"chat"; r_associated; "tigre"*>.

	Tête	**Prédicat**	**Queue**
Évaluation	chat	r_associated	? tigre
Apprentissage	chat	r_isa	félin
Apprentissage	chat	r_associated	félin
Apprentissage	félin	r_associated	jaguar
Apprentissage	tigre	r_associated	lion
Apprentissage	tigre	r_isa	grand fauve

TABLE 4.3 Exemple de présence de triplet de support pour le triplet d'évaluation.

Autre point pris en compte lors de l'échantillonnage des données, vérifier s'il y avait une fuite des triplets d'apprentissage dans l'ensemble des triplets d'évaluation. La fuite des triplets d'apprentissage est fréquente lors d'un échantillonnage aléatoire. Plus formellement, pour un triplet d'évaluation (h, r, t) nous pouvons avoir un ensemble d'apprentissage avec des triplets de la forme $(h, r^{'}, t)$ ou bien $(t, r^{'}, h)$.
Ce genre de cas concerne principalement des relations qui ont des propriétés d'inversion comme les relations *r_carac* ou *r_associated*, ou bien des relations qui indiquent un lien de hiérarchie tel que les relations *r_isa* et *r_associated*.

Le premier type de fuite de données est de la forme : pour chaque triplet d'évaluation $(h, hyponym, t)$, il existe un triplet d'apprentissage $(t, hypernym, h)$ ou vice-versa. Le tableau 4.4 illustre un exemple où il est demandé de prédire la plausibilité du triplet d'évaluation $(abeille, r_isa, insecte)$ alors que l'ensemble des triplets d'apprentissage contient un triplet dans lequel les deux entités apparaissent avec la relation inverse de *r_isa* (qui énumère des **génériques/hyperonymes** du terme) qui est *r_hypo* (qui énumère des **spécifiques/hyponymes** du terme).
Ce type de fuite de données est assez fréquent.

	Tête	**Prédicat**	**Queue**
Évaluation	abeille	r_isa	? insecte
Apprentissage	insecte	r_hypo	abeille

TABLE 4.4 Premier exemple de fuite de donnée.

Le deuxième type de fuite de données et le plus fréquent aussi est dû à la spécificité de *JeuxDeMots* où beaucoup de relations ont des relations inverses telles que *r_carac* et *r_carac-1*, *r_agent* et *r_agent-1*, ...etc.

Ainsi pour chaque triplet d'évaluation (h, r, t) il existe un triple d'apprentissage (t, r^{-1}, h) ou vice-versa, où la relation r^{-1} est la relation inverse de r. À noter que le premier exemple est un cas spécial de ce type de fuite de données.

Le tableau 4.5 montre un exemple où il est demandé d'inférer si la *photographie* a la caractéristique *noir et blanc*, alors que l'ensemble de triplets d'évaluation contient des triplets listant des objets qui ont la propriété *noir et blanc* et parmi ces objets se trouve l'objet *photographie*.

	Tête	**Prédicat**	**Queue**
Évaluation	photographie	r_carac	? noir et blanc
Apprentissage	noir et blanc	r_carac-1	film
Apprentissage	noir et blanc	r_carac-1	photographie

TABLE 4.5 Deuxième type de fuite de donnée.

Un autre type de fuite de données moins fréquent concerne les relations réflexives telles que : r_syn, r_anto et r_cooccurrence .
Pour chaque triplet d'évaluation (h, r, t) il existe un triple d'apprentissage (t, r, h). Le tableau 4.6 montre deux exemples d'inférence de la plausibilité d'un triplet d'évaluation (h, r, t) alors que l'ensemble de triplets d'apprentissage contient un triplet ayant la même relation avec les mêmes entités dans le sens inverse.
Le premier exemple montre que pour le triplet d'évaluation *(inondé, r_syn, gorgé)*, il existe un triplet *(gorgé, r_syn, inondé)* dans l'ensemble d'apprentissage de triplet. Le deuxième exemple illustre le même cas où l'ensemble de triplet d'apprentissage contient le triplet *(contrasté, r_anto, unifié)*, tandis que le triplet d'évaluation est le triplet *(unifié, r_anto, contrasté)*.

	Tête	**Prédicat**	**Queue**
Évaluation	inondé	r_syn	? gorgé
Évaluation	unifié	r_anto	? contrasté
Apprentissage	gorgé	r_syn	inondé
Apprentissage	contrasté	r_anto	unifié

TABLE 4.6 Troisième type de fuite de donnée.

Le dernier type de fuite de données et le moins fréquent concerne seulement les deux relations du graphe de connaissances : *r_associated* et *r_isa*. Ce cas se produit lorsque nous avons un triplet d'apprentissage de la forme (h, r_isa, t) et un triplet d'évaluation $(h, r_associated, t)$.
Le tableau 4.7 montre un exemple de ce cas, où il est demandé de prédire la plausibilité du triplet d'évaluation *(chat, r_associated, félin)* avec la présence du triplet *(chat, r_isa, félin)* dans l'ensemble des triplets d'apprentissage. À noter que ce type de fuite ne se produit pas dans le sens contraire.

	Tête	**Prédicat**	**Queue**
Évaluation	chat	r_associated	? félin
Apprentissage	chat	r_isa	félin

TABLE 4.7 Quatrième type de fuite de donnée.

Nous pouvons nous apercevoir que de simples règles rédigées avec un moindre effort suffisent à prédire la plausibilité des triplets et à les compléter dans la base de connaissances. Ainsi pour prévenir toute forme de fuite de données, nous avons retiré de l'ensemble d'évaluation toute paire d'entité (h, t) qui apparaît dans l'ensemble d'entraînement et qui a un lien direct à travers n'importe quelle relation du graphe de connaissances.
Néanmoins, nous avons appliqué un filtrage flexible des triplets où nous avons seulement supprimé les triplets avec un lien direct apparaissant dans l'ensemble d'entraînement.
Ainsi nous n'avons pas retiré les paires d'entité (h, t) pour lesquelles il existe un chemin indirect dans le graphe de connaissances entre ces deux entités.

	Tête	**Prédicat**	**Queue**
Évaluation	veuve noire	r_associated	? décédé
Apprentissage	veuve noire	r_agent	tuer
Apprentissage	tuer	r_consequences	mort
Apprentissage	mort	r_syn	décédé

TABLE 4.8 Illustration de la flexibilité de la stratégie de filtrage.

Le tableau 4.8 illustre un exemple du filtrage flexible lors de l'échantillonnage des données. Nous pouvons voir que pour l'inférence de la plausibilité du triplet

(veuve noire, r_associated, décédé), les triplets d'entraînement ne constituent pas un cas de fuite de données.
La figure 15 montre qu'il existe un chemin entre la paire d'entité *(veuve noire, décédé)*, mais comme ce chemin n'est pas direct et a une distance supérieure à 1 nous ne déclenchons pas le mécanisme de filtrage pour le triplet d'évaluation *(veuve noire, r_associated, décédé)*.

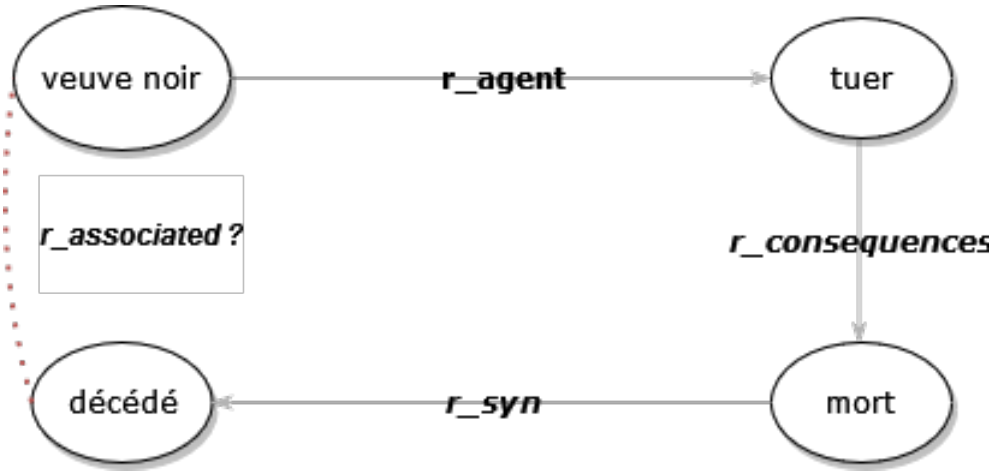


FIGURE 15 Illustration du chemin existant dans le graphe entre le couple d'entité d'évaluation.

Une autre étape de curation des données a été de filtrer les triplets liant *Jeux-DeMots* à d'autres bases de connaissances telles que *UMLS, DBNARY*, etc.
Nous avons aussi supprimé les triplets ayant des termes en anglais comme entité.
Le tableau présente la description des corpus d'entraînement et d'évaluation.

Jeu de données	Triplés	Entités	Relations
Apprentissage	75 934 808	2 919 012	132
Évaluation	16 360 688	2 172 008	132

TABLE 4.9 Description des données d'apprentissage et d'évaluation.

Comme montré dans le tableau 4.9, le nombre de relation est très élevé et cela risque de fortement augmenter la complexité du modèle.
Malgré le fait que cela est dû au fait que beaucoup de ces relations ont des relations inverses, le modèle que nous présentons ci-dessous représente les relations et leurs relations inverses indépendamment. Ce nombre élevé de relations nécessite

aussi d'avoir plus de contre-exemples (triplés invalides) pour apprendre une représentation discriminante des relations. Comme nous sommes limités en termes de ressources *GPU*, nous choisissons de réduire le nombre de relations en enlevant les relations inverses et en filtrant aussi les relations qui ont un nombre insuffisant de triplets.

Jeu de données	**Triplés**	**Entités**	**Relations**
Apprentissage	19 430 004	1 450 430	44
Évaluation	14 280 941	1 450 430	44

TABLE 4.10 Description finale des données d'apprentissage et d'évaluation.

Après avoir généré les données d'entraînement finaux, nous présentons la méthodologie suivie pour augmenter le modèle de langage avec les connaissances de *JeuxdeMots*.

4.6.2 Méthodologie

Nous suivons l'approche proposée par Yao, Mao and Luo (2019) qui repose sur une approche multi tâches pour affiner les représentations contextuelles du modèle de langage *BERT*. Nous commençons par présenter les tâches exploitées pour entraîner le modèle. Cela étant, nous montrons comment nous améliorons ces tâches d'entraînement et aussi nous présentons la tâche que nous ajoutons au processus d'entraînement.

Kim et al. (2020) proposent trois tâches pour affiner le modèle de langage *BERT* : *prédiction de lien, prédiction de relation et classement de pertinence*. Chaque tâche a une couche de classification $W \in R^{K \times H}$ où K est le nombre d'étiquettes et H est la taille de la couche cachée de *BERT*.

— **Prédiction de lien** *(Link Prediction)* :
Étant donné l'ensemble des triplets d'apprentissage P, l'entrée x est une séquence de texte composée du triplet (h, r, t). Chaque entité est représentée par le nom et la description de l'entité, par exemple, pour le triple *(tissu végétal, r_hyperonyme, structure végétale)*,
la séquence d'entrée est la suivante :

[CLS] tissu végétal, le tissu d'une plante [SEP] hyperonyme [SEP] structure végétale, toute partie d'une plante ou un champignon [SEP]

Le jeton *[CLS]* est utilisé par *BERT* pour indiquer le début de la séquence, tandis que le jeton *[SEP]* est utilisé pour séparer les entités dans la séquence. Le modèle est entraîné pour prédire si un triplet donné (h, r, t) est valide ou non. Comme la plupart des bases de connaissances sont seulement constituées de triplets valides, les triples invalides sont créés automatiquement en remplaçant l'entité de début (h) ou de fin (t) par l'une des entités aléatoires. L'ensemble de triplets invalides d'apprentissage est représenté par N.

Soit C le vecteur caché du jeton (*token*) [CLS] appartenant à la dernière couche du modèle, $W_{LP} \in R^2 \times H$ est la couche de classification pour la prédiction de lien. La plausibilité du triplet est alors inférée à travers une couche *Softmax* qui prend en entrée la sortie de W_{LP} comme exprimé par l'équation suivante :

$$f(x) = softmax(CW_{LP}) = |\hat{Y_0}, \hat{Y_1}| \tag{4.1}$$

La validité ou invalidité du triplet est représenté par le label binaire $Y \in \{0, 1\}$. La fonction du coût de l'erreur est définie par l'équation suivante :

$$Loss_{LP} = - \sum_{x \in \{P,N\}} ylog(\hat{Y_1}) + (y-1)log\hat{Y_0} \tag{4.2}$$

— Prédiction de relation (**Relation Prédiction**) :
Dans cette tâche, le modèle est entraîné à prédire la relation qui pourrait lier deux entités. L'entrée du modèle comprend la paire d'entité (h, t) où chacune de ces entités consiste en une séquence textuelle.
Par exemple, la séquence d'entrée peut être la suivante :

[CLS] tissu végétal, le tissu d'une plante [SEP] structure végétale, toute partie d'une plante ou un champignon [SEP].
Une fois l'entrée transmise au modèle, ce dernier est entraîné à inférer la relation la plus pertinente qui pourrait lier la paire d'entité de l'entrée, dans notre exemple la relation est *hyperonyme*.

La tâche de prédiction de relation a aussi une couche de classification spécifique $W_{RP} \in R^{R \times H}$ où R est le nombre d'étiquettes (nombre de relations) et H est la taille de la couche cachée de *BERT*.
Le label de la relation est inféré en transmettant la sortie du jeton *[CLS]* de la dernière couche C à la couche de classification W_{RP} suive d'une couche *softmax* tel que décrit par l'équation suivante :

$$g(x) = softmax(CW_{RP}) \tag{4.3}$$

La tâche de prédiction de relation est optimisée en minimisant la fonction de coût d'erreur définie par la cross-entropie :

$$Loss_{RP} = -\sum_{x \in S} ylog(x) \tag{4.4}$$

Nous tenons à préciser que nous n'avons pas géré le cas où il n'existe aucune relation qui pourrait lier les deux entités candidates.
Si certains travaux introduisent une nouvelle relation souvent nommée *None* pour traiter ce cas, nous n'avons pas fait de même car nous avons considéré le paradigme de monde ouvert. De ce fait, rien ne nous garantissait de générer aléatoirement des exemples négatifs mais qui peuvent être corrects. Aussi, dans cette tâche nous avons plus de chances de générer de faux positifs que la tâche de prédiction de lien car la génération aléatoire était fixée pour une entité et une relation donnée tandis que dans la tâche de prédiction de relation la génération aléatoire est seulement fixée pour une entité donnée.
Pour cela, nous fixons empiriquement un seuil de confiance pour décider de la validité de la prédiction ou non.

— Classement de pertinence (**Relevance Ranking**) :
Le principe de la tâche de classement de pertinence est de faire en sorte que le modèle associe des scores plus élevés au triplets valides par apport aux triplets invalides.
Étant l'ensemble de triplets valides et invalides P et N étant respectivement donnés, l'entrée du modèle se constitue de deux triplets $x = \{x_P, x_N\}$ tels que $x_P \in P$ et $x_N \in N$ et pour $x_P = (h, r, t)$ nous avons $x_N = (h, r, t_N)$ ou $x_N = (h_N, r, t)$.
Par exemple, l'entrée peut être respectivement constituée des triplets valide et

invalide suivants :

[CLS] tissu végétal, le tissu d'une plante [SEP] r_hyperonyme [SEP] structure végétale, toute partie d'une plante ou un champignon [SEP]
[CLS] tissu végétal, le tissu d'une plante [SEP] r_hyperonyme [SEP] végétarien, adepte d'un système d'alimentation (végétarisme) supprimant toute viande. [SEP]

Le modèle infère un score de pertinence en transmettant la sortie du jeton *[CLS]* de la dernière couche C à la couche de classification W_{RR}, la fonction *sigmoid* est ensuite appliquée pour équilibrer les scores entre 0 et 1.

$$h(x) = sigmoid(CW_{RR}) \tag{4.5}$$

La tâche est optimisée à travers la fonction du coût d'erreur de classement à base de marge, et le modèle est encouragé à fournir de plus grands écarts entre le score des triplets valides et invalides.

$$Loss_{RR} = \sum_{p \in P, n \in N} max\{0, h(n) - h(p) + \lambda_{RR}\} \tag{4.6}$$

Où $h(n)$ représente le score de pertinence du triplet invalide, $h(p)$ représente le score de pertinence du triplet valide et λ exprime la valeur de la marge.

Comme un mot peut porter différentes facettes de sémantiques, le contexte du mot est souvent sollicité pour déclencher la préférence de sélection qui permet de rejeter tous les sens non pertinents du mot tout en conservant le sens le plus plausible. Cependant, il est presque impossible de saisir toutes les variations (patrons) contextuelles d'un mot, en plus du signal faible dans lequel le contexte peut être implicitement formulé. Par conséquent, le processus de désambiguïsation s'effondre en raison du manque d'évidences entourant le bonne signification d'un mot.

Nous pouvons voir que tel qu'est actuellement définie la tâche de classement de pertinence (*Relevance Ranking*), elle n'offre pas d'alternative pour pallier ce genre de scénario.

Aussi, la stratégie actuelle de la tâche de classement de pertinence encourage une structure plate car la taxonomie (hiérarchie des mots) est complètement ignorée. En effet, seulement les interactions entre les triplets valides et les triplets invalides sont considérés tandis que l'interconnexion (hiérarchie) des triplets valides n'est pas modélisée. Nous pensons également que le modèle peut bénéficier davantage d'échantillons difficiles alors qu'il n'utilise que d'échantillons aléatoires faciles.

Pour éviter cette limite et pallier tout problème lié à la désambiguïsation, nous nous appuyons sur la propriété probabiliste des mots : *" Certains sens sémantiques sont plus susceptibles d'être portés par un mot ou plus fréquents que d'autres sens sémantiques "*. Ainsi, nous affinons la tâche de classement de pertinence en encouragent le modèle à associer des scores supérieurs au sens sémantique les plus fréquent ou les plus susceptibles en comparaison au triplets valides mais plus générique ou bien moins fréquent .

Aussi, en éloignant dans l'espace de représentation des mots les significations sémantiques spécifiques des significations sémantiques plus génériques ou moins spécifiques, nous pensons que nous avons tendance à séparer le mot d'ancrage d'un nombre plus élevé de mots non pertinents, car les mots porteurs d'un sens plus générique ont tendance à être ceux qui ont le plus haut degré de centralité dans le graphe de connaissances.

Finalement, en encourageant de manière itérative le mot d'ancrage à être plus proche de l'échantillon positif (un de ses sens sémantiques) que de l'échantillon négatif (sens sémantiques non pertinents), une situation peut survenir dans laquelle deux ou plusieurs échantillons positifs (significations) d'un mot soient proches l'un de l'autre tout en n'étant pas sémantiquement liés ou complètement non pertinents l'un à l'autre.
On peut penser par exemple que les mots *justice* et *fruit* peuvent être assez proches l'un de l'autre dans l'espace de représentation des mots car le mot polysémique *avocat* porte le sens de chacun de ces deux mots.
D'où la nécessite de modéliser la hiérarchie des sens pour éviter que cette situation non-désirée n'arrive.

Afin d'affiner la tâche de classement de pertinence comme définie plus haut,

nous tirons profit de la spécificité de *JeuxDeMots* où les triplets sont pondérés. Plus formellement, le graphe de *JeuxDeMots* consiste en un ensemble de quadruplés de la forme (h, r, t, w) associant au triplets standards (h, r, t) un poids w qui exprime la pertinence du triplet. Plus le poids w est petit, plus le triplet (h, r, t) est générique. Nous noterons aussi que lorsque le poids w est negatif, cela indique soit des exceptions ou des triplets invalides.

Ainsi, nous appelons la nouvelle tâche *classement de pertinence hiérarchique* (**Hierachical Relevance Ranking**), où le principe de la tâche est d'associer le score le plus élevé au triplet le plus spécifique des deux triplets valides.
Étant donné l'ensemble de triplets issues de *JeuxDeMots* T, l'entrée du modèle se constitue de deux triplets $x = \{x_1, x_2\}$ pour $Q_1 \in T$ et $Q_2 \in T$ et $Q_1 = (x_1, w_1)$, $Q_2 = (x_2, w_2)$ tel que $x_1 = (h, r, t_1)$ et $x_2 = (h, r, t_2)$ sachant que $w_1 > w_2$ et $w_1 > 0$.
Par exemple, l'entrée peut être respectivement composée des deux triplets spécifiques (plus pertinents) et génériques (moins pertinents) :

[CLS] chat, petit mammifère familier à poil doux, aux yeux oblongs et brillants, à oreilles triangulaires, aux griffes rétractiles [SEP] r_isa [SEP] félin, Un félin ou félidé est un mammifère carnivore digitigrade (qui marche sur ses doigts) et carnivore [SEP]

[CLS] chat, petit mammifère familier à poil doux, aux yeux oblongs et brillants, à oreilles triangulaires, aux griffes rétractiles [SEP] r_isa [SEP] animal, être vivant organisé, doué de sensibilité et qui (en général) peut se mouvoir (opposé aux végétaux) [SEP]

Le modèle infère un score de pertinence en transmettant la sortie du jeton *[CLS]* de la dernière couche C à la couche de classification W_{HRR}, la fonction *sigmoid* est ensuite appliquée pour balancer les scores entre 0 et 1.

$$h(x) = sigmoid(CW_{HRR}) \tag{4.7}$$

La tâche est optimisée à travers la fonction du coût d'erreur de classement à base de marge, et le modèle est encouragé à fournir de plus grands écarts entre le

score des triplets valides et invalides.

$$
\begin{aligned}
Loss_{HRR} = \sum_{x_1 \in Q_1, x_2 \in Q_2} max\{0, h(x_2) - h(x_1) + \lambda_{HRR}\} \\
Q_1 = (x_1, w_1) \\
Q_2 = (x_2, w_2) \\
w_1 > w_2, w_1 > 0
\end{aligned}
\tag{4.8}
$$

Où $h(w_2)$ représente le score de pertinence du triplet le moins pertinent, $h(w_1)$ représente le score de pertinence du triplet le plus pertinent et λ exprime la valeur de la marge.

Finalement, nous introduisons aussi une tâche de réajustement (*Retrofitting*) des vecteurs de mot en exploitant aussi JeuxDeMots. L'idée générale de la tâche porte sur la distance séparant deux vecteurs de mots dans l'espace de représentation, avec trois principales propriétés :

— La distance séparant deux vecteurs de mot synonymes dans l'espace de représentation doit être inférieure à la distance séparant le vecteur de l'un des mots avec le vecteur de son mot antonyme.
— La distance séparant deux vecteurs de mot synonymes dans l'espace de représentation doit être inférieure à la distance séparant le vecteur de l'un des mots avec le vecteur d'un mot échantillonné au hasard.
— La distance séparant deux vecteurs de mot antonymes dans l'espace de représentation doit être supérieure à la distance séparant le vecteur de l'un des mots avec le vecteur d'un mot échantillonné au hasard.

Réajustement (**Retrofitting**) : Dans cette tâche, le modèle est entraîné à minimiser la fonction de distance de deux entités synonymes tout en maximisant la fonction de distance de deux entités antonymes.
Étant donné E_{syn} l'ensemble des triplets des synonymes et E_{anto} l'ensemble des triplets des antonymes, l'optimisation de la tâche d'ajustement peut être considé-

rée comme un problème de satisfaction des contraintes suivantes :

$$\forall w_{anchor}, w_{syn}, w_{rnd}; \tag{4.9}$$
$$(w_{anchor}, r_syn, w_{syn}) \in E_{syn} \tag{4.10}$$
$$(w_{anchor}, r_syn, w_{rnd}) \notin E_{syn} \tag{4.11}$$
$$(w_{anchor}, r_anto, w_{rnd}) \notin E_{anto} \tag{4.12}$$
$$dist(w_{anchor}, w_{syn}) < dist(w_{anchor}, w_{rnd}) + \lambda \tag{4.13}$$

L'équation ci-dessus définit la contrainte spécifiant la similarité entre deux entités synonymes par apport à une fonction de distance, en comparaison de deux entités non synonymes.

$$\forall w_{anchor}, w_{rnd}, w_{anto}; \tag{4.14}$$
$$(w_{anchor}, r_syn, w_{rnd}) \notin E_{syn} \tag{4.15}$$
$$(w_{anto}, r_anto, w_{rnd}) \notin E_{anto} \tag{4.16}$$
$$(w_{anchor}, r_anto, w_{anto}) \in E_{anto} \tag{4.17}$$
$$dist(w_{anchor}, w_{rnd}) < dist(w_{anchor}, w_{anto}) + \lambda \tag{4.18}$$

L'équation ci-dessus définit la contrainte spécifiant la similarité entre deux entités non synonymes par apport à une fonction de distance, en comparaison de deux entités antonymes.

L'entrée du modèle comprend un des deux triplets $(w_{anchor}, w_{syn}, w_{rnd})$ ou $(w_{anchor}, w_{rnd}, w_{anto})$, où chacune de ces entités consiste en une séquence textuelle. Par exemple, la séquence d'entrée peut être le triplet suivant :

— *[CLS] solidité [SEP]*
— *[CLS] robustesse [SEP]*
— *[CLS] fragilité [SEP]*

Chacune de ces séquences textuelles est transmise au modèle *Flaubert* pour obtenir la représentation vectorielle de ces entrées.
Une fois cela fait, nous exploitons une fonction à valeur réelle non negatif pour quantifier la proximité entre le vecteur de l'entité d'ancrage w_{anchor} et le vecteur de l'entité positive w_{syn} ou w_{rnd}.

On fait aussi appel à la fonction à valeur réelle non négative pour quantifier le deuxième score de proximité entre le vecteur de l'entité d'ancrage w_{anchor} et le vecteur de l'entité négative w_{anto} ou w_{rnd}.
Comme une entité peut comporter plusieurs mots, nous calculons d'abord la moyenne de ces vecteurs C^{AVG} à partir des vecteurs de la dernière couche C_i :

$$C^{AVG} = \frac{1}{n} \sum_{i=1}^{n} C_i \tag{4.19}$$

On utilise la fonction mathématique cosinus $cosine - distance$ pour calculer la proximité des deux vecteurs :

$$dist_p(w_{anchor}, w_{pos}) = cosine - distance(C^{AVG}_{anchor}, C^{AVG}_{pos}) \tag{4.20}$$

$$dist_n(w_{anchor}, w_{neg}) = cosine - distance(C^{AVG}_{anchor}, C^{AVG}_{neg}) \tag{4.21}$$

Où $dist_p(w_{anchor}, w_{pos})$ représente la distance cosinus entre la moyenne des vecteurs de l'entité d'ancrage C^{AVG}_{anchor} et la moyenne des vecteurs de l'entité positive C^{AVG}_{pos}, et $dist_n(w_{anchor}, w_{neg})$ représente la distance cosinus entre la moyenne des vecteurs de l'entité d'ancrage C^{AVG}_{anchor} et la moyenne des vecteurs de l'entité négative C^{AVG}_{neg}.

La tâche est optimisée à travers la fonction du coût d'erreur de classement à base de marge, et le modèle est encouragé à fournir de plus grands écarts entre le score de proximité des entités positives et le score de proximité des entités négatives.

$$Loss_{RF} = \sum_{p \in P, n \in N} max\{0, dist(n) - dist(p) + \lambda_{RF}\} \tag{4.22}$$

Où P représente l'ensemble des couples d'entités positives $\{(w_{anchor}, w_{pos})\}$ et N représente l'ensemble des couples d'entités négatives $\{(w_{anchor}, w_{neg})\}$,
avec $dist(n)$ représente le score de proximité du couple d'entités positives, $dist(p)$ représente le score de proximité du couple d'entités négatives et λ exprime la valeur de la marge.

L'optimisation de la tâche de réajustement telle qu'elle est définie actuellement pose un problème, car elle influe négativement sur la convergence des fonctions des autres tâches. En effet, comme la tâche de réajustement change drastiquement la topologie des vecteurs, les autres tâches seront forcément impactées vu que les vecteurs auront changé.

Pour cela, nous introduisons une autre contrainte pour pénaliser les transformations de l'espace vectoriel qui change drastiquement la topologie de la représentation vectorielle des entités.
Ainsi, pour chacune des entités de la tâche de réajustement $w_{anchor}, w_{pos}, w_{neg}$, nous devons quantifier combien la topologie des vecteurs initiaux a changé.
Soit $O_{anchor}, O_{pos}, O_{neg}$ respectivement les vecteurs originaux des entités d'ancrage, entité positive et entité négative obtenue de la couche de projection des mots vers les vecteurs dans BERT.
On commence par calculer la proximité entre le vecteur original et le vecteur transformé de la dernière couche de ***BERT*** en exploitant la fonction mathématique cosinus $cosine - distance$.

$$dist_{anchor} = cosine - distance(C^{AVG}_{anchor}, O^{AVG}_{anchor}) \tag{4.23}$$

$$dist_{pos} = cosine - distance(C^{AVG}_{pos}, O^{AVG}_{pos}) \tag{4.24}$$

$$dist_{neg} = cosine - distance(C^{AVG}_{neg}, O^{AVG}_{neg}) \tag{4.25}$$

Où $dist_{anchor}$ représente la distance cosinus entre la moyenne des vecteurs transformés (à la dernière couche de ***BERT***) de l'entité d'ancrage C^{AVG}_{anchor} et la moyenne des vecteurs originaux de l'entité d'ancrage O^{AVG}_{anchor},
$dist_{pos}$ représente la distance cosinus entre la moyenne des vecteurs transformés (à la dernière couche de ***BERT***) de l'entité positive C^{AVG}_{pos} et la moyenne des vecteurs originaux de l'entité positive O^{AVG}_{pos},
et $dist_{neg}$ représente la distance cosinus entre la moyenne des vecteurs transformés (à la dernière couche de ***BERT***) de l'entité négative C^{AVG}_{neg} et la moyenne des

vecteurs originaux de l'entité négative C_{neg}^{AVG}.

On commence par calculer la fonction du coût d'erreur liée au changement de topologie en additionnant les trois distances définies précédemment :

$$Loss_{topology} = \sum_{a \in A, p \in P, n \in N} (max\{0, dist_{anchor}(a) + dist_{anchor}(p) + dist_{anchor}(n)\}) \tag{4.26}$$

Où A représente l'ensemble de couples des vecteurs d'entité d'ancrage originale et transformée $\{(O_{anchor}, C_{anchor})\}$,
P représente l'ensemble de couples des vecteurs d'entité positive originale et transformé $\{(O_{pos}, C_{pos})\}$,
et N représente l'ensemble de couples des vecteurs d'entité négative originale et transformée $\{(O_{neg}, C_{neg})\}$.

Ainsi, la fonction finale du coût d'erreur de la tâche de réajustement est calculée de la manière suivante :

$$Loss_{retrofitting} = Loss_{rf} + \lambda_{topology} \times Loss_{topology} \tag{4.27}$$

Où $\lambda_{topology}$ est un hyperparamètre qui permet de déterminer à quel point nous préservons la topologie de l'espace vectoriel d'origine.
La figure 16 ci-dessous illustre l'architecture de la méthode d'apprentissage multi-tâche que nous proposons pour *Jeux2BERT*.

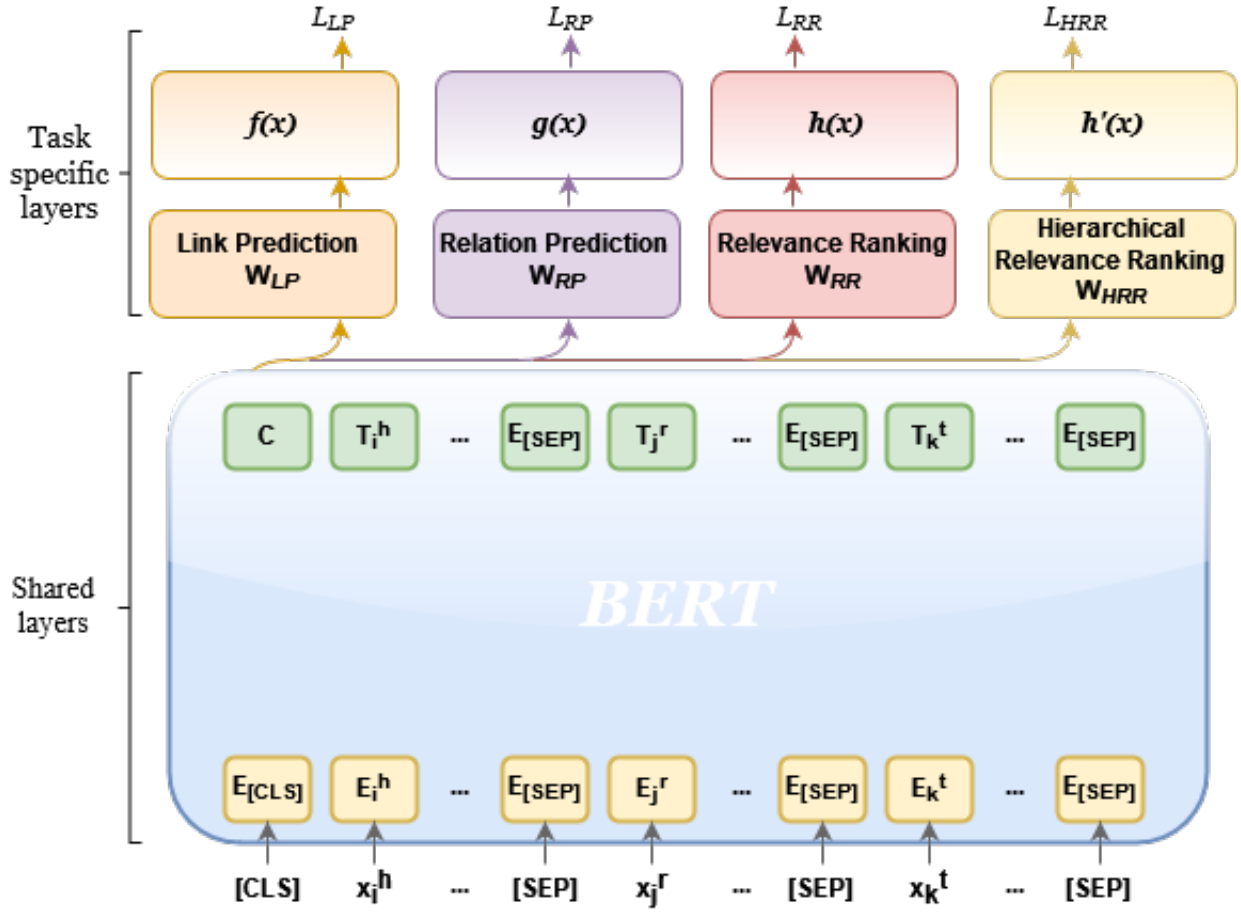


FIGURE 16 Architecture de la méthode d'apprentissage multi-tâche de *Jeux2BERT*.

4.7 Expérimentations

4.7.1 Données d'apprentissage et d'évaluation

Nous divisons aléatoirement le corpus de *JeuxDeMots* en un corpus d'apprentissage et un corpus d'évaluation selon un ratio de 70% et 30% respectivement, nous appliquons ensuite l'étape de traitement des donnés décrite dans la section 4.6.1. Le tableau 4.11 présente la description des corpus d'entraînement et d'évaluation.

Jeu de données	Triplets	Entités	Relations
Apprentissage	19 430 004	1 450 430	44
Évaluation	14 280 941	1 450 430	44

TABLE 4.11 Composition des données d'apprentissage et d'évaluation.

Comme la plupart des bases de connaissances ne contiennent que des triplets

valides et rarement des triplets invalides, (par exemple *JeuxDeMots* ne contient que 2% de triplets invalides) nous générons aléatoirement des triplets invalides.
Pour ce faire, pour chaque triplet valide de la forme (h, r, t), et pour une entité donnée choisie au hasard rnd nous considérons un triplet invalide (rnd, r, t). S'il n'existe pas dans l'ensemble des triplets valides, nous l'ajoutons dans l'ensemble de triplets d'apprentissage. De même, pour le triplet valide (h, r, t) nous considérons le triplet invalide (h, r, rnd) et s'il n'existe pas dans l'ensemble des triplets valides nous l'ajoutons dans l'ensemble d'apprentissage.
Ainsi, nous générons les triplets invalides avec un ratio de quatre triplets invalides pour chaque triplet valide, deux triplets où nous remplaçons la tête du triplet aléatoirement avec une entité donnée et deux triplets où nous remplaçons la queue du triplet avec une entité donnée.
Pour la tâche de prédiction de lien nous générons deux triplets négatifs pour un triplet positif, tandis que pour la tâche de classement de pertinence, nous générons qu'un triplet négatif pour un triplet positif. Pour la tâche de prédiction de relation, nous n'avons pas généré de triplets négatifs.
Dans leurs travaux, Kim et al. (2020) ont généré 8 triplets invalides pour chaque triplet valide. Nous avons gardé le ratio de triplets invalides par triplets valides à 2 afin que les différents systèmes de référence soient comparables dans les expérimentations.
Le tableau 4.12 décrit l'ensemble d'apprentissage pour chaque tâche :

tâche	**taille de l'échantillon**
Prédiction de lien	97 150 020
Prédiction de relation	19 430 004
Classement de pertinence	58 290 012
Classement de pertinence hiérarchique	17 168 980
Réajustement	1 828 438

TABLE 4.12 Description des données d'apprentissage pour chaque tâche.

4.7.2 Modèle et Hyper-paramètres

Comme modèle initial pour *Jeux2BERT* (on utilisera ce nom pour faire référence à notre modèle), nous choisissons *FlauBERT* (Le et al. 2020) qui est l'équivalent du *BERT* français et a été entraîné sur un corpus français très grand et

hétérogène.
Nous choisissons la petite version de *Flaubert* (qui a un nombre de paramètres réduit et donc le temps d'apprentissage sera relativement réduit par rapport au temps d'apprentissage de la version moyenne) qui consiste en 6 couches et 8 têtes d'attention et dont la dimension des vecteurs est de 512.
Dans notre choix de représentation des entrées du modèle, nous avons décidé de traiter les relations comme des jetons spéciaux tels que $[CLS]$. Ainsi, la relation is_a aura un jeton spécial et un vecteur unique qui est $[r_isa]$.
On optimise notre modèle lors de l'ajustement fin en utilisant l'optimiseur *Adam*. Nous définissons les hyperparamètres de *Jeux2BERT* dans le tableau 4.13 ci-dessous :

hyper-paramètre	valeure
taille du batch	500
taux d'apprentissage	$5e^{-5}$
nombre d'itérations	5
dropout	0.1
marge λ_{RF}	0.1
marge λ_{RR}	0.5
marge λ_{HRR}	0.1
marge $\lambda_{topology}$	2

TABLE 4.13 Liste des hyper-paramètres du modèle Jeux2BERT.

4.7.3 Systèmes de référence

Nous comparons notre approché au modèles *KG-BERT* et *MTL-KGC* respectivement proposés par Yao, Mao and Luo (2019) et Kim et al. (2020). Notre méthode correspondant au modèle *Jeux2BERT*, nous considérons aussi deux autres variantes de notre modèle *Jeux2BERT(HRR)* et *Jeux2BERT(RF)*, et qui sont présentés en détail dans la section suivante.

4.7.4 Étude d'ablation

Comme notre approche se distingue de *MTL-KGC* (Kim et al. 2020) en incluant deux tâches additionnelles, classement de pertinence hiérarchique *(Hie-*

rarchical Relevance Ranking (HRR)) et réajustement *Retrofitting (RF)*. Nous assumons deux autres variantes de notre modèle pour connaître l'impact de chaque tâche.
La première variante *Jeux2BERT(HRR)* est la variante de *Jeux2BERT* entraîné sur la tâche de classement de pertinence hiérarchique en plus des trois tâches : prédiction de lien, prédiction de relation et classement de pertinence.
La deuxième variante *Jeux2BERT(RF)* est la variante de *Jeux2BERT* entraîné sur la tâche de réajustement en plus des trois tâches : prédiction de lien, prédiction de relation et classement de pertinence.
On inclut aussi le modèle *MTL-KGC* (Kim et al. 2020) qui correspond à *Jeux2BERT* sans les deux tâches susmentionnées : classement de pertinence hiérarchique et réajustement.

4.7.5 Évaluations

Nous évaluons notre méthode sur la prédiction de lien, où le modèle prédit l'entité de tête étant donnée $(?, r, t)$ et l'entité de queue étant donnée $(h, r, ?)$. Pour comparer les travaux antérieurs, nous suivons le protocole d'évaluation et la configuration filtrée de Bordes et al. (2013), exploité aussi par Yao, Mao and Luo (2019) et Kim et al. (2020) .
Soit E un ensemble d'entités et T un ensemble de triplets englobant les triplets d'apprentissage, de validation et d'évaluation. Alors, l'ensemble des candidats au test U pour prédire l'entité tète h dans le triplet donné (h, r, t) est :

$$U = (h, r, t) \cup \{(h^{'}, r, t) | h^{'} \in E \wedge (h^{'}, r, t) \notin T\} \tag{4.28}$$

et l'ensemble des candidats au test U pour prédire l'entité queue t dans le triplet donnée (h, r, t) est :

$$U = (h, r, t) \cup \{(h, r, t^{'}) | t^{'} \in E \wedge (h, r, t^{'}) \notin T\} \tag{4.29}$$

Une fois que le modèle a calculé les scores de tous les triplets candidats, ils sont triés par ordre décroissant. Les performances sont évaluées en classement moyen *(Mean Rank (MR))*, classement moyen réciproque *(Mean Reciprocal Rank*

(MRR)) et taux de réussite (*Hits*) @1, @3, @10.
Comme le processus d'évaluation tel qu'il est décrit prend beaucoup de temps à s'exécuter, nous avons décidé de choisir aléatoirement 1000 triplets et pour chacun de ces triplets, nous générons aléatoirement 4000 triplets qui peuvent être valides ou invalides. À noter que nous générons des triplets invalides différents pour la prédiction de l'entité de tête $(?, r, t)$ et la prédiction d'entité de queue $(h, r, ?)$.
Pour conclure les expérimentations, nous évaluons aussi les systèmes de référence sur la tâche de classification de triplets et qui correspond à la tache de prédiction de liens. Pour cela, nous générons aussi des triplets invalides aléatoirement pour équilibrer entre les triplets valides et invalides.
On considère aussi la tâche de classification de relations et qui correspond à la tâche de prédiction de type de la relations.

4.7.6 Résultats et Discussions

On aborde, dans cette section, les principaux résultats des expérimentations et nous essayons d'analyser les différentes constatations que nous avons trouvées.
On commence par les résultats de prédiction de lien (la classification des triplets) et les résultats de classification de relations (la classification des relations).

Modèle	Précision↑	Rappel↑	F1↑	Accuracy↑
MTL-KGC	95.82%	96.0%	95.80%	95.79%
Jeux2BERT	95.62%	95.29%	95.29%	95.72%

TABLE 4.14 Résultats de l'évaluation de la tâche de prédiction de lien.

Le tableau 4.14 illustre les résultats de l'évaluation de prédiction de lien. Les résultats montrent que le modèle *MTL-KGC* est légèrement supérieur à notre modèle. Nous expliquons cela par le fait que notre modèle converge moins rapidement car il est optimisé par deux autres tâches additionnelles.

Modèle	Hits@1↑	Hits@3↑	Précision↑	Rappel↑	F1↑
MTL-KGC	83.32%	99.64%	76.69%	73.91%	75.00%
Jeux2BERT	84.2%	99.67%	76.86 %	73.84%	75.00%

TABLE 4.15 Résultats de l'évaluation de la tâche de prédiction de relations.

Le tableau 4.15 illustre les résultats de l'évaluation de prédiction de relations. Nous pouvons voir que les résultats sont approximativement les mêmes. Nous

notons aussi qu'il y a une différence de performance entre la métrique $F1$ et la métrique $Hits@1$, cela est dû au fait que nous avons traité le cas où plusieurs types de relations existent entre deux entités données avec la métrique $Hits@1$.

Modèle	MR↓	MRR↑	Hits@1↑	Hits@3↑	Hits@10↑
MTL-KGC	74	42.62	33.19	48.78	55.85
Jeux2BERT	53	44.03	34.37	49.21	59.37

TABLE 4.16 Résultats de l'évaluation de la tâche de prédiction de liens avec classement.

Le tableau 4.16 montre comment notre méthode améliore les performances par apport au système de l'état de l'art *MTL-KGC* (Kim et al. 2020) et autres systèmes de référence.
Le premier constat est que notre approche a un meilleur impact dans la métrique d'évaluation $Hits@10$ que les deux autres métriques $Hits@3$ et $Hits@1$. Ce qui signifie que notre approche cible les triplets plus au moins difficiles, puisqu'elle permet d'inclure plus de triplets valides dans la liste des meilleurs dix propositions.
Ainsi, nous pouvons voir que la différence est moins significative dans les métriques $Hits@3$ et $Hits@1$, ce qui laisse envisager que les triplets plus ou moins faciles ne sont forcément mieux classés par notre méthode.
Étant donné que des triplets sont considérés comme faciles, ils devraient plus apparaître dans la liste de la meilleures proposition $Hits@1$, et trois meilleures propositions $Hits@3$ propositions que dans la liste des dix meilleures propositions $Hits@10$.

Modèle	MR↓	MRR↑	Hits@1↑	Hits@3↑	Hits@10↑
MTL-KGC	74	42.62	33.19	48.78	55.85
Jeux2BERT (RF)	69	43.17	33.87	48.47	55.90
Jeux2BERT (HRR)	55	43.77	34.24	49.04	59.18
Jeux2BERT	53	44.03	34.37	49.21	59.37

TABLE 4.17 Résultats de l'étude d'ablation.

Le tableau 4.17 illustre l'impact de chaque tâche dans les performances du modèle. Les résultats montrent que la tâche de réajustement n'a pas d'influence significative dans les performances du modèle et la métrique d'évaluation $Hits@k$. Nous nous apercevons que la tâche de réajustement cible principalement la métrique de classement moyen.

Les résultats illustrés par 4.17 montrent que le modèle *Jeux2BERT* bénéficie grandement de la tâche de classement de pertinence hiérarchique, où toutes les métriques $Hits@1$, $Hits@3$ et $Hits@10$ ont un gain de performance, le même cas de figure pour les métriques classement moyen réciproque et classement moyen.

Jeux2BERT			**MTL-KGC**		
rang	***entité***	***score***	***rang***	***entité***	***score***
0	boulot	5.0072	0	condoléances	6.4448
1	mon coeur	4.8244	1	prédire	5.381
2	nanti	4.7912	2	reprendre confiance	5.2342
3	mouton	4.7764	3	communion religion	5.1638
4	stérilité conjugale	4.3775	4	souvenirs	5.0629
5	schall	4.2517	5	soeurs	5.0387
6	porter affection	4.2334	6	raison	5.0329
7	avoir le ballon	4.2096	7	se féliciter de	5.0251
8	guichet automatique	4.1687	8	tenace	5.0074

TABLE 4.18 Exemple des triplets résultant de la requête ***<?, r_sentiment, compassion>*** **.**

Le tableau 4.18 liste les triplets résultants de la requête ***<?, r_sentiment, compassion>***, et correspond aux entités auxquelles sont associés un sentiment de compassion.
Nous pouvons voir que le modèle *Jeux2BERT* associe un score relativement élevé à des triplets non pertinents impliquant les entités : *nanti, schall et guichet automatique*. Il en est de même concernant *MTL-KGC* avec les triplets impliquants les entités : *prédire, raison et se féliciter de*.

Jeux2BERT			**MTL-KGC**		
rang	***entité***	***score***	***rang***	***entité***	***score***
0	costume de marié	7.9068	0	quart>récipient	7.5281
1	guitare à cinq choeurs	7.5255	1	jardinière>récipient	6.8589
2	couvert	7.4262	2	table à neige	6.6937
3	table à neige	7.1273	3	servocommande	6.5178
4	costume de fée	7.06	4	canari>récipient	6.5113
5	marron glacé	6.9903	5	pitepalt	6.4252
6	tenue de soirée	6.8503	6	couvert	6.3728
7	napi	6.7657	7	things	6.338
8	canari>récipient	6.6276	8	porte - équipement	6.3068

TABLE 4.19 Exemple des triplets résultant de la requête ***<?, r_lieu, armoire> .***

Le tableau 4.19 liste les triplets résultant de la requête ***<?, r_lieu, armoire>***, et correspond aux entités qui peuvent se trouver à l'intérieur d'une armoire.
La première erreur que nous constatons dans les inférences est la présence de l'entité *napi* (numéro d'apiculteur) parmi les neuf meilleures prédictionss de *Jeux2BERT* et la présence de l'entité *servocommande* (système de régulation hydraulique, électro-hydraulique ou électrique) parmi les neuf meilleures prédictions de *MTL-KGC*.
Nous remarquons aussi que le modèle *Jeux2BERT* propose des entités de type ustensile de cuisine (*couvert et canari>récipient*) et de type vêtement (*costume de marié, costume de fée et tenue de soirée*), tandis que le modèle *MTL-KGC* propose principalement des ustensiles de cuisine (*quart>récipient, jardinière>récipient, canari>récipient et couvert*).
Aussi, chacun des deux modèles propose une entité de type nourriture, *marron glacé* pour *Jeux2BERT* et *pitepalt* (spécialité culinaire suédoise) pour *MTL-KGC*.

Jeux2BERT			**MTL-KGC**		
rang	***entité***	***score***	***rang***	***entité***	***score***
0	armoire	6.7902	0	armoire	6.1431
1	couvert	4.349	1	tenue de soirée	4.8209
2	cuisine familiale	3.8369	2	piquenique	4.1842
3	tenue de soirée	3.7491	3	chiottes	4.1111
4	cuisine>maison	3.5786	4	couvert	3.9651
5	chiottes	3.5107	5	chaussures à talons	3.7456
6	boulanger	3.2667	6	panetier	3.1663
7	boulot	3.0596	7	salle d' eau	2.7934
8	brassière	2.8921	8	cuisse viande	2.7398

TABLE 4.20 Exemple des triplets résultant de la requête ***< tablier>vêtement, r_lieu, ?> .***

Le tableau 4.20 liste les triplets résultant de la requête ***< tablier>vêtement, r_lieu, ?>***, et correspond aux entités de type lieu dans lesquels nous pouvons trouver un tablier.
Les résultats montrent que *Jeux2BERT* propose plus de triplets pertinents et logiques (*armoire, cuisine familiale, cuisine>maison, boulot et boulanger*) que *MTL-KGC* (*armoire et salle d'eau*).

Jeux2BERT			**MTL-KGC**		
rang	***entité***	***score***	***rang***	***entité***	***score***
0	vêler	6.3123	0	vêler	6.4727
1	couvrir de	5.5973	1	couvrir de	5.8501
2	couvrir de fange	5.4708	2	couvrir de fange	5.778
3	se protéger	5.3272	3	se protéger	5.3715
4	peindre peinture	4.709	4	enficeler	5.1887
5	farder	4.6292	5	cagner	4.7125
6	s' étouffer	4.4221	6	circonscrire	4.6451
7	daler	4.3483	7	coucher allonger	4.4847
8	poiler	4.3098	8	flamber augmenter	4.4637

TABLE 4.21 Exemple des triplets résultant de la requête ***<?, r_syn, couvrir> .***

Le tableau 4.21 liste les triplets résultant de la requête ***<?, r_syn, couvrir>***, et correspond aux entités qui sont synonymes de couvrir.

Jeux2BERT			**MTL-KGC**		
rang	***entité***	***score***	***rang***	***entité***	***score***
0	allergie aux tomates	4.2259	0	anaphylaxiques	5.1432
1	c30 - c34 états d' hypersensibilité et états atopiques	3.9972	1	allergie aux tomates	4.4678
2	anaphylaxiques	3.5771	2	sensation d' irritabilité	4.1365
3	covid - 19	3.3691	3	dyspepsie	3.4788
4	insuffisance parathyroïdienne	2.5515	4	covid - 19	3.2972
5	limiter le sucre	2.5447	5	symptôme de mal de mer	3.1856
6	dyspepsie	2.4922	6	épileptoïdie	2.7286
7	coprostérol	2.1991	7	hystérie	2.594
8	trouble hémorragique	2.1836	8	accident de sport	2.5122

TABLE 4.22 Exemple des triplets résultant de la requête ***<?, r_associated, choc anaphylactique>*** **.**

Le tableau 4.22 liste les triplets résultant de la requête ***<?, r_associated, choc anaphylactique>***, et correspond aux entités qui sont associés à un choc anaphylactique.

Nous remarquons que le modèle *MTL-KGC* privilégie la similarité lexicale au détriment des propriétés relationnelles des mots, cela se concrétise par le fait que *MTL-KGC* classe l'entité *anaphylaxiques* avant et avec un score plus élevé que l'entité *allergie aux tomates*. En effet, on peut remarquer que les deux entités sont plus similaires lexicalement en prenant en compte que le modèle *Flaubert* utilise l'approche *BPE* (sous-mots) comme stratégie de segmentation des mots. Ainsi on aura (*anaphyl/ axiques_*) pour l'entité *anaphylaxiques* et (*choc_/ anaphyl/ actique_*) pour l'entité *choc anaphlactique*.

D'un autre côté, nous pouvons voir que le modèle *Jeux2BERT* ne privilégie pas la similarité lexicale puisque l'entité *anaphylaxiques* n'est proposée qu'en troisième position. Pour finir, les deux modèles incluent l'entité *covid-19* car plusieurs vaccins de ce dernier ont été associés à des cas de réactions d'allergie.

Jeux2BERT			**MTL-KGC**		
rang	***entité***	***score***	***rang***	***entité***	***score***
0	branchie	6.1623	0	branchie	7.4371
1	osphradie	1.2219	1	barre latérale	-0.8435
2	couronne dentaire	0.418	2	couettes	-0.8987
3	plot	-0.5242	3	glandes buccales	-0.9594
4	vitamine	-0.9703	4	fourrure duveteuse	-1.2415

TABLE 4.23 Exemple des triplets résultant de la requête ***< requin nourrice, r_has_part, ?> .***

Le tableau 4.23 liste les triplets résultant de la requête ***< requin nourrice, r_has_part, ?>***, et correspond aux entités qui font partie de ou constituent une partie du requin nourrice.

Jeux2BERT			**MTL-KGC**		
rang	***entité***	***score***	***rang***	***entité***	***score***
0	blastes cd10	7.8752	0	flaer / 100 cellules	7.0231
1	mesure de cd11 / 100 cellules	7.817	1	ny-eso-1 anticorps	6.874
2	cd13 cellules anormales	7.668	2	cephalosporium acremonium ac igg4	6.773
3	mesure d' histamine	7.6385	3	hydromorphone libre	6.7427
4	mesure de blastes cd13	7.6123	4	6-monoacétylmorphine libre	6.6899
5	date début du recueil	7.578	5	valproate libre	6.5974
6	mesure de dichlorvos	7.5459	6	bk virus ac igg	6.5966
7	échantillon de sol	7.5247	7	neisseria meningitidis sérogroupes a+y ag	6.5463
8	mesure du vers	7.4989	8	bêta-2 microglobuline lcr	6.5407

TABLE 4.24 Exemple des triplets résultant de la requête ***<?, r_lieu, liquide céphalo rachidien>.***

Le tableau 4.24 liste les triplets résultant de la requête ***<?, r_lieu, liquide céphalo rachidien>***, et correspond aux entités qui se trouvent à l'intérieur du liquide céphalo rachidien.

4.8 Conclusion

Nous proposons une méthode d'apprentissage multi-tâches efficace pour la complétion de graphes de connaissances et l'affinement de représentations des vecteurs de mots en introduisant une variante de la tâche de classement de pertinence. Les résultats expérimentaux démontrent l'impact de la tâche de classement de pertinence hiérarchique sur l'apprentissage et comment elle améliore les performances de *MRR* et *Hits@*K par à l'état de l'art (Kim et al. 2020).

Nous jugeons qu'il est important et possible de rendre plus explicite les connaissances de chaque entité du triplet lors de l'inférence, en fournissant un contexte pour chaque argument. Par exemple, ce contexte serait formé de tous les triplets dans lesquels l'argument du triplet en question est présente en tant que source ou cible (cela correspondrait à une représentation en étoile pour chaque entité).

Dans le chapitre suivant, nous expliquons comment combiner le modèle *UD2OIE* et le modèle *Jeux2BERT* pour l'extraction de relations sémantiques.

Chapitre 5

Extraction de Relations via OpenIE et Vecteurs de Mots

5.1 Introduction

Dans ce chapitre, nous proposons un modèle visant à effectuer l'extraction de relations sémantiques à partir de textes dans un contexte de veille et d'intelligence d'entreprise. Relativement à ce que nous avons présenté dans l'état de l'art sur l'extraction de relations sémantiques (chapitre 2), notre approche appartient à la catégorie des approches non supervisés. Plus formellement, nous proposons une approche en deux temps qui suit les niveaux stylistiques du langage. Notre système commence par exploiter *UD2OIE* qui se concentre sur la représentation syntaxique afin d'obtenir la représentation prédicat-arguments, et ensuite fait appel à *Jeux2BERT* qui dérive des vecteurs de mots modélisant les relations lexicales qui existent entre les mots et modélisant aussi un certain nombre de relations sémantiques.

La section 5.2 commence par revisiter la problématique étudiée en listant les types de relations sémantiques cibles ainsi que les traitements opérés pour obtenir les entrées de notre système d'extraction de relations sémantiques.
La section 5.3 reprend la méthodologie suivie pour normaliser les triplets obtenus par *UD2OIE*, le passage du paradigme de l'OpenIE vers le paradigme d'extraction de relations sémantiques. Pour cela, la méthodologie suivie pour combiner *UD2OIE* et *Jeux2BERT* est expliquée en détails.
Enfin, nous concluons ce chapitre par la section 5.4 qui présente les résultats obtenus lors de nos expérimentations et fournit une analyse détaillée de ces résultats.

5.2 Redéfinition du problème

Comme notre système d'extraction de relations est destiné au cas d'usage de l'intelligence économique et de la veille technologique, nous nous intéressons à l'extraction de relations d'intérêt en économie principalement prédéfinies entre entreprises *(ORG)* et personnes *(PER)* à partir d'articles de presse. Ces relations permettent l'analyse de la concurrence, la valorisation des entreprises et le suivi de leurs activités. C'est donc un moyen crucial pour identifier les liens entre les acteurs d'un secteur d'activité et pour l'analyse de leurs écosystèmes économiques. Dans le cadre de l'analyse d'adaptation au domaine, nous choisissons de considérer deux sous-domaines de l'intelligence économique et de la veille technologique comme les deux domaines de spécialité à comparer. Le premier sous-domaine couvre principalement la structure de l'entreprise, donc des informations relatives aux employés, aux dirigeants, aux filières et à la société mère. Le deuxième sous-domaine cible les principales activités financières d'une entreprise tels que l'achat d'actions, les acquisitions, les partenariats et la concurrence.
Nous choisissons de commencer par un nombre restreint de relations pour chacun des sous-domaines et qui sont définis dans le tableau 5.1 et le tableau 5.2.

En premier lieu, notre modèle va prendre en entrée un tuple d'OpenIE de la forme *(Sujet, Prédicat, Objet)* et va essayer de prédire la relation parmi les relations définies dans le tableau 5.1 qui pourrait lier les deux entités $Sujet$ et $Objet$. Nous considérons aussi la pseudo-relation $None$ pour indiquer qu'aucune des relations prédéfinies ne lie ces deux entités.
Pour le triplet *(Altran, va se fondre dans, Capgemini)*, ici le sujet correspond à *Altran*, l'objet correspond à *Capgemini* et le prédicat correspond au segment de texte *va se fondre dans* qui est censé exprimer la relation sémantique *Org :subsidiary*. Ainsi, notre approche consiste à normaliser les prédicats des triplets à une relation sémantique donnée.

On applique aussi une étape de reconnaissance d'entité au sujet et à l'objet des triplets. Contrairement aux approches antérieures, la reconnaissance d'entité nommée n'est pas essentielle à notre système d'extraction de relations, mais elle est toujours importante dans la mesure où elle permet de filtrer les relations erronées en se basant sur les contraintes sémantiques des relations.

Type de relation	Description
Org :Subsidiary_of	Organisations qui sont des filiales de l'organisation affectée (l'inverse de Org :Parents).
Org :Founded_by	La personne, l'organisation ou l'entité qui a fondé l'organisation attribué.
Per :CEO_of	La personne qui est le PDG de l'organisation attribuée.
Per :Member_or_employee_of	Les organisations ou entités (gouvernements) dont la personne affectée a été un employé ou un membre.
Per :Top_member_or_employee_of	Les personnes occupant des postes de direction de haut niveau dans l'organisation assignée. Tous les postes de membre supérieur/employé devraient impliquer un niveau d'autorité décisionnelle sur l'ensemble de l'organisation.
Org :Parents	Organisations dont l'organisation affectée est une filiale (l'inverse de Org :Subsidiary_of) .

TABLE 5.1 Tableau des relations pour le domaine de la structure d'entreprise.

Type de relation	Description
Org :Shareholders	Toute organisation, personne ou entité qui détient des actions (majoritaires ou non).
Org :Acquisition_of	La personne, l'organisation ou l'entité qui a acquis l'organisation attribuée.
Org :Partnership_with	L'organisation ou l'entité qui est en collaboration avec l'organisation attribuée.
Org :Competitor_of	L'organisation ou l'entité qui est en compétition avec l'organisation attribuée.
Org :Supplier_of	L'organisation ou l'entité qui est fournisseur de l'organisation attribuée.

TABLE 5.2 Tableau des relations pour le domaine d'activité d'entreprise.

La figure 17 illustre un exemple de reconnaissance d'entité nommée grâce au code la bibliothèque *NER Stanza*, qui comporte un logiciel de reconnaissance d'entités nommés.

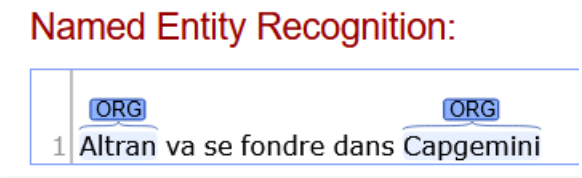


FIGURE 17 Exemple de reconnaissance d'entités nommées dans la plate-forme de démonstration de Stanza. Les termes *'Altran'* et *'Capgemini'* sont identifiés comme étant des organisations. Techniquement le résultat fourni prend la forme d'une table associant des segments des textes (chaînes de caractères et positions) à leurs types.

Par exemple, pour le triplet *(Altran, va se fondre dans, Capgemini)*, nous supposons que le modèle de reconnaissance d'entité nommée associe le type *Organisation* aux entités *Altran* et *Capgemini*. Si par ailleurs, le système d'extraction de relation sémantique infère la relation *Org :Ceo* alors nous pouvons annuler la prédiction à cause de violation de contrainte sémantique. En effet, la relation *Org :Ceo* exige que l'entité sujet ait comme type *Personne* et non *Organisation* comme c'est le cas avec l'entité *Altran*.

À chaque type de relation, sont associés des couples de contraintes sémantiques consistant en deux type d'entités. Par exemple, les contraintes de type posées sur ces relations sémantiques sont :

— *Org :Ceo* : *Personne* et *Organisation*
— *Org :Member_or_employee_of* : *Personne* et *Organisation*
— *Org :Top_member_or_employee_of* : *Personne* et *Organisation*
— *Org :Parents* : *Organisation* et *Organisation*

Nous nous basons donc sur l'identification du NER de Stanza pour vérifier les contraintes et invalider certaines relations. Nous pouvons aussi exploiter la ressource *JDM* si jamais l'entité en question existe dans la base de connaissances.

5.3 Méthode proposée

Notre méthode consiste en l'enchaînement de phase de Prétraitement, d'Inférence et de Normalisation. Le Prétraitement permet de filtrer les relations candidate douteuses qui ont une chance importante de ne pas porter une des relations sémantiques ciblées. Les inférences permettent soit d'ajouter de nouveaux triplets manqués par le système *UD2OIE*, soit d'ajouter des triplets plus complexes que le système *UD2OIE* ne peut pas extraire. Ainsi cette étape peut grandement influer les performances du système d'extraction de relations sémantiques en augmentant les performances de rappel. La normalisation permet de passer de segment textuel tel qu'extrait depuis le document à une des relations cibles.
Le pseudo algorithme 18 de notre approche générale est présenté ci-dessous.

```
def Jeux2BERT+OIE(phrases):
  for p in phrases:
    triplets = UD2OIE(p, Parse_tree(p))
    # exemple triplet
    # (S: Capgemini, P:rachète, O:Altran)

    # Prétraitements
    triplets_processed = Prétraitements(triplets)

    # Inférence avec les appositions et résolution des coréférences
    triplets_augmented = Inference(triplets_processed)

    for t in triplets_augmented
      # Reconnaissance d'entités nommés
      Subject_type,Object_type = NER([t["Subject"],t["Predicate"]])

      # Normalisation
      inferred_relations = list()
      for r in relations:
        for p in patrons[r]:
          # la variable score contient une valeure pour
          # les labels "0" et "1"
          score= Jeux2BERT.Link_Prediction(t["Predicate"],"r_associated",p)

          # exemples de la normalisation
          # Jeux2BERT.Link_Prediction("rachète","r_associated","acquière")
          # Jeux2BERT.Link_Prediction("rachète","r_associated","absorbe")

          # vérifier que les types des entités sujet et objet respectent
```

```
30            # les contraintes de types posés sur la relation inférée r
31            if(score[1]>3 and
32              constraint_satisfaction(r, Subject_type,Object_type)
33            ):
34
35              inferred_relations.append({r:score[1]})
36              break
37
38          if(len(inferred_relations)>0):
39
40            # Si plusieurs relations ont été trouvées alors
41            # on choisit celle avec la valeur max
42
43            relation = max(inferred_relations, key=inferred_relations.get)
44            # exemple de la relation Org:Acquisition
45
46            print(t,": ",relation)
47            # (S: Capgemini, P:rachète, O:Altran) Org:Acquisition
48
```

FIGURE 18 Approche générale pour l'extraction de relations sémantiques.

Avant de détailler notre approche, nous présentons deux figures explicatives du fonctionnement de la méthode d'extraction. La figure 19 résume notre approche générale sur un exemple simple qui n'inclut aucun traitement d'inférence ou de résolution de coréférence. Quant à la figure 20, elle illustre un exemple plus complexe nécessitant une étape de résolution de coréférence dans la chaîne de traitement.

5.3.1 Prétraitement

Comme mentionné précédemment, nous proposons un système d'extraction de relations qui fonctionne en deux temps selon les niveaux stylistiques du langage. Le modèle prend en entrée les triplets d'OpenIE extrait par *UD2OIE* depuis des textes. Ces triplets sont censés respecter le critère d'adaptation de domaine autant que possible, sachant que *UD2OIE* exploite seulement l'analyse syntaxique. Après cela, nous essayons de tirer profit de l'interprétabilité des triplets OpenIE pour introduire une étape de Prétraitement.
Cette étape intervient justement pour apprécier la certitude des faits exprimés par les triplets. Plus précisément nous essayons de cibler les verbes légers pour déterminer la certitude des faits exprimés par les triplets.

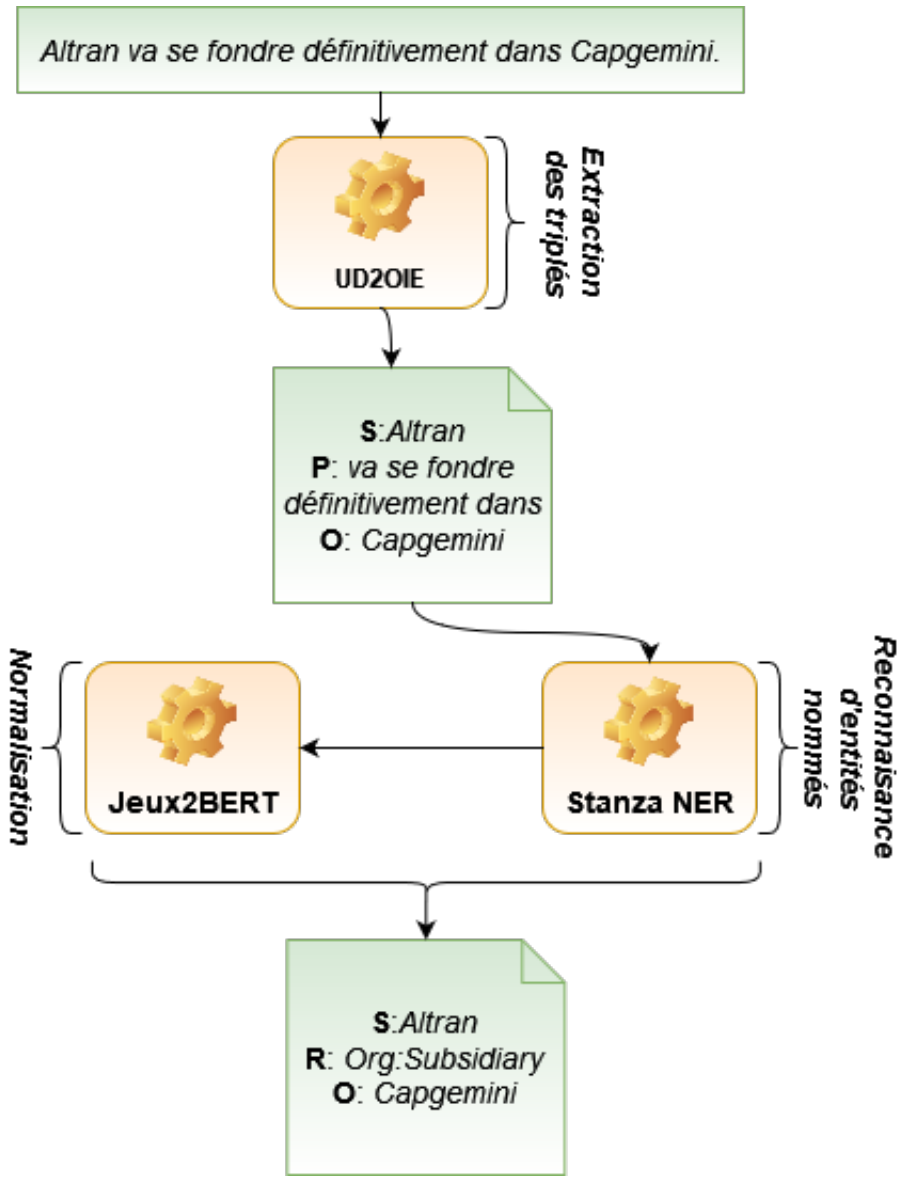


FIGURE 19 Fonctionnement simplifié de notre système d'extraction de relations sémantiques.

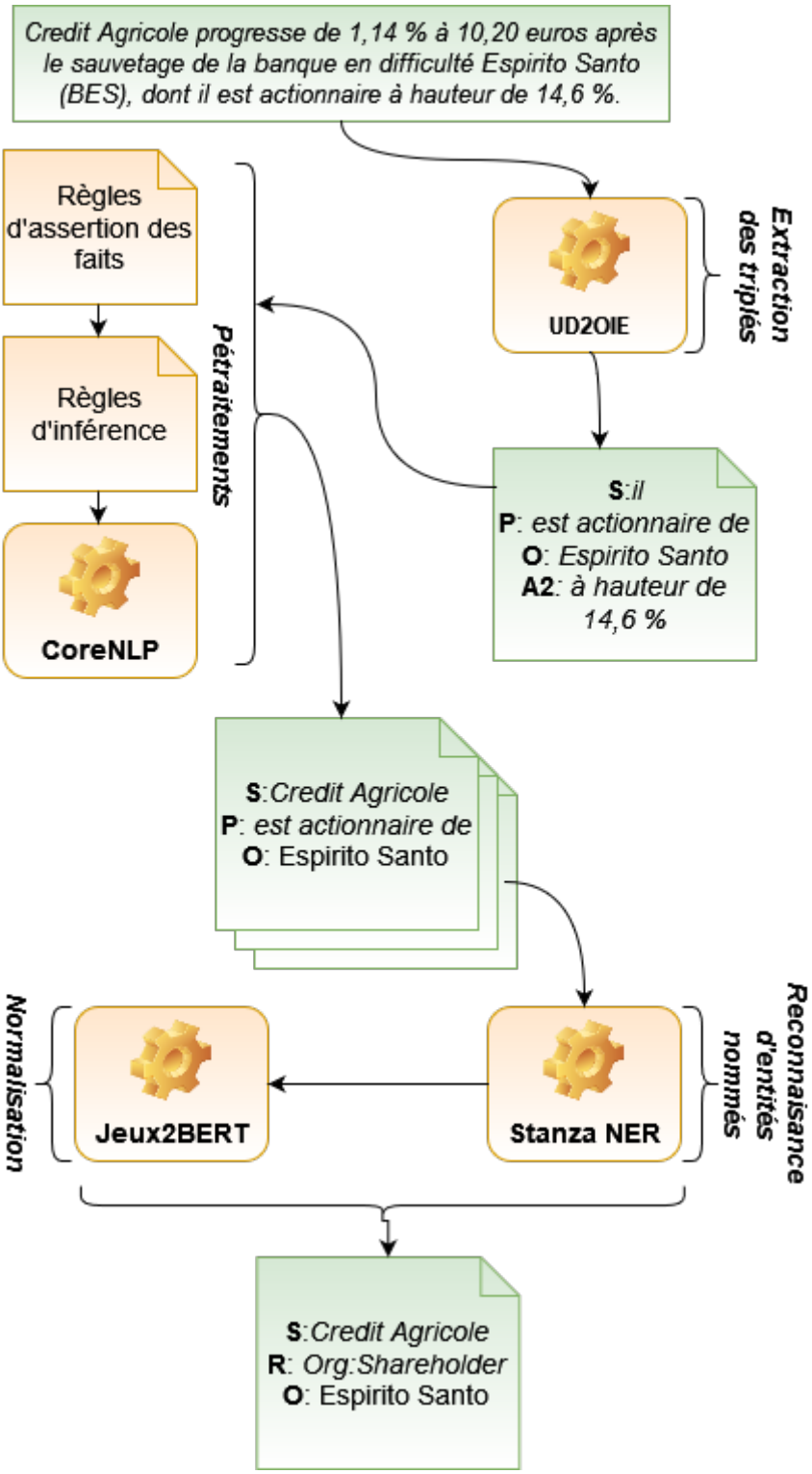


FIGURE 20 Fonctionnement de notre système d'extraction de relations sémantiques. L'exemple inclut la résolution de coréférence dans la chaîne de traitement.

Contrairement aux constructions verbales « prototypiques » où le verbe est le gouverneur syntaxique et sémantique de la phrase et ses dépendances syntaxiques sont aussi ses arguments sémantiques, tandis que dans les verbes légers ne portent que l'inflexion et ne projettent aucune structure argumentative, ils agissent en tant que gouverneur du verbe « prototypique ».

Pour cela, nous vérifions la présence de certains termes ou expressions dans le prédicat du triplet afin de décider de l'assertion de l'évènement exprimé par le prédicat. Les termes considérés sont : *ancien, ex, précédent, annuler, renoncer, résigner, résilier, rompre, abandonner, mettre fin, arrêter, rejeter, démentir, dénier, nier, réfuter, rétracter, ...* Par exemple, supposons le segment "Steve Jobs, l'ancien CEO d'Apple...", le triplet correspondant ne sera pas conservé pour la normalisation à cause de la présence de "ancien". En effet, nous nous focalisons sur des informations qui sont pertinentes au moment de leur extraction. Pour résumer, dans ce cas-là, ça ne nous intéresse pas de savoir que dans le passé Steve Jobs a été CEO de Apple.
Nous essayons d'illustrer cette notion à l'aide des phrases suivantes 21 :

— Le gouvernement Canadien renonce à l'acquisition des nouveaux hélicoptères EH-101 pour des raisons purement politiques.
— Annulation de l'achat des sous-marins français par l'Australie.

FIGURE 21 Exemple de phrases contenant un faux signal de relations sémantiques.

Pour la première phrase, nous aurons le triplet *(**A0 :** Le gouvernement Canadien ; **P :**renonce à l'acquisition de ; **A1 :** les nouveaux hélicoptères EH-101)* extrait par *UD2OIE*. Nous nous apercevons que le mot *acquisition* qui est un mot clé exprimant une relation d'acquisition est aussi gouverné par le verbe léger *renonce*.
Ainsi, la relation *(**Sujet :** Le gouvernement Canadien ; **Org :Acquisition_of** ; **Objet :** hélicoptères EH-101)* est fausse et ne représente pas le sens exprimé par la phrase.

Quant à la deuxième phrase, nous avons le triplet suivant *(**A0 :** Australie ; **P :**Annulation de l'achat de ; **A1 :** les sous-marins français)* extrait par *UD2OIE* où le prédicat est un nominal. Nous nous apercevons aussi que le nominal *achat* qui est un mot clé exprimant une relation d'acquisition est aussi gouverné par le

nominale *annulation*. Ainsi, la relation *(**Sujet :** Australie ; **Org :Acquisition_of** ; **Objet :** sous-marins français)* est fausse et ne représente pas le sens exprimé par la phrase.

Contrairement aux approches à base d'apprentissage ou d'autres approches, les triplets d'OpenIE donnent en sortie les mots clés déclencheurs de la relation et leurs prédicats et gèrent et leur négation et leur certitude.

5.3.2 Inférences

Ensuite, nous appliquons également une étape d'inférences sur les triplets extraits par *UD2OIE* en ciblant les appositions et les coréférences.
Ces inférences ont pour but d'augmenter la couverture de notre système en proposant plus de relations que ce qui serait directement accessible dans les textes.

Nous commençons par illustrer un premier exemple d'inférence en utilisant les **appositions**. Étant donné un premier triplet $t_1 = (s_1, p_1, o_1)$ et un second triplet $t_2 = (s_2, p_2, o_2)$, si le prédicat p_1 du premier triplet correspond à l'auxiliaire *être* et que l'objet du premier triplet o_1 est identique au sujet du deuxième triplet s_2 alors on génère le triplet $t_3 = (s_1, p_2, o_2)$.
Le pseudo-algorithme 22 de ce type d'inférence est présenté ci-dessous :

```
def Inference_apposition(extractions):
  ##extraction : dict(phrase ==> triplets)
  for phrase,triplets in extractions.items():
    for i in range(len(triplets)):
      if(triplet[i]["Predicate"]=="est"):
        for j in range(len(triplets)):
          if(triplet[j]["Subject"]==triplet[i]["Object"]
            and i!=j):
            new_triplet = copy.deepcopy(triplet[j])
            new_triplet["Subject"] = triplet[i]["Subject"]
            triplets.append(new_triplet)

    extractions.update(phrase:triplets})
```

FIGURE 22 Algorithme de l'inférence avec appositions.

Considérons la phrase 23 qui illustre un exemple d'inférence avec les apposi-

« Plus tôt cette année , Blackstone Group , une banque d'investissement de New York , n'a eu aucun mal à vendre une fiducie spéciale de titres hypothécaires de 570 millions de dollars qu'elle a créé pour les investisseurs japonais. »,

FIGURE 23 Exemple d'une phrase liée à l'inférence avec apposition.

tions :

et les triplets extraits par *UD2OIE* :

— *(**A0 :** Blackstone Group ; **P :** [est] ; **A1 :**une banque d' investissement de New York)*
— *(**A0 :** une banque d'investissement de New York ; **P :** n'a eu aucun mal à vendre ; **A1 :** une fiducie spéciale de titres hypothécaires de 570 millions de dollars)*

Comme le premier triplet est exprimé au moyen d'une apposition et que l'objet du premier triplet et sujet du deuxième triple sont similaires, nous pouvons inférer un troisième triplet à partir du deuxième triplet de la manière suivante :

*(**A0 :** Blackstone Group ; **P :** n'a eu aucun mal à vendre ; **A1 :** une fiducie spéciale de titres hypothécaires de 570 millions de dollars)*

L'autre moyen d'appliquer des inférences pour générer des triplets plus informatifs passe par les **coréférences**. Pour cela, nous utilisons le système de résolution de coréférence de *CoreNLP* de la bibliothèque *Stanza*.
La figure 24 illustre un exemple de résolution de coréférence de la bibliothèque *AllenNLP*.

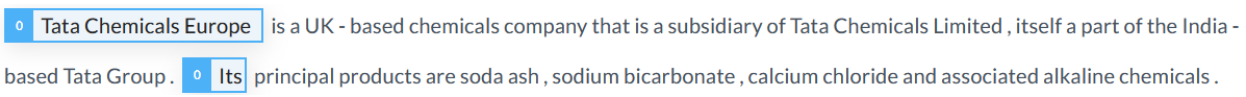


FIGURE 24 Exemple de co-référence dans la plateforme de démonstration AllenNLP.

Étant donné un triplet de la forme $t_1 = (s_1, p_1, o_1)$ et si le système de résolution de coréférences donne en sortie le couple $r_1 = (c_0, c_1)$ alors si le sujet du triplet s_1 est similaire à l'entité c_0 pour laquelle a été associée sa référence c_1 on génère le triplet suivant : $t_2 = (c_1, p_1, o_1)$. Sinon si c'est l'objet du triplet o_1 qui est simi-

laire à l'entité c_0 pour laquelle a été associée sa référence c_1 on génère le triplet suivant : $t_2 = (s_1, p_1, c_1)$. Afin d'éviter l'inférence de triplets erronés suite à une propagation d'erreurs du système de résolution de coréférence, nous avons seulement géré le cas où l'entité mentionnant la référence est un pronom personnel ou un pronom possessif tel que *il* ou lui.
Le pseudo-algorithme de ce type d'inférence 25 est présenté ci-dessous :

```
def Inference_coreference(extractions, corefs):
  ##extraction : dict(phrase ==> triplets)
  ##corefs : dict (phrase ==> coreferences)
  ##coreferences : list((reference,entity))

  anaph = ("nous","il","elles","ils","eux","lui","son","elle",
  "leur","eux-memes","lui-même","moi","tu","je", "lui-même","elle-même")

  for phrase,triplets in extractions.items():
    for i in range(len(triplets)-1):
      if(phrase in corefs.keys()):
        if(triplet[j]["Subject"]==corefs[phrase][0]
          and corefs[phrase][0] in anaph):
            new_triplet = copy.deepcopy(triplet[j])
            new_triplet["Subject"] = corefs[phrase][1]
            triplets.append(new_triplet)

        if(triplet[j]["Object"]==corefs[phrase][0]
          and corefs[phrase][0] in anaph):
            new_triplet = copy.deepcopy(triplet[j])
            new_triplet["Object"] = corefs[phrase][1]
            triplets.append(new_triplet)

    extractions.update({phrase:triplets})
```

FIGURE 25 Algorithme d'inférence avec les coréférences.

Considérons la phrase 26 qui illustre un exemple d'inférence liée à une résolution de coréférence :

« Google et Facebook en embuscade face à Apple, seul Google lui tient un peu tête. »,

FIGURE 26 Exemple de phrase liée à une résolution coréférence.

et le triplet extrait par *UD2OIE* :

— *(**A0 :** Google ; **P :** tient un peu tête ; **A1 :**lui)*

Nous pouvons voir que le pronom *lui* fait référence à l'entité *Apple* (dans notre méthode nous utilisons le système de résolution de coréférence de *CoreNLP*). En le remplaçant par l'entité qu'il référence, nous obtenons le triplet suivant :
*(**A0 :** Google ; **P :** tient un peu tête ; **A1 :**Apple)*

Par ailleurs, nous ciblons certains mots clés (*tel que, comme, inclus, comprend*) pour lancer un mécanisme d'inférence. Si nous avons un triplet de la forme $t_1 = (s_1, p_1, o_1)$ et un autre triplet $t_2 = (s_2, p_2, o_2)$ et que son prédicat p_2 est similaire à un des mots clés suivants (*tel que, comme, inclus*) et si les deux sujets s_1 et s_2 des triplets t_1 et t_2 sont identiques alors on infère le triplet $t_3 = (s_2, p_1, o_1)$. Sinon si c'est l'objet o_2 du deuxième triplet t_2 qui est identique au sujet s_1 du premier triplet t_1 alors infère le triplet $t_3 = (o_2, p_1, o_1)$.
Le pseudo algorithme pour ce type d'inférence 27 est présenté ci-dessous.

```
def Inference_apposition2(extractions):
##extraction : dict(phrase ==> triplets)
    is_a = ("tel que", "tels que", "telle que", "telles que", "comme", "
    inclut", "incluent", "comprend")
    for phrase,triplets in extractions.items():
        for i in range(len(triplets)):
            if(triplet[i]["Predicate"] in is_a):
                for j in range(len(triplets)):
                    if (triplet[j]["Subject"]==triplet[i]["Object"]
                    and i!=j):
                        new_triplet = copy.deepcopy(triplet[j])
                        new_triplet["Subject"] = triplet[i]["Subject"]
                        triplets.append(new_triplets)

                    elif (triplet[j]["Subject"]==triplet[i]["Subject"])
                    and i!=j):
                        new_triplet = copy.deepcopy(triplet[j])
                        new_triplet["Subject"] = triplet[i]["Object"]
                        triplets.append(new_triplets)

                    elif (triplet[j]["Object"]==triplet[i]["Object"])
                    and i!=j):
                        new_triplet = copy.deepcopy(triplet[j])
                        new_triplet["Object"] = triplet[i]["Subject"]
```

```
                        triplets.append(new_triplets)

                elif (triplet[j]["Object"]==triplet[i]["Subject"])
                and i!=j):
                    new_triplet = copy.deepcopy(triplet[j])
                    new_triplet["Object"] = triplet[i]["Object"]
                    triplets.append(new_triplets)

        extractions.update({phrase:triplets})
```

FIGURE 27 Algorithme d'inférence avec les synonymes.

Considérons l'exemple 28 avec la phrase et triplets suivants :

« De nombreuses marques de mode, comme Calvin Klein, Armani, Versace, Burberry, Ferragamo, Valentino, Hilfiger, ou Nautica lancent des collections de montres saisonnières et signent des licences avec des groupes horlogers tels que Swatch, Fossil, Movado ou Timex .»

FIGURE 28 Exemple de phrases liée à l'inférence par synonymes

— *(**A0 :** De nombreuses marques de mode ; **P :** signent des licences avec ; **A1 :**des groupes horlogers)*
— *(**A0 :** De nombreuses marques de mode ; **P :** comme ; **A1 :**Calvin Klein)*
— *(**A0 :** des groupes horlogers ; **P :** tels que ; **A1 :**Swatch)*

En exploitant les deux premiers triplets, nous pouvons inférer le triplet suivant :
— *(**A0 :** Calvin Klein ; **P :** signent des licences avec ; **A1 :**des groupes horlogers)*

Ensuite en utilisant ce triplet que nous venons d'inférer avec le troisième triplet, nous arrivons à obtenir le triplet suivant :
— *(**A0 :** Calvin Klein ; **P :** signent des licences avec ; **A1 :**Swatch)*

Un autre aspect pris en compte par *UD2OIE* est le contexte, c'est-à-dire la source qui appuie le fait exprimé par le triplet. Pour cela, le système *UD2OIE* construit des relations imbriquées par inférence et qui permettent de déterminer le contexte du fait exprimé par le triplet. Si nous avons un triplet $t_1 = (s_1, p_1, o_1)$ et un autre triplet $t_2 = (s_2, p_2, o_2)$ et que le syntagme du triplet $t_1 = (s_1 + p_1 + o_1)$ correspondant à la concaténation du sujet avec le prédicat et le triple est une sous-chaîne du sujet s_2 du triplet t_2 alors nous déduisons que le triplet t_1 est imbriqué

dans le triplet t_2. Considérons la phrase 29 :

« Selon Bloomberg, Facebook a annoncé son intention d'acquérir CTRL-labs, une startup créée il y a quatre ans et considérée comme une pionnière dans le domaine émergant du brain computing, pour un montant non divulgué. »,

FIGURE 29 Phrase comprenant des tuples d'OpenIE imbriqués

et les triplets extraits par *UD2OIE* :

— *(**A0 :** Facebook ; **P :** a annoncé son intention d'acquérir ; **A1 :** CTRL-labs)*
— *(**A0 :** Facebook a annoncé son intention d'acquérir CTRL-labs ; **P :** Selon ; **A1 :** Bloomberg)*

On peut voir que le premier triplet représente une relation d'acquisition entre *Facebook* et *CTRL-labs* à cause de la présence du mot déclencheur *acquérir*.
Nous nous apercevons aussi que ce triplet même est imbriqué dans le deuxième triplet, c'est-à-dire qu'il est inclus dans le sujet du deuxième triplet. Ainsi nous pouvons dire que la relation *(**Sujet :** Facebook ; **Org :Acquisition_of** ; **Objet :** CTRL-labs)* a comme source *Bloomberg* et peut être exprimé ainsi *(**Sujet :** Facebook ; **Org :Acquisition_of** ; **Objet :** CTRL-labs ; **Context :** Bloomberg)*.
Le pseudo-algorithme de la capture du contexte 30 d'un triplet est présenté ci-dessous.

```
def Contextualisation(extractions):
##extraction : dict(phrase ==> triplets)
    for phrase,triplets in extractions.items():
        for i in range(len(triplets)-1):
            for j in range(i+1,len(triplets)):
                if(i!=j):
                    if (
                        (
                         triplet[j]["Subject"]
                         + triplet[j]["Predicate"]
                         + triplet[j]["Object"]
                        )
                        in triplet[i]["Subject"]
                       ):
                        triplet[j]["Context"] = triplet[i]["Object"]
                        triplets.remove(j)
                        triplets.append(triplet[j])
```

```
18
19
20                      elif (
21                          (
22                           triplet[j]["Subject"]
23                           + triplet[j]["Predicate"]
24                           + triplet[j]["Object"]
25                          )
26                          in triplet[i]["Object"]
27                         ):
28                          triplet[j]["Context"] = triplet[i]["Suject"]
29                          triplets.remove(j)
30                          triplets.append(triplet[j])
31
32
33                      elif (
34                          (
35                           triplet[i]["Subject"]
36                           + triplet[i]["Predicate"]
37                           + triplet[i]["Object"]
38                          )
39                          in triplet[j]["Subject"]
40                         ):
41                          triplet[i]["Context"] = triplet[j]["Object"]
42                          triplets.remove(i)
43                          triplets.append(triplet[i])
44
45
46                      elif (
47                          (
48                           triplet[i]["Subject"]
49                           + triplet[i]["Predicate"]
50                           + triplet[i]["Object"]
51                          )
52                          in triplet[j]["Object"]
53                         ):
54                          triplet[i]["Context"] = triplet[j]["Suject"]
55                          triplets.remove(i)
56                          triplets.append(triplet[i])
57
58
59          extractions.update({phrase:triplets})
```

FIGURE 30 Algorithme de détection du contexte.

Nous avons seulement ajouté la capture du contexte dans notre chaîne de trai-

tement et qui correspond à la catégorie de *la perspective* rapportée dans Jean et al. (2017), nous pouvons aussi penser à inclure la catégorie du *niveau de la certitude* et la catégorie du *temps* afin d'évaluer la certitude des triplets.
À noter que contrairement à la première proposition, ce prétraitement n'est pas utilisé pour filtrer les triplets mais est inclus dans le traitement pour enrichir l'information et contextualiser l'assertion.

5.3.3 Normalisation

L'étape qui suit l'inférence consiste à normaliser les prédicats des triplets vers l'une des relations cibles. À partir de la phrase relationnelle qui correspond au prédicat d'un triplet donné, nous devons d'abord trouver quel type de relation sémantique est exprimé par le prédicat. Par exemple, les prédicats *acquiert* et *rachète* tous deux sont sensés exprimer une relation d'acquisition. Pour cela, nous devons identifier les synonymes des phrases relationnelles (les prédicats) étant donné qu'un type de relation sémantique peut être exprimé de plusieurs manières (plusieurs variantes). Pour un prédicat donné, nous devons trouver le type de relation lui correspondant le plus. C'est-à-dire un ou plusieurs mots clés qui sont aussi proche que possible du prédicat. Pour cela, nous adoptons une approche par similarité cosinus entre le vecteur calculé pour le prédicat et les vecteurs des mots clés associés à un type de relation.

Notre approche peut être réduite à une approche à base de règles améliorées du fait que les patrons sémantiques sont appliqués à des triplets et non à des segments de phrases entiers. Avant d'en présenter les détails, nous donnons d'abord une vue d'ensemble de la limite des approches basées sur l'hypothèse distributionnelle et les raisons qui ont motivé notre choix de concevoir un système tel qu'il est présenté dans ce chapitre.

L'hypothèse distributionnelle, qui stipule que les mots qui apparaissent dans des contextes similaires ont tendance à avoir des significations similaires, a inspiré plusieurs algorithmes d'exploration du Web pour paraphraser des phrases sémantiquement équivalentes. Cependant, l'hypothèse distributionnelle présente plusieurs inconvénients. Premièrement, elle peut confondre les antonymes et les synonymes, car les expressions antonymes apparaissent dans des contextes simi-

laires à ceux des expressions synonymes. Pour les mêmes raisons, il confond aussi souvent les causes avec les effets. Deuxièmement, l'hypothèse de distribution repose sur des statistiques sur de grands corpus pour produire des statistiques de similarité précises. Il reste difficile de savoir comment paraphraser avec précision les relations moins fréquentes avec l'hypothèse distributionnelle.

Il est à noter aussi que les vecteurs de mots souffrent de ces limites étant donné qu'ils suivent un principe similaire : *You shall know a word by the company it keeps*. Un autre constat est que à quelques exceptions près, la plupart des approches de vecteurs de mots favorisent la similarité de domaine *(pâtissier :**gâteaux**)* au détriment de la similarité fonctionnelle *(pâtissier :**boulanger**)*.

Pour ces raisons, nous avons exploité le réseau lexico-sémantique *JeuxDeMots* pour affiner les représentations vectorielles des mots et discriminer les propriétés relationnelles des mots. Ainsi le modèle *Jeux2BERT* présenté au chapitre 4 a l'avantage d'exploiter les propriétés distributionnelles et relationnelles des mots et permet d'alléger ces problématiques à un certain degré.

Dans notre système d'extraction de relations, *Jeux2BERT* agit en tant qu'intermédiaire pour passer du paradigme de l'OpenIE vers le paradigme d'extraction de relations sémantiques en calculant un score de similarité entre le prédicat du triplet d'OpenIE et les mots clés d'une relation donnée. Pour cela, nous devons d'abord associer manuellement un petit ensemble de mots clés pour chaque relation, cet ensemble n'a pas besoin d'être exhaustif.

Le tableau 5.3 illustre les mots clés les plus représentatifs de chaque relation.

TABLE 5.3 Tableau des relations et leurs mots clés.

Relation	Mots clés
Org :Subsidiary_of	filiale de unité de division de branche de subordonné de appartient détenue par intègre fait partie de
Org :Acquisition_of	fait l'acquisition de rachète absorbe achète acquière
Org :Shareholders	investit dans détient des actions de prend le contrôle actionnaire de subventionne alloue des fonds entre au capital lève finance
Org :Founded_by	fondée par crée par à l'origine de lancée par
	suite page suivante

TABLE 5.3 – suite de la page précédente

Relation	Mots clés
Per :CEO_of	exécutif de leader de pdg de directeur de patron de chef de
Per :Member_or_employee_of	employé de membre de affilié chez collaborateur chez
Per :Top_member_or_employee_of	associé chez responsable chez représentant chez délégué chez
Org :Partnership_with	partenaire de en partenariat avec en coopération avec associée de fusionne avec signe un accord avec
Org :Competitor_of	rival de concurrent de en concurrence avec défie dans le viseur de
Org :Supplier_of	fournisseur de constructeur fabricant
suite page suivante	

« Mme Marie-Anne BARBAT-LAYANI fut nommée en avril 2010, directeur adjoint de le cabinet de le Premier ministre, M. François FILLON. »,

FIGURE 31 Exemple de normalisation

TABLE 5.3 – suite de la page précédente

Relation	Mots clés
Org :Parent_of	parent de société mère de une joint-venture de englobe propriétaire

Plus formellement, afin d'évaluer la correspondance entre le prédicat du triplet et les mots clés d'une relation donnée, nous construisons le triplet suivant $(h, r_associated, t)$, où h est le prédicat du triplet et t est un des mots clés de la relation. Par exemple, considérons la phrase 31 :

pour lesquels le système *UD2OIE* extrait les tuples OpenIE suivants :

— *(**A0 :** Marie-Anne BARBAT-LAYANI ; **P :** fut nommée ; **A1 :** directeur adjoint de le cabinet de le Premier ministre ; **A2 :** en avril 2010)*
— *(**A0 :** Marie-Anne BARBAT-LAYANI ; **P :** directeur adjoint de ; **A1 :** le cabinet de le Premier ministre)*
— *(**A0 :** Marie-Anne BARBAT-LAYANI ; **P :** directeur adjoint de ; **A1 :** M. François FILLON)*
— *(**A0 :** le Premier ministre ; **P :** [est] ; **A1 :** M. François FILLON)*

Étant donnée une relation R et son ensemble de patrons sémantiques $X = \{x_1, x_2, ..., x_n\}$ et le triplet $t = (s, p, t)$, nous essayons de normaliser le prédicat p du triplet t vers la relation R en comparant ce même prédicat p avec chacun des patrons sémantiques x_i de la relation R.
Ainsi, nous construisons une entrée pour le modèle *Jeux2BERT* pour chaque patron sémantique de la relation et qui est de la forme *(p, r_associated , x_i)* où p est le prédicat du triplet t et x_i un des patrons sémantiques de la relation R.
Par exemple, afin de normaliser le triplet :
*(**A0 :** Marie-Anne BARBAT-LAYANI ; **P :** directeur adjoint de ; **A1 :** M. François*

FILLON),
nous construisons l'ensemble de triplets pour chaque mots clés d'une relation :
— *(directeur adjoint, r_associated , associé)*
— *(directeur adjoint, r_associated , responsable)*
— *(directeur adjoint, r_associated , représentant)*
— *(directeur adjoint, r_associated , délégué)*

Ainsi, nous injectons les triplets comme entrée au modèle *Jeux2BERT* pour prédire leur plausibilité. Nous reconstruisons un ensemble de triplets pour chaque relation. On choisit l'ensemble de relations ayant au moins un triplet prédit en tant que plausible (vrai).
Nous filtrons ensuite les triplets violant les contraintes sémantiques associées aux relations. Par exemple, si notre modèle infère que la relation sémantique qui lie le couple d'entité (s, o) est la relation de *Org :CEO* et que le modèle de reconnaissance d'entité nommée associe au couple (s, o) le type *Organisation* alors il est fort probable que la classification de la relation est erronée puisque la relation *Org :CEO* implique que l'entité sujet s soit de type *Personne* et non *Organisation*.

Si l'ensemble de triplets restant appartient à plusieurs relations distinctes, et afin de déterminer la relation liant les deux entités, nous choisissons la relation avec le triplet ayant le plus grand score d'inférence.
Dans le cas où notre système d'extraction de relations sémantiques classe deux relations distinctes R_1 et R_2 comme étant valides, alors nous considérons les scores d'inférence y_1 et y_2 de R_1 et R_2 respectivement. Nous sélectionnons alors la relation donc le score d'inférence est le plus grand.
Le système *Jeux2BERT* infère des scores avec une plage de valeurs allant de $-\infty$ jusqu'a $+\infty$ pour chacune des classes lors de l'évaluation de la plausibilité d'un triplet. Pour un triplet donné, le système va produire un couple de score (x, y) où x représente le score de la classe négative et y représente le score de la classe positive.
Ainsi si le score x est supérieur à y alors le triplet sera considéré invalide (*faux*) sinon le triplet sera considéré comme valide (*vrai*). Puisque le système produit des scores en valeur réelle, nous fixons un seuil minimal supérieur à 3 pour la classe positive afin que le triplet soit considéré valide.
Comme nous avons choisi de suivre le paradigme de monde ouvert, nous gérons

aussi l'incertitude du modèle en définissant une contrainte définie telle que : si la valeur absolue de la différence arithmétique entre le score des deux classes ($\mid x - y \mid$) est inférieure à 2 où x et y sont les scores des classes positive et négatives, alors nous considérons que le modèle *ne sais pas* et nous n'associons aucune classe au triplé. Par exemple, pour les couples de score $(2, 0.5)$ et $(-1, 1)$, le modèle *Jeux2BERT* ne va pas produire de prédiction et nous considérons tout simplement qu'il ne sait pas.
Par exemple, pour la phrase : *Dan Cupkovic, gestionnaire du fonds négocié en bourse Amplify BlackSwan Growth and Treasury Core*, si le modèle infère que pour le couple d'entité *(Dan Cupkovic, Amplify BlackSwan Growth and Treasury Core)* les deux relations *Org :Top_Member_or_employee_of* et *Org :Member_or_employee_of* sont pertinentes et valides avec respectivement un score de $(-9.2, 10)$ et $(-7, 8)$ alors nous choisissons de garder la relation sémantique *Org :Top_Member_or_employee_of* et qui a le score d'inférence le plus élevé.
Pour finir, nous considérons qu'il n'existe pas de relation sémantique si tous les scores d'inférences sont inférieures à un seuil de confiance (égal à 3) fixé empiriquement pour décider de la validité de l'inférence.

5.4 Expérimentations et Analyse

Dans cette section, nous commençons par présenter les données d'expérimentation et les moyens utilisés pour les obtenir. Ensuite, nous détaillons les résultats des expérimentations menées. Finalement, nous concluons par l'analyse des erreurs produites par notre système.

5.4.1 Données d'évaluation et protocole

Mis à part le corpus français *BizRel* (Khaldi et al. 2020) regroupant un ensemble de relations du domaine de l'économie, nous avons aucune connaissance d'autre corpus français spécifique au domaine de l'intelligence d'entreprise. Comme aucun référentiel d'évaluation englobant notre ensemble de relations n'existe pour le français et que nous n'avons pas pu accéder au corpus *BizRel*, nous avons dû nous même manuellement constituer un petit référentiel d'évaluation de 301 phrases à partir du web. Le tableau 5.4 ci-dessous illustre les détails du corpus d'évaluation et nous incluons aussi le lien vers le document : corpus url.

Type de la relation	Nombre de relations
Org :Subsidiary_of	35
Org :Shareholders	26
Org :Acquisition_of	51
Org :Founded_by	42
Per :CEO_of	22
Per :Member_or_employee_of	19
Per :Top_member_or_employee_of	40
Org :Partnership_with	18
Org :Competitor_of	13
Org :Supplier_of	15
Org :Parents	20

TABLE 5.4 Tableau des types des relations et leurs descriptions.

Le tableau 5.5 illustre chaque type de relation avec un exemple de ce type de relation.

TABLE 5.5 Tableau des types des relations et leurs exemples.

Type de la relation	**Exemple de relations**
Org :Shareholders	Capgemini détient désormais 98% du capital d'Altran.
Org :Acquisition_of	Plastic Omnium a pris les pleins pouvoirs chez Inergy.
Org :Founded_by	Richie a travaillé avec Dave Elliott et Garry Leach pour relancer Atomeka Press en 2004.
Per :CEO_of	Eric Patrux, le patron d'Iorga, une SSII qui côtoie moult PME et ETI.
Per :Member_or_employee_of	Chez Tata Motors, Debasis Ray, porte-parole, revendique aussi l'indépendance des décisions.
Per :Top_member_or_employee_of	Gilles Vermot-Desroches directeur du développement durable de Schneider Electric.

suite page suivante

TABLE 5.5 – suite de la page précédente

Type de la relation	Exemple de relations
Org :Partnership_with	En 2006, Hermès qui collaborait déjà avec Audemars Piguet ou Vacheron Constantin, s'invite au capital de Vaucher.
Org :Competitor_of	Comment Apple a perdu face à Qualcomm ?
Org :Supplier_of	Landbot, basé à Barcelone, un constructeur de chatbot "sans code", a levé une série A de 8 millions de dollars dirigée par Swanlaab.
Org :Parents	Meta, la société mère de Facebook, a perdu plus d'un quart de sa valeur.

Comme mentionné précédemment, l'inexistence de corpus d'évaluation pour le français nous limite sévèrement dans notre expérimentation. Ainsi, nous ne pouvons pas comparer notre méthode à une méthode basée sur l'apprentissage par des exemples. Nous décidons de nous concentrer sur notre méthode et nous considérons d'autres approches dérivées de notre modèle afin d'apporter des éléments d'analyse. Afin de considérer plus de variantes pour lesquels nous comparons notre système principal, nous considérons une autre variante de notre modèle *Flaubert+OIE* dans lequel nous n'exploitons pas la base de connaissance *JeuxDeMots* et nous ne basons seulement sur une représentation basique de vecteurs de mots à travers le modèle *Flaubert*. Afin d'évaluer la correspondance entre le prédicat du triplet et les mots clés d'une relation donnée nous calculons simplement la distance *cosinus* des deux vecteurs.
Nous incluons aussi dans nos expérimentations l'adaptation au domaine en constituant un ensemble d'évaluation pour chacun des sous-domaines. Pour rappel, les deux sous-domaines sont la structure d'une entreprise et l'activité d'entreprise.

Finalement, nous décidons aussi d'ajouter une approche basée sur le paradigme *few-shot* étant donnée qu'une quantité de donnée d'apprentissage est nécessaire. Pour cela, nous décidons de collecter un petit ensemble d'exemples de support pour chaque relation, et nous suivons la stratégie : 11way-5shot qui spécifie que pour chacune des 11 relations nous lui associons 5 exemples d'apprentissage.
Pour le choix du modèle, nous décidons de prendre le modèle de réseaux prototy-

piques (Snell et al. 2017) comme système de référence. Cependant, nous différons de l'approche originale en exploitant *Flaubert* comme encodeur au lieu de *Glove*. Nous avons décidé de ne pas prendre le système *MTB* (Ye and Ling 2019) et qui est actuellement l'état de l'art à cause de la non-disponibilité de corpus en français pour l'étape de préapprentissage.

5.4.2 Résultats

Nous commençons cette section en analysant les résultats de l'évaluation qui sont rapportés dans le tableau 5.6. Nous rappelons les formules de *Précision*, *Rappel* et *F1*.

$$Précision = \frac{nombred'extractionscorrectementclassés}{nombretotald'extractionsdumodèle} \tag{5.1}$$

$$Rappel = \frac{nombred'extractionscorrectementclassés}{nombretotald'extractionsducorpus} \tag{5.2}$$

$$F1 = 2 \times \frac{Précision \times Rappel}{Précision + Rappel} \tag{5.3}$$

Système	Précision	Rappel	F1
Jeux2BERT+OIE	83.9	57.4	68.2
Flaubert+OIE	63.3	55.5	59.2
Réseaux prototypiques (11way-1shot)(Snell et al. 2017)	57.2	54.0	55.6
Réseaux prototypiques (11way-5shot)(Snell et al. 2017)	74.5	68.9	71.6

TABLE 5.6 Résultats de l'évaluation.

Dans notre modèle proposé, la première observation est que la performance de précision est élevée en comparaison à la performance de rappel qui peut être considérée comme moyenne. Aussi, la deuxième variante de notre modèle (*Flaubert+OIE*) présentée précédemment a une baisse de performance importante dans la métrique de précision.
Comme la variante *Flaubert+OIE* se base uniquement sur la version basique de *Flaubert* pour normaliser les triplets de *UD2OIE* et n'exploite pas les connaissances de *JeuxdeMots*, une autre observation est que le système *Jeux2BERT* améliore principalement les performances de précision. Ainsi, notre modèle a une

meilleure performance de précision (la confusion entre les relations est moindre) grâce au système *JeuxdeMots* qui arrive à démêler les propriétés relationnelles des mots. En effet, comme les relations de l'ensemble d'évaluation sont du même domaine et sont assez proche, il est moins facile de séparer les mots clés et les expressions discriminantes de chaque relation sans avoir des vecteurs de mots avec un espace de représentation assez affiné. Cette observation est aussi confirmée par les résultats du tableau 5.7, où le système proposé par Snell et al. (2017) a une meilleure performance de précision en utilisant *Jeux2BERT* comme encodeur au lieu de la version basique de *Flaubert*.

En comparant notre modèle aux deux variantes du système proposé par Snell et al. (2017), nous nous apercevons que notre modèle surpasse largement le modèle de réseaux prototypiques avec la stratégie 1shot mais a des performances inférieures en comparaison avec la stratégie 5shot. Nous déduisons que notre modèle se situe entre les deux stratégies 1shot et 5shot en termes de performance bien que la différence de performance avec la stratégie 5shot n'est pas importante, aussi la performance de précision est toujours meilleure.

Système	Précision	Rappel	F1
Réseaux prototypiques (11way-1shot)(Flaubert)	57.2	54.0	55.6
Réseaux prototypiques (11way-1shot)(Jeux2BERT)	64.0	58.3	61.0
Réseaux prototypiques (11way-5shot)(Flaubert)	74.5	68.9	71.6
Réseaux prototypiques (11way-5shot)(Jeux2BERT)	85.2	76.9	80.8

TABLE 5.7 Résultats de l'évaluation des variantes du modèle de réseaux prototypiques (Snell et al. 2017) .

Les résultats du tableau 5.7 confirment les conclusions tirées des résultats du tableau 5.6, et qui indique un gain de performance important apporté par le système *Jeux2BERT*. Contrairement à notre système, nous nous apercevons que *Jeux2BERT* cible les deux métriques de performance (précision et rappel) du fait de la complexité du modèle de réseaux prototypiques.

Le tableau 5.8 résume les performances du modèle sur les deux sous-domaines de veille technologique et veille d'entreprise. Les résultats sont relativement proches,

Système	Précision	Rappel	F1
domaine de structure d'entreprise	82.78	61.25	70.40
domaine d'activité d'entreprise	85.24	52.14	64.70

TABLE 5.8 Résultats de l'évaluation de Jeux2BERT+OIE.

excepté pour la performance de rappel du deuxième sous-domaine qui est plus basse. À noter aussi qu'il nous a fallu approximativement deux heures pour rédiger les différentes variantes des patrons des relations sémantiques avec leurs contraintes sémantiques, et une meilleure couverture des patrons des relations sémantiques pourrait en principe améliorer les performances.
Ces résultats laissent envisager qu'il est possible d'adapter notre méthode à différents domaines de spécialité avec un petit effort manuel lors de la création des patrons pour chaque relation et avec des connaissances externes du domaine de spécialité.

5.4.3 Analyse des erreurs

Dans cette section, nous présentons nos résultats d'analyse, en apportant une réponse à la question : lesquelles des erreurs restantes peuvent être attribuées aux modèles, et quelles sont les raisons potentielles de ces erreurs ?
Nous discutons d'abord des résultats de notre analyse manuelle des erreurs dues au modèle *UD2OIE*, suivis des résultats de l'analyse manuelle des erreurs dues au modèle *Jeux2BERT*.

La première et principale source d'erreur est propagée à partir du modèle *UD2OIE*, en effet la conception de notre modèle d'extraction de relations implique une dépendance au modèle *UD2OIE*.
En partant de l'analyse de la section précédente, il en convient que notre système d'extraction de relation a plus du mal à extraire un nombre important de relations qu'à trouver le type de relation liant les deux entités.
Cela s'explique par le fait que la granularité de l'entrée du modèle d'extraction de relations est le triplet obtenu par *UD2OIE*, ainsi le système d'extraction de relation ne pourra jamais associer une relation entre deux entités si elles ne se trouvent pas dans un même triplet de *UD2OIE*. À noter qu'il est toujours possible d'inférer de nouveaux triplets incluant les deux entités en appliquant des règles d'inférence selon les dépendances des relations du schéma.

Pour cela, nous essayons de nous concentrer sur les raisons qui font que le système d'extraction a une performance de rappel relativement moyenne. La première raison est due aux co-références où le triplet ne capture pas l'entité concernée mais juste sa référence.
L'exemple du tableau 5.9 illustre deux cas où le système de résolution de co-référence n'arrive pas à trouver l'antécédent du pronom *lui* est l'entité *Google*, le deuxième triplet présente le cas où la résolution de la co-référence est erronée (le pronom *lui* fait référence à *Apple* et non à *Google*). Ainsi une relation de concurrence ne peut être établie entre les entités *Google* et *Apple*.
Le deuxième exemple d'entités non connectées puisque les entités Orange et *Atos Origin* n'apparaissent pas dans le même triplet. La raison est souvent soit due à un signal faible entre les entités ou à des constructions linguistiques plus complexes entre les deux entités. Ainsi, une relation de partenariat ne peut être établie entre les entités *Orange* et *Atos Origin*.

Aussi, nous présentons un autre type d'erreur qui cible cette fois ci la performance de précision et non la performance de rappel. Le troisième exemple du tableau 5.9 montre un exemple d'inversion d'argument qui se produit qu'avec les relations nominales. L'exemple illustre un exemple ou l'entité *LVMH* acquière *Zenith* et non le contraire. À noter qu'il est toujours possible de recouvrir ce type d'erreurs dans certain cas en vérifiant les prépositions (*par,de,...*). Finalement, nous présentons non pas un type d'erreur mais un exemple englobant une proposition et sa contre-proposition en même temps.
Nous pouvons voir que le quatrième exemple du tableau 5.9 exprime une relation de *CEO_Of* entre l'entité *WandCo* et *Guillaume Chifflot*, mais le deuxième triplet indique que cette dernière a été démise de ses fonction chez l'entité *WandCo*.

Type d'erreur	Exemple
Co-référence	Google et Facebook en embuscade face à Apple, seul Google lui tient un peu tête. (**A0 :**Google **; P :**tient un peu tête **; A1 :**lui)) (**A0 :**Google **; P :**tient un peu tête **; A1 :**Facebook))
Entités non connectés	SFR , Bouygues et Orange viennent de signer un accord de partenariat qui permet à Atos Origin de se placer dans le marché de la téléphonie et d'internet. (**A0 :**Orange ; **P :**viennent de signer ; **A1 :**un accord de partenariat) (**A0 :**accord de partenariat ; **P :**permet à ; **A1 :**Atos Origin ; **A2 :**de se placer dans le marché de la téléphonie et d'internet)
Inversion d'arguments	Apparues en 1975 , les montres Dior développent des pièces plus proches de la haute horlogerie dès le rachat par LVMH de Zenith en 1999. (**A0 :**Zenith **; P :**rachat par **; A1 :**LVMH **; A2 :**en 1999)
Contre proposition	Par ailleurs , l' agence WandCo vient de se séparer de son directeur de la création , Guillaume Chifflot. (**A0 :**Guillaume Chifflot **; P :**directeur de la création **;** **A1 :**l' agence WandCo) (**A0 :**l' agence WandCo **; P :**vient de se séparer **;** **A1 :**Guillaume Chifflot)

TABLE 5.9 Analyse et types des erreurs les plus récurrentes.

Maintenant, nous enchaînons sur les types d'erreurs qui peuvent être attribués au modèle *Jeux2BERT*. Nous tenons à préciser que les erreurs au niveau du modèle *Jeux2BERT* impactent plus la performance de précision, la performance de rappel dépend plus du seuil du score de confiance de *Jeux2BERT* pour décider si la relation entre les deux entités est valide. Ainsi, nous listons quelques-uns des types d'erreurs dans le tableau 5.10.

Le premier exemple montre un exemple où le modèle considère qu'une relation d'employé existe entre les deux entités en se basant sur le prédicat qui est à la fois ambigu et avec un contexte incomplet (l'argument de préposition).
Le second exemple illustre un exemple où le mot clé déclencheur exprimant une relation de coopération est trop spécifique et ne peut être normalisé vers aucune des relations cibles.

Type d'erreur	Exemple
Signal ambigu	UBS a intégré l'Oréal à sa liste de valeurs préférées en estimant que le groupe a la capacité de faire mieux que son marché. --- (**A0 :**UBS **;P :**a intégré **; A1 :**l'Oréal **;** **A2 :**à sa liste de valeurs préférées) (**A0 :**UBS **;Per :Member_or_employee_of ; A1 :**l'Oréal)
Signal trés spécifique	Depuis mars, Facebook loue son réseau social à Warner Bros qui y diffuse ses films à 3 dollars l'unité. --- (**A0 :**Facebook ; **P :**loue son réseau social à ; **A1 :**Warner Bros)

TABLE 5.10 Analyse et types des erreurs les plus récurrentes.

Nous incluons aussi quelques exemples de prédicats (phrases relationnelles) qui n'ont pas été normalisé par le modèle *Jeux2BERT* dans le tableau 5.11 . Ces segments de phrases relationnelles peuvent être considérés très spécifiques.

Type de relations	**Prédicats spécifiques**
Org :Subsidiary	sous la bannière de comme un essaimage du
Org :Shareholder	se partagent
Org :Competitor_with	perd face à a été devancé par
Org :Partnership_with	loue soutenu par

TABLE 5.11 Exemple de phrases relationnelles pour lesquels le modèle *Jeux2BERT* n'arrive pas à les projeter vers leurs types de relations correspondants.

5.5 Conclusion

Dans ce chapitre, nous avons présenté et illustré comment la méthode que nous avons proposée combine les deux systèmes *UD2OIE* et *Jeux2BERT* afin de construire un système d'extraction de relations sémantiques. Nous montrons que nous pouvons passer du paradigme de l'OpenIE vers le paradigme d'extraction de relations grâce à un ensemble de règles lexico-sémantiques augmentés d'une base de connaissances. Nous arrivons également à montrer l'adaptation de notre méthode à différents domaines de spécialité avec un petit effort manuel en l'appliquant sur les deux sous-domaines de l'intelligence d'entreprise : la structure de l'entreprise et l'activité de l'entreprise. La méthode que nous avons présentée semble donc raisonnablement générique et permet d'obtenir des résultats satisfaisants.

Chapitre 6

Conclusion générale

Nous résumons brièvement les principaux enseignements tirés de nos travaux sur l'apport des connaissances externes (dictionnaires, bases de connaissances) à la tâche d'extraction de relations sémantiques depuis des textes.

6.1 Principales conclusions

Dans le troisième chapitre sur l'Open Information Extraction, nous avons montré l'utilité de bien saisir les facteurs de variation dans le cadre de l'apprentissage automatique. Nous arrivons aussi à justifier la relation entre les facteurs de variation et la quantité de données nécessaire pour entraîner un modèle d'apprentissage. Nous confirmons notre hypothèse en proposant un système OpenIE à partir d'un ensemble de données d'apprentissage relativement petit en comparaison à l'état de l'art. Nous démontrons aussi que les approches récentes exploitant les modèles de langage ne respectent pas le critère d'adaptation au domaine et doivent être utilisés judicieusement.

Finalement, nous présentons dans le quatrième chapitre une nouvelle méthode pour affiner les modèles de langage à l'aide d'une base de connaissances. Nous proposons une approche qui introduit la tâche de classement de pertinence hiérarchique et qui arrive à améliorer les performances sur les différentes métriques d'évaluation. Dans le cadre d'apprentissage contrastif (apprentissage des similarités et différences entre les objets), nous concluons qu'apprendre les différences entre deux triplets valides apporte un meilleur gain d'apprentissage que d'apprendre les différences entre deux triplets invalides. Nous introduisons aussi une approche permettant d'amoindrir la dépendance au contexte pour les modèles

de langage car elle introduit une hiérarchie entre les différents sens possibles d'un mot.

6.2 Apport des connaissances à l'extraction de relations

Dans cette section, nous discutons de l'impact des bases de connaissances dans la tâche d'extraction de relations. Le premier aspect est l'utilisation des bases de connaissances pour la tâche de reconnaissance d'entités nommées. En effet, mis à part les systèmes de reconnaissance d'entités nommées, les bases de connaissances peuvent englober différentes entités avec leurs types. Autre aspect lié aux entités nommées est l'analyse syntaxique. Ainsi, une phrase donnée peut être prétraitée de sorte que l'analyseur syntaxique considère l'entité nommée comme un nom propre composé ou simple pour obtenir une meilleure représentation syntaxique.
Dans nos travaux nous n'avons pas pu utiliser la base de connaissances *JeuxdeMots* pour détecter les entités nommées car elle ne contient pas assez de connaissances sur le domaine d'intelligence d'entreprise. Cependant, nous avons montré l'utilité des bases de connaissances dans la normalisation des prédicats ou mots clés des relations sémantiques. Comme il est impossible de lister toutes les variations linguistiques exprimant une certaine relation, les bases de connaissances apportent en partie une solution à ce problème.
Nous avons évalué notre système d'extraction de relations sémantiques sur deux sous-domaines de l'intelligence d'entreprise : la structure de l'entreprise et l'activité de l'entreprise. Nous avons choisi de procéder de cette manière pour cause de manque de corpus d'évaluation, toutefois une évaluation sur un autre domaine de spécialité reste le meilleur cas d'étude pour évaluer l'adaptation au domaine. Afin de passer du domaine d'intelligence d'entreprise à un autre domaine de spécialité, nous pensons qu'un domaine de spécialité tel que l'histoire ou des articles de bibliographie sont plus pertinents que le domaine de la médecine. On explique cela par la similarité lexicale entre les domaines tels que les entités nommées de type personne ou lieu géographique, et nous avons déjà des outils de reconnaissance d'entités de ce type ainsi que *JeuxDeMots* pour faciliter la transition vers un de ces domaines de spécialité.

6.3 Résumé de nos contributions au domaine

Nous listons ci-dessous nos principales contributions :

— *OpenIE* : Nous proposons une approche d'OpenIE appuyée par des expérimentations qui démontrent la supériorité des approches syntaxiques par apport aux approches exploitant la sémantique lexicale (*BERT*) pour l'adaptation au domaine. Les résultats des expérimentations démontrent aussi que c'est la qualité plutôt que la quantité des données qui va impacter les performances, et que corréler l'analyse syntaxique à un petit ensemble de données de qualité est plus bénéfique pour l'apprentissage que l'état de l'art actuel.

— *Affinement des vecteurs de mots* : Nous introduisons une tâche de classement de pertinence hiérarchique pour l'affinement des représentations contextualisées des mots. Suivant le principe d'apprentissage contrastif, cette tâche se distingue par le fait de baser son apprentissage sur des couples d'exemples pertinents mais difficiles que des couples d'exemples faciles mais non pertinents. Les expérimentations montrent que les exemples difficiles sont plus bénéfiques pour l'apprentissage contrastif que les exemples faciles.

— *Extraction de relations sémantiques* : Nous présentons une approche en deux temps qui renvoie des triplets avec une représentation prédicat-arguments, l'extraction de n'importe quelle relation sémantique peut être amorcée à partir de ces triplets du moment que nous listons quelques mots clés de la relation et en exploitant le modèle de vecteur de mots affiné *Jeux2BERT*.

6.4 Quelques perspectives

Nous pouvons entrevoir quelques perspectives et points d'améliorations applicables à notre approche proposée. En premier lieu, nous pensons qu'il serait intéressant d'avoir un modèle de classification automatique capable de juger si un triplet dans le contexte d'une phrase particulière est correct ou non. Ce modèle serait orthogonal au modèle d'extraction de triplet depuis des phrases. Le modèle de vérification permettrait de filtrer les triplets erronés et éventuellement de classer la pertinence des triplets extraits quand il y a une multiplicité des triplets avec des

arguments plus ou moins précis.

Quant au modèle affiné avec les données de *JeuxDeMots*, nous jugeons qu'il est important et possible de rendre plus explicite les connaissances de chaque entité du triplet lors de l'inférence, en fournissant un contexte pour chaque argument. Par exemple, ce contexte serait formé de tous les triplets dans lesquels l'argument du triplet en question est présente en tant que source ou cible (cela correspondrait à une représentation en étoile pour chaque entité). De plus, cette tâche pourrait conjointement sélectionner le sens correct de l'argument si celui-ci est identifié comme polysémique dans *JDM*.

Finalement, nous pensons à évaluer notre approche sur un domaine éloigné de celui de la veille d'entreprise et d'évaluer l'effort nécessaire à l'adaptation et les résultats obtenus.

6.5 Pour finir

Pour résumer, nous avons montré que les approches actuelles de bout-en-bout pour l'extraction de relations sémantiques sont un peu trop simplistes dans la mesure où elles ignorent beaucoup d'aspects linguistiques, comme la syntaxe et la sémantique lexicale, et n'incorporent pas de connaissances externes, pourtant indispensables, dans leur raisonnement.
Nous proposons un système d'extraction relativement simple mais 1) qui a l'avantage d'offrir un raisonnement interprétable, 2) qui exploite des connaissances externes et 3) qui peut être paramétré facilement pour s'adapter à un domaine spécialité.

Bibliographie

Agichtein, E. and Gravano, L. (2000), Snowball : Extracting relations from large plain-text collections, *in* 'Proceedings of the fifth ACM conference on Digital libraries', pp. 85–94.

Agrawal, R., Imieliński, T. and Swami, A. (1993), Mining association rules between sets of items in large databases, *in* 'Proceedings of the 1993 ACM SIGMOD international conference on Management of data', pp. 207–216.

Baker, C. F., Fillmore, C. J. and Lowe, J. B. (1998), The berkeley framenet project, *in* '36th Annual Meeting of the Association for Computational Linguistics and 17th International Conference on Computational Linguistics, Volume 1', pp. 86–90.

Baldini Soares, L., FitzGerald, N., Ling, J. and Kwiatkowski, T. (2019), Matching the blanks : Distributional similarity for relation learning, *in* 'Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics', Association for Computational Linguistics, Florence, Italy, pp. 2895–2905.
URL: *https ://www.aclweb.org/anthology/P19-1279*

Banko, M. and Etzioni, O. (2008), The tradeoffs between open and traditional relation extraction, *in* 'Proceedings of ACL-08 : HLT', Association for Computational Linguistics, Columbus, Ohio, pp. 28–36.
URL: *https ://www.aclweb.org/anthology/P08-1004*

Bekoulis, G., Deleu, J., Demeester, T. and Develder, C. (2018), 'Joint entity recognition and relation extraction as a multi-head selection problem', *Expert Systems with Applications* **114**, 34–45.

Bhardwaj, S., Aggarwal, S. and Mausam, M. (2019), CaRB : A crowdsourced benchmark for open IE, *in* 'Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference

on Natural Language Processing (EMNLP-IJCNLP)', Association for Computational Linguistics, Hong Kong, China, pp. 6262–6267.
URL: *https ://www.aclweb.org/anthology/D19-1651*

Bojanowski, P., Grave, E., Joulin, A. and Mikolov, T. (2017), 'Enriching word vectors with subword information', *Transactions of the Association for Computational Linguistics* **5**, 135–146.
URL: *https ://www.aclweb.org/anthology/Q17-1010*

Bollacker, K., Evans, C., Paritosh, P., Sturge, T. and Taylor, J. (2008), Freebase : a collaboratively created graph database for structuring human knowledge, *in* 'Proceedings of the 2008 ACM SIGMOD international conference on Management of data', pp. 1247–1250.

Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J. and Yakhnenko, O. (2013), 'Translating embeddings for modeling multi-relational data', *Advances in neural information processing systems* **26**.

Brin, S. (1998), Extracting patterns and relations from the world wide web, *in* 'International workshop on the world wide web and databases', Springer, pp. 172–183.

Bunescu, R. and Mooney, R. J. (2005), Subsequence kernels for relation extraction, *in* 'NIPS', Citeseer, pp. 171–178.

Carlson, A., Betteridge, J., Hruschka Junior, E. R. and Mitchell, T. M. (2009), Coupling semi-supervised learning of categories and relations, *in* 'Proceedings of the NAACL HLT 2009 Workshop on Semi-supervised Learning for Natural Language Processing', Association for Computational Linguistics, Boulder, Colorado, pp. 1–9.
URL: *https ://www.aclweb.org/anthology/W09-2201*

Chen, J., Ji, D., Tan, C. L. and Niu, Z. (2005), Unsupervised feature selection for relation extraction, *in* 'Companion Volume to the Proceedings of Conference including Posters/Demos and tutorial abstracts'.
URL: *https ://www.aclweb.org/anthology/I05-2045*

Chen, Q., Zhu, X., Ling, Z., Wei, S., Jiang, H. and Inkpen, D. (2016), 'Enhanced lstm for natural language inference', *arXiv preprint arXiv :1609.06038* .

Christensen, J., Mausam, Soderland, S. and Etzioni, O. (2010), Semantic role labeling for open information extraction, *in* 'Proceedings of the NAACL HLT 2010 First International Workshop on Formalisms and Methodology for Learning by Reading', Association for Computational Linguistics, Los Angeles, California, pp. 52–60.
URL: *https ://www.aclweb.org/anthology/W10-0907*

Collins, M. and Duffy, N. (2001), Convolution kernels for natural language, *in* 'Advances in neural information processing systems', pp. 625–632.

Collins, M. and Miller, S. (1998), Semantic tagging using a probabilistic context free grammar, Technical report, PENNSYLVANIA UNIV PHILADELPHIA DEPT OF COMPUTER AND INFORMATION SCIENCE.

Cui, L., Wei, F. and Zhou, M. (2018*a*), Neural open information extraction, *in* 'Proceedings of the56th Annual Meeting of the Association for Compu-tational Linguistics', pp. 407–413.

Cui, L., Wei, F. and Zhou, M. (2018*b*), Neural open information extraction, *in* 'Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers)', Association for Computational Linguistics, Melbourne, Australia, pp. 407–413.
URL: *https ://www.aclweb.org/anthology/P18-2065*

Culotta, A., McCallum, A. and Betz, J. (2006), Integrating probabilistic extraction models and data mining to discover relations and patterns in text, *in* 'Proceedings of the Human Language Technology Conference of the NAACL, Main Conference', Association for Computational Linguistics, New York City, USA, pp. 296–303.
URL: *https ://www.aclweb.org/anthology/N06-1038*

Culotta, A. and Sorensen, J. (2004), Dependency tree kernels for relation extraction, *in* 'Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)', Barcelona, Spain, pp. 423–429.
URL: *https ://www.aclweb.org/anthology/P04-1054*

Curran, J. R., Murphy, T. and Scholz, B. (2007), Minimising semantic drift with mutual exclusion bootstrapping, *in* 'Proceedings of the 10th Conference of the Pacific Association for Computational Linguistics', Vol. 6, Citeseer, pp. 172–180.

Day, D., Aberdeen, J., Hirschman, L., Kozierok, R., Robinson, P. and Vilain, M. (1997), Mixed-initiative development of language processing systems, *in* 'Fifth Conference on Applied Natural Language Processing', pp. 348–355.

Del Corro, L. and Gemulla, R. (2013), Clausie : clause-based open information extraction, *in* 'Proceedings of the 22nd international conference on World Wide Web', pp. 355–366.

Desjardins, G., Courville, A. and Bengio, Y. (2012), 'Disentangling factors of variation via generative entangling', *arXiv preprint arXiv :1210.5474* .

Devlin, J., Chang, M.-W., Lee, K. and Toutanova, K. (2019), 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)', Association for Computational Linguistics, Minneapolis, Minnesota, pp. 4171–4186.
URL: *https ://www.aclweb.org/anthology/N19-1423*

Doddington, G., Mitchell, A., Przybocki, M., Ramshaw, L., Strassel, S. and Weischedel, R. (2004), The automatic content extraction (ACE) program – tasks, data, and evaluation, *in* 'Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)', European Language Resources Association (ELRA), Lisbon, Portugal.
URL: *http ://www.lrec-conf.org/proceedings/lrec2004/pdf/5.pdf*

Dozat, T. and Manning, C. D. (2016), 'Deep biaffine attention for neural dependency parsing', *arXiv preprint arXiv :1611.01734* .

Du, J., Han, J., Way, A. and Wan, D. (2018), 'Multi-level structured self-attentions for distantly supervised relation extraction', *arXiv preprint arXiv :1809.00699* .

Etzioni, O., Cafarella, M., Downey, D., Kok, S., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D. S. and Yates, A. (2004), Web-scale information extraction in knowitall : (preliminary results), *in* 'Proceedings of the 13th international conference on World Wide Web', pp. 100–110.

Etzioni, O., Cafarella, M., Downey, D., Popescu, A.-M., Shaked, T., Soderland, S., Weld, D. S. and Yates, A. (2005), 'Unsupervised named-entity extraction from the web : An experimental study', *Artificial intelligence* **165**(1), 91–134.

Fabre, C. and Bourigault, D. (2006), Extraction de relations sémantiques entre noms et verbes au-delà des liens morphologiques, *in* 'TALN', Vol. 6, pp. 10–13.

Fader, A., Soderland, S. and Etzioni, O. (2011*a*), Identifying relations for open information extraction, *in* 'Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing', Association for Computational Linguistics, Edinburgh, Scotland, UK., pp. 1535–1545.
URL: *https ://www.aclweb.org/anthology/D11-1142*

Fader, A., Soderland, S. and Etzioni, O. (2011*b*), Identifying relations for open information extraction, *in* 'Proceedings of the conference on empirical methods in natural language processing', Association for Computational Linguistics, pp. 1535–1545.

Falke, T., Stanovsky, G., Gurevych, I. and Dagan, I. (2016), Porting an open information extraction system from english to german, *in* 'Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing', pp. 892–898.

Faruqui, M., Dodge, J., Jauhar, S. K., Dyer, C., Hovy, E. and Smith, N. A. (2015), Retrofitting word vectors to semantic lexicons, *in* 'Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies', Association for Computational Linguistics, Denver, Colorado, pp. 1606–1615.
URL: *https ://www.aclweb.org/anthology/N15-1184*

Forney, G. (1973), 'The viterbi algorithm', *Proceedings of the IEEE* **61**(3), 268–278.

Fried, D. and Duh, K. (2014), 'Incorporating both distributional and relational semantics in word representations', *arXiv preprint arXiv :1412.4369* .

Fu, T.-J., Li, P.-H. and Ma, W.-Y. (2019), GraphRel : Modeling text as relational graphs for joint entity and relation extraction, *in* 'Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics', Association for Computational Linguistics, Florence, Italy, pp. 1409–1418.
URL: *https ://www.aclweb.org/anthology/P19-1136*

Gamallo, P. and Garcia, M. (2015), Multilingual open information extraction, *in* 'Portuguese Conference on Artificial Intelligence', pp. 711–722.

Ganitkevitch, J., Van Durme, B. and Callison-Burch, C. (2013), Ppdb : The paraphrase database, *in* 'Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies', pp. 758–764.

Giorgi, J., Wang, X., Sahar, N., Shin, W. Y., Bader, G. D. and Wang, B. (2019), 'End-to-end named entity recognition and relation extraction using pre-trained language models', *arXiv preprint arXiv :1912.13415* .

Girju, R., Badulescu, A. and Moldovan, D. (2006), 'Automatic discovery of part-whole relations', *Computational Linguistics* **32**(1), 83–135.

Goodfellow, I. J., Bengio, Y. and Courville, A. (2016), *Deep Learning*, MIT Press, Cambridge, MA, USA. `http://www.deeplearningbook.org`.

Gotti, F. and Langlais, P. (2016), 'From french wikipedia to erudit : A test case for cross-domain open information extraction', *Computational Intelligence* .
URL: *https ://onlinelibrary.wiley.com/doi/abs/10.1111/coin.12120*

Han, X., Zhu, H., Yu, P., Wang, Z., Yao, Y., Liu, Z. and Sun, M. (2018), FewRel : A large-scale supervised few-shot relation classification dataset with state-of-the-art evaluation, *in* 'Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing', Association for Computational Linguistics, Brussels, Belgium, pp. 4803–4809.
URL: *https ://www.aclweb.org/anthology/D18-1514*

Harabagiu, S., Bejan, C. A. and Morarescu, P. (2005), Shallow semantics for relation extraction, *in* 'Proceedings of the 19th international joint conference on Artificial intelligence', pp. 1061–1066.

Hasegawa, T., Sekine, S. and Grishman, R. (2004), Discovering relations among named entities from large corpora, *in* 'Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics (ACL-04)', Barcelona, Spain, pp. 415–422.
URL: *https ://www.aclweb.org/anthology/P04-1053*

He, T., Zhao, J. and Li, J. (2006), Discovering relations among named entities by detecting community structure, *in* 'Proceedings of the 20th Pacific Asia Conference on Language, Information and Computation', Tsinghua University Press,

Huazhong Normal University, Wuhan, China, pp. 42–48.
URL: *https ://www.aclweb.org/anthology/Y06-1006*

Hearst, M. A. (1992), Automatic acquisition of hyponyms from large text corpora, *in* 'COLING 1992 Volume 2 : The 15th International Conference on Computational Linguistics'.
URL: *https ://www.aclweb.org/anthology/C92-2082*

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

Hoffart, J., Yosef, M. A., Bordino, I., Fürstenau, H., Pinkal, M., Spaniol, M., Taneva, B., Thater, S. and Weikum, G. (2011), Robust disambiguation of named entities in text, *in* 'Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing', Association for Computational Linguistics, Edinburgh, Scotland, UK., pp. 782–792.
URL: *https ://www.aclweb.org/anthology/D11-1072*

Hoffmann, R., Zhang, C., Ling, X., Zettlemoyer, L. and Weld, D. S. (2011), Knowledge-based weak supervision for information extraction of overlapping relations, *in* 'Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics : Human Language Technologies', Association for Computational Linguistics, Portland, Oregon, USA, pp. 541–550.
URL: *https ://www.aclweb.org/anthology/P11-1055*

Hong, Y., Liu, Y., Yang, S., Zhang, K. and Hu, J. (2020), 'Joint extraction of entities and relations using graph convolution over pruned dependency trees', *Neurocomputing* **411**, 302–312.

Howard, J. and Ruder, S. (2018), Universal language model fine-tuning for text classification, *in* 'Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)', Association for Computational Linguistics, Melbourne, Australia, pp. 328–339.
URL: *https ://www.aclweb.org/anthology/P18-1031*

Jabbari, A., Sauvage, O., Zeine, H. and Chergui, H. (2020), A French corpus and annotation schema for named entity recognition and relation extraction of financial news, *in* 'Proceedings of the 12th Language Resources and Evaluation Conference', European Language Resources Association, Marseille, France, pp. 2293–

2299.
URL: *https ://www.aclweb.org/anthology/2020.lrec-1.279*

Jacques, M. and Aussenac-Gilles, N. (2006), 'Variabilité des performances des outils de TAL et genre textuel', *Trait. Autom. des Langues* **47**(1), 11–32.
URL: *http ://www.atala.org/Variabilite-des-performances-des*

Jean, P.-A., Harispe, S., Bellot, P., Ranwez, S. and Montmain, J. (2017), Gérer l'incertitude lors de l'extraction de relations et lors de l'inférence de nouvelles connaissances, *in* 'COnférence en Recherche d'Informations et Applications-CORIA 2017'.

Kambhatla, N. (2004), Combining lexical, syntactic, and semantic features with maximum entropy models for information extraction, *in* 'Proceedings of the ACL Interactive Poster and Demonstration Sessions', Association for Computational Linguistics, Barcelona, Spain, pp. 178–181.
URL: *https ://www.aclweb.org/anthology/P04-3022*

Kambhatla, N. (2006), Minority vote : At-least-n voting improves recall for extracting relations, *in* 'Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions', Association for Computational Linguistics, Sydney, Australia, pp. 460–466.
URL: *https ://www.aclweb.org/anthology/P06-2060*

Kate, R. J. and Mooney, R. (2010), Joint entity and relation extraction using card-pyramid parsing, *in* 'Proceedings of the Fourteenth Conference on Computational Natural Language Learning', Association for Computational Linguistics, Uppsala, Sweden, pp. 203–212.
URL: *https ://www.aclweb.org/anthology/W10-2924*

Katiyar, A. and Cardie, C. (2016), Investigating LSTMs for joint extraction of opinion entities and relations, *in* 'Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)', Association for Computational Linguistics, Berlin, Germany, pp. 919–929.
URL: *https ://www.aclweb.org/anthology/P16-1087*

Khaldi, H., Abdaoui, A., Benamara, F., Sigel, G. and Aussenac-Gilles, N. (2020), Classification de relations pour l'intelligence économique et concurrentielle, *in* '6e conférence conjointe Journées d'Études sur la Parole (JEP, 33e édition),

Traitement Automatique des Langues Naturelles (TALN, 27e édition), Rencontre des Étudiants Chercheurs en Informatique pour le Traitement Automatique des Langues (RÉCITAL, 22e édition). Volume 2 : Traitement Automatique des Langues Naturelles', ATALA : Association pour le traitement automatique des langues ; AFCP . . . , pp. 27–39.

Kim, B., Hong, T., Ko, Y. and Seo, J. (2020), Multi-task learning for knowledge graph completion with pre-trained language models, *in* 'Proceedings of the 28th International Conference on Computational Linguistics', International Committee on Computational Linguistics, Barcelona, Spain (Online), pp. 1737–1743.
URL: *https ://www.aclweb.org/anthology/2020.coling-main.153*

Kingsbury, P. R. and Palmer, M. (2002), From treebank to propbank., *in* 'LREC', Citeseer, pp. 1989–1993.

Kolluru, K., Adlakha, V., Aggarwal, S., Mausam and Chakrabarti, S. (2020), OpenIE6 : Iterative Grid Labeling and Coordination Analysis for Open Information Extraction, *in* 'Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)', Association for Computational Linguistics, Online, pp. 3748–3761.
URL: *https ://www.aclweb.org/anthology/2020.emnlp-main.306*

Kolluru, K., Aggarwal, S., Rathore, V., Mausam and Chakrabarti, S. (2020), IMoJIE : Iterative memory-based joint open information extraction, *in* 'Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics', Association for Computational Linguistics, Online, pp. 5871–5886.
URL: *https ://www.aclweb.org/anthology/2020.acl-main.521*

Lafferty, J., McCallum, A. and Pereira, F. C. (2001), Conditional random fields : Probabilistic models for segmenting and labeling sequence data, *in* 'Proceedings of the 18th International Conference on Machine Learning 2001 (ICML 2001)', pp. 282–289.

Lafourcade, M. and Ramadier, L. (2016), Semantic relation extraction with semantic patterns experiment on radiology reports, *in* 'Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)', European Language Resources Association (ELRA), Portorož, Slovenia, pp. 4578–4582.
URL: *https ://www.aclweb.org/anthology/L16-1725*

Le, H., Vial, L., Frej, J., Segonne, V., Coavoux, M., Lecouteux, B., Allauzen, A., Crabbé, B., Besacier, L. and Schwab, D. (2020), FlauBERT : Unsupervised language model pre-training for French, *in* 'Proceedings of the 12th Language Resources and Evaluation Conference', European Language Resources Association, Marseille, France, pp. 2479–2490.
URL: *https ://www.aclweb.org/anthology/2020.lrec-1.302*

Lechelle, W., Gotti, F. and Langlais, P. (2019), WiRe57 : A fine-grained benchmark for open information extraction, *in* 'Proceedings of the 13th Linguistic Annotation Workshop', Association for Computational Linguistics, Florence, Italy, pp. 6–15.
URL: *https ://www.aclweb.org/anthology/W19-4002*

Lewis, M. and Steedman, M. (2013), Unsupervised induction of cross-lingual semantic relations, *in* 'Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing', Association for Computational Linguistics, Seattle, Washington, USA, pp. 681–692.
URL: *https ://www.aclweb.org/anthology/D13-1064*

Li, H., Bollegala, D., Matsuo, Y. and Ishizuka, M. (2011), Using graph based method to improve bootstrapping relation extraction, *in* 'International Conference on Intelligent Text Processing and Computational Linguistics', Springer, pp. 127–138.

Li, J., He, R., Ye, H., Ng, H. T., Bing, L. and Yan, R. (2020), 'Unsupervised domain adaptation of a pretrained cross-lingual language model', *arXiv preprint arXiv :2011.11499* .

Li, Q. and Ji, H. (2014), Incremental joint extraction of entity mentions and relations, *in* 'Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)', Association for Computational Linguistics, Baltimore, Maryland, pp. 402–412.
URL: *https ://www.aclweb.org/anthology/P14-1038*

Li, X., Yin, F., Sun, Z., Li, X., Yuan, A., Chai, D., Zhou, M. and Li, J. (2019), Entity-relation extraction as multi-turn question answering, *in* 'Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics', Association for Computational Linguistics, Florence, Italy, pp. 1340–1350.
URL: *https ://www.aclweb.org/anthology/P19-1129*

Lin, D. and Pantel, P. (2001), Dirt@ sbt@ discovery of inference rules from text, *in* 'Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining', pp. 323–328.

Lodhi, H., Saunders, C., Shawe-Taylor, J., Cristianini, N. and Watkins, C. (2002), 'Text classification using string kernels', *Journal of Machine Learning Research* **2**(Feb), 419–444.

Mausam, M. (2016), Open information extraction systems and downstream applications, *in* 'Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence', pp. 4074–4077.

McCann, B., Bradbury, J., Xiong, C. and Socher, R. (2017), 'Learned in translation : Contextualized word vectors', *arXiv preprint arXiv :1708.00107* .

McIntosh, T. and Curran, J. R. (2009), Reducing semantic drift with bagging and distributional similarity, *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', Association for Computational Linguistics, Suntec, Singapore, pp. 396–404.
URL: *https ://www.aclweb.org/anthology/P09-1045*

Mesquita, F., Schmidek, J. and Barbosa, D. (2013), Effectiveness and efficiency of open relation extraction, *in* 'Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing', pp. 447–457.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S. and Dean, J. (2013), Distributed representations of words and phrases and their compositionality, *in* 'Advances in neural information processing systems', pp. 3111–3119.

Miller, G. A. (1995), 'Wordnet : a lexical database for english', *Communications of the ACM* **38**(11), 39–41.

Miller, S., Fox, H., Ramshaw, L. and Weischedel, R. (2000), A novel use of statistical parsing to extract information from text, *in* '1st Meeting of the North American Chapter of the Association for Computational Linguistics'.
URL: *https ://www.aclweb.org/anthology/A00-2030*

Mintz, M., Bills, S., Snow, R. and Jurafsky, D. (2009), 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', Association for Computational Linguistics, Suntec, Singapore, pp. 1003–1011.
URL: *https ://www.aclweb.org/anthology/P09-1113*

Miwa, M. and Bansal, M. (2016), End-to-end relation extraction using LSTMs on sequences and tree structures, *in* 'Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)', Association for Computational Linguistics, Berlin, Germany, pp. 1105–1116.
URL: *https ://www.aclweb.org/anthology/P16-1105*

Miwa, M. and Sasaki, Y. (2014), Modeling joint entity and relation extraction with table representation, *in* 'Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)', Association for Computational Linguistics, Doha, Qatar, pp. 1858–1869.
URL: *https ://www.aclweb.org/anthology/D14-1200*

Mou, L., Meng, Z., Yan, R., Li, G., Xu, Y., Zhang, L. and Jin, Z. (2016), How transferable are neural networks in NLP applications ?, *in* 'Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing', Association for Computational Linguistics, Austin, Texas, pp. 479–489.
URL: *https ://www.aclweb.org/anthology/D16-1046*

Nakashole, N., Weikum, G. and Suchanek, F. (2012), PATTY : A taxonomy of relational patterns with semantic types, *in* 'Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning', Association for Computational Linguistics, Jeju Island, Korea, pp. 1135–1145.
URL: *https ://www.aclweb.org/anthology/D12-1104*

Neelakantan, A., Shankar, J., Passos, A. and McCallum, A. (2014), Efficient non-parametric estimation of multiple embeddings per word in vector space, *in* 'Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)', Association for Computational Linguistics, Doha, Qatar, pp. 1059–1069.
URL: *https ://www.aclweb.org/anthology/D14-1113*

Niklaus, C., Cetto, M., Freitas, A. and Handschuh, S. (2018), 'A survey on open information extraction', *arXiv preprint arXiv :1806.05599* .

Nivre, J., de Marneffe, M.-C., Ginter, F., Goldberg, Y., Hajič, J., Manning, C. D., McDonald, R., Petrov, S., Pyysalo, S., Silveira, N., Tsarfaty, R. and Zeman, D. (2016), Universal Dependencies v1 : A multilingual treebank collection, *in* 'Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)', European Language Resources Association (ELRA), Portorož, Slovenia, pp. 1659–1666.
URL: *https ://www.aclweb.org/anthology/L16-1262*

Ouchi, H., Shindo, H. and Matsumoto, Y. (2018), A span selection model for semantic role labeling, *in* 'Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing', Association for Computational Linguistics, Brussels, Belgium, pp. 1630–1642.
URL: *https ://www.aclweb.org/anthology/D18-1191*

Pal, H. and Mausam (2016), Demonyms and compound relational nouns in nominal open IE, *in* 'Proceedings of the 5th Workshop on Automated Knowledge Base Construction', Association for Computational Linguistics, San Diego, CA, pp. 35–39.
URL: *https ://www.aclweb.org/anthology/W16-1307*

Pal, H. et al. (2016), Demonyms and compound relational nouns in nominal open ie, *in* 'Proceedings of the 5th Workshop on Automated Knowledge Base Construction', pp. 35–39.

Pantel, P. and Pennacchiotti, M. (2006), Espresso : Leveraging generic patterns for automatically harvesting semantic relations, *in* 'Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics', Association for Computational Linguistics, Sydney, Australia, pp. 113–120.
URL: *https ://www.aclweb.org/anthology/P06-1015*

Pennington, J., Socher, R. and Manning, C. (2014), GloVe : Global vectors for word representation, *in* 'Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)', Association for Computational Linguistics, Doha, Qatar, pp. 1532–1543.
URL: *https ://www.aclweb.org/anthology/D14-1162*

Peters, M., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K. and Zettlemoyer, L. (2018), Deep contextualized word representations, *in* 'Proceedings of

the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long Papers)', Association for Computational Linguistics, New Orleans, Louisiana, pp. 2227–2237.
URL: *https ://www.aclweb.org/anthology/N18-1202*

Poon, H. and Domingos, P. (2009), Unsupervised semantic parsing, *in* 'Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing', Association for Computational Linguistics, Singapore, pp. 1–10.
URL: *https ://www.aclweb.org/anthology/D09-1001*

Qi, P., Zhang, Y., Zhang, Y., Bolton, J. and Manning, C. D. (2020), Stanza : A python natural language processing toolkit for many human languages, *in* 'Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics : System Demonstrations', Association for Computational Linguistics, Online, pp. 101–108.
URL: *https ://www.aclweb.org/anthology/2020.acl-demos.14*

Qian, L., Zhou, G., Kong, F., Zhu, Q. and Qian, P. (2008), Exploiting constituent dependencies for tree kernel-based semantic relation extraction, *in* 'Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)', Coling 2008 Organizing Committee, Manchester, UK, pp. 697–704.
URL: *https ://www.aclweb.org/anthology/C08-1088*

Qian, L., Zhou, G., Zhu, Q. and Qian, P. (2007), Relation extraction using convolution tree kernel expanded with entity features, *in* 'Proceedings of the 21st Pacific Asia Conference on Language, Information and Computation', The Korean Society for Language and Information (KSLI), Seoul National University, Seoul, Korea, pp. 415–421.
URL: *https ://www.aclweb.org/anthology/Y07-1043*

Radford, A., Narasimhan, K., Salimans, T. and Sutskever, I. (2018), 'Improving language understanding by generative pre-training'.

Ro, Y., Lee, Y. and Kang, P. (2020), Multi^2OIE : Multilingual open information extraction based on multi-head attention with BERT, *in* 'Findings of the Association for Computational Linguistics : EMNLP 2020', Association for Computational Linguistics, Online, pp. 1107–1117.
URL: *https ://www.aclweb.org/anthology/2020.findings-emnlp.99*

Rosenfeld, B. and Feldman, R. (2007), Clustering for unsupervised relation identification, *in* 'Proceedings of the sixteenth ACM conference on Conference on information and knowledge management', pp. 411–418.

Roth, D. and Yih, W.-t. (2002), Probabilistic reasoning for entity and relation recognition, *in* 'COLING 2002 : The 19th International Conference on Computational Linguistics'.
URL: *https ://www.aclweb.org/anthology/C02-1151*

Roth, D. and Yih, W.-t. (2004), A linear programming formulation for global inference in natural language tasks, *in* 'Proceedings of the Eighth Conference on Computational Natural Language Learning (CoNLL-2004) at HLT-NAACL 2004', Association for Computational Linguistics, Boston, Massachusetts, USA, pp. 1–8.
URL: *https ://www.aclweb.org/anthology/W04-2401*

Roth, D. and Yih, W.-t. (2007), 'Global inference for entity and relation identification via a linear programming formulation', *Introduction to statistical relational learning* pp. 553–580.

Sanh, V., Wolf, T. and Ruder, S. (2019), A hierarchical multi-task approach for learning embeddings from semantic tasks, *in* 'Proceedings of the AAAI Conference on Artificial Intelligence', pp. 6949–6956.

Snell, J., Swersky, K. and Zemel, R. S. (2017), 'Prototypical networks for few-shot learning', *arXiv preprint arXiv :1703.05175* .

Soares, L. B., FitzGerald, N., Ling, J. and Kwiatkowski, T. (2019), 'Matching the blanks : Distributional similarity for relation learning', *arXiv preprint arXiv :1906.03158* .

Socher, R., Huval, B., Manning, C. D. and Ng, A. Y. (2012), Semantic compositionality through recursive matrix-vector spaces, *in* 'Proceedings of the 2012 joint conference on empirical methods in natural language processing and computational natural language learning', pp. 1201–1211.

Soderland, S., Gilmer, J., Bart, R., Etzioni, O. and Weld, D. S. (2013), Open information extraction to kbp relations in 3 hours., *in* 'TAC'.

Soderland, S., Roof, B., Qin, B., Xu, S., Etzioni, O. et al. (2010), 'Adapting open information extraction to domain-specific relations', *AI magazine* **31**(3), 93–102.

Srikant, R. and Agrawal, R. (1996), Mining sequential patterns : Generalizations and performance improvements, *in* 'International conference on extending database technology', Springer, pp. 1–17.

Stanovsky, G. and Dagan, I. (2016), Creating a large benchmark for open information extraction, *in* 'Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing', Association for Computational Linguistics, Austin, Texas, pp. 2300–2305.
URL: *https ://www.aclweb.org/anthology/D16-1252*

Stanovsky, G., Michael, J., Zettlemoyer, L. and Dagan, I. (2018*a*), Supervised open information extraction, *in* 'Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long Papers)', pp. 885–895.

Stanovsky, G., Michael, J., Zettlemoyer, L. and Dagan, I. (2018*b*), Supervised open information extraction, *in* 'Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long Papers)', Association for Computational Linguistics, New Orleans, Louisiana, pp. 885–895.
URL: *https ://www.aclweb.org/anthology/N18-1081*

Sun, C., Gong, Y., Wu, Y., Gong, M., Jiang, D., Lan, M., Sun, S. and Duan, N. (2019), Joint type inference on entities and relations via graph convolutional networks, *in* 'Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics', Association for Computational Linguistics, Florence, Italy, pp. 1361–1370.
URL: *https ://www.aclweb.org/anthology/P19-1131*

Sung, F., Yang, Y., Zhang, L., Xiang, T., Torr, P. H. and Hospedales, T. M. (2018), Learning to compare : Relation network for few-shot learning, *in* 'Proceedings of the IEEE conference on computer vision and pattern recognition', pp. 1199–1208.

Surdeanu, M., Tibshirani, J., Nallapati, R. and Manning, C. D. (2012), Multi-instance multi-label learning for relation extraction, *in* 'Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning', Association for Computational Linguistics, Jeju Island, Korea, pp. 455–465.
URL: *https ://www.aclweb.org/anthology/D12-1042*

Takanobu, R., Zhang, T., Liu, J. and Huang, M. (2019), A hierarchical framework for relation extraction with reinforcement learning, *in* 'Proceedings of the AAAI Conference on Artificial Intelligence', pp. 7072–7079.

Turian, J., Ratinov, L.-A. and Bengio, Y. (2010), Word representations : A simple and general method for semi-supervised learning, *in* 'Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics', Association for Computational Linguistics, Uppsala, Sweden, pp. 384–394.
URL: *https ://www.aclweb.org/anthology/P10-1040*

Turney, P. D. (2008), 'The latent relation mapping engine : Algorithm and experiments', *Journal of Artificial Intelligence Research* **33**, 615–655.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L. and Polosukhin, I. (2017), Attention is all you need, *in* 'NIPS'.
URL: *https ://arxiv.org/pdf/1706.03762.pdf*

Verga, P., Strubell, E. and McCallum, A. (2018), 'Simultaneously self-attending to all mentions for full-abstract biological relation extraction', *arXiv preprint arXiv :1802.10569* .

Wang, J. and Lu, W. (2020), Two are better than one : Joint entity and relation extraction with table-sequence encoders, *in* 'Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)', Association for Computational Linguistics, Online, pp. 1706–1721.
URL: *https ://www.aclweb.org/anthology/2020.emnlp-main.133*

Wang, X., Han, X., Lin, Y., Liu, Z. and Sun, M. (2018), Adversarial multi-lingual neural relation extraction, *in* 'Proceedings of the 27th International Conference on Computational Linguistics', pp. 1156–1166.

White, A. S., Reisinger, D., Sakaguchi, K., Vieira, T., Zhang, S., Rudinger, R., Rawlins, K. and Van Durme, B. (2016), Universal decompositional semantics on Universal Dependencies, *in* 'Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing', Association for Computational Linguistics, Austin, Texas, pp. 1713–1723.
URL: *https ://www.aclweb.org/anthology/D16-1177*

Wieting, J., Bansal, M., Gimpel, K. and Livescu, K. (2016), Charagram : Embedding words and sentences via character n-grams, *in* 'Proceedings of the 2016

Conference on Empirical Methods in Natural Language Processing', Association for Computational Linguistics, Austin, Texas, pp. 1504–1515.
URL: *https ://www.aclweb.org/anthology/D16-1157*

Wu, F. and Weld, D. S. (2010), Open information extraction using wikipedia, *in* 'Proceedings of the 48th annual meeting of the association for computational linguistics', Association for Computational Linguistics, pp. 118–127.

Xu, Y., Jia, R., Mou, L., Li, G., Chen, Y., Lu, Y. and Jin, Z. (2016), 'Improved relation classification by deep recurrent neural networks with data augmentation', *arXiv preprint arXiv :1601.03651* .

Yahya, M., Whang, S., Gupta, R. and Halevy, A. (2014), ReNoun : Fact extraction for nominal attributes, *in* 'Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)', Association for Computational Linguistics, Doha, Qatar, pp. 325–335.
URL: *https ://www.aclweb.org/anthology/D14-1038*

Yan, Y., Okazaki, N., Matsuo, Y., Yang, Z. and Ishizuka, M. (2009), Unsupervised relation extraction by mining Wikipedia texts using information from the web, *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', Association for Computational Linguistics, Suntec, Singapore, pp. 1021–1029.
URL: *https ://www.aclweb.org/anthology/P09-1115*

Yao, H., Zhu, D.-l., Jiang, B. and Yu, P. (2019), Negative log likelihood ratio loss for deep neural network classification, *in* 'Proceedings of the Future Technologies Conference', Springer, pp. 276–282.

Yao, L., Haghighi, A., Riedel, S. and McCallum, A. (2011), Structured relation discovery using generative models, *in* 'Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing', Association for Computational Linguistics, Edinburgh, Scotland, UK., pp. 1456–1466.
URL: *https ://www.aclweb.org/anthology/D11-1135*

Yao, L., Mao, C. and Luo, Y. (2019), 'Kg-bert : Bert for knowledge graph completion', *arXiv preprint arXiv :1909.03193* .

Yao, L., Riedel, S. and McCallum, A. (2012), Unsupervised relation discovery with sense disambiguation, *in* 'Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)', Association for Computational Linguistics, Jeju Island, Korea, pp. 712–720.
URL: *https ://www.aclweb.org/anthology/P12-1075*

Yao, L., Riedel, S. and McCallum, A. (2013), Universal schema for entity type prediction, *in* 'Proceedings of the 2013 Workshop on Automated Knowledge Base Construction', AKBC '13, Association for Computing Machinery, New York, NY, USA, p. 79–84.
URL: *https ://doi.org/10.1145/2509558.2509572*

Yates, A., Banko, M., Broadhead, M., Cafarella, M., Etzioni, O. and Soderland, S. (2007*a*), TextRunner : Open information extraction on the web, *in* 'Proceedings of Human Language Technologies : The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)', Association for Computational Linguistics, Rochester, New York, USA, pp. 25–26.
URL: *https ://www.aclweb.org/anthology/N07-4013*

Yates, A., Banko, M., Broadhead, M., Cafarella, M. J., Etzioni, O. and Soderland, S. (2007*b*), Textrunner : open information extraction on the web, *in* 'Proceedings of Human Language Technologies : The Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT)', pp. 25–26.

Ye, Z.-X. and Ling, Z.-H. (2019), 'Multi-level matching and aggregation network for few-shot relation classification', *arXiv preprint arXiv :1906.06678* .

Yu, M. and Dredze, M. (2014), Improving lexical embeddings with semantic knowledge, *in* 'Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2 : Short Papers)', Association for Computational Linguistics, Baltimore, Maryland, pp. 545–550.
URL: *https ://www.aclweb.org/anthology/P14-2089*

Zelenko, D., Aone, C. and Richardella, A. (2003), 'Kernel methods for relation extraction', *Journal of machine learning research* **3**(Feb), 1083–1106.

Zhan, J. and Zhao, H. (2019), 'Span based open information extraction', *arXiv preprint arXiv :1901.10879* .

Zhang, M., Zhang, J. and Su, J. (2006), Exploring syntactic features for relation extraction using a convolution tree kernel, *in* 'Proceedings of the Human Language Technology Conference of the NAACL, Main Conference', Association for Computational Linguistics, New York City, USA, pp. 288–295.
URL: *https ://www.aclweb.org/anthology/N06-1037*

Zhang, M., Zhang, Y. and Fu, G. (2017), End-to-end neural relation extraction with global optimization, *in* 'Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing', Association for Computational Linguistics, Copenhagen, Denmark, pp. 1730–1740.
URL: *https ://www.aclweb.org/anthology/D17-1182*

Zheng, S., Wang, F., Bao, H., Hao, Y., Zhou, P. and Xu, B. (2017), Joint extraction of entities and relations based on a novel tagging scheme, *in* 'Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)', Association for Computational Linguistics, Vancouver, Canada, pp. 1227–1236.
URL: *https ://www.aclweb.org/anthology/P17-1113*

Zhou, G., Su, J., Zhang, J. and Zhang, M. (2005), Exploring various knowledge in relation extraction, *in* 'Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)', Association for Computational Linguistics, Ann Arbor, Michigan, pp. 427–434.
URL: *https ://www.aclweb.org/anthology/P05-1053*

Zhou, G., Zhang, M., Ji, D. H. and Zhu, Q. (2007), Tree kernel-based relation extraction with context-sensitive structured parse tree information, *in* 'Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)', Association for Computational Linguistics, Prague, Czech Republic, pp. 728–736.
URL: *https ://www.aclweb.org/anthology/D07-1076*

Appendices

Annexe A

Liste des architectures

La figure 32 montre l'architecture du module d'extraction d'arguments et du prédicat appartenant au système *UD2OIE*. Après l'analyse syntaxique et la segmentation de la phrase, les vecteurs d'entrée du modèle sont obtenus en suivant la méthodologie décrite au chapitre 3. Ces vecteurs sont ensuite concaténés et transmis à la couche *BiLSTM*, chacun de ces vecteurs sont transmis à la couche dense qui permet de passer de vecteurs de taille n à des vecteurs de taille k correspondant au nombre de classes à prédire. Les k valeurs du vecteur correspondent aux probabilités de chacune des k classes. Les vecteurs de la couche dense sont transmise à la couche *CRF* responsable de l'inférence des bornes du prédicat. Ces bornes inférés sont concaténés avec les vecteurs de la couche *LSTM* afin de conditionner l'inférence des bornes des arguments sur les bornes du prédicat extraits à l'étape précédente. Une fois ces vecteurs concaténés, ils sont transmis à une couche dense pour obtenir en sortie des vecteurs de taille k' correspondant au nombre de classes des arguments. La sortie de la couche dense est alors transmise à la couche *CRF* responsable de l'inférence des bornes des arguments pour inférer les bornes des arguments. Nous tenons à préciser qu'a cette étape, seulement les bornes et non les types des arguments qui sont extraits.

La figure 33 montre l'architecture du module d'étiquetage des arguments appartenant au système *UD2OIE*. En plus des entrées qui sont les mêmes de celles du module d'extraction d'arguments et du prédicat, les entrées incluent aussi les bornes des arguments et du prédicat qui sont extraits par le module d'extraction d'arguments et du prédicat. Les vecteurs de ces entrées sont ensuite concaténés et sont transmis à la couche *LSTM*. La sortie de la couche *LSTM* est transmise à la couche dense qui est-elle même transmise à la couche *CRF* pour inférer les types de chaque argument extrait à l'étape précédente.

FIGURE 32 Extraction de la structure prédicat-arguments.

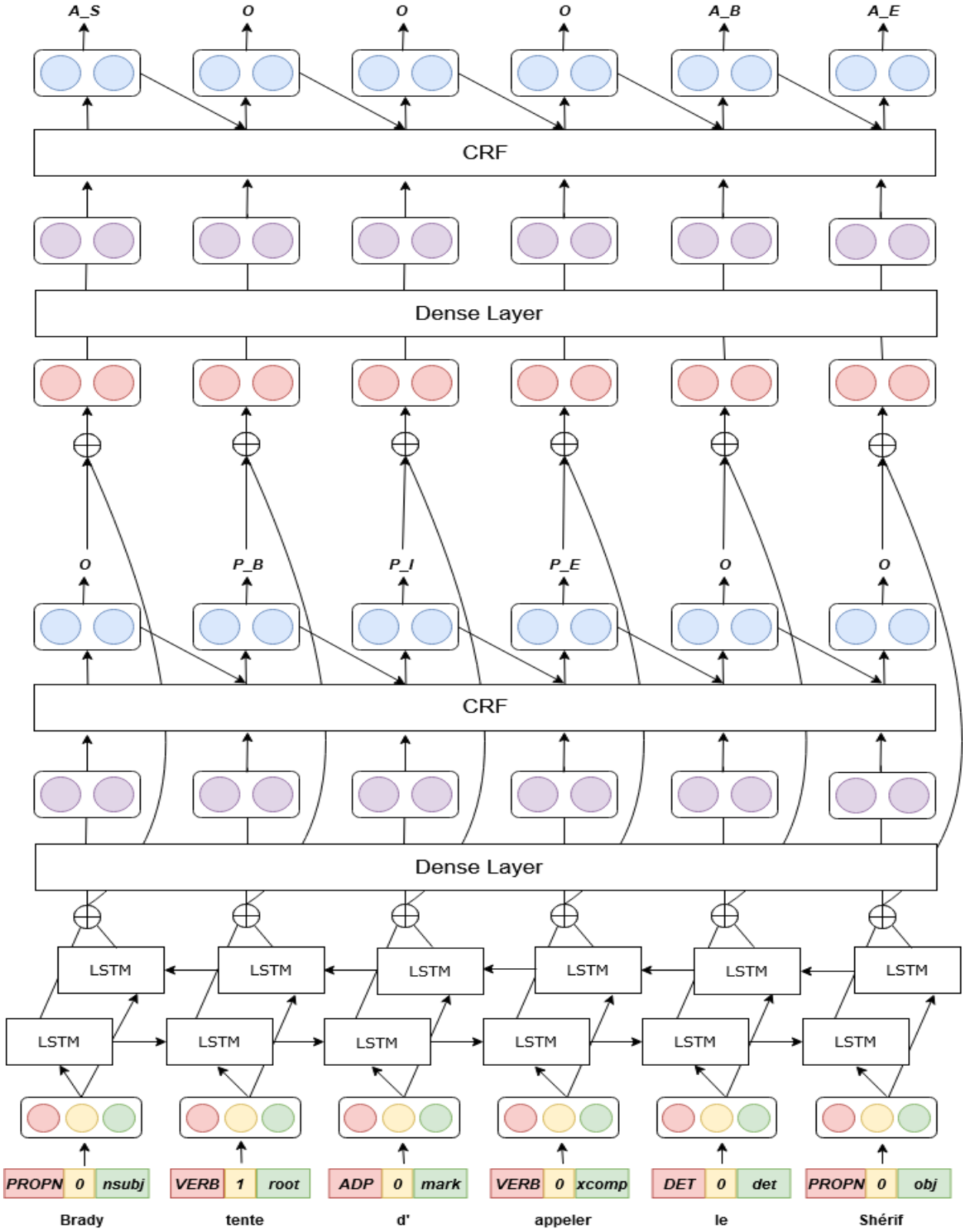
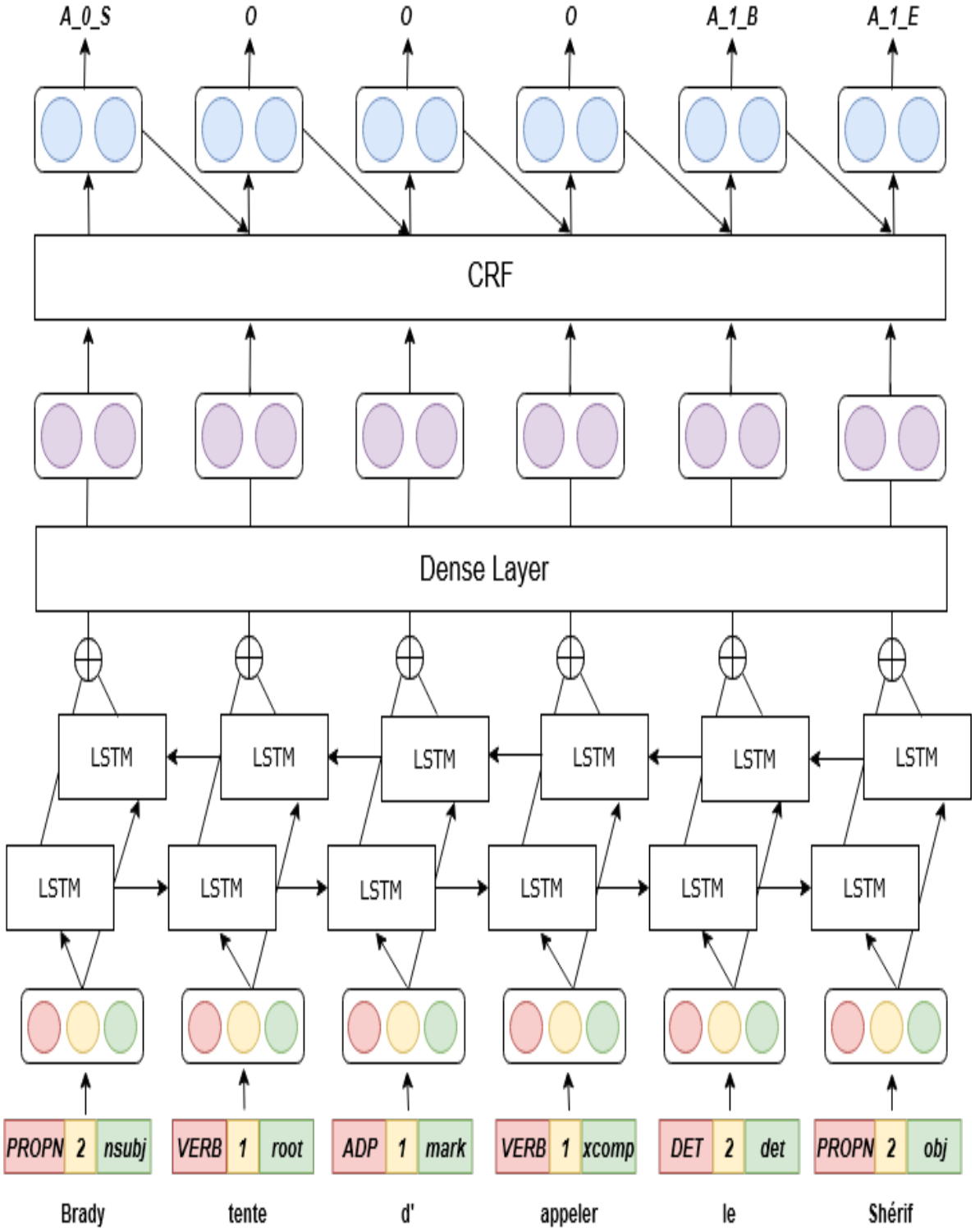


FIGURE 33 Étiqueteur des arguments.



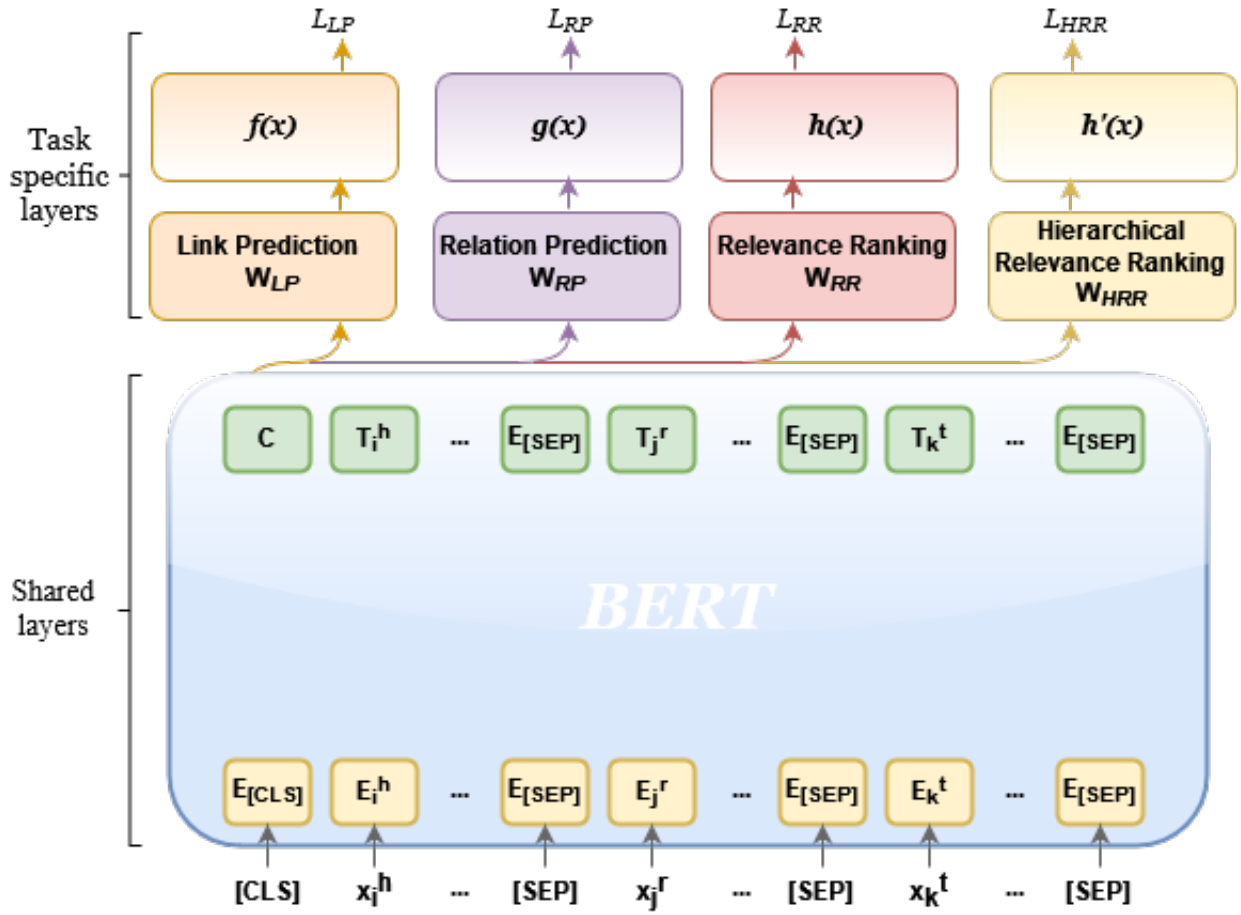


FIGURE 34 Architecture de la méthode d'apprentissage multi-tâche de *Jeux2BERT*.

La figure 34 illustre l'architecture du modèle *Jeux2BERT*, cette architecture prend en compte quatre tâches à optimiser : prédiction de liens, prédiction de relations, classement de pertinence et classement de pertinence hiérarchique. Le modèle prend en entrée un triplé issu de *JeuxDeMots* ou un couple si la tâche en cours et la prédiction de relations. Une fois l'entrée transmise au modèle, la sortie de la dernière couche du modèle *Flaubert* est considérée en calculant la moyenne des k vecteurs qui correspondent aux k jetons de l'entrée (k correspond à la longueur de la séquence d'entrée). Ensuite, le vecteur obtenu est transmis à une des quatre couches du modèle selon la tâche en cours. Chacune des couches correspond à une fonction optimisable qui va donner en sortie des probabilités ou un score de confiance selon la nature de la tâche.

La figure 35 illustre un exemple d'extraction de relations sémantiques nécessitant une étape de résolution de coréférence dans la chaîne de traitement.

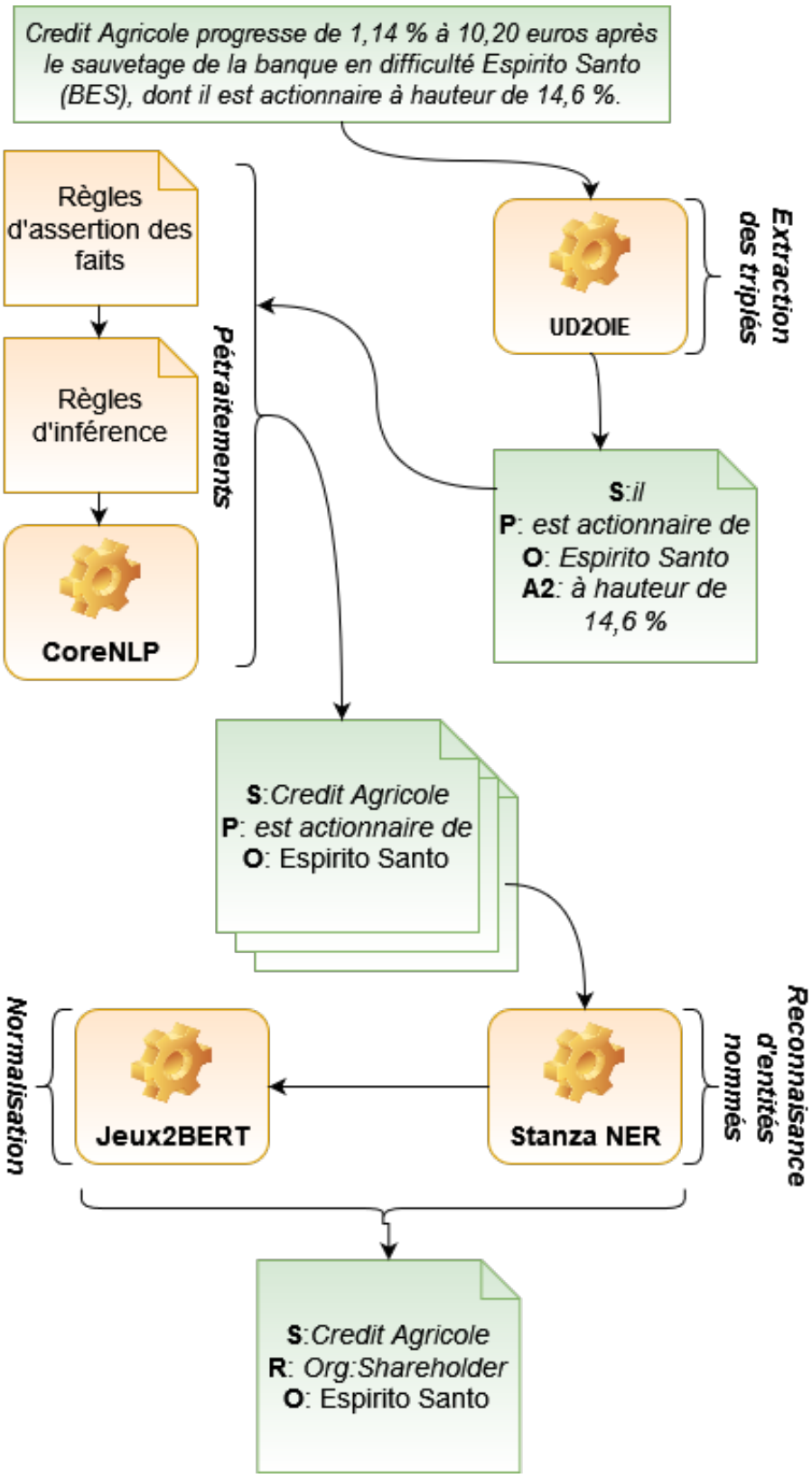


FIGURE 35 Figure illustrant le fonctionnement de notre système d'extraction de relations sémantiques. L'exemple inclut la résolution de coréférence dans la chaîne de traitement.

Annexe B

Sélection des relations de JDM pour l'apprentissage

Comme indiqué au chapitre 4, nous avons décidé de ne pas inclure toutes les relations de *JeuxDeMots* pour éviter d'avoir un modèle trop complexe. La difficulté d'apprentissage du modèle augmente avec le nombre de relations à modéliser. Empiriquement selon nos expériences pour entraîner le modèle sur deux types de relations (il faut deux classes au minimum), il est nécessaire d'avoir au moins 100 exemples par relation. Pour chaque exemple, il est nécessaire de fournir conjointement entre 2 et 3 contre-exemples. Cependant, cet ordre de grandeur peut varier selon la difficulté sémantique du type de relation.
Une relation lexico-sémantique (c'est à dire un type de relation de *JDM*) est d'autant plus difficile à apprendre par le système qu'il y a une grande variabilité de catégories d'attributs (features) pertinents pour le terme d'entrée et pour le terme de sortie. Par exemple pour la relation x *r_can_eat* y, si on suppose que globalement x doit être un animal et y un aliment, cette relation peut s'avérer relativement simple à apprendre et ne nécessiter d'un nombre raisonnable d'exemples. Cependant, pour *r_isa* ou pour *r_syn* il y a une grande variabilité de x et de y, rendant ces relations difficiles à discriminer (relativement aux autres) et demandant donc un plus grand nombre d'exemples et de contre-exemples.
Le nombre d'exemples à fournir pour chaque type de relation (un type de relation est finalement une classe qui va être oui ou non sélectionnée lors de l'utilisation du modèle neuronal pour la tâche de prédiction de relation) augmente en fonction du nombre de classes. Si nous devons entraîner le modèle sur trois types de relations, l'ordre de grandeur du nombre d'exemples nécessaires pour ce type passe de 100 à environ 150. Pour finir, si nous devons entraîner le modèle sur k types de relations, le nombre d'exemple sera de l'ordre de $50 \times k$ (et 2 à 3

fois lus de contre-exemples). Outre la difficulté potentielle à trouver suffisamment d'exemples, le temps de calcul pour l'apprentissage augmente (linéairement) avec le nombre d'exemples et de contre-exemples.
Une autre raison pour réduire le nombre de type de relations est due à une limite du modèle qui ne considère pas les dépendances qui existent entre un type de relation et sa relation inverse. Le modèle considère qu'il s'agit de deux types de relations sans rapport. La stratégie est donc d'entraîner le modèle sur une relation en ignorant la relation inverse. Lors de l'utilisation du modèle, on fait deux prédictions de type $a\ R\ b$ et $b\ R'\ a$. Selon les prédictions pour R et R', on pourra en déduire le vrai type de relation entre a et b. Par exemple, pour *chat* R *sauvage*, la meilleure prédiction est R = *r_carac*, et pour *sauvage* R' *chat* la meilleure prédiction est R' = *r_associated* (il y a bien un rapport associatif entre les deux termes), on peut raisonnablement en déduire que R est effectivement *r_carac*.

Pour finir, notre modèle est entraîné avec un grand nombre d'exemples et de contre-exemples et nous avons cherché à réduire autant que possible la taille des données d'apprentissage. Étant donné que nous sommes limités en termes de GPU (trois cartes Tesla V100), il nous a fallu nous "contenter" de 180 millions de relations (triplés) positives (environ 60 millions d'exemples) et négatives (environ 120 millions de contre-exemples, dont environ $350\ 000$ ne sont pas générés et viennent directement de JDM).
Il nous a fallu une semaine pour entraîner notre modèle *Jeux2BERT* pendant une itération, ainsi le temps total d'apprentissage du modèle est de un mois.

Dans le tableau ci-dessous, nous donnons la description des relations et nous indiquons celles qui ont été retenues (un *X* dans la première colonne du tableau). Les symboles suivants indiquent les raisons pour lesquelles la relation n'a pas été retenu : '*S*' pour une relation symétrique avec une autre, '*P*' pour une relation trop peu productive dans les corpus, '*I*' pour une relation non pertinente a priori dans le cadre de nos expérimentations

TABLE B.1 Définition et description des relations de JeuxDeMots.

	Type de la relation	Description
X	r_associated	Il est demandé d'énumérer les termes les plus étroitement associés au mot cible... Ce mot vous fait penser à quoi ?
X	r_raff_sem	Raffinement sémantique vers un usage particulier du terme source
I	r_domain	Il est demandé de fournir des domaines relatifs au mot cible. Par exemple, pour 'corner', on pourra donner les domaines 'football' ou 'sport'.
I	r_pos	Partie du discours (Nom, Verbe, Adjectif, Adverbe, etc.)
X	r_syn	Il est demandé d'énumérer les synonymes ou quasi-synonymes de ce terme.
X	r_isa	Il est demandé d'énumérer les GENERIQUES/hyperonymes du terme. Par exemple, 'animal' et 'mammifère' sont des génériques de 'chat'.
X	r_anto	Il est demandé d'énumérer des contraires du terme. Par exemple, 'chaud' est le contraire de 'froid'.
X	r_hypo	Il est demandé d'énumérer des SPECIFIQUES/hyponymes du terme. Par exemple, 'mouche', 'abeille', 'guêpe' pour 'insecte'.

suite page suivante

TABLE B.1 – suite de la page précédente

	Type de la relation	Description
X	r_has_part	Il faut donner des PARTIES/constituants/éléments (a pour méronymes) du mot cible. Par exemple, 'voiture' a comme parties : 'porte', 'roue', 'moteur',
X	r_holo	Il est demandé d'énumérer des 'TOUT' (a pour holonymes) de l'objet en question. Pour 'main', on aura 'bras', 'corps', 'personne', etc... Le tout est aussi l'ensemble comme 'classe' pour 'élève'.
I	r_locution	A partir d'un terme, il est demandé d'énumérer les locutions, expression ou mots composés en rapport avec ce terme. Par exemple, pour 'moulin', on pourra avoir 'moulin à vent', 'moulin à eau', 'moulin à café'. Pour 'vendre', on pourra avoir 'vendre la peau de l'ours avant de l'avoir tué', 'vendre à perte', etc..
X	r_agent	L'agent (qu'on appelle aussi le sujet) est l'entité qui effectue l'action, Ou la subit pour des formes passives ou des verbes d'état. Par exemple, dans - Le chat mange la souris -, l'agent est le chat. Des agents typiques de 'courir' peuvent être 'sportif', 'enfant',... (manger r_agent chat)

suite page suivante

TABLE B.1 – suite de la page précédente

	Type de la relation	Description
X	r_patient	Le patient (qu'on appelle aussi l'objet) est l'entité qui subit l'action. Par exemple dans - Le chat mange la souris -, le patient est la souris. Des patients typiques de 'manger' peuvent être 'viande', 'légume', 'pain', ... (manger r_patient pain)
X	r_lieu	Il est demandé d'énumérer les LIEUX typiques où peut se trouver le terme/objet en question. (carotte r_lieu potager)
X	r_instr	L'instrument est l'objet avec lequel on fait l'action. Dans - Il mange sa salade avec une fourchette -, fourchette est l'instrument. Des instruments typiques de 'tuer' peuvent être 'arme', 'pistolet', 'poison', ... (couper r_instr couteau)
I	r_carac	Pour un terme donné, souvent un objet, il est demandé d'en énumérer les Caractéristiques (adjectifs) possibles/typiques. Par exemple, 'liquide', 'froide', 'chaude', pour 'eau'.
X	r_lemma	Le lemme (par exemple 'mangent a pour lemme 'manger' ; 'avions' a pour lemme 'avion' ou 'avoir').
I	r_has_magn	La magnification ou amplification, par exemple - forte fièvre - ou - fièvre de cheval - pour fièvre. Ou encore - amour fou - pour amour, - peur bleue - pour peur.

suite page suivante

TABLE B.1 – suite de la page précédente

	Type de la relation	Description
I	r_has_antimagn	L'inverse de la magnification, par exemple - bruine - pour pluie.
X	r_family	Des mots de la même famille lexicale sont demandés (dérivation morphologique, par exemple). Par exemple, pour 'lait' on pourrait mettre 'laitier', 'laitage', 'laiterie', etc.
I	r_carac-1	Quels sont les objets (des noms) possédant typiquement/possiblement la caractéristique suivante ? Par exemple, 'soleil', 'feu', pour 'chaud'.
S	r_agent-1	Que peut faire ce SUJET ? (par exemple chat => miauler, griffer, etc.) (chat r_agent-1 manger)
S	r_instr-1	L'instrument est l'objet avec lequel on fait l'action. Dans - Il mange sa salade avec une fourchette -, fourchette est l'instrument. On demande ici, ce qu'on peut faire avec un instrument donné... (scie r_instr-1 scier)
S	r_patient-1	(inverse de r_patient) Que peut-on faire à cet OBJET. Pour 'pomme', on pourrait avoir 'manger', 'croquer', couper', 'éplucher', etc. (pomme r_patient-1 manger)
I	r_domain-1	inverse de r_domain : à un domaine, on associe des termes

suite page suivante

TABLE B.1 – suite de la page précédente

	Type de la relation	Description
S	r_lieu-1	A partir d'un lieu, il est demandé d'énumérer ce qui peut typiquement s'y trouver. Par exemple : Paris r_lieu-1 tour Eiffel
P	r_lieu_action	A partir d'un lieu, énumérer les actions typiques possibles dans ce lieu.
S	r_action_lieu	A partir d'une action (un verbe), énumérer les lieux typiques possibles où peut être réalisée cette action.
X	r_sentiment	Pour un terme donné, évoquer des mots liés à des SENTIMENTS ou des EMOTIONS que vous pourriez associer à ce terme. Par exemple, la joie, le plaisir, le dégoût, la peur, la haine, l'amour, l'indifférence, l'envie, avoir peur, horrible, etc.
I	r_manner	De quelles MANIERES peut être effectuée l'action (le verbe) proposée. Il s'agira d'un adverbe ou d'un équivalent comme une locution adverbiale , par exemple : 'rapidement', 'sur le pouce', 'goulûment', 'salement' ... pour 'manger'.
X	r_meaning/glose	Quels SENS/SIGNIFICATIONS pouvez-vous donner au terme proposé. Il s'agira de termes (des gloses) évoquant chacun des sens possibles, par exemple : 'forces de l'ordre', 'contrat d'assurance', 'police typographique', ... pour 'police'.

suite page suivante

TABLE B.1 – suite de la page précédente

	Type de la relation	Description
X	r_telic_role	Le rôle télique indique la fonction du nom ou du verbe. Par exemple, couper pour couteau, scier pour scie, etc. C'est le rôle qu'on lui destine communément pour un artéfact, ou bien un rôle qu'on peut attribuer à un objet naturel (réchauffer, éclairer pour soleil).
X	r_agentif_role	De quelle(s) manière(s) peut être CRÉE/CONSTRUIT le terme suivant. On demande des verbes transitifs (le terme en est un complément d'objet) qui DONNENT NAISSANCE à l'entité désignée par le terme, par exemple, 'construire' pour 'maison', 'rédiger'/'imprimer' pour 'livre' ou 'lettre'.
X	r_verbe-action	du verbe vers l'action. Par exemple, construire -> construction , jardiner -> jardinage . C'est un terme directement dérivé (ayant la même racine). Applicable que pour un verbe et inverse de la relation 34 (action vers verbe).

suite page suivante

TABLE B.1 – suite de la page précédente

	Type de la relation	Description
X	r_action-verbe	de l'action vers le verbe. Par exemple, construction -> construire, jardinage -> jardiner. C'est un terme directement dérivé (ayant la même racine). Applicable que pour un nom et inverse de la relation 33 (verbe vers action).
X	r_conseq	B (que vous devez donner) est une CONSÉQUENCE possible de A. A et B sont des verbes ou des noms. Exemples : tomber -> se blesser ; faim -> voler ; allumer -> incendie ; négligence –> accident ;
X	r_causatif	B (que vous devez donner) est une CAUSE possible de A. A et B sont des verbes ou des noms. Exemples : se blesser -> tomber ; vol -> pauvreté ; incendie -> négligence ; mort –> maladie ;
X	r_adj-verbe	Pour un adjectif de potentialité/possibilité, son verbe correspondant. Par exemple pour 'lavable' -> 'laver'
X	r_verbe-adj	Pour un verbe, son adjectif de potentialité/possibilité correspondant. Par exemple pour 'laver' -> 'lavable'

suite page suivante

TABLE B.1 – suite de la page précédente

	Type de la relation	Description
X	r_time	Donner une valeur temporelle -quel moment- peut-on associer au terme indiqué (par exemple 'dormir' -> nuit, 'bronzer' -> été, 'fatigue' -> 'soir')
I	r_object>mater	Quel est la ou les MATIÈRE/SUBSTANCE pouvant composer l'objet qui suit. Par exemple, 'bois' pour 'poutre'.
I	r_mater>object	Quel est la ou les CHOSES qui sont composés de la MATIÈRE/SUBSTANCE qui suit (exemple 'bois' -> poutre, table, ...).
P	r_successeur-time	Qu'est ce qui peut SUIVRE temporellement (par exemple Noêl -> jour de l'an, guerre -> paix, jour -> nuit, pluie -> beau temps, repas -> sieste, etc) le terme suivant
X	r_make	Que peut PRODUIRE le terme ? (par exemple abeille -> miel, usine -> voiture, agriculteur -> blé, moteur -> gaz carbonique ...)
X	r_product_of	Le terme est le RÉSULTAT/PRODUIT de qui/quoi ?
X	r_against	A quoi le terme suivant S'OPPOSE/COMBAT/EMPÊCHE ? Par exemple, un médicament s'oppose à la maladie.

suite page suivante

TABLE B.1 – suite de la page précédente

	Type de la relation	Description
S	r_against-1	Inverse de r_against (s'oppose à) - a comme opposition active (S'OPPOSE/COMBAT/EMPÊCHE). Par exemple, une bactérie à comme opposition antibiotique.
X	r_implication	Qu'est-ce que le terme implique logiquement ? Par exemple : ronfler implique dormir, courir implique se déplacer, câlin implique contact physique. (attention ce n'est pas la cause ni le but...)
I	r_quantificateur	Quantificateur(s) typique(s) pour le terme, indiquant une quantité. Par exemples, sucre -> grain, morceau - sel -> grain, pincée - herbe -> brin, touffe - ... -
X	r_masc	L'équivalent masculin du terme : lionne –> lion.
X	r_fem	L'équivalent féminin du terme : lion –> lionne.
I	r_equiv	Termes strictement équivalent/identique : acronymes et sigles (PS -> parti socialiste), apocopes (ciné -> cinéma), entités nommées (Louis XIV -> Le roi soleil), etc. (attention il ne s'agit pas de synonyme)
I	r_manner-1	Quelles ACTIONS (verbes) peut-on effectuer de cette manière ? Par exemple, rapidement -> courir, manger, ...

suite page suivante

TABLE B.1 – suite de la page précédente

	Type de la relation	Description
X	r_agentive_implication	Les verbes ou actions qui sont impliqués dans la création de l'objet. Par exemple pour 'construire' un livre, il faut, imprimer, relier, brocher. Il s'agit des étapes nécessaires à la réalisation du rôle agentif.
X	r_verb_real	Pour un verbe, celui qui réalise l'action (par dérivation morphologique). Par exemple, chasser -> chasseur, naviguer -> navigateur.
X	r_similar	Similaire/ressemble à ; par exemple le congre est similaire à une anguille
I	r_set>item	Quel est l'ELEMENT qui compose l'ENSEMBLE qui suit (par exemple, un essaim est composé d'abeilles)
I	r_item>set	Quel est l'ENSEMBLE qui est composé de l'ELEMENT qui suit (par exemple, un essaim est composé d'abeilles)
X	r_processus>agent	Quel est l'acteur de ce processus/événement ? Par exemple, 'nettoyage' peut avoir comme acteur 'technicien de surface'.
P	r_variante	Variantes du termes cible. Par exemple, yaourt, yahourt, ou encore évènement, événement.

suite page suivante

TABLE B.1 – suite de la page précédente

	Type de la relation	Description
X	r_syn_strict	Termes strictement substituables, pour des termes hors du domaine général, et pour la plupart des noms (exemple : endométriose intra-utérine –> adénomyose)
I	r_is_smaller_than	Qu'est-ce qui est physiquement plus gros que... (la comparaison doit être pertinente)
I	r_is_bigger_than	Qu'est-ce qui est physiquement moins gros que... (la comparaison doit être pertinente)
P	r_accomp	Est souvent accompagné de, se trouve avec... Par exemple : Astérix et Obelix, le pain et le fromage, les fraises et la chantilly
P	r_processus>patient	Quel est le patient de ce processus/événement ? Par exemple, 'découpe' peut avoir comme patient 'viande'.
I	r_verb_ppas	Le participe passé (au masculin singulier) du verbe infinitif. Par exemple, pour manger => mangé
P	r_cohypo	Il est demandé d'énumérer les CO-HYPONYMES du terme. Par exemple, 'chat' et 'tigre' sont des co-hyponymes (de 'félin').

suite page suivante

TABLE B.1 – suite de la page précédente

	Type de la relation	Description
I	r_verb_ppre	Le participe présent (au masculin singulier) du verbe infinitif. Par exemple, pour manger => mangeant
X	r_processus>instr	Quel est l'instrument/moyen de ce processus/événement ? Par exemple, 'découpe' peut avoir comme instrument 'couteau'.
I	r_pref_form	Indique une forme préférée
P	r_der_morpho	Des termes dérivés morphologiquement sont demandés. Par exemple, pour 'lait' on pourrait mettre 'laitier', 'laitage', 'laiterie', (mais pas 'lactose'). Pour 'jardin', on mettra 'jardinier', 'jardinage', 'jardiner'.
I	r_has_auteur	Quel est l'auteur de l'oeuvre suivante ?
I	r_has_personnage	Quels sont les personnages présents dans l'oeuvre qui suit ?
I	r_can_eat	De quoi peut se nourrir l'animal suivant ?
I	r_has_actors	A comme acteurs (pour un film ou similaire).
I	r_deplac_mode	Mode de déplacement. chat r_deplac_node marche
I	r_has_interpret	Interprète de personnages (cinéma ou théâtre)
I	r_color	A comme couleur(s)... chat r_color noir
X	r_cible	Cible de la maladie : myxomatose => lapin, rougeole => enfant,

suite page suivante

TABLE B.1 – suite de la page précédente

	Type de la relation	Description
X	r_symptomes	Symptômes de la maladie : myxomatose => yeux rouges, rougeole => boutons, ...
I	r_predecesseur-time	Qu'est ce qui peut PRÉCÉDER temporelle ment (par exemple - inverse de successeur) le terme suivant :
P	r_diagnostique	Diagnostique pour la maladie : diabète => prise de sang, rougeole => examen clinique, ...
I	r_predecesseur-space	Qu'est ce qui peut PRÉCÉDER spatialement (par exemple - inverse de successeur spatial) le terme suivant :
I	r_successeur-space	Qu'est ce qui peut SUIVRE spatialement (par exemple Locomotive à vapeur -> tender, wagon etc.) le terme suivant
I	r_social_tie	Relation sociale/familiale entre les individus... (annotation pour la nature exacte : frère, mari, etc.)
I	r_tributary	Tributaire de (physique ou spatial)
S	r_sentiment-1	Pour un SENTIMENT ou ÉMOTION donné, il est demandé d'énumérer les termes que vous pourriez associer. Par exemple, pour 'joie', on aurait 'cadeau', 'naissance', 'bonne nouvelle'

suite page suivante

TABLE B.1 – suite de la page précédente

	Type de la relation	Description
P	r_linked-with	A quoi est-ce relié (un wagon est relié à un autre wagon ou à une locomotive) ?
I	r_foncteur	La fonction de ce terme par rapport à d'autres. Pour les prépositions notamment, 'chez' => relation r_location.
X	r_but	But de l'action (nom ou verbe)
S	r_but-1	Quel sont les actions ou verbes qui ont le terme cible comme but ?
P	r_own	Que POSSÈDE le terme suivant ? (un soldat possède un fusil, une cavalière des bottes, ... soldat r_own fusil, ...)
P	r_own-1	Par qui ou quoi EST POSSÈDE le terme suivant ? (par exemple, fusil r_own-1 soldat)
I	r_verb_aux	Auxiliaire utilisé pour ce verbe
I	r_predecesseur-logic	Qu'est ce qui peut PRECEDER logiquement (par exemple : A précède B - inverse de successeur logique) le terme suivant
I	r_successeur-logic	Qu'est ce qui peut SUIVRE logiquement (par exemple A -> B, C etc.) le terme suivant

suite page suivante

TABLE B.1 – suite de la page précédente

	Type de la relation	Description
P	r_isa-incompatible	Relation d'incompatibilité pour les génériques. Si A r_isa-incompatible B alors X ne peut pas être à la fois A et B ou alors X est polysémique. Par exemple, poisson r_isa-incompatible oiseau. Colin est à la fois un oiseau et un poisson, donc colin est polysémique.
X	r_incompatible	Relation d'incompatibilité, ne doivent pas être présents ensembles. Par exemple, alcool r_incompatible antibiotique.
P	r_require	Il est demandé d'énumérer les termes nécessaires au mot mot cible... Par exemple, 'se reposer' => 'calme', ou 'pain' => 'farine'.
P	r_is_concerned_by	A peut être concerné par B. Par exemple, une personne a un rendez-vous a une maladie, une idée, une opinion
S	r_symptomes-1	Inverse de symptômes de la maladie : myxomatose => yeux rouges, rougeole => boutons,
P	r_promote	Qu'est-ce que le terme suivant FAVORISE ? Par exemple, un catalyseur favorise une réaction chimique.
I	r_circumstances	Les circonstances possibles pour un événement, ou un objet

suite page suivante

TABLE B.1 – suite de la page précédente

	Type de la relation	Description
I	r_has_auteur-1	Quel sont les oeuvres de l'auteur suivant ?
S	r_processus>agent-1	
S	r_processus>patient-1	
S	r_processus>instr-1	
P	r_pourvoyeur	Le pourvoyeur est l'entité qui fournit l'objet de l'action (le pourvoyeur est un complément d'objet indirect introduit par 'à',). Par exemple dans - Le client demande une pizza à la serveuse - le pourvoyeur est ici la serveuse
P	r_compl_agent	Le complément d'agent est celui qui effectue l'action dans les formes passives. Par exemple, pour 'être mangé', la souris est l'agent et le chat le complément d'agent
P	r_beneficiaire	Le bénéficiaire est l'entité qui tire bénéfice/préjudice de l'action (un complément d'objet indirect introduit par 'à', 'pour',). Par exemple dans - La sorcière donne une pomme à Blanche Neige -, la bénéficiaire est Blanche Neige ... .
I	r_descend_de	Descend de (évolution)...
P	r_domain_subst	Quels sont le ou les domaines de substitution pour ce terme quand il est utilisé comme domaine (par exemple, 'muscle' => 'anatomie du système musculaire')

suite page suivante

TABLE B.1 – suite de la page précédente

	Type de la relation	Description
X	r_prop	Pour le terme donné, il faut indiquer les noms de propriétés pertinents (par exemple pour 'voiture', le 'prix', la 'puissance', la 'longueur', le 'poids'. On ne met que des noms et pas des adjectifs).
I	r_activ_voice	Pour un verbe à la voix passive, sa voix active. Par exemple, pour 'être mangé' on aura 'manger'
P	r_make_use_of	Peut utiliser un objet ou produit (par exemple électricité pour frigo).
P	r_is_used_by	Est utilisé par (par exemple essence pour voiture).
I	r_adj-nomprop	Pour un adjectif, donner le nom de propriété correspondant. Par exemple, pour 'friable' -> 'friabilité'
I	r_nomprop-adj	Est utilisé par (par exemple essence pour voiture).
I	r_adj-adv	Pour un adjectif, donner l'adverbe correspondant. Par exemple, pour 'rapide' -> 'rapidement'
I	r_adv-adj	Pour un adverbe, donner l'adjectif correspondant. Par exemple, pour 'rapidement' -> 'rapide'
I	r_homophone	Il est demandé d'énumérer les homophones ou quasi-homophones de ce terme.
P	r_potential_confusion	Confusion potentielle avec un autre terme (par exemple, acre et âcre, détonner et détoner)

suite page suivante

TABLE B.1 – suite de la page précédente

	Type de la relation	Description
P	r_concerning	Qui concerne quelque chose ou quelqu'un. Par exemple : maladie r_concerning personne, ou disparition r_concerning emploi.
I	r_adj>nom	Le nom associé à l'adjectif. Par exemple , 'urinaire' -> 'urine'
I	r_nom>adj	L'adjectif associé au nom. Par exemple, 'urine' -> 'urinaire'
I	r_opinion_of	L'opinion de tel groupe ou telle personne. Utilisé comme relation d'annotation.
I	r_pos_seq	Séquence de parties du discours (Nom, Verbe, Adjectif, Adverbe,). Exemple : belle maison => Adj : Nom :
I	r_cooccurrence	co-occurences (non utilisée)
I	r_aki	(TOTAKI) equivalent pour TOTAKI de l'association libre
P	r_annotation_exception	Relation pour indiquer qu'il s'agit d'une exception par rapport à la cible. L'autruche ne vole pas, et c'est une exception par rapport à l'oiseau prototypique
X	r_inhib	Relation d'inhibition, le terme inhibe les termes suivants... ce terme a tendance à exclure le terme associé.
S	r_raff_sem-1	Inverse de r_raff_sem (automatique)

Annexe C

Lien vers les ressources

Lien vers les corpus d'évaluation de la tâche de classification de relations sémantiques:
https://github.com/atmani-massinissa/UD2OIE/blob/master/dataset/sem_rels.txt

Lien vers le système *UD2OIE* (https://github.com/atmani-massinissa/UD2OIE/), il englobe le code, le modèle et tout les corpus utilisés lors de la conception du modèle.

Lien vers le système *Jeux2BERT* (https://github.com/atmani-massinissa/Jeux2BERT), il contient le code nécessaire à la conception du modèle.

Lien vers le modèle *Jeux2BERT* (shorturl.at/foLR7), il contient le modèle final présenté dont les performances ont été rapportés lors des expérimentations.