Actes de la 28e Conférence sur le Traitement Automatique des Langues Naturelles
Lille, France, 28 juin au 2 juillet 2021

Open Information Extraction: Approche Supervisée et Syntaxique pour le Français

Massinissa Atmani[1, 2] Mathieu Lafourcade[1]
(1) LIRMM, 860 rue de St Priest, 34095 Montpellier, France
(2) Amaris Research Unit, 25 boulevard Eugène Deruelle, 69003 Lyon, France
massinissa.atmani@etu.umontpellier.fr, mathieu.lafourcade@lirmm.fr

RÉSUMÉ

L' Open Information Extraction, est un paradigme d'extraction conçu pour gérer l'adaptation de domaine, la principale difficulté des approches traditionnelles pour l'extraction d'informations. Cependant, la plupart des approches se concentrent sur l'anglais. Ainsi, nous proposons une approche supervisée pour l'OpenIE pour le français, nous développons également un corpus d'entraînement et un référentiel d'évaluation. Nous proposons un nouveau modèle basé en deux étapes pour l'étiquetage de séquence, qui identifie d'abord tous les arguments de la relation avant de les étiqueter. Les expérimentations montrent non seulement que l'approche que nous proposons obtient les meilleurs résultats, mais aussi que l'état de l'art actuel n'est pas assez robuste pour s'adapter à un domaine différent du domaine du corpus d'entraînement.

ABSTRACT

Supervised Syntactic Approach for French Open Information Extraction.

Most of Open Information Extraction approaches focus on English. Hence, we propose a supervised OpenIE for French, we also derive a training set and an evaluation benchmark for French OpenIE. We propose a new two-stage pipeline model for sequence labeling, that first identifies all the arguments of the relation and only then classifies them according to their most likely label. The experiments not only show that our proposed approach achieves the best results, but also that the current state-of-the-art approach is not cross-domain friendly and fails when facing out-of-domain data (their domain is different from the training-set's domain).

MOTS-CLÉS : Extraction d'information, Apprentissage machine, Syntaxe.

KEYWORDS: Information Extraction, Machine Learning, Syntax.

1 Introduction

L'Open Information Extraction (OpenIE) (Yates *et al.*, 2007) consiste à extraire des faits et des événements exprimés dans une phrase, à travers une représentation prédicat-argument. (Yates *et al.*, 2007) le présente comme *"un nouveau paradigme d'extraction qui facilite l'extraction de relations à partir de texte en considérant* ***l'indépendance de domaine*** *et qui s'adapte facilement à la diversité et à la taille du corpus du Web"*. De nombreuses tâches de TALN (Mausam, 2016) ont bénéficié de l'OpenIE telles que les réponses aux questions avec documents multiples (Fan *et al.*, 2019), l'induction de schéma d'événement (Balasubramanian *et al.*, 2013) et la génération de vecteur de mots (Stanovsky *et al.*, 2015).

Compte tenu de l'inexistence d'approches qui se focalisent sur l'OpenIE pour le Français, nous proposons une approche supervisée pour le Français à base de réseaux neuronaux. Pour ce faire, nous construisons un corpus d'entraînement pour le Français en traduisant des corpus Anglais exploités par (Corro & Gemulla, 2013). Nous annotons également un référentiel d'évaluation issu d'articles de journaux du domaine de la finance, qui est différent du domaine du corpus d'entraînement pour vérifier le critère d'indépendance de domaine.

Notre modèle proposé consiste en deux sous-modules faiblement couplés, le premier module est un modèle multi-tâches qui extrait la relation de 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 ou tag le 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 & 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. Dans notre configuration, les arcs prédits indiquent les arguments extraits de la relation de prédicat, ces arguments extraits seront raffinés et étiquetés à l'étape suivante.

Finalement, les résultats des expérimentations montrent que les approches basées sur le modèle de langage BERT (Devlin *et al.*, 2019) sont beaucoup moins performantes sur des échantillons de données issues de domaines qui sont différents de celui sur lequel le modèle de langage a été entraîné.

2 État de l'art d'OpenIE

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; Fader *et al.*, 2011). Des systèmes plus avancés ont considérablement amélioré les performances en exploitant un traitement linguistique plus avancé. (Corro & Gemulla, 2013) ont utilisés 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 prédéfinis. Le *Semantic Role Labelling* (SRL) consiste à étiqueter les mots d'une phrase avec leur rôle sémantique, tel que agent, thème et artefact. La tâche SRL est quelque peu similaire à la tâche OpenIE, et en raison de la disponibilité des ressources, (Christensen *et al.*, 2010) ont utilisés un analyseur SRL 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 proposés *RENOUN* pour extraire les relations nominales. (Pal & Mausam, 2016) ont conçus un système OpenIE adapté aux relations exprimées par des démonymes et des noms composés relationnels.
OPENIE4 a été conçu de la fusion des systèmes *SRLIE* (Christensen *et al.*, 2010) et *RelNoun* (Pal & Mausam, 2016). Ils ont augmentés *OpenIE4* avec un système d'OpenIE adapté aux relations numériques ainsi qu'un système analysant les conjonctions de coordination afin de concevoir *OpenIE5*.

(Gotti & Langlais, 2016) ont proposés un système d'OpenIE pour le français en adaptant le système *Reverb* (Fader *et al.*, 2011) afin qu'il extrait des fait simples à partir de Wikipédia français.

2.2 OpenIE multilingue

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 & Garcia, 2015) présentent ArgOIE qui prend comme entrée l'analyse syntaxique de dépendances au format CoNLL-X, identifie les structures d'argument dans l'analyse des dépendances et extrait un ensemble de propositions basique de chaque structure d'argument. ArgOIE supporte 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 syntaxique et peut en principe supporter toutes les langues supportés par Universal Dependency.
(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 supporter l'OpenIE dans toutes les langues supportées 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.

3 Méthodologie

Nous présentons notre méthode en détail dans cette section. Tout d'abord, nous présentons la manière dont nous formulons la tache de l'OpenIE ainsi qu'un aperçu de notre approche supervisée de l'OpenIE dans 3.1 et 3.2 respectivement. Enfin, nous décrivons la représentation des entrées et notre nouvelle architecture pour notre modèle d'OpenIE respectivement dans 3.3 & 3.4.

3.1 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*).

TABLE 1 – Exemple de la représentation de la sortie du modèle.

Phrase	Brady tente d' appeler 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)

3.2 Approche

S'inspirant de (Stanovsky *et al.*, 2018), 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 le 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 on déduit un masque binaire $M = (m_1, m_2, ..., m_n)$.

3.3 Représentation des entrées

Nos deux sous-modules exploitent les mêmes entrées, à l'exception du modèle d'argument qui attend les bornes inféré 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 obtenues en utilisant l'encodage *one-hot encoding* (encodage 1 parmi n) où chaque catégorie est assignée à un vecteur différent.

3.4 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 précédente.
Les entrées pour le sous-module sont la concaténation des trois vecteurs de caractéristiques : $E_{pos}, E_{dep}, E_{masque}$. Le premier représente le plongement vectoriel de l'étiquetage grammatical, le second représente le plongement vectoriel de la relation syntaxique et le troisième 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 Bi-LSTM pour les deux tâches et exploite une couche de champs aléatoires conditionnels (CRF) (Lafferty *et al.*, 2001) pour chaque tâche.
Étant donné 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 :

$$x_i = E_{pos}(w_i) \oplus E_{dep}(w_i) \oplus E_{masque}(w_i) \tag{1}$$

Le vecteur de caractéristiques 1 est transmis au Bi-LSTM, 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 précédent et suivant du Bi-LSTM est calculée pour chaque mot et est transmise à une couche dense.

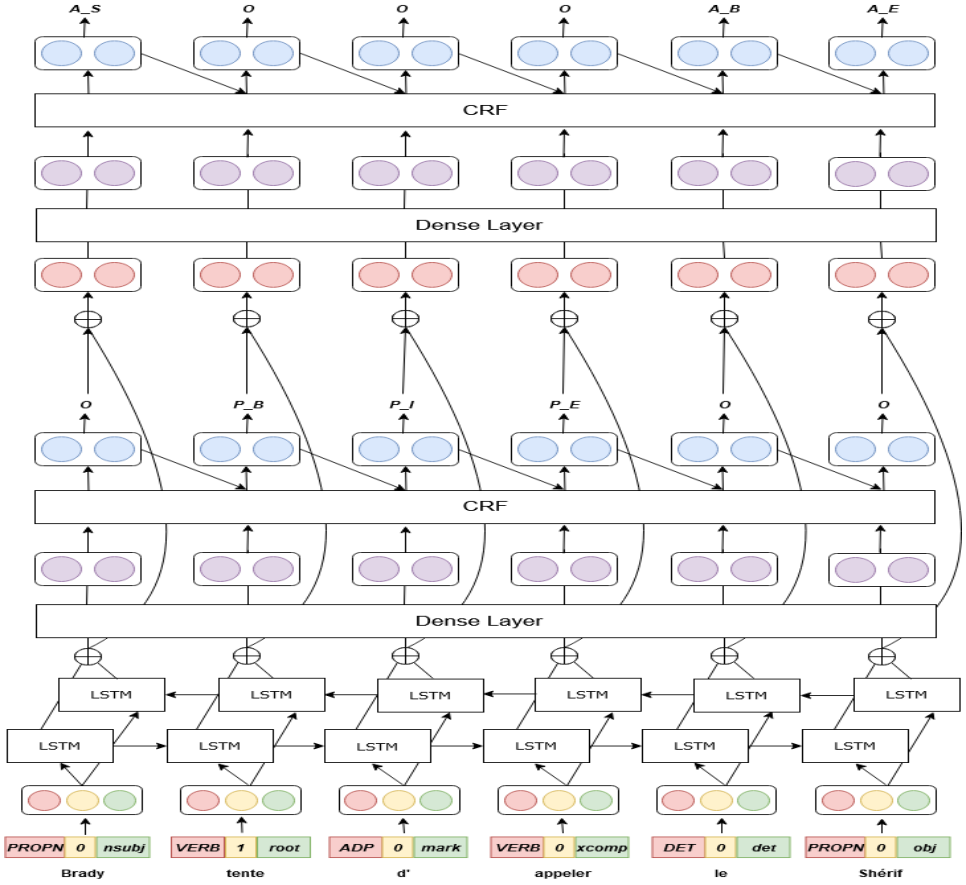
$$v_i^{\rightarrow}, v_i^{\leftarrow} = Bi - LSTM(x_i) \tag{2}$$

$$u_i = AVG(v_i^{\rightarrow}, v_i^{\leftarrow}) \tag{3}$$

$$h_i = Wu_i + b \tag{4}$$

Ensuite, la représentation est transmise au décodeur de chaque tâche. Puisque les deux tâches utilisent comme décodeur les champs aléatoires conditionnels (CRF) , nous introduisons le décodeur CRF.

FIGURE 1 – Extraction des prédicat-arguments



3.4.1 Décodeur CRF

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 $S(H, Y)$.

$$S(H, Y) = \sum_{i=1}^{n-1} A_{y_i, y_{i+1}} + \sum_{i=1}^{n} H_{i, y_i} \tag{5}$$

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 calculé, une probabilité conditionnelle sur toutes les séquences d'étiquettes possibles Y en utilisant Softmax, où Y_H représente les séquences d'étiquettes possibles pour H.

$$p(Y|H) = \frac{e^{S(H,Y)}}{\sum_{Y' \in Y_H} e^{S(H,Y')}} \tag{6}$$

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

$$y^* = argmax_{Y \in Y_H} S(H, Y) \tag{7}$$

La sortie de l'encodeur 4 est d'abord transmise à l'extracteur de prédicat, qui identifie le prédicat. Après avoir extrait le prédicat 7, 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 4 est concaténée avec la sortie du décodeur CRF du prédicat 7 et est envoyée au décodeur CRF de l'extracteur d'arguments. La nouvelle représentation est donnée par l'équation suivante :

$$h_i(Argument) = h_i \oplus y_i(Predicat) \tag{8}$$

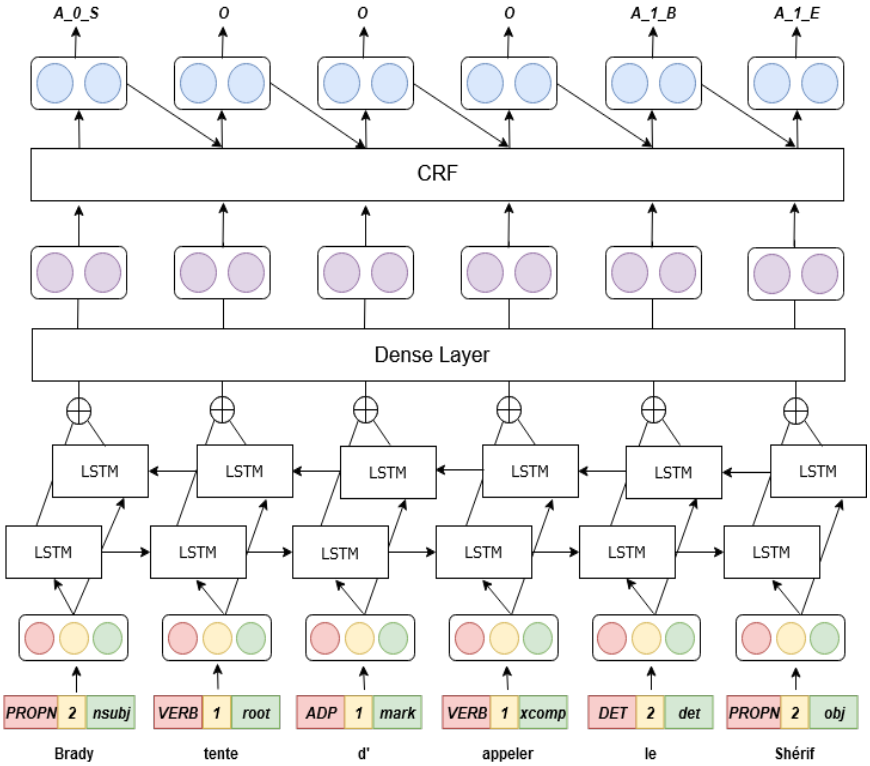
Les deux tâches sont optimisées conjointement et nous maximisons la vraisemblance logarithmique de la séquence d'étiquettes correcte de chaque tâche sur l'ensemble d'apprentissage $\{(H_j, Y_j)\}$, en minimisant la fonction de coût : la Negative Log Likelihood (NLL) (Yao *et al.*, 2019).

$$NLL = -\sum_j \log p(Y|H) \tag{9}$$

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

$$NLL = -\sum_j \log p(Y|H)_{predicat} - \sum_j \log p(Y|H)_{argument} \tag{10}$$

FIGURE 2 – Étiqueteur des arguments



3.5 Étiquetage des arguments

Après l'extraction de la structure prédicat-argument, le premier sous-module alimente le prédicat et les arguments extraits par le deuxième sous-module. L'entrée du modèle est le vecteur de caractéristiques défini par 11, et consiste en la concaténation des vecteurs représentant le plongement vectoriel de

l'étiquetage grammatical, le plongement vectoriel de la relation de dépendance syntaxique et E_{pr-arg}, le vecteur inféré dans la première étape qui représente le prédicat et les arguments extraits.

$$x_i = E_{pos}(w_i) \oplus E_{dep}(w_i) \oplus E_{pr-arg}(w_i) \tag{11}$$

Le sous-module exploite la même architecture que le premier sous-module qui consiste en un décodeur CRF empilé sur une couche Bi-LSTM 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.

4 Expérimentations

Dans cette section, les corpus de données d'entraînement et les hyper-paramètres sont respectivement présentés dans 4.2 et 4.1, puis 4.3 et 4.4 décrivent le référentiel d'évaluation et la stratégie d'évaluation. Finalement, nous présentons l'étude d'impact de l'architecture et les systèmes de référence dans 4.5 et 4.6.

4.1 Hyperparameters

Le tableau 2 ci-dessous, reprend les hyper-paramètres de notre modèle, qui sont les mêmes pour les deux sous-modules. Nous avons entraîné notre modèle à l'aide de l'optimiseur Adam.

TABLE 2 – 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

Dataset	Domaine	#Phrases	#Relations
ReVerb	Yahoo	500	1 551
NYT	New York Times	200	642
Wiki	Wikipedia	200	568

TABLE 3 – Données d'apprentissage

4.2 Données d'apprentissage

Comme aucun corpus n'existe pour le Français dans la tâche d'OpenIE, nous décidons de construire nous même un corpus d'apprentissage pour le Français. Pour ce faire, nous optons pour une approche

semi-supervisée dans laquelle nous traduisons automatiquement des corpus en anglais (Corro & Gemulla, 2013) vers le Français en utilisant l'API de Google [1], nous faisons une deuxième passe pour manuellement corriger les erreurs de traduction. La description du corpus obtenu est décrite dans le tableau 3.

4.3 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 décrit 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 extractions, 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 meilleur annotation de la relation. Nous avons décidé de laisser 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).

— Anaphore : Contrairement aux recommandations d'annotation, nous avons opté pour la non-résolution d'anaphores dans les extractions.

— Appositions : nous avons aussi décidé d'inclure l'extraction introduite par l'apposition alors que cette extraction peut être considéré 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 se compose de 506 phrases et 1783 relations.

4.4 Évaluations

Nous évaluons les différentes baselines en utilisant le framework d'évaluation proposé avec le référentiel d'évaluation standard CARB (Bhardwaj *et al.*, 2019). Nous rapportons le F1 et l'AUC (Area Under the Curve) score. Les systèmes de référence sont évaluées en exploitant le code de (Kolluru *et al.*, 2020).

4.5 Impact de l'architecture

Nous considérons une étude additionnelle pour étudier l'impact de notre nouvelle architecture, qui vise à séparer l'identification et l'étiquetage des arguments. Par conséquent, nous considérons comme système de référence le modèle *SpanOIE* présenté par (Zhan & Zhao, 2019). L'architecture de notre modèle est la même que l'architecture utilisé par (Zhan & Zhao, 2019) sauf que la notre se distingue en dissociant l'identification et la classification des arguments. En effet, notre architecture introduit une étape auxiliaire pour identifier les arguments du prédicat extrait avant d'étiqueter ces arguments,

1. https ://github.com/ssut/py-googletrans

tandis que (Zhan & Zhao, 2019) identifie et étiquette les arguments du prédicat extrait simultanément.

4.6 Systèmes de référence

Comme systèmes de référence, nous avons choisis le système à base de règle PredPatt(White *et al.*, 2016) et Multi2OIE(Ro *et al.*, 2020). Nous considérons deux variants pour notre modèle, le premier *FR-OIE* notre modèle entraîné en utilisant le corpus d'entraînement en français tandis que le deuxième *FR-OIE(En)* est le modèle entraîné sur le corpus d'entraînement original et qui est en anglais. Nous avons choisis d'inclure cet autre variant afin d'avoir une évaluation plus équitable et correcte avec Multi2OIE.

5 Résultats et discussion

Cette section présente les principales conclusions des résultats de l'expérience dans 5.1. Les résultats d'indépendance de domaine et de l'étude de l'impact de l'architecture sont discutés dans 5.2 et 5.3. Enfin, 5.4 présente une analyse des erreurs du modèle.

5.1 Performances

Les performances de chaque système par rapport au référentiel d'évaluation avec les différentes métriques sont rapportées dans le tableau 4. Les résultats de l'évaluation montrent que la méthode que nous proposons surpasse largement les autres baselines. Un autre résultat est que nous constatons est que le deuxième variant de notre modèle obtient de meilleurs résultats que Multi2OIE, qui est entraîné sur un corpus d'entraînement en anglais comme le deuxième variant de notre modèle. Ils semblent démontrer aussi que Multi2OIE basé sur BERT obtient des résultats légèrement meilleurs que l'approche à base de règles, PredPatt.

Système	Précision	Rappel	score F1	AUC
PredPatt (White *et al.*, 2016)	0.323	0.524	0.42	0.347
Multi2OIE (Ro *et al.*, 2020)	0.688	0.315	0.432	0.245
FR-OIE	0.727	0.627	**0.673**	0.496
FR-OIE(En)	0.702	0.596	**0.644**	0.461

TABLE 4 – Résultats d'évaluation des baselines et de notre approche sur le référentiel d'évaluation

5.2 L'indépendance de domaine

Afin de vérifier notre hypothèse, nous comparons aussi Multi2OIE et PredPatt sur un de nos corpus d'apprentissage issu de Wikipedia, le même domaine de donnée sur lequel a été pré-entraîné le modèle de langage BERT. Les résultats qui sont rapportés dans la table 5, montrent que contrairement au corpus issu du domaine de la finance, Multi2OIE dépasse largement PredPatt pour la précision et le score F1. Ces résultats montrent que les approches actuelles basées sur BERT ne supportent pas

Système	Précision	Rappel	F1	AUC
PredPatt (White *et al.*, 2016)	0.318	0.461	0.376	0.304
Multi2OIE (Ro *et al.*, 2020)	0.686	0.44	**0.536**	0.329

TABLE 5 – Résultats d'évaluation des baselines sur le corpus Wikipedia.

l'adaptation au domaine, qui est pourtant un critère essentiel dans l'OpenIE. Comme rapporté par (Li *et al.*, 2020), malgré leur habilité à extraire des représentations multilingues, les modèles de langage tel que BERT ne capturent que les caractéristiques spécifiques au domaine d'échantillon de données et n'extraient pas des caractéristiques indépendantes du domaine d'échantillon de données.

5.3 Résultats de l'étude de l'impact de l'architecture

L'architecture du modèle *FR-OIE(-Identification arguments)* correspond à celle utilisé par *SpanOIE* (Zhan & Zhao, 2019). Les résultats de l'impact de l'architecture rapportés dans 6, montrent que l'architecture proposée offre un gain de performance notable. Notre architecture proposée cible la performance du rappel, elle améliore la performance du rappel tout en entraînant une baisse de performance de la précision. Nous attribuons cela au fait de rechercher 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 du rappel augmente à mesure que le nombre de relations prédicat-argument augmente. Cependant, des relations prédicat-argument plus erronées seront propagées au module chargé d'étiqueter les arguments, ce dernier cherche uniquement à étiqueter les arguments extraits et ne peut pas rejeter ou détecter les arguments erronés, ce qui entraîne une baisse de performance de la précision.

Système	Précision	Rappel	F1	AUC
FR-OIE	0.727	**0.627**	**0.673**	**0.496**
FR-OIE(-Identification des arguments)	**0.746**	0.563	0.642	0.447

TABLE 6 – Résultats de l'étude d'impact de l'architecture

5.4 Analyse des erreurs

Comme prévu, la principale source d'erreurs était due aux erreurs de propagation de l'analyseur. Nous constatons que notre système échoue face aux constructions linguistiques complexes.
Le troisième exemple de 7 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 n'arrive pas à détecter l'ellipse dans la phrase et alimente un arbre syntaxique incorrect dans notre modèle.
Une autre source d'erreur importante 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 de l'objet. Le premier exemple de 7 montre un exemple dû à l'ambiguïté de l'attachement de préposition, où *en juillet 2010* est extrait dans le champ de l'objet *par la Hong Kong Monetary Authority*.
De plus, notre système échoue plus souvent à extraire des relations ayant comme prédicat des nominales, comme indiqué dans 7.
Enfin, le dernier exemple de 7 est relatif à au deuxième variant de notre modèle et qui est entraîné sur un corpus en anglais, il montre une construction linguistique spécifique au français (*objet indirect agentif* (exprimé par **iobj :agent** dans l'arbre syntaxique au format Universal Dependency) où l'agent initial (le pronom *lui* dans l'exemple) a été rétrogradé et est devenu un objet indirect. Puisque le deuxième variant de notre modèle a été entraîné sur un corpus en anglaises, il échouera naturellement face à des constructions spécifiques au français.

Type d'erreurs	Exemple
Arguments n-aire	Chinese Yuan Offshore (abréviation : CNH) monnaie chinoise lancée en juillet 2010 par la Hong Kong Monetary Authority (HKMA) **Extraction :** (**A0 :**Chinese Yuan Offshore **; P :**lancée **;** **A1 :**en juillet 2010 par la Hong Kong Monetary Authority) **Référence :** (**A0 :**Chinese Yuan Offshore **; P :**lancée **;** **A1 :**par la Hong Kong Monetary Authority **; A2 :**en juillet 2010)
Relations nominales	L'appétit de les entreprises pour la devise chinoise est bridé par les contrôles des capitaux. (**A0 :**les entreprises ; **P :**[ont un] appétit ; **A1 :**pour la devise chinoise)
Constructions linguistiques complexes ou ambiguës	L'objectif de cours de l' équipementier passe de 22 à 25,5 euros et celui du constructeur du Rafale de 1.100 à 1.260 euros. **Extraction :** (**A0 :**L'objectif de cours de l' équipementier **;** **P :**passe **; A1 :**de 22 **; A2 :**à 25,5 euros) **Référence :** (**A0 :**L'objectif de cours de l' équipementier **;** **P :**passe **; A1 :**de 22 **; A2 :**à 25,5 euros) **Référence** (**A0 :**L'objectif de cours du constructeur du Rafale **;** **P :**passe **; A1 :**de 1.100 **; A2 :**à 1.260 euros)
Constructions linguistiques propres au français	Google et Facebook en embuscade face à Apple, seul Google lui tient un peu tête. **Extraction :** (**A0 :**Google **; P :**tient un peu tête **;**) **Référence** (**A0 :**Google **; P :**tient un peu tête **; A1 :**lui)

TABLE 7 – Analyse et types des erreurs les plus récurrentes

6 Conclusion

Dans cet article, nous avons proposé une première approche d'OpenIE pour le Français, à base de réseaux de neurones. Notre modèle proposé introduit une étape auxiliaire pour identifier les arguments avant leur étiquetage et obtient les meilleurs résultats. Comme les référentiels d'évaluation actuels ne sont pas assez divers (Wikipedia et actualités), nous proposons un référentiel d'évaluation issu du domaine de la finance pour évaluer la performance et la robustesse des systèmes d'une manière plus précise. Le code du prototype ainsi que les données sont disponibles dans le répertoire du projet. [2]

Références

BALASUBRAMANIAN N., SODERLAND S., ETZIONI O. *et al.* (2013). Generating coherent event schemas at scale. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, p. 1721–1731.

BHARDWAJ S., AGGARWAL S. & 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)*, p. 6262–6267, Hong Kong, China : Association for Computational Linguistics. DOI : 10.18653/v1/D19-1651.

CHRISTENSEN J., MAUSAM, SODERLAND S. & 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*, p. 52–60, Los Angeles, California : Association for Computational Linguistics.

CORRO L. D. & GEMULLA R. (2013). Clausie : clause-based open information extraction. In *Proceedings of the 22nd international conference on World Wide Web*, p. 355–366.

DEVLIN J., CHANG M.-W., LEE K. & 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)*, p. 4171–4186, Minneapolis, Minnesota : Association for Computational Linguistics. DOI : 10.18653/v1/N19-1423.

DOZAT T. & MANNING C. D. (2016). Deep biaffine attention for neural dependency parsing. *arXiv preprint arXiv :1611.01734*.

FADER A., SODERLAND S. & ETZIONI O. (2011). Identifying relations for open information extraction. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, p. 1535–1545, Edinburgh, Scotland, UK. : Association for Computational Linguistics.

FAN A., GARDENT C., BRAUD C. & BORDES A. (2019). Using local knowledge graph construction to scale seq2seq models to multi-document inputs. *arXiv preprint arXiv :1910.08435*.

FORNEY G. (1973). The viterbi algorithm. *Proceedings of the IEEE*, **61**(3), 268–278. DOI : 10.1109/PROC.1973.9030.

GAMALLO P. & GARCIA M. (2015). Multilingual open information extraction. In *Portuguese Conference on Artificial Intelligence*, p. 711–722 : Springer.

2. https://github.com/atmani-massinissa/UD2OIE

GOTTI F. & LANGLAIS P. (2016). From french wikipedia to erudit : A test case for cross-domain open information extraction. *Computational Intelligence*. DOI : 10.1111/coin.12120.

JABBARI A., SAUVAGE O., ZEINE H. & 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*, p. 2293–2299, Marseille, France : European Language Resources Association.

KOLLURU K., ADLAKHA V., AGGARWAL S., MAUSAM & 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)*, p. 3748–3761, Online : Association for Computational Linguistics.

LAFFERTY J., MCCALLUM A. & 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)*, p. 282–289.

LECHELLE W., GOTTI F. & LANGLAIS P. (2019). WiRe57 : A fine-grained benchmark for open information extraction. In *Proceedings of the 13th Linguistic Annotation Workshop*, p. 6–15, Florence, Italy : Association for Computational Linguistics. DOI : 10.18653/v1/W19-4002.

LI J., HE R., YE H., NG H. T., BING L. & YAN R. (2020). Unsupervised domain adaptation of a pretrained cross-lingual language model. *arXiv preprint arXiv :2011.11499*.

MAUSAM M. (2016). Open information extraction systems and downstream applications. In *Proceedings of the twenty-fifth international joint conference on artificial intelligence*, p. 4074–4077.

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. & ZEMAN D. (2016). Universal Dependencies v1 : A multilingual treebank collection. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, p. 1659–1666, Portorož, Slovenia : European Language Resources Association (ELRA).

PAL H. & MAUSAM (2016). Demonyms and compound relational nouns in nominal open IE. In *Proceedings of the 5th Workshop on Automated Knowledge Base Construction*, p. 35–39, San Diego, CA : Association for Computational Linguistics. DOI : 10.18653/v1/W16-1307.

QI P., ZHANG Y., ZHANG Y., BOLTON J. & 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*, p. 101–108, Online : Association for Computational Linguistics. DOI : 10.18653/v1/2020.acl-demos.14.

RO Y., LEE Y. & 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*, p. 1107–1117, Online : Association for Computational Linguistics.

STANOVSKY G., DAGAN I. *et al.* (2015). Open ie as an intermediate structure for semantic tasks. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2 : Short Papers)*, p. 303–308.

STANOVSKY G., MICHAEL J., ZETTLEMOYER L. & DAGAN I. (2018). 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)*, p. 885–895, New Orleans, Louisiana : Association for Computational Linguistics. DOI : 10.18653/v1/N18-1081.

WHITE A. S., REISINGER D., SAKAGUCHI K., VIEIRA T., ZHANG S., RUDINGER R., RAWLINS K. & VAN DURME B. (2016). Universal decompositional semantics on Universal Dependencies. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, p. 1713–1723, Austin, Texas : Association for Computational Linguistics. DOI : 10.18653/v1/D16-1177.

YAHYA M., WHANG S., GUPTA R. & HALEVY A. (2014). ReNoun : Fact extraction for nominal attributes. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, p. 325–335, Doha, Qatar : Association for Computational Linguistics. DOI : 10.3115/v1/D14-1038.

YAO H., ZHU D.-L., JIANG B. & YU P. (2019). Negative log likelihood ratio loss for deep neural network classification. In *Proceedings of the Future Technologies Conference*, p. 276–282 : Springer.

YATES A., BANKO M., BROADHEAD M., CAFARELLA M., ETZIONI O. & SODERLAND S. (2007). 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)*, p. 25–26, Rochester, New York, USA : Association for Computational Linguistics.

ZHAN J. & ZHAO H. (2019). Span based open information extraction. *arXiv preprint arXiv :1901.10879*.