

## Relation, es-tu là ? Détection de relations par LSTM pour améliorer l'extraction de relations

Cyrielle Mallart<sup>1, 2, 3</sup> Michel Le Nouy<sup>1</sup> Guillaume Gravier<sup>3</sup> Pascale Sébillot<sup>2</sup>

(1) SIPA Ouest-France, 10 rue du Breil, 35000 Rennes, France

(2) INSA Rennes, IRISA, Campus de Beaulieu, 35042 Rennes, France

(3) CNRS, IRISA, Campus de Beaulieu, 35042 Rennes, France

cyrielle.mallart@ouest-france.fr, michel.lenouy@ouest-france.fr,  
guig@irisa.fr, pascale.sebillot@irisa.fr

### RÉSUMÉ

---

De nombreuses méthodes d'extraction et de classification de relations ont été proposées et testées sur des données de référence. Cependant, dans des données réelles, le nombre de relations potentielles est énorme et les heuristiques souvent utilisées pour distinguer de vraies relations de co-occurrences fortuites ne détectent pas les signaux faibles pourtant importants. Dans cet article, nous étudions l'apport d'un modèle de détection de relations, identifiant si un couple d'entités dans une phrase exprime ou non une relation, en tant qu'étape préliminaire à la classification des relations. Notre modèle s'appuie sur le plus court chemin de dépendances entre deux entités, modélisé par un LSTM et combiné avec les types des entités. Sur la tâche de détection de relations, nous obtenons de meilleurs résultats qu'un modèle état de l'art pour la classification de relations, avec une robustesse accrue aux relations inédites. Nous montrons aussi qu'une détection binaire en amont d'un modèle de classification améliore significativement ce dernier.

### ABSTRACT

---

#### **Relation, are you there ? LSTM-based relation detection to improve knowledge extraction**

Various methods for relation extraction and classification have been proposed and benchmarked on standard academic datasets. In real-life data however, the number of potential relations is enormous and the heuristics and count-based methods often used to separate actual relations from meaningless co-occurrences fail to detect weak signals of importance. In this paper, we investigate the use of a computationally-light binary detection model to identify whether a couple of entities in a sentence bears a relation, as a preliminary step prior to relation classification. Our model is based on the shortest dependency path between two entities analyzed with a LSTM recurrent network and combined with information on the entities types. On the binary relation detection task, we achieve results better than a state-of-the-art relation classification model adapted to detection with increased robustness to relations unseen in training. We finally show that binary detection as a pre-processing step to classification of relations is effective in significantly improving the latter.

**MOTS-CLÉS** : extraction d'informations, détection de relations, classification de relations, LSTM, plus court chemin de dépendances.

**KEYWORDS**: information extraction, relation detection, relation classification, LSTM, shortest dependency path.

---

# 1 Motivation

Les articles de journaux sont une source particulière de données, de par leur quantité – des milliers d’articles écrits chaque jour – et le grand nombre d’entités qui interagissent dans leurs textes au sein de multiples relations. Cette masse de textes, souvent confinée aux archives, regorge d’informations, de connaissances, de détails qui peuvent aider à la mise en contexte de nouveaux événements. Afin de pouvoir exploiter une telle richesse, l’utilisation de bases de connaissances est un atout majeur. Il s’agit de remplir une base contenant des entités et les relations qu’elles entretiennent pour mettre en lien des concepts à plus grande échelle que celle d’un seul texte. Pour ce faire, de nombreux aspects de l’extraction d’informations sont mis en œuvre, un aspect crucial étant l’extraction de relations entre entités.

Une entité est un objet, soit tangible soit un concept, auquel on peut référer par un nom propre, comme « Barack Obama » ou « Paris ». Une relation est une paire d’entités issues d’une même phrase et entre lesquelles la phrase exprime un lien. Elle est spécifiée avec un nom, comme « capitale de » ou « époux de ». Extraire cette relation permet de structurer l’information, en expliquant en quoi les entités d’une phrase sont liées, si elles le sont. Ceci est un défi dans le cas de données journalistiques, où une grande masse d’entités cohabite dans les articles et peu sont réellement en relation. L’extraction doit donc être opérée de façon sûre, rapide, sans omettre les signaux faibles qui sont potentiellement porteurs d’informations inédites pour les journalistes, tout en étant capable de s’adapter au flux constant d’articles aux thématiques variées.

De nombreux extracteurs et classifieurs de relations existent déjà, issus principalement de quatre types d’approches : les schémas universels (Riedel *et al.*, 2013), l’*open information extraction* (Banko *et al.*, 2007; Mesquita *et al.*, 2013; Del Corro & Gemulla, 2013), la classification non supervisée (Hasegawa *et al.*, 2004; Wang *et al.*, 2011; Takase *et al.*, 2015), et la classification supervisée (Kambhatla, 2004; Xu *et al.*, 2015; Wang *et al.*, 2016; Cai *et al.*, 2016; Zhang *et al.*, 2018). Certaines approches combinent d’ailleurs plusieurs de ces paradigmes (Banko & Etzioni, 2008). Cependant, ces diverses méthodes ne sont souvent testées et évaluées que sur des données de référence, en anglais, propres et peu bruitées. Quand elles sont testées sur des données réelles, le nombre de relations candidates est habituellement fortement réduit par l’utilisation d’heuristiques, pour ne garder que les relations les plus fréquentes. Dans le cas du journalisme, où les relations potentielles dans les textes sont extrêmement nombreuses tout en cachant de précieux signaux faibles, il devient d’une part coûteux d’appliquer des modèles de classification sur toutes les relations candidates ; d’autre part, les signaux faibles se retrouvent noyés dans de telles quantités de données que les systèmes de classification ne les distinguent pas : des relations potentiellement cruciales sont classifiées dans de très vastes classes « poubelle », ou tout simplement ignorées.

Les systèmes existants ne peuvent donc pas être appliqués directement à notre cas d’application journalistique. Nous proposons par conséquent de procéder en deux étapes, en ajoutant un modèle préliminaire pour détecter si une phrase contient ou non une relation. Les phrases pour lesquelles un couple d’entités a été identifié comme étant bien en relation peuvent ensuite être traitées par un modèle de classification de relations. Cette division permet tout d’abord d’augmenter la qualité des données fournies au modèle de classification, en réduisant le bruit apporté par des entités sans relation, et ensuite de diminuer le temps d’exécution d’une classification de relations, puisque seule une partie des données est traitée par un modèle de classification. Nous inspirant de l’état de l’art, nous proposons un modèle simple mais efficace de détection binaire de relations, fondé sur un réseau de neurones LSTM exploitant la syntaxe des phrases, que nous testons sur des données réelles que

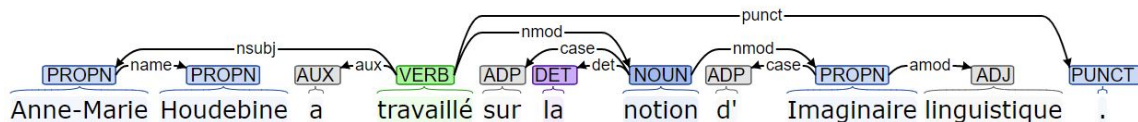


FIGURE 1 – Exemple de graphe de dépendances pour la phrase « Anne-Marie Houdebine a travaillé sur la notion d’Imaginaire linguistique. ».

nous avons collectées. Nous montrons la performance de notre modèle sur la détection de relations, ses capacités de généralisation et la possibilité qu’il offre d’améliorer les performances d’un modèle de classification de relations avec de la détection de relations en amont, démontrant ainsi la pertinence de notre approche en deux temps.

## 2 Méthodologie

Notre modèle doit être capable de détecter des relations au sein d’articles de style journalistique, c’est-à-dire identifier si deux entités sont ou non en relation, en analysant la phrase dans laquelle elles apparaissent. Dans chaque phrase, tous les couples d’entités potentiellement en relation (les relations candidates) sont donc extraits. Notre modèle doit apprendre si les mots se trouvant entre deux des entités signifient qu’une relation incluant ces deux entités existe ou non.

La caractéristique principale de notre modèle est le recours au plus court chemin de dépendances entre deux entités, c’est-à-dire le chemin syntaxique liant deux entités dans le graphe de dépendances qui traverse le moins de mots. Un graphe de dépendances consiste en un graphe où les nœuds correspondent aux mots, et les arêtes aux dépendances syntaxiques entre les mots (voir un exemple en figure 1). Suivant (Bunescu & Mooney, 2005), nous faisons l’hypothèse que le plus court chemin entre deux entités du graphe de dépendances contient l’essentiel de la relation, car il traverse les mots nécessaires à la compréhension de la phrase, omettant l’information auxiliaire et réduisant ainsi le bruit potentiel. Ce plus court chemin de dépendances est produit par l’algorithme de Dijkstra appliqué sur le graphe de dépendances obtenu avec le *Stanford dependency parser* (Manning *et al.*, 2014), dont nous avons modifié quelques aspects (regroupé les entités « multi-mots » (« Barack Obama » ou « université d’Harvard ») en un seul nœud, réparé les phrases auxquelles il manque un lien, et retiré les doublons créés à cause d’une erreur d’implémentation de l’analyseur). Sur ce plus court chemin de dépendances, nous collectons des variables sur chaque mot : les plongements du mot obtenu grâce à un modèle *skip-gram* pré-entraîné (Mikolov *et al.*, 2013), la catégorie morpho-syntaxique et l’étiquette de dépendance. La moyenne des plongements de mots est prise pour les entités formées de plusieurs mots. À ceci, nous ajoutons le type des entités donné par un système de détection d’entités, tel « parti politique », « ong », « personne » ou encore « autre ». Le plus court chemin exclut donc les entités elles-mêmes, car elles sont déjà prises en compte dans notre modèle *via* leur type.

Le modèle est constitué de deux branches, l’une modélisant les entités à travers leur type avec une couche cachée entièrement connectée, et l’autre prenant en compte l’information syntaxique sur le plus court chemin entre les deux entités avec une couche LSTM. Une variante du LSTM proposée dans (Graves *et al.*, 2013) est utilisée. Les deux branches sont fusionnées par un produit scalaire, puis une couche entièrement connectée dotée d’une activation *sigmoïde* prédit la probabilité qu’il y ait une relation entre les deux entités. Une telle couche de sortie a été préférée à une activation *softmax* avec deux neurones de sortie, car cette dernière approche nécessite de comparer les deux probabilités de sortie et garder la classe correspondant à la plus grande, tandis que, dans notre cas,

une unique probabilité de sortie permet d’appréhender le degré de certitude de la relation. La figure 2 présente l’architecture de notre modèle appliqué à une phrase exemple. Le choix d’intégrer les types des entités est motivé par l’observation que certaines relations n’existent pas entre certains types d’entités. Ainsi une relation potentielle « fils de » ne peut exister entre une personne et un lieu. Nous cherchons donc à confronter les mots du chemin et les types des entités pour s’assurer de leur compatibilité. Nous avons de plus séparé l’information sur les entités de l’information séquentielle par souci de simplicité et de rapidité. Nos résultats préliminaires ont en effet montré qu’entraîner une couche entièrement connectée uniquement pour les types des entités réduit le temps d’entraînement par rapport à un LSTM où les entités sont incorporées dans le plus court chemin de dépendances, pour des performances identiques.

## 3 Expériences

### 3.1 Création d’un jeu de données

Les données de référence standards pour l’extraction de relations ne sont pas compatibles avec notre volonté de détecter des relations pour des données réelles : premièrement, elles ne sont généralement pas en français ; ensuite, ces données sont déjà propres, annotées à la main, et les effectifs dans les classes de relations sont équilibrés, ce qui est éloigné des données réelles de notre problème initial.

Nous avons donc créé un jeu de données dérivé de Wikipédia, car le « style Wikipédia » est assez proche de celui des journaux d’information, les phrases sont généralement syntaxiquement correctes et de nombreux sujets sont abordés. De plus, ces données sont *open-source* et sont disponibles pour une future utilisation <sup>1</sup>. 200 000 articles ont ainsi été pris aléatoirement dans le *dump* français de Wikipédia en date d’avril 2019. Après nettoyage, découpage en phrases, liage des entités et appariement des paires d’entités comme relations candidates, un étiquetage distant des relations présentes dans les phrases est rendu possible grâce à la base de connaissances Wikidata, en partant du principe qu’une relation existe lorsqu’un triplet connectant les deux entités dans Wikidata existe. Le jeu de données est ensuite sous-échantillonné en termes des types d’entités, pour équilibrer les couples de types d’entités, ce qui réduit drastiquement le nombre total de relations candidates. Ce sous-échantillonnage assure qu’aucun couple de types d’entités (personne-personne, ong-lieu, etc.) ne soit majoritaire. Nous évitons ainsi que nos données soient dominées par des relations entre types « autres », ou entre lieux (communes, départements, lieux, pays), bien plus courantes dans Wikidata que les relations entre autres types (par ex. ong-parti politique). Nos modèles n’apprennent donc pas que les relations existent uniquement entre lieux et ne rejettent pas le reste des données sur ce seul critère. Nous n’équilibrons en revanche pas les types de relations, car certaines relations (par ex. « capitale de » ou « pays ») sont plus communes que d’autres (par ex. « employeur »), et certaines si peu fréquentes qu’elles sont considérées signaux faibles (par ex. « éditeur scientifique » ou « élève de »). Au final, nous obtenons 230 exemples positifs et 876 exemples négatifs, pour un total de 1106 paires. Ces données sont réparties en 55% de données d’entraînement, 20% de validation et 25% de test. Malgré cette réduction, notre jeu de données a été créé pour respecter les propriétés de signaux faibles et de déséquilibre des relations positives et négatives qui sont les défis réels que notre modèle doit relever.

---

1. <https://github.com/CMallart/RelationDetectionFrench>

## 3.2 Modèle de référence

Nous comparons tous nos résultats à ceux du modèle proposé par *Xu et al. (2015)*. Ce modèle a été choisi comme base pour une comparaison équitable car ce système est l'état de l'art pour la classification de relations parmi les modèles ayant recours à un réseau LSTM exploitant les informations sur le plus court chemin de dépendances. Notre modèle s'inscrit dans le même esprit syntaxique, ce qui ne le rend pas fondamentalement différent de ce modèle de référence, à l'exception toutefois notable des modifications que nous avons dû faire pour prendre en compte les types des entités. L'option que nous avons retenue d'effectuer une comparaison entre des modèles similaires a été guidée par la possibilité de juger non seulement la pertinence des choix de modélisation spécifiques – tels l'ajout des types d'entités – mais surtout celle de la tâche de détection de relations par rapport à la tâche de classification.

Le modèle de *Xu et al.* est fondé sur la séparation du plus court chemin de dépendances en deux, d'un côté et de l'autre de l'ancêtre commun des deux entités sur l'arbre de dépendances, afin de prendre en compte la direction des dépendances. Le long du chemin de dépendances sont collectées la catégorie morpho-syntaxique, l'étiquette de dépendance, les plongements et les hyperonymes WordNet de chaque mot. Chaque variable est traitée par un LSTM distinct. Pour chaque variable, deux LSTM collectent l'information sur chaque moitié de chemin, puis les couches cachées de ces LSTM sont fusionnées par *max-pooling*. Les quatre canaux, un pour chaque variable, sont ensuite aussi fusionnés par *max-pooling*. Une couche entièrement connectée avec une fonction d'activation *softmax* renvoie la probabilité d'appartenir à chaque catégorie de relation.

Bien que relevant donc de la même philosophie syntaxique, notre modèle diffère du modèle précédent selon cinq aspects. Premièrement, nous voyons les dépendances syntaxiques comme un graphe plutôt qu'un arbre, ce qui permet de mieux réparer les erreurs de l'analyseur en ajoutant des arêtes. De plus, nous ne nous soucions pas de la direction des dépendances, et ne séparons donc pas le plus court chemin en deux morceaux de part et d'autre d'un ancêtre commun, afin de garder entières les expressions figées et figures de style. En troisième lieu, le plongement du mot et les catégories morpho-syntaxiques et de dépendance sont corrélées et donnent ensemble son sens global au mot ; ces variables ne sont donc pas séparées par canaux dans le LSTM. Quatrièmement, les hyperonymes WordNet, qui auraient pu être obtenus en français grâce à une traduction de la base lexicale telle que WoNeF (*Pradet et al., 2013*), sont omis pour éviter le recours à une ressource externe qui prend du temps supplémentaire et ajoute du bruit, dû à la granularité très fine des sens de WordNet, à une incomplétude dans certains domaines et à la difficulté de choisir le sens approprié. Enfin, seul le type des entités est pris en compte, au lieu de traiter ces entités comme des mots du chemin de dépendances, ce qui permet de réduire le temps de traitement pour les entités et de généraliser.

## 3.3 Résultats

Nous rapportons nos résultats pour quatre expériences conduites sur les données que nous avons construites pour la tâche. Ces expériences ont pour but de prouver que notre modèle répond aux défis de l'extraction de relations dans des articles, à savoir qu'il sépare correctement les couples d'entités en relation de co-occurrences fallacieuses des autres, en étant capable de généraliser à des relations rares ou inédites, et qu'il apporte de la valeur ajoutée par rapport à une classification de relations.

Le but de notre première expérience est de vérifier qu'une architecture simple comme celle de notre modèle permet d'atteindre des performances au moins aussi élevées que le modèle plus complexe

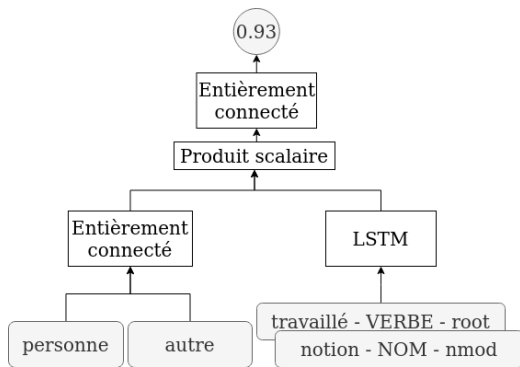


FIGURE 2 – Architecture générale du modèle appliqué à la phrase « Anne-Marie Houdebine a travaillé sur la notion d’Imaginaire linguistique. ».

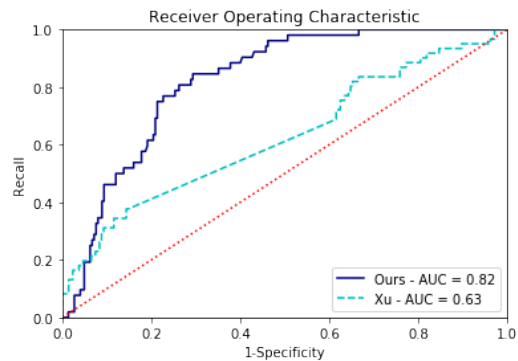


FIGURE 3 – Courbes ROC pour la détection binaire de relations (Xu-binaire et Nous).

de Xu *et al.* sur une tâche de détection. Pour ce faire, ce dernier est transformé d’un modèle de classification multi-classes à un modèle de détection binaire (Xu-binaire) en changeant l’activation de la dernière couche d’un *softmax* en un *sigmoïde*. Xu-binaire et notre modèle rendant tous deux des probabilités, le seuil pour décider qu’une relation existe est la valeur qui maximise le score *rappel+spécificité*. La courbe ROC est donnée en figure 3. Notre rappel de 0,73 est un peu plus bas que le rappel de Xu-binaire de 0,75, mais l’apport en précision est important, notre modèle atteignant 0,84 tandis que Xu-binaire obtient une précision de 0,67. En outre, l’aire sous la courbe ROC est plus grande pour notre modèle, indiquant qu’il a de meilleures performances que Xu-binaire indépendamment du seuil choisi. Un saut dans la courbe ROC de Xu-binaire montre que ce modèle est très sensible au seuil, à la différence du nôtre, plus robuste à une variation dans le choix de celui-ci. Notre modèle est donc particulièrement pertinent pour détecter l’existence ou non de relations. De plus, ce modèle spécialisé, créé pour la détection de relations, détecte mieux les cas fallacieux que Xu-binaire, qui classe de nombreuses relations négatives dans d’autres catégories que « none ».

Une deuxième expérience vérifie la capacité à détecter des relations inédites, qui n’ont pas été vues lors de l’entraînement. Notre modèle de détection binaire est entraîné avec des labels binaires. L’architecture originale de Xu *et al.* (2015) (Xu-multiclass) n’est pas modifiée, et le modèle est entraîné avec les noms des relations, les différentes étiquettes en sortie étant binarisées *a posteriori*. Nous avons conscience que ce protocole défavorise Xu-multiclass pour ce qui est des performances en classification binaire, car il s’agit d’un modèle entraîné à spécifier et non à détecter. Cependant, le but ici n’est pas de comparer les performances de deux modèles binaires – ce qui a été fait dans l’expérience précédente – mais de montrer qu’un modèle spécialisé dans la détection de relations permet de mieux découvrir des relations inédites qu’un modèle plus « généraliste » de classification, et donc de mieux les traiter ultérieurement. Le jeu d’entraînement consiste en notre jeu de données auquel on a ôté les relations dont le nom est « capitale » et « capitale de ». Notre modèle détecte ces relations inédites (« capitale » et « capitale de ») avec un rappel de 0,65 et une précision de 0,8, alors que le modèle de Xu *et al.* (2015) a un rappel de 0,57 et une précision de 0,71. Notre modèle a donc rempli sa promesse initiale de pouvoir mieux détecter les relations inédites, et donc de pouvoir mieux s’adapter à un flux de textes variés qu’un modèle de classification.

Notre troisième expérience teste la détection de signaux faibles. Nous cherchons à nouveau à prouver l’intérêt d’un modèle de détection binaire pour ce type de tâche par rapport à un modèle de classification dont le rôle principal n’est pas la reconnaissance des signaux faibles. Dans les mêmes conditions que l’expérience précédente, nous avons ôté les signaux faibles des données d’entraînement de l’expérience 1, et redistribué les exemples de signaux faibles dans les données de test. Notre modèle, pour

l'ensemble des données de test, a un rappel de 0,71, égal à celui de Xu-multiclass, mais une meilleure précision, à savoir 0,67 contre 0,72. Mais, lorsque nous considérons uniquement les exemples de signaux faibles, nous obtenons un rappel de 0,34, presque le triple du rappel de Xu-multiclass, 0,12. La précision pour ces résultats est égale à 1, ce qui est normal car elle mesure la proportion d'exemples classifiés comme relation qui sont réellement des relations ; or nous nous concentrons ici sur les relations rares qui existent toutes, donc toutes les relations candidates classifiées comme « en relation » sont bien pertinentes. La détection des relations rares par notre modèle est donc significativement plus élevée, avec un score F1 de 0,51 contre 0,21 pour Xu-multiclass ; ceci est certainement dû au fait qu'il repose sur la syntaxe en utilisant le plus court chemin de dépendances sans le scinder, permettant de reconnaître des constructions syntaxiques en passant outre le vocabulaire inédit.

Notre modèle étant pensé comme une étape préliminaire à la classification de relations, une dernière expérience teste le bénéfice d'ajouter ce modèle de détection comme étape de pré-traitement à un modèle de classification. Les deux systèmes testés et comparés sont un *pipeline* combinant notre modèle et le modèle de classification (Nous+Xu-multiclass) d'une part et, d'autre part, le modèle de classification de relations Xu-multiclass seul. Pour Nous+Xu-multiclass, lors de l'inférence, les relations prédites comme existantes par notre modèle sont passées en entrée de Xu-multiclass, tandis que les relations fallacieuses sont assignées à la classe « none » ; 39% des relations en entrée de notre modèle de détection sont considérées comme pertinentes pour être classifiées. Xu-multiclass prédit, lui, l'entièreté du jeu de test. Nous obtenons pour Nous+Xu-multiclass une précision de 0,74 et un rappel de 0,62, contre une précision de 0,68 et un rappel de 0,62 pour Xu-multiclass seul. Les relations étant en nombre très inégal, le modèle de classification de relations n'apprend à classer que dans les classes les plus peuplées, les exemples de test n'étant jamais classifiés dans les autres classes. Nous+Xu-multiclass reconnaît six classes, tandis que Xu-multiclass n'en reconnaît que 4. Les rappels présentés pour les deux modèles testés, bien qu'identiques, sont donc issus de phénomènes différents. Plus précisément, le rappel de Nous+Xu-multiclass sur la classe « none » est inférieur à celui de Xu-multiclass seul (0,66 contre 0,75), mais pour une précision accrue (0,93 contre 0,25). Nous+Xu-multiclass obtient également des valeurs de précision et rappel supérieures à Xu-multiclass pour les 4 classes reconnues par les deux systèmes, à l'exception de la classe « contient les subdivisions territoriales administratives » pour laquelle la précision est plus faible (0,18 contre 0,22) mais le rappel plus que doublé. Ces résultats montrent qu'ôter les relations qui ne sont pas pertinentes aide le classifieur, et apporte plus de subtilité dans la classification des relations, tout en limitant l'attribution erronée de la classe « none ».

## 4 Conclusion

Nous avons créé un système de détection de relations efficace dans le cas de textes journalistiques, et avons expérimentalement prouvé sa pertinence pour le problème de détection et de classification de relations sur des données réelles. Notre modèle peut être utilisé en amont d'un système de classification dont il améliore les résultats, tout en étant capable de repérer des signaux faibles et des relations inédites. Ce modèle seul permet, lorsqu'un journaliste recherche des informations sur une entité, de pouvoir éviter les contresens d'une recherche par co-occurrences : nous ne présentons pas les cas où les relations ne sont pas avérées, évitant d'associer des concepts qui ne sont liés que de façon lointaine. Notre modèle peut de plus être intégré dans la chaîne de traitement des articles d'un journal, juste avant le typage des relations, dans le but d'améliorer à la fois la qualité et la rapidité de l'alimentation d'une base de connaissances journalistique.

## Références

- BANKO M., CAFARELLA M. J., SODERLAND S., BROADHEAD M. & ETZIONI O. (2007). Open information extraction from the web. In *Proceedings of the 20th International Joint Conference on Artificial Intelligence*, p. 2670–2676 : Morgan Kaufmann Publishers Inc. DOI : [10.1145/1409360.1409378](https://doi.org/10.1145/1409360.1409378).
- BANKO M. & ETZIONI O. (2008). The tradeoffs between open and traditional relation extraction. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics : Human Language Technologies*, p. 28–36 : Association for Computational Linguistics. Anthologie ACL : [P08-1004](#).
- BUNESCU R. & MOONEY R. (2005). A shortest path dependency kernel for relation extraction. In *Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing*, p. 724–731 : Association for Computational Linguistics. DOI : [10.3115/1220575.1220666](https://doi.org/10.3115/1220575.1220666).
- CAI R., ZHANG X. & WANG H. (2016). Bidirectional recurrent convolutional neural network for relation classification. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1-Long Papers)*, p. 756–765 : Association for Computational Linguistics. DOI : [10.18653/v1/P16-1072](https://doi.org/10.18653/v1/P16-1072).
- DEL CORRO L. & GEMULLA R. (2013). Clausie : Clause-based open information extraction. In *Proceedings of the 22nd International Conference on World Wide Web*, p. 355–366 : Association for Computing Machinery. DOI : [10.1145/2488388.2488420](https://doi.org/10.1145/2488388.2488420).
- GRAVES A., MOHAMED A. & HINTON G. (2013). Speech recognition with deep recurrent neural networks. In *Proceedings of the 38th IEEE International Conference on Acoustics, Speech and Signal Processing*, p. 6645–6649 : IEEE Signal Processing Society. DOI : [10.1109/ICASSP.2013.6638947](https://doi.org/10.1109/ICASSP.2013.6638947).
- HASEGAWA T., SEKINE S. & GRISHMAN R. (2004). Discovering relations among named entities from large corpora. In *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics* : Association for Computational Linguistics. DOI : [10.3115/1218955.1219008](https://doi.org/10.3115/1218955.1219008).
- KAMBHATLA N. (2004). Combining lexical, syntactic, and semantic features with maximum entropy models for extracting relations. In *Proceedings of the ACL 2004 on Interactive Poster and Demonstration Sessions* : Association for Computational Linguistics. DOI : [10.3115/1219044.1219066](https://doi.org/10.3115/1219044.1219066).
- MANNING C. D., SURDEANU M., BAUER J., FINKEL J., BETHARD S. J. & MCCLOSKEY D. (2014). The Stanford CoreNLP natural language processing toolkit. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics : System Demonstrations*, p. 55–60 : Association for Computational Linguistics. DOI : [10.3115/v1/P14-5010](https://doi.org/10.3115/v1/P14-5010).
- MESQUITA F., SCHMIDEK J. & BARBOSA D. (2013). Effectiveness and efficiency of open relation extraction. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, p. 447–457 : Association for Computational Linguistics. Anthologie ACL : [D13-1043](#).
- MIKOLOV T., CHEN K., CORRADO G. & DEAN J. (2013). Efficient estimation of word representations in vector space. In *Proceedings of the 1st International Conference on Learning Representations, Workshop Track Proceedings*. arXiv : [1301.3781](https://arxiv.org/abs/1301.3781).
- PRADET Q., BAGUENIER-DESORMEAUX J., DE CHALENDAR G. & DANLOS L. (2013). WoNeF, an improved, extended and evaluated automatic French translation of WordNet (WoNeF : amélioration, extension et évaluation d’une traduction française automatique de WordNet) [in French]. In *Proceedings of TALN 2013 (Volume 1 : Long Papers)*, p. 76–89 : ATALA. HAL : [cea-00932340](https://hal.archives-ouvertes.fr/cea-00932340).



- RIEDEL S., YAO L., MCCALLUM A. & MARLIN B. M. (2013). Relation extraction with matrix factorization and universal schemas. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, p. 74–84 : Association for Computational Linguistics. Anthologie ACL : [N13-1008](#).
- TAKASE S., OKAZAKI N. & INUI K. (2015). Fast and large-scale unsupervised relation extraction. In *Proceedings of the 29th Pacific Asia Conference on Language, Information and Computation*, p. 96–105. Anthologie ACL : [Y15-1012](#).
- WANG L., CAO Z., DE MELO G. & LIU Z. (2016). Relation classification via multi-level attention CNNs. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1-Long Papers)*, p. 1298–1307 : Association for Computational Linguistics. DOI : [10.18653/v1/P16-1123](#).
- WANG W., BESANÇON R., FERRET O. & GRAU B. (2011). Filtering and clustering relations for unsupervised information extraction in open domain. In *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, p. 1405–1414, New York, NY, USA : Association for Computing Machinery. DOI : [10.1145/2063576.2063780](#).
- XU Y., MOU L., LI G., CHEN Y., PENG H. & JIN Z. (2015). Classifying relations via long short term memory networks along shortest dependency paths. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, p. 1785–1794 : Association for Computational Linguistics. DOI : [10.18653/v1/D15-1206](#).
- ZHANG Y., QI P. & MANNING C. D. (2018). Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, p. 2205–2215 : Association for Computational Linguistics. DOI : [10.18653/v1/D18-1244](#).