Architecture siamoise et embeddings de triplet pour la validation de relations

Jose G. Moreno, Rashedur Rahman, Charlotte Rudnik, Cong Wang, Brigitte Grau

Architecture siamoise et embeddings de triplet pour la validation de relation

Jose G. Moreno* — **Rashedur Rahman**** — **Charlotte Rudnik**** — **Cong Wang**** — **Brigitte Grau *****

* *IRIT UMR 5505 CNRS, Université de Toulouse*
** *LIMSI, CNRS, Université Paris-Saclay*
** *LIMSI, CNRS, ENSIIE, Université Paris-Saclay*

RÉSUMÉ. *La reconnaissance qu'une relation existe entre deux entités mentionnées dans un texte joue un rôle vital en extraction d'information (EI). Pour répondre à la nécessité d'annoter manuellement de nombreux exemples, des paradigmes de supervision distante et d'EI non supervisée ont été proposés. Le point crucial dans ces approches est de pouvoir évaluer la validité des relations extraites. Dans cet article, nous proposons une nouvelle architecture neuronale pour modéliser la validation de relations, inspirée des modèles neuronaux pour l'implication textuelle. Nous encodons le texte et le triplet correspondant à la relation dans une architecture siamoise afin de décider si le texte supporte ou non la relation. Nous proposons différentes représentations d'une relation qui tirent profit de l'apprentissage joint de mots et d'entités dans un espace commun.*

ABSTRACT. *Recognizing if a relation holds between two entities in a text plays a vital role in information extraction (IE). To overcome the need to annotate many examples manually, open IE and distant supervision paradigms were proposed. In these two last settings, the crucial point is to be able to assess the validity of the extracted relations. In this paper, we propose a new NN architecture for modelling relation validation, inspired by NN entailment models, that encodes the text and the relation triplet in a siamese architecture in order to decide whether or not the text supports the relation. We propose different representations of a relation that take advantage of learning word and entity embeddings in a common space.*

MOTS-CLÉS : *validation de relation, réseau de neurone siamois, embedding d'entités*

KEYWORDS: *Relation validation, Siamese Neural Network, entity embedding*

1. Introduction

Reconnaître si une relation existe entre deux entités dans un texte joue un rôle vital en extraction d'information, le peuplement de bases de connaissances ou la réponse automatique à des questions. Quelques exemples de relations typiques que l'on trouve dans les bases de connaissances (BC) sont *conjoint, PDG, lieu de naissance, profession*, etc. La plupart des approches modélisent la tâche d'extraction de relations (dos Santos *et al.*, 2015 ; Nguyen et Grishman, 2015) comme un problème de classification multi-classes pour prédire si un passage contient un type de relation entre deux entités données. Cette approche nécessite des exemples annotés pour chaque classe, c'est-à-dire pour chaque type de relation, annotations qui peuvent être difficiles à obtenir. Pour résoudre ce problème, le principe de supervision distante a été proposé (Mintz *et al.*, 2009) pour l'annotation automatique des textes étant donné les triplets de relation existant dans une BC, avec la contrepartie de devoir traiter des exemples faussement annotés. La difficulté de la tâche est illustrée par les résultats à l'évaluation TAC KBP (Knowledge Base Population). Par exemple, en 2014, le meilleur score était une F1-mesure de $0,3672$ (Surdeanu et Ji, 2014). Une autre possibilité consiste à collecter des informations directement sur le Web dans une approche non supervisée, c'est le paradigme de l'extraction d'information ouverte (Banko *et al.*, 2007). Dans ces deux derniers contextes, un point crucial est de pouvoir évaluer la validité des relations extraites. Cette problématique a motivé la création d'une tâche à TAC KBP en 2015 qui consiste à valider les relations extraites[1] par les systèmes d'extraction de relations afin d'améliorer leurs scores finaux. En question-réponse (QR), l'évaluation AVE (Answer Validation Exercise) (Giampiccolo *et al.*, 2007) est une tâche similaire. (Rodrigo *et al.*, 2019) montre l'impact de l'étape de validation sur l'amélioration des systèmes de QR et surtout sur la réduction du nombre de réponses fausses qui permettront aux utilisateurs de plus faire confiance à un système de QR.

Le but de la validation de relations est de tirer parti de plusieurs hypothèses, fournies par un ou plusieurs systèmes, pour améliorer la reconnaissance des relations dans les textes et éliminer les propositions fausses. Cette tâche peut être définie comme un problème de classification binaire qui, étant donnée une relation candidate sous forme de triplet $(e1, R, e2)$ et un passage, vise à décider si le passage supporte la relation ou non. Dans (Yu *et al.*, 2014), les auteurs montrent qu'il est nécessaire de modéliser la relation à valider par des caractéristiques linguistiques profondes en plus de scores de crédibilité du système et du document. Des mots déclencheurs et des patrons de relations constituent généralement les caractéristiques choisies pour représenter le type de la relation. Dans (Wang et Neumann, 2008), la validation de relation est modélisée comme un problème d'implication textuelle, où le système apprend si le texte implique la relation, en se basant sur des caractéristiques linguistiques.

Les travaux récents sur les réseaux de neurones ont été appliqués pour reconnaître la proximité sémantique de deux textes courts, avec la proposition d'architectures sia-

1. `https://tac.nist.gov/2015/KBP/SFValidation/index.html`

moises fondés sur des réseaux convolutifs (CNN) ou récurrents (RNN)[2]. Ces systèmes apprennent une représentation de chaque entrée qui sont ensuite jointes dans une couche donnée en entrée d'une classification binaire.

Dans cet article, pour modéliser la validation de relation selon un texte, nous proposons non seulement d'apprendre la représentation du type de relation, mais aussi d'apprendre la représentation du triplet au travers d'une architecture siamoise, inspirée des modèles neuronaux d'implication textuelle. Notre but est de décider si le texte supporte ou non la relation en encodant le texte et le triplet comme dans (Severyn et Moschitti, 2015). Nous avons en outre augmenté le modèle d'un mécanisme d'attention pour mieux aligner les deux entrées. Nous encodons également la position des mots par rapport aux entités dans le texte pour capturer les mots qui expriment la relation par leur proximité aux entités comme dans (dos Santos *et al.*, 2015 ; Nguyen et Grishman, 2015).

Nous proposons également différentes représentations d'un triplet pour tester différents modèles d'embeddings, qu'il s'agisse d'embeddings de mot et entité ou d'embeddings de mot seulement. Selon le modèle utilisé, les entités d'un triplet sont encodées par un embedding d'entité ou de mot, tandis que seuls des embeddings de mots sont utilisés pour encoder le type de la relation.

Nous évaluons notre approche sur un corpus construit par supervision distante qui contient 25-40 fois plus de types de relations différents que les jeux de données existants. Nos modèles neuronaux surpassent les baselines, sans nécessiter de prétraitement par des outils de reconnaissance d'entités nommées ou d'analyse de phrase, ce qui montre l'intérêt de l'apprentissage explicite de la représentation du triplet de la relation à valider. Nous montrons également que l'utilisation d'embeddings d'entités au lieu d'embeddings de mots lors de la représentation des triplets donne de meilleurs résultats dans tous les modèles étudiés.

2. État de l'art

Différents systèmes de vote (Viswanathan *et al.*, 2015 ; Wang *et al.*, 2013 ; Sammons *et al.*, 2014 ; Rodriguez *et al.*, 2015) ont été appliqués pour la tâche de validation de relation de KBP sur la base des prédictions faites par les systèmes d'extraction de relation. Cependant, (Yu *et al.*, 2014) montre que la validation des relations nécessite de tenir compte de caractéristiques linguistiques avancées pour reconnaître qu'une relation est exprimée dans un texte. Les auteurs ont utilisé un modèle multidimensionnel de validation qui représente la véracité de la proposition à partir de la crédibilité d'un système ou d'un document, et de l'évaluation du fait que le texte justifie la relation proposée par l'exploitation de connaissances linguistiques des niveaux lexical, syntaxique et sémantique. Dans (Wang et Neumann, 2008), la relation à valider est transformée par de simples patrons en une phrase et l'alignement des deux textes est

2. Pour une évaluation comparative de ces modèles sur un jeu de données de QR voir `https://aclweb.org/aclwiki/Question_Answering_(State_of_the_art)`

appris par une méthode à noyau. Ce travail a été évalué sur trois types de relations uniquement.

En fait, les caractéristiques utilisées pour la validation de relation sont les mêmes qu'en extraction de relations. Les méthodes traditionnelles d'extraction des relations sont fondées sur des approches à base de traits qui s'appuient sur des informations lexicales et syntaxiques. Les arbres de dépendance sont souvent exploités pour fournir des indices permettant de décider de la présence d'une relation, par exemple en extraction de relation non supervisée (Culotta et Sorensen, 2004 ; Bunescu et Mooney, 2005 ; Fundel *et al.*, 2007). (Gamallo *et al.*, 2012) définit des patrons de relations en analysant les relations de dépendances. Les mots autour des entités mentionnées dans le texte donnent quant à eux des indices pour caractériser la sémantique d'une relation, i.e. identifier les termes déclencheurs (Niu *et al.*, 2012 ; Hoffmann *et al.*, 2011 ; Yao *et al.*, 2011 ; Riedel *et al.*, 2010 ; Mintz *et al.*, 2009).

En plus des informations linguistiques, des informations globales sur les entités et leurs relations ont été exploitées dans (Rahman *et al.*, 2018), en ajoutant des traits calculés sur un graphe d'entités, et (Augenstein, 2016) qui intègre des informations statistiques liées à l'objet de la relation. Ces derniers travaux montrent l'importance d'ajouter des informations sur les entités du triplet.

Les approches ci-dessus s'appuient sur des outils de TAL pour l'analyse syntaxique et sur des connaissances lexicales pour identifier les déclencheurs de relation. Il reste donc difficile de combler le fossé sémantique entre les textes et le nom de la relation lors de l'apprentissage de modèles destinés à reconnaître de nombreux types de relations en domaine ouvert.

Récemment, les systèmes neuronaux ont obtenu de bons résultats pour la tâche de classification de relations (dos Santos *et al.*, 2015 ; Nguyen et Grishman, 2015 ; Vu *et al.*, 2016 ; Dligach *et al.*, 2017 ; Zheng *et al.*, 2016). Ces méthodes utilisent des CNN et/ou des LSTM et apprennent les patrons et la sémantique des relations à partir de données annotées, par exemple SemEval, ACE etc. Cependant, ils ne tirent pas parti de la représentation des triplets pour apprendre la relation entre le texte et le triplet.

De nombreux modèles neuronaux ont été proposés pour évaluer la similarité de deux phrases[3]. Ils apprennent une représentation de chacune des phrases données en entrée à un CNN ou un RNN (LSTM ou BiLSTM), et calculent une similarité entre ces représentations comme dans (Severyn et Moschitti, 2015) ou les interactions entre les mots des deux phrases par un mécanisme d'attention comme dans (Yin *et al.*, 2016). Dans le domaine des questions-réponses, avec une modélisation proche de notre tâche, la représentation d'une réponse candidate extraite de la base de connaissance (BC) est comparée à la représentation de la question en langage naturel afin d'être validée (Bordes *et al.*, 2014). Les réponses candidates sont des sous-graphes de la BC dont les embeddings sont apprises à partir du graphe de la base.

3. Voir (Lan et Xu, 2018) pour une étude comparative de certains d'entre eux

3. Modèle de validation de relation

Dans cette section, nous présentons notre modèle, qui est basé sur une architecture siamoise, avec en entrée, d'un côté le triplet de la relation et de l'autre le texte, qui justifie ou non la véracité du triplet. L'architecture globale de notre modèle est illustrée dans la figure 1.

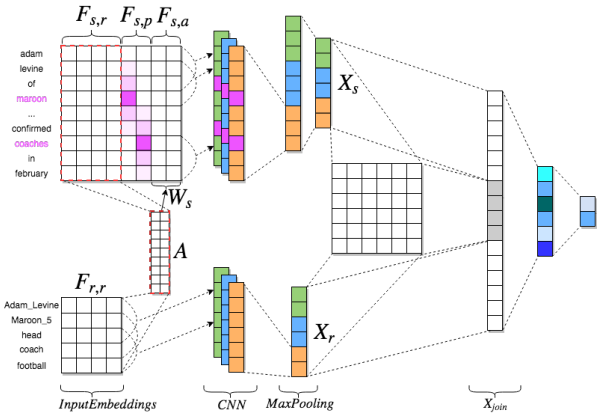

Figure 1 – Architecture siamoise pour la validation de relation

Afin de modéliser les interactions entre le texte et le triplet, nous calculons une matrice d'attention A comme dans (Yin *et al.*, 2016). A est calculé en faisant correspondre les unités de la représentation de phrase $(F_{s,r})$ et de la représentation de la relation $(F_{r,r})$. A_{ij} dans la matrice est le score correspondant au i-$ème$ élément dans $F_{s,r}$ et au j-$ème$ dans $F_{r,r}$.

$$A_{i,j} = match\_score(F_{s,r}[:, i], F_{r,r}[:, j])$$

Nous avons choisi la fonction cosinus comme fonction $match\_score$ pour mesurer la similarité de deux éléments. Ensuite, nous utilisons une couche linéaire pour générer une carte d'attention sur les éléments $F_{s,a}$ comme information supplémentaire pour la phrase.

$$F_{s,a} = W_s \cdot A^T$$

Comme les mots proches des deux entités dans le texte sont généralement importants, nous utilisons des embeddings de position comme dans (Nguyen et Grishman, 2015) pour représenter la distance entre chaque mot et l'entité tête, respectivement l'entité queue, des entités quand le mot se trouve sur le côté droit, respectivement le côté gauche. La distance est définie comme la valeur absolue de la distance relative.

Nous concaténons les vecteurs de représentation ($F_{s,r}$), les vecteurs d'attention ($F_{s,a}$) et les embeddings de position ($F_{s,p}$) en entrée du CNN du côté texte. Comme le triplet n'a pas besoin d'information de position et non plus du mécanisme d'attention, seule la représentation ($F_{r,r}$) est utilisée en entrée du CNN du côté relation. Les représentations vectorielles X_s et X_r sont générées par les couches de convolution suivies par la fonction d'activation ReLU et max-pooling.

Pour calculer la similarité des deux entrées, X_{sim} est la similarité entre X_s et X_r comme dans (Severyn et Moschitti, 2015) et est définie comme suit :

$$sim(X_s, X_r) = X_s^T M X_r$$

où M est une matrice de similarité qui est un paramètre du réseau, c'est-à-dire qu'elle est optimisée pendant l'apprentissage.

La couche de jointure consiste à concaténer tous les vecteurs intermédiaires en un seul vecteur comme suit :

$$X_{join} = [X_s^T; X_{sim}; X_r^T]$$

Enfin, nous utilisons une couche standard entièrement connectée avec $softmax$ comme fonction d'activation pour engendrer la probabilité de validation finale.

Sur la base de la description précédente, nous définissons nos quatre modèles avec différentes interactions entre le triplet et le texte d'entrée :

– ABSMCNN-POS et ABSMCNN : correspondent au modèle décrit précédemment, mais le dernier n'inclut pas $F_{s,p}$.

– SMCNN-POS et SMCNN : sont analogues aux systèmes précédents, mais sans la matrice d'attention, donc ils n'incluent pas A et par conséquent $F_{s,a}$.

4. Représentation des relations

Pour représenter un triplet $t = (e_1, r, e_2)$, deux types d'éléments doivent être représentés : entités, e, et type de relation, r. Chacun d'eux est associé à un label, l, composée d'une séquence de mots où $l = [w_1..w_n]$. Ainsi, une entité ou un type de relation peut être transformé en la séquence des mots de son label. Nous proposons de représenter un triplet de la façon suivante :

– MOTS : Une séquence de mots, $t = [we_{1,1}...we_{1,n}, we_{2,1}...we_{2,m}, wr_1...wr_r]$. Les mots sont ceux des labels des entités, e_1 puis e_2, suivis des mots du label de relation ;

– ENTITÉS+MOTS : Une séquence composée d'entités, suivis des mots du label de la relation, $t = [e_1, e_2, wr_1..wr_r]$;

Nous transformons ensuite t en vecteurs, où chaque unité est représentée par un vecteur. Afin d'obtenir une représentation fiable des entités et de pouvoir les combiner

avec des représentations de mots, nous avons utilisé un modèle qui apprend conjointement leurs embeddings (voir 4.1). Notre but est de tester ces différents schémas de représentation du triplet et d'étudier si l'utilisation d'embeddings d'entités permet de calculer une représentation plus informée de la relation.

4.1. *Embeddings de mots et d'entités*

Représenter conjointement des connaissances apprises sur des graphes et des mots appris à partir de textes à l'aide de techniques d'embeddings s'est récemment révélée utile dans plusieurs tâches, comme la liaison d'entités, la population de base de connaissances, etc. Malgré l'existence de travaux proposés pour combiner différentes représentations (Fang *et al.*, 2016 ; Yamada *et al.*, 2016), nous optons pour l'algorithme EAT récemment proposé par (Moreno *et al.*, 2017) pour sa simplicité et sa disponibilité. Ce modèle apprend conjointement les mots et les entités dans un espace vectoriel unique et se limite aux entités des pages Wikipédia. L'idée sous-jacente est d'exploiter les mentions d'entités, i.e. les textes d'ancrage, dans les pages Wikipédia pour calculer des embeddings d'entités selon (Mikolov *et al.*, 2013). Lorsqu'un texte d'ancrage est trouvé dans une fenêtre, EAT traite deux fois celle-ci, une fois pour le mot et une autre pour l'entité. De cette façon, les embeddings de mots ne sont pas dégradés et des embeddings d'entités non ambigus sont appris. Cependant, des étapes de prétraitement supplémentaires sont nécessaires pour s'assurer que les textes d'ancrage ne sont pas supprimés, que les entrées des entités dans le vocabulaire sont identifiées et normalisées pour éviter des redondances (par exemple, en appliquant toutes les redirections possibles dans Wikipédia). Nous avons utilisé une implémentation accessible au public basée sur TensorFlow[4] avec les paramètres suggérés par (Moreno *et al.*, 2017) (taille des vecteurs = 200, taux d'apprentissage = 0.025, 5 époques, configuration Skipgram et taille de la fenêtre = 5). Le vocabulaire obtenu est composé de plus de 5,2 millions d'entrées dont 1,8 million d'entités.

5. Expériences et résultats

5.1. *Les données*

Dans cette étude, nous faisons les expériences sur deux jeux de données différents provenant de deux tâches : *question-réponse* (dataQA) et *population de base de connaissance* (dataKBP).

5.1.1. *Jeu de données construit à partir de Webquestion*

Les embeddings d'entités sont calculés à partir des pages Wikipédia. Nous avions donc besoin d'un corpus pour valider des relations entre des entités Wikipédia. Nous

4. `https://github.com/jgmorenof/EAT-tensorflow`

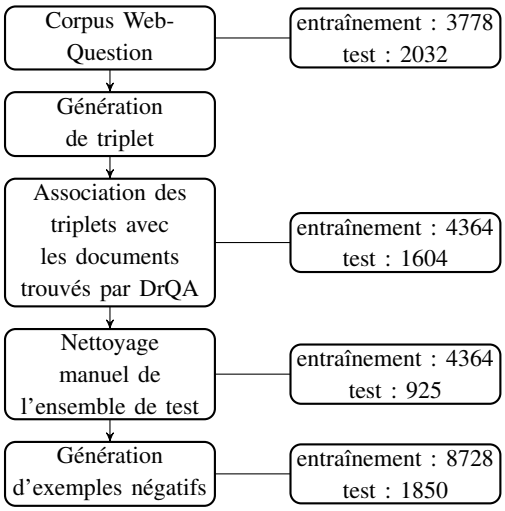

Figure 2 – Construction du corpus dataQA par supervision distante

avons d'abord étudié le corpus fourni par (Culotta et Sorensen, 2004). Cependant, la plupart des relations sont des relations familiales, qui correspondent à un sous-ensemble limité des relations existantes. De ce fait, nous avons décidé de construire un corpus, nommé dataQA, qui comporte de multiples types de relation, en exploitant WebQuestion (Berant *et al.*, 2013) dans un processus de supervision distante décrit dans la figure 2. WebQuestion est un jeu de données bien connu sur la base Freebase contenant des paires de questions et réponses (QR). Ce corpus est bien adapté à notre problème car les questions ont été posées sans connaître le schéma de la base de connaissance et les réponses sont des entités de Freebase, ce qui nous permet de faire la correspondance avec les entités Wikipédia via DBpedia.

Afin d'extraire les chemins de relation de Freebase, nous calculons les sous-graphes de longueur un ou deux qui existent entre l'entité réponse et les entités de la question, qui sont fournies avec le corpus[5]. Nous sélectionnons les graphes qui sont valides pour toutes les réponses à la question. Par exemple, à la question,

"What countries are part of the UK ?", les chemins récupérés sont
$United\_Kingdom$,
$/location/location/contains, Northern\_Ireland$,
$/location/location/contains, England$,
$/location/location/contains, Scotland$,
$/location/location/contains, Wales$
communs aux quatre réponses. Nous avons converti les entités Freebase en entités

5. https://github.com/brmson/dataset-factoid-webquestions

Wikipédia. Les labels des entités sont donc ceux de Wikipédia. Par exemple, l'entité Freebase $Waterloo$ devient $Waterloo, \_Belgium$

En utilisant le module[6] fourni par (Chen *et al.*, 2017), nous associons des passages à chaque paire de QR, en interrogeant Wikipédia. Les passages sélectionnés sont ceux qui sont les plus proches de la question et qui contiennent en plus la réponse. Ce processus peut engendrer de faux exemples positifs, car le texte sélectionné peut ne pas justifier la relation. Cependant, nous avons conservé tous les exemples dans le jeu d'apprentissage. Le jeu de test, lui, a été validé manuellement.

La dernière étape consiste à engendrer des exemples négatifs. Nous avons remplacé les relations dans les triplets par des relations choisies au hasard. Comme il existe de nombreux types de relations, il y a peu de chances d'obtenir une relation similaire. Cependant, nous avons vérifié que les nouvelles relations ne partagent pas de mot avec la relation correcte. Des exemples ont été supprimés pour assurer un équilibre entre l'ensemble d'entraînement et l'ensemble de test. Les mots utilisés comme étiquettes de relation sont ceux que l'on trouve dans le chemin Freebase.

5.1.2. *Jeu de données KBP*

Dans nos expérimentations, nous avons utilisé le jeu de données de (Rahman *et al.*, 2018), que nous nommons dataKBP. Il consiste en un sous-ensemble du corpus proposé pour la tâche *Slot Filler Validation* à TAC KBP 2016. Le jeu de données initial contient des relations dont l'objet peut être une valeur (âge, numéro, date, titre, etc.) qui ne fait pas référence à une entité alors que le corpus dataKBP ne contient que des relations où les sujets et les objets sont des entités nommées. Il comporte les

	Ensemble d'entraînement		**Ensemble de test**	
Nom de la relation	**# Positif**	**# Total**	**# Positif**	**# Total**
person :parents	148	377	94	480
person :spouse	112	330	25	131
person :children	67	160	37	667
person :country_of_birth	108	248	5	265
person :country_of_death	77	225	72	261
person :city_of_birth	487	1,303	139	229
person :city_of_death	237	635	30	257
person :employee_or_member_of	2,501	5,770	287	1,825
org :country_of_headquarters	473	1,295	140	502
org :city_of_headquarters	638	1,640	192	565
org :member_of	575	1,492	27	416
org :top_members_employees	461	1204	61	338
Total	**5,884**	**14,679**	**1,109**	**5,936**

Tableau 1 – Types de relation et leur distribution dans le corpus dataKBP. Le préfixe indique sur quel type d'entité source porte la relation

6. https://github.com/facebookresearch/DrQA

triplets de relation et les phrases associées à ces relations avec les étiquettes correctes ou fausses. Un exemple est correct si la phrase exprime la relation, et incorrect sinon. Une entité nommée a été associée à une entité Wikipédia selon sa proximité avec le label de l'entité. Certains exemples ne font donc pas référence à des entités connues dans Wikipédia. Des informations détaillées de la composition du corpus dataKBP sont données dans le tableau 1. Il comporte 12 types de relation.

Jeu de données		**# Positif**	**# Négatif**	**# Total**	**# Relation**
dataQA	Entraînement	4364	4364	8,728	501
	Test	925	925	1,850	311
dataKBP	Entraînement	5,884	8,795	14,679	12
	Test	1,109	4,827	5,936	12

Tableau 2 – Statistiques des corpus dataQA et dataKBP

Contrairement à dataQA, le corpus dataKBP est déséquilibré comme indiqué dans le tableau 2. Le ratio d'instances positives et négatives est de $1:1.5$. En revanche, leur ratio dans les données de test est de $1:4.4$. Pour toutes les relations sauf *city_of_birth*, cf. tableau 1, le nombre d'exemples négatifs est plus élevé que le nombre d'exemples positifs dans les données d'entraînement et de test. Nous pouvons également noter que dataQA contient un nombre de types de relations différents nettement plus important que dataKBP.

5.2. *Protocole expérimental*

5.2.1. *Métriques d'évaluation*

Nous exécutons chaque modèle de validation de relation N=5 fois et évaluons la performance de classification par le score F1 moyen et l'exactitude moyenne. Le score F1 indique la proportion de relations correctement reconnues sur la base de la précision et du rappel, tandis que l'exactitude mesure l'étiquetage correct. Nous mesurons le score F1 moyen et l'exactitude moyenne en utilisant Eq. 1 où F_{1i} et A_i font référence au score F1 et à l'exactitude standard.

$$F_{1avg} = \frac{\sum_{i=1}^{N} F_{1i}}{N}; \quad A_{avg} = \frac{\sum_{i=1}^{N} A_i}{N} \qquad [1]$$

5.2.2. *Baselines*

Afin de montrer la contribution d'une architecture siamoise, notre première baseline est fondée sur une architecture non-siamoise proposée d'abord pour l'extraction des relations et utilisée ici dans une configuration de validation de relation. Le modèle CR-CNNN-POS (dos Santos *et al.*, 2015) a été adapté pour obtenir en sortie une prédiction binaire de l'existence ou non de la relation entre les deux entités dans le texte. Les caractéristiques de position sont utilisées car elles se sont révélées utiles

pour améliorer les prédictions. Pour les expériences sur dataKBP, comme le corpus est déséquilibré, nous ajoutons en plus un classificateur aléatoire comme référence.

5.2.3. *Nos modèles*

Nous avons testé les quatre architectures NN décrites dans la section 3 : SMCNNN, ABSMCNNN, SMCNN-POS et ABSMCNNN-POS. Les deux premiers modèles n'ont pas besoin d'informations sur les entités du texte et ne font donc pas appel à un prétraitement. Les deux derniers modèles nécessitent l'annotation dans le texte des deux entités à l'étude. Tous nos modèles ainsi que la baseline ont été développés sur la base d'une mise en œuvre publique de modèles neuronaux proches[7].

5.2.4. *Hyper-paramètres*

Tous les modèles et la baseline ont été configurés ainsi : $batch\_size = 64$, $nb\_filter = 100$, $filter\_length = 3$, $hidden\_dims = 100$, et $dropout = 0.50$. Aucune optimisation individuelle des paramètres n'a été effectuée. Cependant, le nombre d'époques est différent pour les deux jeux de données. Nous obtenons de meilleurs résultats moyens de classification aux 20ème et 10ème époques pour dataQA et dataKBP, respectivement.

5.2.5. *Modèles d'embeddings*

Comme indiqué plus haut, le triplet correspondant à la relation peut être représenté par deux modèles. Afin de comparer la qualité des différents modèles d'embeddings et l'utilisation d'embeddings d'entités au lieu d'embeddings de mots, nous avons appris nos modèles avec quatre modèles de relation :

– EAT : correspond à la représentation du triplet ENTITY+WORDS avec le modèle EAT pour les embeddings de mots et d'entités ;

– EAT.NOE : la représentation du triplet WORDS avec le modèle EAT pour les embeddings de mots, donc sans utiliser les embeddings d'entités ;

– glove : la représentation du triplet WORDS avec les embeddings glove ;

– word2vec : la représentation triplet WORDS avec les embeddings word2vec ;

Le texte est représenté par une séquence de mots qui sont transformés en vecteurs selon le modèle d'embeddings utilisé pour représenter le triplet. Si un token (un mot dans le triplet ou le texte) ou une entité (le sujet ou l'objet dans un triplet) manque dans le vocabulaire des embeddings utilisé, nous assignons un vecteur aléatoire à ce token ou cette entité manquant. Les statistiques des tokens et entités inconnus dans les différents modèles d'embeddings et jeux de données sont indiqués dans le tableau 3[8].

7. `https://github.com/UKPLab/deeplearning4nlp-tutorial`

8. Chacun des jeux de données comporte de l'ordre de 2000 (dataQA) à 2300 (dataKBP) entités différentes.

	dataQA		dataKBP	
Embeddings	**Token**	**Entité**	**Token**	**Entité**
EAT	1.30%	4.88%	0.88%	39.14%
EAT.NOE	1.30%	0.37%	0.88%	1.40%
glove	1.38%	0.07%	1.98%	1.55%
word2vec	31.17%	5.67%	28.38%	6.42%

Tableau 3 – Statistiques des tokens inconnus (mot et entités) dans les différents modèles d'embeddings et jeux de données

5.3. *Résultats*

Les résultats de nos systèmes sur dataQA sont présentés Tableau 4. Les perfor-

Modèle	**Représentation de la relation**	**F1-score**	**Exactitude**
Baselines			
CR-CNN-POS	EAT	**0.6025**	0.4991
	EAT.NOE	0.5820	0.4962
	glove	0.5859	0.4979
	word2vec	0.5941	**0.5067**
Nos modèles			
SMCNN	EAT	0.9157	0.9142
	EAT.NOE	0.9043	0.9011
	glove	**0.9177**	**0.9150**
	word2vec	0.9135	0.9113
ABSMCNN	EAT	0.9150	0.9120
	EAT.NOE	0.9080	0.9061
	glove	**0.9195***	**0.9189***
	word2vec	0.9076	0.9047
SMCNN-POS	EAT	0.9079	0.9037
	EAT.NOE	0.9071	0.9039
	glove	0.9090	0.9055
	word2vec	**0.9107**	**0.9092**
ABSMCNN-POS	EAT	**0.9155**	**0.9128**
	EAT.NOE	0.9040	0.9003
	glove	0.9115	0.9097
	word2vec	0.9078	0.9049

Tableau 4 – Performances de validation de la relation (score F1 moyen et exactitude moyenne sur 5 runs à l'époque 20) par différents modèles NN et modèles d'embeddings sur dataQA

mances de tous nos modèles sur le corpus dataQA sont assez élevées. Ils surpassent la baseline de plus de 0.30 en termes de score F1 et d'exactitude. Les meilleurs résultats sont obtenus avec les trois modèles d'embeddings, EAT, glove et word2vec. Ces résultats confirment que l'utilisation d'une représentation explicite du triplet en entrée est

beaucoup plus efficace que de laisser le modèle l'apprendre. Dans nos architectures, le modèle ne doit apprendre que ce que sont des représentations similaires pour valider la relation.

Si on compare les modèles d'embeddings, le modèle EAT.NOE donne des résultats légèrement inférieurs à ceux de glove et word2vec. Il peut y avoir un effet dû aux différentes dimensions des vecteurs, 300 pour glove et word2vec, et 200 pour EAT. Cependant, l'utilisation de représentations d'entité dans EAT permet d'augmenter les résultats de EAT.NOE dans tous les modèles, ce qui montre que les embeddings d'entités apportent des informations supplémentaires pour représenter la relation.

L'ajout d'un mécanisme d'attention pour aligner le texte avec la relation améliore légèrement les résultats avec certains modèles (glove et EAT.NOE), mais pas avec EAT, comme nous aurions pu nous y attendre. C'est peut-être parce que l'interaction est mieux reconnue avec les mêmes mots pour les entités dans le texte et le triplet.

Les deux modèles qui utilisent la position des entités dans les textes n'obtiennent pas de meilleurs résultats, contrairement à nos attentes. Cela signifie que, comme les entités sont représentées avec le type de la relation d'un côté, sans information supplémentaire, les convolutions conduisent à une représentation assez bonne du texte pour permettre de décider de leur similarité.

Pour savoir dans quelle mesure la matrice de similarité M apporte aux performances de nos modèles, nous avons testé le réseau SMCNN sans elle, sur les modèles EAT et glove. Les deux modèles ont obtenu des résultats un peu plus faibles : EAT a obtenu un score $F1 = 0.9115$ et une $exactitude = 0.9090$, et glove un score $F1 = 0.9097$ et une $exactitude = 0.9066$.

Nous avons tracé la courbe de précision sur 20 époques dans la figure 3. Elle montre que notre modèle atteint rapidement de bonnes performances.

Le tableau 5, quant à lui, montre les résultats de nos différentes architectures neuronales sur dataKBP. La plupart de nos modèles surpassent la baseline aléatoire. Cependant, les résultats sont plus contrastés dans cette expérience. ABSMCNNN-POS et SMCNNN-POS en association avec le modèle d'embeddings EAT permet d'obtenir les meilleurs scores F1, soit 0.017 de plus que le CR-CNNN-POS de référence. Nous pouvons également voir que la baseline CR-CNNN-POS obtient son meilleur score F1 avec EAT. Cependant, les embeddings EAT ont un pourcentage élevé d'entités manquantes qui est de 39.14% (voir Tableau 3). Nous rappelons que ce corpus n'est pas conçu pour les entités Wikipédia. Ainsi, nous pouvons nous attendre à une augmentation des performances si nous pouvions apprendre les embeddings pour les entités manquantes. Pour ce jeu de données, l'information sur les positions des entités améliore considérablement les résultats en matière de classification.

Ces résultats indiquent que les embeddings EAT sont utiles pour les tâches de validation de relations lorsque l'information de position est utilisée. Le modèle linguistique de base de (Rahman *et al.*, 2018) a obtenu un score F1 de $0,4964$, soit $0,0593$ de plus que notre meilleur modèle. Cependant, il calcule des caractéristiques sur l'arbre

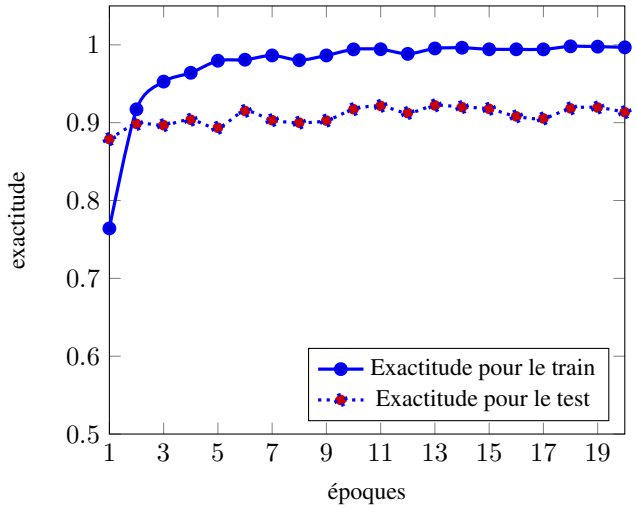

Figure 3 – Courbes de précision pour l'entraînement et le test du modèle ABSMCNNN-POS avec EAT sur dataQA

des dépendances issu de l'analyse syntaxique de la phrase tandis que ABSMCNNN-POS obtient un score comparable sans utiliser d'analyse linguistique avancée.

Pour mieux comprendre la performance de classification du meilleur modèle ABSMCNNN-POS, nous présentons les scores par type de relation dans le tableau 6. Le meilleur score est obtenu pour la relation *city_of_birth* qui compte au total 229 exemples où le ratio des exemples positifs et négatifs est $1 : 0.65$. Ces scores indiquent que le modèle de classification préserve la plupart du temps les relations correctes et rejette les erronées. En revanche, le pire résultat est obtenu pour la relation *spouse* où tous les exemples positifs, 25 (au total 131 avec les exemples négatifs), sont classés comme faux. Le comportement du modèle ne semble pas lié au nombre d'exemples par classe, qui est, en revanche, un facteur critique dans les modèles de classification multi-classes. Cette observation est corrélée avec les résultats obtenus sur l'ensemble de données de dataQA, où il existe un nombre considérable de types de relations.

Nous avons également calculé des valeurs de corrélation simples entre la similarité des entités et les vrais labels. Ils sont présentés dans le tableau 7. Pour chaque exemple du jeu de test, nous avons calculé la similarité cosinus pour un modèle donné d'embeddings. Ensuite, tous les exemples sont classés par similarité et comparés au classement basé sur les étiquettes (positives en haut, négatives en bas). Les corrélations de Spearman et de Pearson sont utilisées pour comparer les classements. Les résultats montrent un ordre cohérent de la similarité explicite des embeddings, mais à des échelles différentes. Pour dataKBP, le modèle d'embeddings le plus informatif est glove, mais les embeddings d'entités arrivent en deuxième position. Pour dataQA,

Modèle	Représentation de la relation	F1-score	Exactitude
Baselines			
CR-CNN-POS	EAT	**0.4201**	0.8183
	EAT.NOE	0.3911	0.8139
	glove	0.3097	0.8184
	word2vec	0.3721	**0.8253**
Random (Dummy)	Uniform	0.2665	0.5020
	Stratified	0.2631	0.5612
Nos modèles			
SMCNN	EAT	**0.3926**	0.7769
	EAT.NOE	0.2992	0.7768
	glove	0.3242	**0.7987**
	word2vec	0.3262	0.7930
ABSMCNN	EAT	0.3574	**0.8029**
	EAT.NOE	0.2529	0.7834
	glove	**0.3630**	0.7973
	word2vec	0.2869	0.7920
SMCNN-POS	EAT	**0.4370**	0.7978
	EAT.NOE	0.3745	**0.8266***
	glove	0.3748	0.7990
	word2vec	0.3841	0.8108
ABSMCNN-POS	EAT	**0.4371***	0.8067
	EAT.NOE	0.3956	0.7797
	glove	0.3621	0.8042
	word2vec	0.3600	**0.8082**

Tableau 5 – Performances de validation de relation (score F1 moyen et exactitude moyenne sur 5 runs à l'époque 10) avec différents modèles neuronaux et modèles d'embeddings sur dataKBP

les embeddings d'entités sont plus informatifs que tous les autres embeddings. Cependant, ces classements ne peuvent pas être directement comparés aux classements des tableaux 4 et 5, car les interactions codées dans les architectures neuronales peuvent aider certains embeddings plus que d'autres à mieux classer les exemples.

6. Conclusion

Dans cet article, nous avons proposé une architecture siamoise pour la validation de relation selon un passage de texte[9]. Au lieu de modéliser cette tâche avec un classifieur sur une seule entrée qui doit apprendre la représentation de la relation ainsi que ses expressions dans les textes, nous avons proposé un modèle à partir d'entrées plus adéquates. Suivant une architecture largement utilisée en implication textuelle, nous avons proposé de représenter le triplet d'un côté et de comparer sa représenta-

9. Ce travail a été en partie financé par le projet FUI PULSAR

Nom de la relation	Précision	Rappel	F1-score	Exactitude
person :parents	0.4669	0.4723	0.4446	0.7721
person :spouse	0.0	0.0	0.0	0.7847
person :children	0.0228	0.0108	0.0133	0.8951
person :country_of_birth	0.1233	0.8400	0.2144	0.8823
person :country_of_death	0.8611	0.2889	0.4103	0.7870
person :city_of_birth	0.9625	0.9741	0.9675	0.9607
person :city_of_death	0.5328	0.4267	0.4443	0.8848
person :employee_or_member_of	0.2990	0.3233	0.2997	0.7770
org :country_of_headquarters	0.5213	0.2014	0.2799	0.7124
org :city_of_headquarters	0.7171	0.5438	0.5997	0.7710
org :member_of	0.0319	0.0296	0.0293	0.8923
org :top_members_employees	0.1292	0.0623	0.0799	0.7373

Tableau 6 – Résultats de classification, relation par relation (scores moyens à l'époque 10), par ABSMCNN-POS avec les embeddings EAT sur dataKBP.

		EAT	EAT.NOE	glove	word2vec
dataKBP	Spearman	-0.0567	0.0004	-0.0867	-0.0346
	Pearson	-0.0364	-0.0162	-0.0932	-0.0324
dataQA	Spearman	-0.0244	-0.0058	0.0154	-0.0053
	Pearson	-0.0274	-0.0051	0.0118	-0.0054

Tableau 7 – Corrélations de Spearman et Pearson entre la similarité des deux entités à valider et la vérité terrain sur les données de test. Les valeurs proches de (-1,1) et (0) indiquent respectivement une corrélation supérieure ou inférieure

tion avec une représentation du texte de l'autre côté, dans un modèle neuronal siamois. Plusieurs modèles d'embeddings ont été testés, et en particulier un modèle joint d'embeddings d'entité et de mots en comparaison avec des modèles d'embeddings de mots uniquement. L'architecture que nous proposons présente l'avantage de ne pas nécessiter d'ingénierie des traits ou le recours à des ressources externes, ce qui peut s'avérer coûteux ou non disponible.

D'après nos résultats expérimentaux, la représentation explicite des triplets est robuste et permet de valider l'existence d'une relation dans un texte, même dans le cas d'un très grand nombre de relations. En ce qui concerne la question portant sur les modèles d'embeddings, les résultats montrent que l'utilisation d'embeddings d'entités améliore la représentation des triplets et donc les performances des modèles.

Afin de tester nos modèles, nous avons construit un corpus de validation de relation, qui est disponible publiquement[10]. A notre connaissance, c'est le premier corpus publiquement disponible pour cette tâche.

10. `https://github.com/jgmorenof/dataQA`

7. Bibliographie

Augenstein I., Web Relation Extraction with Distant Supervision, PhD thesis, University of Sheffield, 2016.

Banko M., Cafarella M. J., Soderland S., Broadhead M., Etzioni O., « Open Information Extraction from the Web. », *IJCAI*, vol. 7, p. 2670-2676, 2007.

Berant J., Chou A., Frostig R., Liang P., « Semantic parsing on freebase from question-answer pairs », *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, p. 1533-1544, 2013.

Bordes A., Chopra S., Weston J., « Question Answering with Subgraph Embeddings », *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Association for Computational Linguistics, Doha, Qatar, p. 615-620, October, 2014.

Bunescu R. C., Mooney R. J., « A shortest path dependency kernel for relation extraction », *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, p. 724-731, 2005.

Chen D., Fisch A., Weston J., Bordes A., « Reading Wikipedia to Answer Open-Domain Questions », *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1 : Long Papers)*, Association for Computational Linguistics, p. 1870-1879, 2017.

Culotta A., Sorensen J., « Dependency Tree Kernels for Relation Extraction », *Proceedings of the 42Nd Annual Meeting on Association for Computational Linguistics*, ACL '04, Association for Computational Linguistics, Stroudsburg, PA, USA, 2004.

Dligach D., Miller T., Lin C., Bethard S., Savova G., « Neural Temporal Relation Extraction », *EACL 2017*p. 746, 2017.

dos Santos C., Xiang B., Zhou B., « Classifying Relations by Ranking with Convolutional Neural Networks », *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1 : Long Papers)*, Association for Computational Linguistics, p. 626-634, 2015.

Fang W., Zhang J., Wang D., Chen Z., Li M., « Entity Disambiguation by Knowledge and Text Jointly Embedding », *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, Association for Computational Linguistics, p. 260-269, 2016.

Fundel K., Küffner R., Zimmer R., « RelEx—Relation extraction using dependency parse trees », *Bioinformatics*, vol. 23, n° 3, p. 365-371, 2007.

Gamallo P., Garcia M., Fernández-Lanza S., « Dependency-based open information extraction », *Proceedings of the joint workshop on unsupervised and semi-supervised learning in NLP*, Association for Computational Linguistics, p. 10-18, 2012.

Giampiccolo D., Forner P., Herrera J., Peñas A., Ayache C., Forascu C., Jijkoun V., Osenova P., Rocha P., Sacaleanu B. *et al.*, « Overview of the CLEF 2007 multilingual question answering track », *Workshop of the Cross-Language Evaluation Forum for European Languages*, Springer, p. 200-236, 2007.

Hoffmann R., Zhang C., Ling X., Zettlemoyer L., Weld D. S., « Knowledge-based weak supervision for information extraction of overlapping relations », *Proceedings of the 49th Annual*

Meeting of the Association for Computational Linguistics : Human Language Technologies-Volume 1, Association for Computational Linguistics, p. 541-550, 2011.

Lan W., Xu W., « Neural Network Models for Paraphrase Identification, Semantic Textual Similarity, Natural Language Inference, and Question Answering », *Proceedings of the 27th International Conference on Computational Linguistics*, Association for Computational Linguistics, p. 3890-3902, 2018.

Mikolov T., Sutskever I., Chen K., Corrado G. S., Dean J., « Distributed representations of words and phrases and their compositionality », *Advances in neural information processing systems*, p. 3111-3119, 2013.

Mintz M., Bills S., Snow R., Jurafsky D., « Distant supervision for relation extraction without labeled data », *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP : Volume 2-Volume 2*, Association for Computational Linguistics, p. 1003-1011, 2009.

Moreno J. G., Besançon R., Beaumont R., D'hondt E., Ligozat A.-L., Rosset S., Tannier X., Grau B., « Combining word and entity embeddings for entity linking », *European Semantic Web Conference*, Springer, p. 337-352, 2017.

Nguyen T. H., Grishman R., « Relation extraction : Perspective from convolutional neural networks », *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, p. 39-48, 2015.

Niu F., Zhang C., Ré C., Shavlik J. W., « DeepDive : Web-scale Knowledge-base Construction using Statistical Learning and Inference. », *VLDS*, vol. 12, p. 25-28, 2012.

Rahman R., Grau B., Rosset S., « Impact of Entity Graphs on Extracting Semantic Relations », *in* J. A. Lossio-Ventura, H. Alatrista-Salas (eds), *Information Management and Big Data*, Springer International Publishing, Cham, p. 31-47, 2018.

Riedel S., Yao L., McCallum A., « Modeling relations and their mentions without labeled text », *Machine Learning and Knowledge Discovery in Databases*, Springer, p. 148-163, 2010.

Rodrigo A., Herrera J., Peñas A., « The effect of answer validation on the performance of Question-Answering systems », *Expert Systems with Applications*, vol. 116, p. 351-363, 2019.

Rodriguez M., Goldberg S., Wang D. Z., « University of Florida DSR lab system for KBP slot filler validation 2015 », *Proceedings of the Eighth Text Analysis Conference (TAC2015)*, 2015.

Sammons M., Song Y., Wang R., Kundu G., Tsai C.-T., Upadhyay S., Ancha S., Mayhew S., Roth D., « Overview of UI-CCG systems for event argument extraction, entity discovery and linking, and slot filler validation », *Urbana*, vol. 51, p. 61801, 2014.

Severyn A., Moschitti A., « Learning to Rank Short Text Pairs with Convolutional Deep Neural Networks », *Proceedings of the 38th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '15, p. 373-382, 2015.

Surdeanu M., Ji H., « Overview of the english slot filling track at the tac2014 knowledge base population evaluation », *Proc. Text Analysis Conference (TAC2014)*, 2014.

Viswanathan V., Rajani N. F., Bentor Y., Mooney R., « Stacked Ensembles of Information Extractors for Knowledge-Base Population », *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1 : Long Papers)*, vol. 1, p. 177-187, 2015.

Vu N. T., Adel H., Gupta P., Schütze H., « Combining Recurrent and Convolutional Neural Networks for Relation Classification », *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*, Association for Computational Linguistics, San Diego, California, p. 534-539, June, 2016.

Wang I.-J., Liu E., Costello C., Piatko C., « JHUAPL TAC-KBP2013 Slot Filler Validation System », *Proceedings of the Sixth Text Analysis Conference (TAC 2013)*, vol. 24, 2013.

Wang R., Neumann G., « Relation validation via textual entailment », *Ontology-based information extraction systems (obies 2008)*, 2008.

Yamada I., Shindo H., Takeda H., Takefuji Y., « Joint Learning of the Embedding of Words and Entities for Named Entity Disambiguation », *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, Association for Computational Linguistics, p. 250-259, 2016.

Yao L., Haghighi A., Riedel S., McCallum A., « Structured relation discovery using generative models », *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, Association for Computational Linguistics, p. 1456-1466, 2011.

Yin W., Schütze H., Xiang B., Zhou B., « ABCNN : Attention-Based Convolutional Neural Network for Modeling Sentence Pairs », *Transactions of the Association for Computational Linguistics*, vol. 4, p. 259-272, 2016.

Yu D., Huang H., Cassidy T., Ji H., Wang C., Zhi S., Han J., Voss C., Magdon-Ismail M., « The Wisdom of Minority : Unsupervised Slot Filling Validation based on Multi-dimensional Truth-Finding », *Proceedings of 2014 International Conference on Computational Linguistics*, August, 2014.

Zheng S., Xu J., Zhou P., Bao H., Qi Z., Xu B., « A neural network framework for relation extraction : Learning entity semantic and relation pattern », *Knowledge-Based Systems*, vol. 114, p. 12-23, 2016.