Graphe de communauté pour la validation de relations dans le cadre de la population de bases de connaissances

Rashedur Rahman[1,2,3], Brigitte Grau[2,3,4], Sophie Rosset[2,3]

[1] *IRT SystemX*
[2] *LIMSI, CNRS*
[3] *University Paris Saclay*
[4] *ENSIIE*

RÉSUMÉ. *L'extraction de relations entre entités à partir de textes est une étape importante pour des tâches d'extraction d'information ou de découverte de connaissances. Les systèmes produisent de nombreux candidats et la tâche de validation de relation consiste à décider si une relation candidate est correcte ou non en fonction des informations fournies par les systèmes. Dans cet article, nous proposons un nouvel ensemble de traits fondés sur l'analyse des graphes engendrés par les relations entre entités, qui complète ceux provenant d'une analyse linguistique.*

ABSTRACT. *Relation extraction between entities from text plays an important role in information extraction and knowledge discovery related tasks. A relation validation method justifies a claimed relation based on the provided information. In this paper we propose a relation validation method with some linguistic and world knowledge-based graph features for validating a claimed relation.*

MOTS-CLÉS : *Validation de relation$_1$, Population de base de connaissance$_2$, Analyse de graphe de communauté$_3$.*

KEYWORDS: *Relation Validation$_1$, Knowledge Base Population$_2$, Community Graph Analysis$_3$.*

1. Introduction

Extraire des relations exprimées entre entités dans des documents est une tâche importante pour de nombreuses applications comme le peuplement automatique de bases de connaissance ou encore la réponse automatique à des questions précises. Cette tâche est d'autant plus complexe lorsqu'on s'intéresse à des entités en domaine ouvert décrites par un très grand nombre de relations sémantiques. Les relations peuvent être aussi diverses que les liens familiaux d'une personne (conjoint, enfants, etc) ou les caractéristiques d'une société (filiale, taille du personnel, dirigeants, etc), etc. Cette tâche fait l'objet d'une campagne d'évaluation internationale, TAC-KBP[1], pour laquelle les systèmes doivent reconnaître une soixantaine de relations sémantiques reliant différentes sortes d'entités (essentiellement des lieux, des personnes, des organisations et leur différentes sous-catégories). Une tâche de validation des relations fournies par les différents systèmes a été créée afin d'étudier si l'extraction de relation peut bénéficier des apports de chacun des systèmes. La méthode décrite dans cet article entre dans le cadre de cette dernière tâche, qui consiste donc, étant donné une entité, une relation, et une réponse fournie par un système (sa valeur et l'extrait de texte qui la supporte), de décider si la valeur est correcte ou non, c'est-à-dire de valider ou non la relation proposée.

Différentes approches ont été étudiées pour la validation de relations notamment par l'évaluation de la confiance que l'on peut avoir dans la source de la réponse : le document qui justifie la réponse et la fiabilité du système (Yu *et al.*, 2014 ; Viswanathan *et al.*, 2015). D'autres critères sont également proposés, et concernent la validation sémantique d'une relation par des caractéristiques linguistiques (Yu *et al.*, 2014 ; Niu *et al.*, 2012 ; Hoffmann *et al.*, 2011 ; Yao *et al.*, 2011 ; Riedel *et al.*, 2010), caractéristiques analogues à celles utilisées en extraction de relation.

Néanmoins, le plus souvent, les différentes méthodes ne tiennent pas compte des informations disponibles au niveau de l'entité, qui peuvent être calculées sur l'ensemble de la collection de documents, et elles reposent plutôt sur des caractérisations de l'expression de la relation au niveau des mentions, c'est-à-dire de la phrase. Or de telles informations permettent d'introduire un type de connaissance sur le monde, autorisant la prise de décision sur des critères indépendants de l'expression linguistique de la relation et qui viennent les compléter. Nous faisons l'hypothèse que si deux entités sont en relation, elle sont liées à plus d'entités communes et ce de manière récurrente, que si elles ne sont pas liées par cette même relation. Par exemple le conjoint d'une personnalité partagera plus des mêmes lieux et des relations avec des mêmes personnes avec son conjoint qu'avec d'autres personnes. De ce fait, nous avons extrait un graphe d'entités de la collection qui nous a permis de proposer de nouvelles caractérisations des relations. Nous introduisons différentes mesures calculées sur ce graphe d'entités, qui ont été, pour certaines, utilisées avec succès dans d'autres tâches, comme l'entropie pour la détection de connaissances dans des réseaux de publication (Holzinger *et al.*, 2013) ou l'information mutuelle pour la validation de réponses

1. https ://tac.nist.gov/

dans des systèmes de questions-réponses (Magnini *et al.*, 2002 ; Cui *et al.*, 2005). Nous avons également exploré l'utilisation de caractéristiques fondées sur l'analyse de graphes communautaires (Han *et al.*, 2011 ; Friedl *et al.*, 2010 ; Solá *et al.*, 2013).

Nous proposons d'effectuer la validation de relations proposées par des systèmes par une classification binaire reposant sur trois catégories d'information pour caractériser les réponses : des informations linguistiques associées à l'expression des relations dans les textes, des informations issues des graphes d'entités construits sur la collection, et enfin des informations liées aux systèmes et aux propositions faites. Nos modèles sont évalués sur les données des campagnes 2015 de KBP. Nous montrons que l'ensemble des caractéristiques permettent de dépasser la baseline de 12 points et que les mesures calculées sur les graphes d'entités permettent d'apporter des informations complémentaires aux informations linguistiques (plus 5 points).

2. Etat de l'art

La validation de relations candidates, que nous nommerons aussi hypothèses de relations, repose sur différents types de caractéristiques permettant de décider si une hypothèse de relation est valide ou non.

Les caractéristiques linguistiques sont principalement les chemins dans l'arbre syntaxique, l'existence de mots amorces entre les paires de mentions d'entités ainsi que les types d'entités (Yu *et al.*, 2014). Il s'agit là des traits habituellement utilisés par les systèmes d'extraction de relation. Les travaux de (Culotta et Sorensen, 2004 ; Bunescu et Mooney, 2005 ; Fundel *et al.*, 2007) utilisent les arbres de dépendance pour décider de la présence ou de l'absence d'une relation de manière non supervisée. (Gamallo *et al.*, 2012) ont proposé une analyse en dépendance pour l'extraction d'information en domaine ouvert qui s'appuie sur des règles prédéfinies. Ils ont défini des patrons de relation en analysant les types de propositions verbales. Enfin, (Chowdhury et Lavelli, 2012) ont proposé un noyau hybride combinant des patrons de dépendance et des mots amorces pour extraire des relations en domaine médical. Nous nous appuierons également sur la construction de motifs de dépendances pour valider les hypothèses de relation.

La détection du type sémantique d'une relation repose quant à elle sur l'information lexicale comme les mots présents dans l'entourage des mentions. Ces informations ont été utilisées dans différentes approches semi-supervisées pour l'extraction d'information et l'extraction de relations (Niu *et al.*, 2012 ; Hoffmann *et al.*, 2011 ; Yao *et al.*, 2011 ; Riedel *et al.*, 2010 ; Mintz *et al.*, 2009). L'un des problèmes principaux réside dans l'emploi de vocabulaires différents pour exprimer une relation, et requiert de s'appuyer sur des ressources explicites (e.g. listes de synonymes ou de paraphrases) ou des représentations distribuées pour rapprocher des mots de sens proche.

Pour valider des hypothèses de relation provenant de systèmes différents, les méthodes proposées s'appuient sur le vote et l'estimation de la crédibilité des systèmes et des sources d'information. Ainsi (Sammons *et al.*, 2014) ont utilisé un vote majoritaire

pour valider les réponses proposées pour compléter une base de connaissance (tâche Slot Filling dans KBP). (Wang *et al.*, 2013) ont modélisé le problème sous forme d'un système de satisfaction de contraintes en se fondant sur les scores de confiance fournis pour valider les réponses. (Yu *et al.*, 2014) ont proposé une mesure de la crédibilité de la réponse, de la source et du système qui permettent une optimisation multidimensionnelle. (Viswanathan *et al.*, 2015) ont eux utilisé la combinaison de multiples classifieurs et leurs scores de confiance.

Des mesures de l'information sur des graphes ou sur l'ensemble d'une collection ont également été proposées pour des tâches connexes. Ainsi, (Holzinger *et al.*, 2013) s'appuient sur l'entropie pour découvrir des connaissances dans des réseaux de publication. L'information mutuelle a également été utilisée pour la validation de réponses dans des systèmes de questions-réponses (Magnini *et al.*, 2002 ; Cui *et al.*, 2005) pour exploiter la redondance. Dans (Friedl *et al.*, 2010), des mesures de centralité permettent de trouver des nœuds importants et influents dans un réseau social. (Solá *et al.*, 2013) ont exploré le concept de centralité de vecteur propre ("eigenvector centrality"). Nous avons appliqué ces différentes mesures pour l'extraction de relation sur des graphes d'entités construits à partir des textes de la collection.

3. Les tâches de la conférence KBP

La tâche "Slot Filling" (SF) de la campagne d'évaluation TAC-KBP existe depuis 2009. Environ 60 types de relations (i.e. les slots) sont définies[2], comme *conjoint, fondé_par, etc.*, dont certaines sont des relations inverses, comme *pays de naissance* et *naissances d'un pays*. Certaines de ces relations sont monovaluées (par exemple *date de naissance*), et d'autres multivaluées (par exemple, *enfants*). Plus spécifiquement, la tâche "Cold Start Slot Filling" (CSSF) consiste, pour un système participant, à répondre à un ensemble de requêtes en respectant un format défini. Le NIST, organisateur de la campagne d'évaluation, fournit une collection de documents dans laquelle les éléments de la requête et l'élément de réponse figurent potentiellement. Une requête, cf Exemple 1 consiste en une entité (<name>), son type, (<enttype>) et une relation (<slot>) à trouver. Un document et la position de l'entité dans le document sont également fournis. Les systèmes doivent trouver les réponses à ces requêtes (tour 1), et rechercher la relation mentionnée dans <slot1> sur les entités réponses (tour 2).

Exemple 1 : Requête pour Cold Start Slot Filling

```
<query_id="CSSF15_ENG_SampleQueryRound-1">
<name>Barack Obama</name>
<enttype>PER</enttype>
<slot>per:spouse</slot>
<slot1>per:age</slot1>
</query>
```

2. https ://tac.nist.gov//2015/KBP/ColdStart/guidelines.html

Une réponse doit contenir la valeur de l'élément réponse, son type, sa position dans un texte, la position dans un texte de l'expression de la relation avec le texte associé et un score de confiance. Le segment de texte doit justifier la réponse fournie par le système. Le NIST organise dans ce cadre une tâche de validation de réponses[3], appelée Slot Filler Validation (SFV) dont l'objectif est d'évaluer la capacité d'un système à valider ou non les hypothèses fournies par différents système de SF. D'un point de vue général, on peut considérer que la tâche de validation est une tâche de classification binaire permettant de déterminer si une hypothèse est correcte ou non. Un système de validation dispose en entrée des réponses des différents systèmes CSSF, des requêtes et de la collection de textes.

4. Graphe de communauté d'entités

4.1. *Définition du graphe*

Considérons un graphe $G = (E, R)$, la relation de la requête (slot) r_q, l'entité de la requête $e_q \epsilon E$ et des candidats réponses $e_c = \{e_{c1}, e_{c2}, \ldots, e_{cn}\} \epsilon E$ où $r_q = r(e_q, e_c) \epsilon R$. La liste des candidats est générée par les systèmes d'extraction de relation. Supposons également qu'il existe d'autres relations sémantiques $r_o \epsilon R$ où $r_o \neq r_q$. Nous voulons donc savoir si un candidat e_{ci} de E_c est correct ou incorrect pour une relation exprimée dans la requête r_q en analysant les communautés X_q et X_c formées par l'entité requête et chaque candidat réponse. Une communauté X_i contient les voisins de e_i, et ce pour plusieurs pas éventuellement.

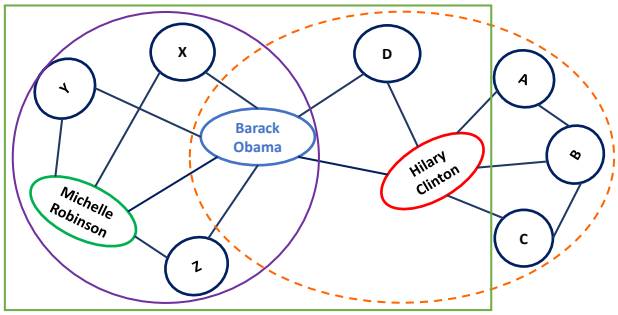

Figure 1. *Graphe de communauté*

La figure 1 montre un exemple de ce type de graphe où l'entité de la requête, son type et sa valeur sont respectivement *Barack Obama*, *person* and *spouse*. Les candidats réponses sont *Michelle Robinson* et *Hilary Clinton*, les deux étant reliés à *Barack Obama* par une hypothèse de relation de type *spouse*. L'objectif est de pouvoir classer la réponse *Michelle Robinson* comme correcte en s'appuyant sur une analyse de

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

communauté. Les communautés de *Barack Obama* (rectangle vert), *Michelle Robinson* (cercle violet) et *Hilary Clinton* (ellipse en pointillé orange) sont définies à partir des relations *in_same_sentence* qui signifient que les paires d'entités cooccurrent dans une même phrase dans les documents. Le graphe est donc construit à partir de relations sémantiques non typées, fondées sur les cooccurrences. Il serait possible d'utiliser des relations sémantiques typées fournies par un système d'extraction de relations.

4.2. *Création du graphe de communauté*

Le graphe sur les entités tel qu'il est illustré figure 1 est créé à partir d'un graphe représentant les connaissances extraites des textes, voir figure 2, partie basse, nommé graphe de connaissance. Ce graphe de connaissance est généré après avoir appliqué des systèmes de reconnaissance des entités nommées, avoir découpé les textes en phrases et extrait les relations sémantiques sur l'ensemble du corpus. La reconnaissance des entités nommées est effectuée par *Luxid* [4], qui est capable de décomposer les mentions en composants, tels que le prénom et la fonction, et par le système de Stanford (Manning *et al.*, 2014). Lorsque les deux systèmes sont en désaccord, nous choisissons l'annotation produite par Luxid. Le graphe de connaissance représente les documents, phrases, mentions et entités comme des nœuds et les arêtes entre ces nœuds représentent des relations entre ces éléments.

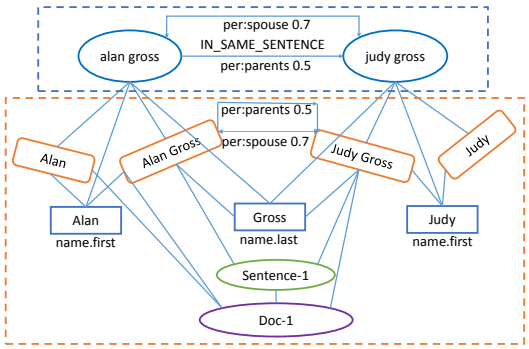

Figure 2. *Graphe de connaissance*

Les multiples mentions d'une même entité trouvées dans un même document sont reliées à un même nœud entité dans le graphe de connaissance, en se fondant sur la similarité textuelle des mentions et de ses composants éventuels, ce qui correspond à une première étape de liaison référentielle sur des critères locaux. Cette opération est réalisée par Luxid. Cependant une entité peut être mentionnée dans différents documents, qui plus est sous différentes formes (par exemple *Barack Obama, President Barack Obama, President Obama* etc.) ce qui crée des nœuds redondants dans le graphe

4. `http://www.expertsystem.com/fr/`

de connaissance. Une étape de regroupement des entités en fonction de la similarité de leurs noms et de la densité de leurs entités voisines calculées par l'équation (1) regroupe les entités considérées comme identiques en une seule entité dans le graphe de communauté (Fig. 2, partie haute). Le graphe est donc construit à partir des informations sur les entités et relations présentes dans le graphe de connaissance et le lien avec les documents est toujours maintenu. Il est ainsi possible de connaître le nombre d'occurrences de chaque entité et chaque relation. Le graphe est stocké dans une base Neo4j, une base de données orientée graphe, qui permet d'extraire les sous-graphes liés à une entité par des requêtes. Nous n'avons considéré que les types d'entité *person, location et organization* comme membres des communautés.

5. Validation de relation

Afin de prédire la validité ou non d'une relation, nous avons considéré ce problème comme une classification binaire et défini des traits fondés sur trois catégories d'informations. Nous avons calculé un ensemble de traits exploitant les graphes (voir section 5.1) auxquels nous avons ajouté des traits fondés sur une analyse linguistique du passage de texte qui justifie le candidat et décrit la relation (voir section 5.2) ainsi qu'une estimation de la confiance dans la provenance des candidats (voir section 5.3). La table 1 récapitule tous les traits utilisés.

5.1. *Traits fondés sur les graphes*

Nous supposons qu'un candidat correct pour une requête est un membre important dans la communauté de l'entité de la requête. La communauté X_e d'une entité est définie par le sous-graphe formé par ses voisins à plusieurs pas. La communauté de deux entités fusionne les communautés des deux entités. Nous avons donc défini différents traits en relation avec cette hypothèse.

Nous faisons l'hypothèse que la densité du réseau (voire équation 1) de la communauté d'un candidat correct avec l'entité de la requête doit être plus élevée que la densité de la communauté d'un candidat incorrect avec cette même entité.

$$\rho_{X_e} = \frac{\#\, de\, liens\, existant\, sur\, e}{\#\, de\, liens\, possibles} \qquad [1]$$

Ceci est illustré sur la figure 1 où la communauté de *Barack Obama* avec *Michelle Robinson* est plus dense que celle avec *Hilary Clinton*.

La centralité de vecteur propre (Bonacich et Lloyd, 2001) mesure l'influence d'un nœud dans un graphe. Un nœud sera d'autant plus influent si il est connecté à d'autres nœud influents. Nous faisons l'hypothèse que l'entité de la requête sera plus influencée par l'entité réponse que par d'autres candidats. Nous mesurons l'influence des candidats dans la communauté de l'entité de la requête en calculant la différence absolue entre le score de centralité des vecteurs propres de l'entité de la requête et de chaque

Catégorie	Traits
Graphe	Densité du réseau
	Centralité de vecteur propre
	Information mutuelle
	Similarité cosinus
	6 ratios sur la collection de documents
Linguistique	Distance d'édition minimale entre des motifs de dépendances
	Longueur du motif de dépendances
	Présence dans la même proposition
	Présence de mot amorce entre les mentions
	Présence de mot amorce dans le chemin de dépendances
	Présence de mot amorce dans le sous-arbre minimal
	Relation sans mot amorce
Baseline (vote)	Crédibilité de la mention et du document
	Crédibilité du système
	Score de confiance du candidat

Tableau 1. *Traits pour la validation de relation*

candidat. Nous faisons donc l'hypothèse, ici, que cette différence sera plus petite pour un candidat correct que pour un candidat incorrect.

Soit $A = (a_{i,j})$ la matrice d'adjacence du graphe G. Le score de centralité x_i d'un nœud i est calculé récursivement par l'équation 2.

$$x_i = \frac{1}{\lambda} \sum_k a_{k,i} x_k \qquad [2]$$

où, $\lambda \neq 0$ est une constante et l'équation peut être exprimée sous la forme : $\lambda x = xA$

Nous faisons également l'hypothèse que l'information mutuelle (voire équation 3) et la similarité (voire équation 4) de la communauté de l'entité de la requête avec la communauté d'un candidat correct doivent être plus élevées qu'avec la communauté d'un candidat incorrect.

$$MI(X_q, X_c) = H(X_q) + H(X_c) - H(X_q, X_c) \quad [3]$$

$$\text{où } H(X) = -\sum_{i=1}^{n} p(e_i)\ log_2(p(e_i))\ et\ p(e) = \frac{\#\ de\ liens\ de\ e}{\#\ de\ liens\ de X}$$

où X_q et X_c sont les communautés de l'entité de la requête et de celle du candidat et p(e), la probabilité du degré de centralité d'un membre de la communauté.

$$cosine\ similarity = \frac{|X_q \cap X_c|}{\sqrt{|X_q||X_c|}} \quad [4]$$

où, X_q et X_c sont resp. l'ensemble des membres de la communauté de la requête et de celle du candidat.

La communauté d'une entité (entité de la requête ou candidat) est étendue jusqu'au troisième pas pour mesurer la centralité des vecteurs propres et l'information mutuelle.

De plus, nous définissons six ratios au niveau de la collection de documents pour tenir compte de la redondance des informations extraites et de leur distribution sur les documents, les phrases et les relations. Ces ratios sont définis par les équations 5 à 10. L'équation 5 rend compte de la fréquence de la relation entre un candidat et la requête ramenée à la fréquence d'apparition du candidat dans les textes.

$$r_{mention}(v_c) = \frac{\#\ de\ mentions\ associées\ à\ v_q}{\#\ total\ de\ mentions de v_c} \quad [5]$$

L'équation 6 s'intéresse au nombre de documents justifiant l'hypothèse de relation que l'on cherche à valider et l'équation 7 aux phrases qui la justifient.

$$r_{hyp_doc}(e_q, e_c) = \frac{\#\ de\ documents\ avec\ r_q(e_q, e_c)}{\#\ total\ de\ r_q(e_q, e_c)} \quad [6]$$

$$r_{hyp_sent}(e_q, e_c) = \frac{\#\ total\ de\ r_q(e_q, e_c)\ hyp.}{\#\ de\ phrases\ contenant\ e_q\ et\ e_c} \quad [7]$$

L'équation 8 mesure la probabilité d'apparition de la relation dans des documents différents et 9 indique la fréquence de cette relation.

$$r_{doc}(e_q, e_c) = \frac{\#\ de\ documents\ avec\ r_q(e_q, e_c)}{\#\ total\ de\ documents\ avec\ tous\ les\ candidats} \quad [8]$$

$$relFreq_{hyp}(e_c) = \frac{\#\ de\ r_q(e_q, e_c)\ hyp.}{\#\ total\ de\ r_q\ avec\ tous\ les\ candidats} \quad [9]$$

Dans 10, on s'intéresse aux entités autres présentes dans les phrases où la relation figure, qui peuvent rendre compte du bruit, car il est plus difficile d'extraire une relation en présence de nombreuses entités.

$$r_{entity}(e_q, e_c) = \frac{\#\, d'\, entités\, diff\acute{e}rentes\, dans\, toutes\, les\, r_q(e_q, e_c)}{\#\, total\, d'\, entités\, dans\, toutes\, les\, phrases de\, r_q(e_q, e_c)} \quad [10]$$

5.2. *Traits linguistiques*

On considère généralement que la relation entre deux entités est exprimée au niveau de la phrase. De ce fait, la phrase constitue notre unité pour définir des critères liés à l'expression linguistique d'une relation. Nous avons défini des critères syntaxiques pour déterminer si une relation existe ou non entre deux mentions d'entités. Pour déterminer la catégorie sémantique d'une relation, nous nous sommes appuyés sur des mots amorces.

Les traits syntaxiques sont calculés à partir du résultat d'une analyse en dépendances, i.e. l'analyseur syntaxique (Manning *et al.*, 2014) fournit un arbre où les nœuds sont les mots de la phrase et les arcs entre eux sont étiquetés par leur fonction syntaxique. Nous avons extrait une liste de motifs de dépendances pour chaque relation à partir d'un ensemble d'exemples annotés. Par exemple, dans la phrase *Paola, Queen of the Belgians is the wife of King Albert of Belgium.*, les dépendances sont *nn(Queen, Paola), nsubj(wife, Queen), prep_of(wife, Albert)* et le motif de dépendances entre *Paola* et *King Albert* est *[nn, nsubj, prep_of]*. Pour réduire le nombre de motifs, nous simplifions le motif *[nn, nsubj, prep_of]* en *[nsubj, prep_of]* en supprimant les dépendances *nn* aux extrémités. Certains motifs de dépendances contiennent des répétitions consécutives d'étiquettes comme *[nsubj, dobj, prep_of, prep_of, poss]*. Dans de tels cas, nous simplifions le motif en substituant la succession de labels par un seul label. Ainsi *[nsubj, dobj, prep_of, prep_of, poss]* est simplifié en *[nsubj, dobj, prep_of, poss]*. Cette simplification permet de généraliser les motifs de dépendances.

Les motifs ainsi formés sont comparés, en utilisant une distance d'édition, avec le chemin de dépendance simplifié d'une phrase. Par exemple, si nous avons une liste de motifs de dépendances (a,b,c), (a,c,d), (b,c,d) pour une relation R et un motif de dépendances (a,c,b) extrait d'une phrase contenant une relation R détectée entre la requête et le candidat, nous calculons la distance d'édition entre chaque paire [(a,c,b), (a,b,c)], [(a,c,b), (a,c,d)],[(a,c,b), (b,c,d)] et considérons la plus faible comme un trait. Les relations étant souvent exprimées avec des chemins de dépendances courts, la longueur du motif simplifié est également prise comme trait.

De plus, le fait que les deux mentions soient ou non dans la même proposition est également considéré, comme cela a été proposé par (Chowdhury et Lavelli, 2012).

L'analyse sémantique est réalisée en s'appuyant sur des mots amorces associés au type de relation considéré. Les traits sémantiques sont des valeurs booléennes qui s'appuient sur des mots amorces positifs et négatifs. Les mots positifs sont les mots qui sont très fortement associés à une relation donnée et une seule alors que les mots

négatifs nient la relation considérée. Par exemple, les mots *wife, husband, married* sont des mots amorces positifs de la relation *spouse* alors que les mots *parent, children, brother* sont des mots amorces négatifs de cette même relation. Ces mots amorces ont été collectés sur un corpus annoté en relations.

Les mots exprimant une relation étant très variés, il est peu probable de pouvoir tous les collecter. Nous avons donc associé à chaque mot amorce un vecteur de type plongement lexical en utilisant un modèle *word2vec* pré-entraîné. Ainsi décider si un mot est ou non une amorce dépend de la similarité entre leurs vecteurs. Prenons *[a, b]* deux mots situés entre les mentions des entités de la requête et du candidat pour la relation R et *[x, y, z]* les mots amorces positifs pour cette relation. La similarité cosinus entre les vecteurs de chacune des paires *[(a,x), (a,y), (a,z), (b,x), (b,y), (b,z)]* est calculée. Si l'un des scores satisfait un seuil prédéfini, alors nous considérons qu'il existe un mot amorce positif. Le même calcul est effectué avec les mots négatifs. La présence de mots amorces, positifs ou négatifs, est vérifiée pour trois cas : 1) entre les mentions dans la forme de surface de la phrase, 2) dans le chemin de dépendances et 3) dans l'arbre minimal qui contient les deux mentions comme cela a été proposé par (Chowdhury et Lavelli, 2012).

Certaines relations peuvent être exprimées sans la présence de mot amorce. Par exemple, le passage *Mr. David, from California won the prize* exprime la relation *country of residence* sans utiliser explicitement de mot amorce. Les relations sont donc classées en deux catégories : peut être exprimée sans mot amorce ou non. Un indicateur booléen est utilisé (*Relation sans mot amorce*) comme trait.

5.3. *Votes et scores de confiance*

Nous utilisons et calculons un score de crédibilité pour les candidats, les documents et les systèmes en nous appuyant sur toutes les réponses données par les différents systèmes à une requête.

Soient F les candidats d'une requête Q fournis par les systèmes S ; ces candidats sont trouvés dans des documents D . La crédibilité d'un candidat F_i et celle d'un document D_i sont calculées avec les équations 11 et 12. Si un candidat pour une requête existe dans plusieurs documents alors la plus haute valeur de crédibilité des documents est retenu pour ce candidat.

$$filler\ credibility(F_i) = \frac{\#\ occurrences\ de\ F_i}{\#\ occurrences\ de\ tous\ les\ candidats} \quad [11]$$

$$document\ credibility(D_i, Q) = \frac{\#\ de\ références\ à\ D_i}{\#\ total\ de\ documents\ référencés} \quad [12]$$

La crédibilité des systèmes est estimée en fonction du nombre de candidats communs entre tous les systèmes. Notre hypothèse est en effet que plus un système trouve de candidats trouvés par d'autres système, plus celui-ci est fiable. Une matrice à deux

dimensions est construite où les lignes et les colonnes correspondent aux systèmes. Dans un premier temps la matrice est initialisé à zéro. Si une paire de systèmes propose le même candidat pour une requête, la crédibilité des deux systèmes est augmenté de 1. Ce processus est répété pour toutes les requêtes. À l'issue du processus, la crédibilité totale de chaque système est estimée en additionnant les valeurs des lignes et colonnes correspondantes. Enfin, les valeurs de crédibilité sont normalisées par la crédibilité du système le mieux évalué. Le score de confiance fourni par les systèmes est également utilisé comme trait. Notre baseline est constituée de l'ensemble des scores de confiance que nous venons de décrire.

6. Expérimentations et résultats

6.1. *Données*

Pour nos expériences, nous avons utilisé les données de la campagne TAC-KBP Cold Start Slot Filling (CSSF) en anglais. Comme nous utilisons les données de test de 2014 pour extraire les chemins de dépendance et les mots amorces, nous avons utilisé les jeux de données de 2015 pour l'entraînement et le test. Une collection de 45 000 documents était fournie. Ces documents incluent des textes journalistiques et des textes issus de forums de discussion. Ces documents ont été analysés pour construire le graphe de connaissance.

Nos données d'entraînement et de test sont constituées à partir des réponses fournies par les systèmes participants à la tâche CSSF 2015, réponses qui ont été évaluées par le NIST. Au total, il y a 9 339 requêtes pour le premier tour et 330 314 requêtes générées à partir des réponses des systèmes aux requêtes du premier tour (2ème tour). Environ 2 000 requêtes du tour 1 et 2 500 du tour 2 ont été évaluées par le NIST. Nous n'avons retenu que les requêtes contenant des candidats corrects et incorrects et avons donc écarté de notre jeu de données les requêtes dont les réponses sont toutes évaluées comme incorrectes. Au total notre jeu de données contient 1 296 requêtes (1 080 du tour 1 et 216 du tour 2).

Pour construire un ensemble d'exemples positifs et négatifs par requête, nous avons extrait les réponses des fichiers réponses des systèmes. Ces fichiers contiennent la valeur du candidat évalué, l'identifiant du document et les positions début et fin du passage justificatif qui permettent de retourner au document, et extraire les phrases justificatives. L'évaluation d'un candidat peut être correct (C), faux (W) ou inexact (X). Celle d'un passage justificatif est correct (C), faux (W), court (S) ou long (L) avec S et L considérés comme inexact (X). Un candidat est correct si la valeur est correcte et que le passage le justifie. Il est inexact si le passage est correct et que la valeur n'est pas tout à fait exacte (trop longue ou trop courte). Une valeur incorrecte extraite d'un passage correct est fausse. Pour construire l'ensemble de candidats, nous avons sélectionné les candidats évalués C ou W et extrait les passages justificatifs leur correspondant. Au total, notre jeu de données contient 68 076 candidats. La plupart des traits que nous calculons ne peuvent l'être que sur les phrases complètes et pas

sur les passages qui sont souvent des phrases tronquées, pour respecter la limite de taille exigée. Les phrases complètes correspondant aux passages sont donc extraites des documents sources grâce aux offsets contenus dans le fichier.

Les traits linguistiques sont calculés à partir de phrases analysées où les mentions des deux entités (requête et candidat) doivent être repérées. Or notre système ne peut trouver les deux mentions dans toutes les phrases sélectionnées. Cela se produit soit lorsque l'entité de la requête ou l'entité candidate sont mentionnées par un pronom ou une anaphore nominale, dans la mesure où nous n'utilisons aucun mécanisme de résolution de co-référence. De plus, le système de détection d'entités nommées, qui résulte pourtant de deux systèmes assez performants, ne détecte pas toutes les entités présentes dans les requêtes ou les hypothèses. Cette restriction s'applique aussi pour le calcul des traits sur les graphes qui sont construits à partir des entités nommées reconnues. Ce comportement correspond à une tendance généralement observée dans les systèmes de reconnaissance d'entité nommées lorsqu'ils sont appliqués sur des documents différents de ceux sur lesquels ils ont été entraînés (ici des documents Web et des blogs au lieu d'articles de journaux or de pages Wikipedia). De plus, le fait de rajouter la contrainte de trouver deux entités dans la même phrase provoque cette baisse supplémentaire des performances. Au total 55 276 hypothèses (sur les 68 076 initiales) ont pu être traitées pour extraire les traits linguistiques et nous ne pouvons calculer des graphes de communautés que pour les réponses de 1 296 requêtes. En résumé, nous pouvons extraire à la fois les traits linguistiques et les traits sur les graphes pour 4 321 réponses provenant de 260 requêtes, (213 du tour 1 et 47 du tour 2).

6.2. *Résultats*

Nous avons entraîné un modèle en utilisant le classifieur de type *random forest*. La validation de relation est évaluée en calculant précision, rappel et F-mesure sur les résultats obtenus selon différents ensembles de traits. Nous avons procédé à une validation croisée (sur 10 échantillons). Nous avons constitué les échantillons par requête, de manière à ce qu'une même réponse donnée à une requête mais par des systèmes différents, ne puisse être présente à la fois dans les jeux d'apprentissage et de test.

Le tableau 2 présente les résultats des modèles. Nous pouvons observer que le système qui regroupe l'ensemble des traits (ceux fondés sur les graphes, les traits linguitiques et les traits fondés sur les scores de confiance) obtient la meilleure F-mesure par comparaison avec le modèle baseline (scores de confiance). De plus les traits fondés sur les graphes se montrent utiles pour améliorer les résultats, et ce même si ils sont appliqués seuls (sans les traits qui caractérisent sémantiquement la relation). Cela peut sembler surprenant, les graphes étant les mêmes pour des hypothèses portant sur des couples d'entités identiques mais liées par des relations différentes. Ils faut noter que les hypothèses de relation résultent déjà d'une sélection par les systèmes fondée sur la sémantique des relations. Ces résultats illustrent donc la complémentarité des deux types de caractérisation des hypothèses de relations. Rappelons que les traits issus de l'analyse des graphes rendent compte à la fois du contexte global des hypo-

Ensemble de traits	Précision	Rappel	F-mesure
sur 55276 réponses			
Baseline	0,870	0,860	0,865
(1) Baseline+Linguistique	0,923	0,928	**0,925**
sur 4321 réponses			
Baseline	0,871	0,826	0,848
(2) Baseline+Linguistique	0,921	0,906	0,913
(3) Baseline+Graphe	0,943	0,927	0,935
(4) Baseline+Linguistique+Graphe	0,952	0,937	**0,945**

Tableau 2. *Évaluation de la validation de relation*

thèses de relations et de leur distribution sur la collection. Ces traits se montrent donc très intéressants et il nous faudra améliorer leur couverture.

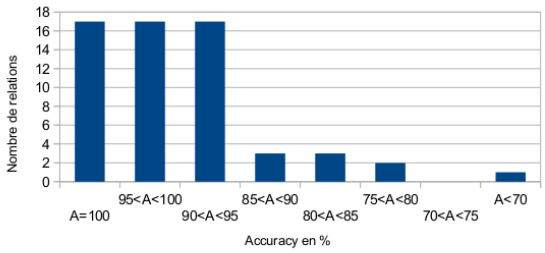

Figure 3. *Rappel du système par relations*

Le but d'un système de validation de résultats de systèmes d'extraction de relation est de pouvoir améliorer la réponse aux requêtes, donc en choisissant pour chacune la (les) réponse(s) exacte(s) parmi les propositions. Il est donc nécessaire que le système n'écarte pas toutes les propositions correctes par relation. Si l'on regarde le rappel du système, c'est-à-dire le pourcentage de réponses correctes validées[5], on obtient un score total de 92,8%, pour le système (1), où il y a 15 720 réponses correctes à valider. La figure 3 indique le nombre de relations pour lesquelles on obtient un score situé dans l'intervalle en abscisse avec ce même système. Sur 60 relations, 51 obtiennent de très bons scores (> 90%) et on peut s'apercevoir que certaines semblent plus difficiles à caractériser. C'est le cas notamment pour les relations "births in country" et "country

5. On ne tient pas compte ici des réponses fausses reconnues à juste titre comme fausses

of death" et "residents of stateorprovince", qui figurent dans les derniers intervalles. Parmi les mieux reconnues, figurent "deaths in country", "country of birth", "cause of death" ou "age". Il est difficile de savoir pourquoi les performances diffèrent par relation alors que certaines comme "country of birth" et "country of death" pourraient sembler être de difficulté analogue. Une étude approfondie des traits et de la distribution des relations dans le corpus devra être menée pour essayer de le découvrir.

7. Conclusion

Nous avons proposé dans cet article une méthode de validation de relations provenant de sorties de systèmes. Nous avons proposé un ensemble de traits permettant de caractériser l'expression d'une relation dans les textes, mais aussi, nous avons introduit des traits calculés à un niveau global. Ceux-ci sont calculés sur la présence des entités et relations dans l'ensemble de la collection et, ce qui constitue une nouveauté pour cette tâche, sur des graphes de communauté permettant de rendre compte de connaissances générales sur les entités provenant des entités auxquelles elles sont liées. Nous avons montré que nos ensembles de traits permettent d'améliorer une baseline construite à partir de votes et scores de confiance sur les réponses des systèmes.

Le calcul des différentes caractéristiques est dépendant de la capacité d'analyse des textes, notamment des résultats du système d'annotation en entité nommées. Il faudra améliorer cette partie de manière à pouvoir évaluer l'apport des graphes de communauté sur plus d'exemples.

8. Bibliographie

Bonacich P., Lloyd P., « Eigenvector-like measures of centrality for asymmetric relations », *Social networks*, vol. 23, n° 3, p. 191-201, 2001.

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.

Chowdhury M. F. M., Lavelli A., « Combining tree structures, flat features and patterns for biomedical relation extraction », *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, Association for Computational Linguistics, p. 420-429, 2012.

Cui H., Sun R., Li K., Kan M.-Y., Chua T.-S., « Question answering passage retrieval using dependency relations », *Proceedings of the 28th annual international ACM SIGIR conference on Research and development in information retrieval*, ACM, p. 400-407, 2005.

Culotta A., Sorensen J., « Dependency tree kernels for relation extraction », *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, Association for Computational Linguistics, p. 423, 2004.

Friedl D.-M. B., Heidemann J. *et al.*, « A critical review of centrality measures in social networks », *Business & Information Systems Engineering*, vol. 2, n° 6, p. 371-385, 2010.

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.

Han X., Sun L., Zhao J., « Collective entity linking in web text : a graph-based method », *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, ACM, p. 765-774, 2011.

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.

Holzinger A., Ofner B., Stocker C., Valdez A. C., Schaar A. K., Ziefle M., Dehmer M., « On graph entropy measures for knowledge discovery from publication network data », *Availability, reliability, and security in information systems and HCI*, Springer, p. 354-362, 2013.

Magnini B., Negri M., Prevete R., Tanev H., « Is it the right answer ? : exploiting web redundancy for Answer Validation », *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, Association for Computational Linguistics, p. 425-432, 2002.

Manning C. D., Surdeanu M., Bauer J., Finkel J., Bethard S. J., McClosky D., « The Stanford CoreNLP Natural Language Processing Toolkit », *Association for Computational Linguistics (ACL) System Demonstrations*, p. 55-60, 2014.

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.

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.

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.

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.

Solá L., Romance M., Criado R., Flores J., del Amo A. G., Boccaletti S., « Eigenvector centrality of nodes in multiplex networks », *Chaos : An Interdisciplinary Journal of Nonlinear Science*, vol. 23, n° 3, p. 033131, 2013.

Viswanathan V., Rajani N. F., Bentor Y., Mooney R., « Stacked Ensembles of Information Extractors for Knowledge-Base Population », *Proceedings of ACL*, 2015.

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.

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.

Yu D., Huang H., Cassidy T., Ji H., Wang C., Zhi S., Han J., Voss C. R., Magdon-Ismail M., « The Wisdom of Minority : Unsupervised Slot Filling Validation based on Multi-dimensional Truth-Finding. », *COLING*, p. 1567-1578, 2014.