Prédiction de liens dans les graphes de connaissances avec les concepts de plus proches voisins

Sébastien Ferré

Prédiction de liens dans les graphes de connaissances avec les concepts de plus proches voisins

Sébastien Ferré*

*Univ Rennes, CNRS, IRISA
Campus de Beaulieu, 35042 Rennes cedex, France
ferre@irisa.fr

Résumé. La nature ouverte des graphes de connaissances implique souvent qu'ils soient incomplets. La prédiction de liens consiste à inférer de nouveaux liens entre entités sur la base des liens existants. La plupart des approches existantes s'appuient sur l'apprentissage de vecteurs de traits latents pour l'encodage des entités et des relations. En général cependant, les traits latents ne sont pas facilement interprétables. Les approches à base de règles sont interprétables mais un ensemble de règles différent doit être appris pour chaque relation. Nous proposons une nouvelle approche qui n'a pas besoin de phase d'apprentissage et qui peut fournir des explications intelligibles pour chaque inférence. Elle repose sur le calcul de Concepts de plus proches voisins (*Concepts of Nearest Neighbours*, CNN) pour identifier des entités similaires fondées sur des motifs de graphe communs. La théorie de Dempster-Shafer est ensuite utilisée pour tirer des inférences à partir des CNN. Nous évaluons notre approche sur FB15k-237, un *benchmark* classique en prédiction de liens, où elle obtient de meilleures performances que les approches existantes.

1 Introduction

Les graphes de connaissances (GC) sont de plus en plus utilisés pour représenter et partager les données sur le Web. Le Web sémantique définit des standards pour la représentation (RDF), l'interrogation (SPARQL) et le raisonnement (RDFS/OWL) sur ces GC. Des milliers de GC sont disponibles : ex., DBpedia, Wikidata (préc. Freebase), YAGO, WordNet. La nature ouverte des GC implique souvent qu'ils soient incomplets, et différentes techniques d'apprentissage automatique ont été étudiées pour les compléter.

La tâche de *prédiction de liens* (Nickel et al., 2016) consiste à prédire les (parties d')arcs manquants. Supposons que le directeur du film *Avatar* soit absent du GC, on souhaite le prédire, c-à-d. l'identifier parmi tous les nœuds du GC. Le principe est de trouver des régularités dans les connaissances existantes et de les exploiter afin de classer les nœuds du GC. Plus haut est un nœud dans le classement, meilleure est la prédiction. La prédiction de liens a été introduite pour les réseaux sociaux avec un seul type d'arc (Liben-Nowell et Kleinberg, 2007) puis a ensuite été étendu à des données multi-relationnelles et appliquée aux GC (Nickel et al., 2016). Comparé à la classification supervisée, la prédiction de liens affronte plusieurs défis. Premièrement, il y a autant de problèmes de classification qu'il y a de relations, qui comptent souvent dans les centaines. Deuxièmement,

pour chaque relation, le nombre de "classes" est le nombre d'entités distinctes dans le co-domaine de la relation, lesquelles comptent souvent dans les milliers (ex., relation *époux*). Troisièmement, certaines relations sont multi-valuées, comme la relation entre films et acteurs.

Dans ce papier, nous rapportons de premiers résultats expérimentaux sur une nouvelle approche de prédiction de liens basée sur les *Concepts de plus-proches voisins (CNN, pour Concepts of Nearest Neighbours)* (Ferré, 2017). Cette approche est une forme symbolique des k-plus proches voisins où les distances numériques sont remplacées par des motifs de graphe qui offre une représentation intelligible de la similarité entre deux nœuds. Notre hypothèse est que le partitionnement des nœuds du GC en CNN (voir section 3) fournit une bonne base pour différents types d'inférence. Nous nous concentrons ici sur la prédiction de liens, c-à-d. sur l'inférence du nœud manquant d'un arc incomplet. La contribution de ce travail est une nouvelle approche de la prédiction de liens avec les qualités suivantes :

1. une forme d'apprentissage à base d'instances, donc *sans phase d'entraînement* ;
2. une approche symbolique, donc donnant des *explications* pour chaque inférence ;
3. des performances supérieures à l'état de l'art sur un benchmark difficile.

Le reste du papier est organisé comme suit. La section 2 discute l'état de l'art en prédiction de liens. La section 3 rappelle la définition des CNN et leur calcul efficace. La section 4 présente notre méthode de prédiction de liens, utilisant les CNN et la théorie de Dempster-Shafer. La section 5 présente des résultats expérimentaux positifs sur le benchmark FB15k-237 ainsi que sur la base Mondial. Enfin, la section 6 conclut et ouvre des pistes de travaux futurs.

2 État de l'art

Nickel et al. (2016) ont récemment publié une synthèse sur l'apprentissage automatique pour les graphes de connaissances, où la prédiction de liens est la tâche d'inférence principale. Ils identifient deux types d'approches qui diffèrent par le type de modèle : les modèles à base de *traits latents* et ceux à base de *traits observés*. Les premiers sont de loin les plus étudiés. Avant d'aller plus loin, il est utile de poser le vocabulaire utilisé dans le domaine. Les nœuds sont appelés *entités*, les étiquettes d'arcs sont appelés *relations* et les arcs sont des *triplets* (e_i,r_k,e_j), où e_i est la *tête*, e_j est la *queue* et r_k est la relation de e_i à e_j. Les approches existantes ignorent les valeurs littérales (nombres, dates, chaînes, etc.) ou bien les traitent comme des entités, sans prendre en compte la sémantique de leur domaine.

Les modèles à base de traits latents apprennent des *embeddings* des entités et relations dans des espaces vectoriels de petite dimension, puis font des inférences sur un triplet (e_i,r_k,e_j) en combinant les embeddings des deux entités et de la relation. Les méthodes existantes varient selon la façon d'apprendre ces embeddings et selon la façon des les combiner. Ces méthodes sont fondés sur différentes techniques telles que : factorisation de matrices ou de tenseurs, réseaux de neurones et descente de gradients. Par exemple, une des premières méthodes pour les GC, TransE (Bordes et al., 2013), modélise une relation comme une translation dans l'espace d'embedding des entités, et évalue un triplet candidat d'après la distance entre la tête translatée et la queue. Bordes *et al* ont aussi introduit deux jeux de données, FB15k et WN18 respectivement extraits de Freebase et Wordnet, qui sont devenus des benchmarks de référence pour la prédiction de liens. Toutanova et Chen (2015) ont cependant montré qu'une méthode très simple pouvait battre les méthodes existantes à cause d'un biais dans les jeux de données : de nombreux triplets de test ont leur inverse parmi les triplets d'entraînement. Ils ont extrait

un sous-ensemble plus exigeant de FB15k, appelé FB15k-237, où tous les triplets inverses sont supprimés. Récemment, Schlichtkrull et al. (2018) ont amélioré de façon significative les performances sur FB15k-237 en utilisant des *réseaux convolutifs de graphes* pour apprendre les embeddings.

Les modèles à base de traits observés font leurs inférences directement à partir des arcs du GC. L'inférence par marche aléatoire (Lao et al., 2011) utilise des séquences de relations comme traits et définit la valeur de chaque trait par des marches aléatoires dans le GC. Des poids sur les traits sont appris par régression logistique pour chaque relation cible, et ensuite utilisés pour évaluer les triplets candidats. La méthode a démontré une amélioration par rapport à la génération de règles avec la PLI (Programmation Logique Inductive, Muggleton (1995)). AMIE (Galárraga et al., 2015) parvient à générer de telles règles de façon plus efficace en taillant des algorithmes de PLI sur mesure pour les GC. Ils introduisent aussi une nouvelle mesure de règles qui améliore la précision des inférences dans l'hypothèse du monde ouvert en vigueur dans les GC. Ces deux méthodes ont l'avantage de produire des explications intelligibles pour les inférences, contrairement aux traits latents. Cependant, elles requièrent une phase d'entraînement distincte pour chacune des centaines de relations cibles, tandis que les traits latents sont généralement appris conjointement en une seule phase.

Une différence clé de notre approche est qu'il n'y a pas de phase d'entraînement et que tout l'apprentissage est fait lors de l'inférence. Il s'agit donc d'une approche à base d'instances (ou de raisonnement par cas) plutôt qu'une approche à base de modèles. Étant donné un triplet incomplet $(e_i,r_k,?)$ nous calculons des Concepts de plus proches voisins (CNN) à partir des traits observés de la tête e_i, où les CNN ont une représentation équivalente aux corps des règles d'AMIE. À partir de là, nous inférons un classement d'entités candidates pour la queue de la relation r_k. En fait, comme r_k n'est pas utilisé dans le calcul des CNN, plusieurs relations cibles peuvent être inférées pour presque le même coût qu'une seule relation. En effet le coût principal réside dans le calcul des CNN, lequel est facilement contrôlé car l'algorithme est *any-time*.

3 Concepts de plus proches voisins (CNN)

Dans cette section, nous rappelons brièvement les définitions relatives aux CNN, ainsi que les aspects algorithmiques et pratiques du calcul de leur approximation dans un temps donné. Plus de détails sont disponibles dans des publications précédentes (Ferré, 2017, 2018).

3.1 Définitions théoriques

Un *graphe de connaissance* (CG) est défini par une structure $K = \langle E,R,T \rangle$, où E est l'ensemble des nœuds, aussi appelés *entités*, R est l'ensemble des étiquettes d'arcs, aussi appelées *relations*, et $T \subseteq E \times R \times E$ est l'ensemble des arcs orientés et étiquetés, aussi appelés *triplets*. Chaque triplet (e_i,r_k,e_j) représente le fait que la relation r_k relie l'entité e_i à l'entité e_j. Un exemple de triplet est `(France, capitale, Paris)`. Cette définition est très proche des graphes RDF, où les entités sont des URI ou des littéraux (ou des *blank nodes*) et les relations sont des URI appelées propriétés. Elle est aussi équivalente aux ensembles de faits logiques où les entités sont des constantes et les relations des prédicats binaires.

Les *requêtes* basés sur des *motifs de graphe* jouent un rôle central dans notre approche car elles sont utilisées pour caractériser les CNN, et peuvent être utilisées comme explications des inférences. Un *motif de triplet* $(x,r,y) \in V \times R \times V$ est un triplet avec des variables (prises dans V) à la place des entités. Un *filtre* exprime une condition booléenne sur les variables. On ne considère ici que

les égalités entre une variable et une entité : $x = e$. Les *éléments de requêtes* sont l'ensemble des motifs de triplet et des filtres qui peuvent être composés à partir des entités, relations et variables. Un *motif de graphe* P est un ensemble d'éléments de requêtes. Les égalités sont équivalentes à autoriser des entités dans les motifs de triplets mais ont deux avantages : (1) simplifier le traitement de motifs de triplet qui n'ont ainsi qu'une seule forme ; (2) ouvrir des perspectives pour des filtres plus riches (ex., intervales de valeurs : $x \in [a,b]$). Une *requête* $Q = (x_1,...,x_n) \leftarrow P$ est la projection d'un motif de graphe sur un sous-ensemble de ses variables. De telles requêtes ont une forme concrète en SPARQL avec la syntaxe `SELECT ?x1...?xn WHERE { graph pattern }`. Les requêtes peuvent être vues comme des règles anonymes, c-à-d. des règles similaires à celles d'AMIE (Galárraga et al., 2015) mais sans la relation dans la tête de la règle. Par exemple, la requête $(x,y) \leftarrow (x,parent,z),(z,sibling,y),(y,sex,w),w = female$ sélectionne tous les couples personne-tante, c-à-d. les paires (x,y) où y est une sœur d'un parent de x.

Nous définissons maintenant les *réponses* à une requête. Un *matching* d'un motif P sur le GC $K = \langle E,R,T \rangle$ est une fonction μ des variables de P vers des entités de E telle que $\mu(t) \in T$ pour tout motif de triplet $t \in P$ et $\mu(f)$ s'évalue à vrai pour tout filtre $f \in P$, où $\mu(t)$ et $\mu(f)$ sont obtenus à partir de t et f en remplaçant chaque variable x par $\mu(x)$. Les *réponses* $ans(Q,K)$ d'une requête $Q = (x_1,...,x_n) \leftarrow P$ est l'ensemble des n-uplets $(\mu(x_1),...,\mu(x_n))$ pour tout matching μ de P dans K. On notera que plusieurs matchings peuvent produire la même réponse et que les doublons sont ignorés. Dans la suite, nous considérons des requêtes avec une seule variable projetée, dont les réponses sont assimilées à un ensemble d'entités.

Dans Graph-FCA (Ferré, 2015), un *concept de graphe* est defini comme une paire $C = (A,Q)$ telle que $A = ans(Q)$ et Q est la requête la plus spécifique qui vérifie $A = ans(Q)$. Cette requête la plus spécifique Q peut être calculée à partir de A avec le produit catégoriel de graphes (voir intersection PGP dans (Ferré, 2015)), ou avec l'anti-unification de Plotkin (1971). A est appelé l'*extension* du concept et Q l'*intension* du concept. Un concept $C_1 = (A_1,Q_1)$ est plus spécifique qu'un concept $C_2 = (A_2,Q_2)$ si $A_1 \subseteq A_2$.

Dans un précédent travail (Ferré, 2017), nous avons utilisé les concepts de graphe pour définir la *distance conceptuelle* entre deux entités e_1 et e_2 comme le plus petit concept $\delta(e_1,e_2) = (A,Q)$ dont l'extension A contient à la fois e_1 et e_2, et où l'intension $Q = x \leftarrow P$ est une requête qui caractérise ce que les deux entités ont en commun. Cela est bien défini car les concepts sont organisés en treillis (comme prouvé dans (Ferré, 2015)). Les "valeurs de distances" ont donc une représentation symbolique via l'intension de concept Q. Les distances conceptuelles sont organisées en ordre partiel plutôt qu'en ordre total, contrairement aux mesures de distance classiques. Une distance numérique $dist(e_1,e_2) = |ext(\delta(e_1,e_2))|$ peut être dérivée de la taille de l'extension parce que plus e_1 et e_2 sont proches, plus leur distance conceptuelle est spécifique et plus l'extension est petite. De façon duale, une similarité numérique $sim(e_1,e_2) = |int(\delta(e_1,e_2))|$ peut être dérivée de la taille de l'intension (nombre d'éléments de requête) parce que plus e_1 et e_2 sont similaires, plus leur distance conceptuelle est spécifique et plus l'intension est grande.

Partant d'une entité e dont on veut trouver des entités similaires, les distances conceptuelles entre e et chacune des autres entités induisent une partition de l'ensemble d'entités E, où deux entités e_1,e_2 appartiennent au même cluster si elles partagent la même distance conceptuelle, c-à-d. $\delta(e,e_1) = \delta(e,e_2)$. Chaque cluster S_l est représenté symboliquement par la distance conceptuelle partagée δ_l. S_l est un sous-ensemble de l'extension de δ_l et est appelé l'*extension propre* de δ_l : $S_l = proper(\delta_l) \subseteq ext(\delta_l)$. Chaque paire (S_l,δ_l) est appelé un *Concept de plus proches voisins (CNN, Concept of Nearest Neighbours)* et nous notons $CNN(e,K)$ leur collection pour un graphe K.

Discussion. Comme $CNN(e,K)$ est une partition de l'ensemble des entités, le nombre de CNN ne peut qu'être inférieur au nombre d'entités, et en pratique il est très inférieur. C'est intéressant parce qu'en comparaison le nombre de concepts de graphe est exponentiel dans le nombre d'entités dans le pire cas. L'espace de recherche des approches à base de PLI est l'ensemble des règles, lequel est encore plus large que l'ensemble des concepts de graphe. Calculer les CNN pour une entité donnée est donc une tâche nettement plus abordable que la fouille de règles, bien que l'espace de représentations soit le même. La question que nous commençons à étudier dans ce papier est de savoir si ces CNN sont utiles pour l'inférence et comment ils se comparent avec les autres approches.

Comparé aux mesures numériques utilisées dans les approches de plus proches voisins, les CNN définissent un ordre plus subtil sur les entités. Tout d'abord, parce que les distances conceptuelles sont seulement partiellement ordonnées, il se peut que parmi deux entités aucune ne soit plus similaire à e que l'autre. Cela reflète le fait qu'il peut y avoir plusieurs façons d'être similaire à quelque chose, sans qu'une soit nécessairement préférée à l'autre. Par exemple, laquelle est la plus similaire à une "grande vieille maison"? une "petite vieille maison" ou une "grande maison neuve"? Ensuite, il se peut que deux entités soit à exactement la même distance et donc soit indiscernables en terme de similarité. Enfin, l'intension de concept fournit une explication intelligible de la similarité avec e.

3.2 Aspects algorithmiques et pratiques

Nous esquissons ici les aspects algorithmiques et pratiques du calcul de $CNN(e,K)$. Plus de détails sont disponibles dans (Ferré, 2018). Le principe de l'algorithme est de raffiner de façon itérative une partition de l'ensemble des entités, convergeant vers la partition induite par les extensions propres des CNN. Chaque cluster S_l est associé à une requête $Q_l = x \leftarrow P_l$ et à une ensemble d'éléments de requête candidats H_l. La correspondance avec les CNN est que lorsque H_l est vide, (S_l,δ_l) où $\delta_l = (ans(Q_l),Q_l)$ est un CNN, c-à-d. S_l est l'extension propre d'un CNN dont la distance conceptuelle a pour intension Q_l. Tant que H_l n'est pas vide, S_l est susceptible d'être l'union de plusieurs extensions propres (manque de discernement) et Q_l n'est pas nécessairement la requête la plus spécifique qui matche toutes les entités de S_l (manque de précision dans la similarité conceptuelle). Dans ce cas, on obtient une surestimation des distances conceptuelles pour certaines entités de S_l.

Initialement, il y a un seul cluster $S_0 = E$ où P_0 est le motif vide et H_0 est la *description* de e. La description d'une entité e est le motif de graphe obtenu en extrayant le sous-graphe autour de e et, pour toute entité e_i dans le sous-graphe, en remplaçant e_i par une variable y_i et en ajoutant le filtre $y_i = e_i$. Ici, nous choisissons d'extraire le sous-graphe qui contient tous les arcs partant de e jusqu'à une certaine profondeur.

Ensuite à chaque itération, un cluster S de motif P et d'ensemble d'éléments candidats H est partagé en deux clusters S_1,S_0 en utilisant un élément $h \in H$ comme trait discriminant. L'élément h doit être choisi de telle sorte que $P \cup \{h\}$ forme un motif connecté et contenant la variable x. Actuellement, cet élément est choisi de façon à faire un compromis entre exploration en profondeur et en largeur de la description de e, mais d'autres stratégies sont possibles. Les nouveaux clusters sont définis comme suit :

$$\begin{array}{lll} P_1 = P \cup \{h\} & S_1 = S \cap ans(Q_1 = x \leftarrow P_1,K) & H_1 = H \setminus \{h\} \\ P_0 = P & S_0 = S \setminus S_1 & H_0 = H \setminus \{h\} \end{array}$$

Les équations pour S_1,S_0 assurent qu'on a bien une partition après chaque partage. Les éventuels clusters vides ($S_i = \emptyset$) sont ôtés de la partition. En conséquence, bien que l'espace de recherche

soit l'ensemble des sous-graphes de la description de e, qui a une taille exponentielle dans la taille de la description, le nombre de clusters reste toujours inférieur au nombre d'entités.

Discussion. L'algorithme ci-dessus termine car l'ensemble H décroit à chaque partage. Cependant, dans le cas de grandes descriptions ou de grands GC, le temps de calcul peut être trop long pour une utilisation au moment de l'inférence. Nous pouvons facilement contrôler ce temps d'exécution avec un *timeout* car l'algorithme est *any-time*. En effet, il peut produire à tout moment une partition des entités, avec une surestimation de la distance conceptuelle pour chaque cluster. Des expériences passées (Ferré, 2018) ont montré que l'algorithme a la bonne propriété de produire plus de la moitié des concepts dans une petite fraction du temps total de calcul.

En fait, l'algorithme converge vers une approximation des CNN, dans le sens où la distance conceptuelle peut encore être une surestimation pour certaines entités après termination. La raison vient de ce que les motifs de graphe sont contraints à être des sous-ensembles de la description de e. Afin d'avoir des résultats exacts, il faudrait autoriser la duplication des variables et de leurs arcs adjacents, ce qui augmenterait considérablement l'espace de recherche.

Des expériences sur des GC ayant jusqu'à un million de triplets ont montré que l'algorithme peut calculer tous les clusters pour des descriptions de centaines d'arcs en quelques secondes ou minutes. En comparaison, dans le même délai, les approches par relâchement de requêtes ne parviennent pas à faire plus de 3 relâchements, ce qui est largement insuffisant pour identifier des entités similaires dans la plupart des cas ; et les approches calculant les similarités symboliques avec chaque entité ne passent pas à l'échelle des GC qui ont des dizaines de milliers de nœuds.

4 Prédiction de liens

Le problème de la *prédiction de liens* est d'inférer une entité manquante dans un triplet (e_i,r_k,e_j), c-à-d. soit d'inférer la queue à partir de la tête et de la relation, soit d'inférer la tête à partir de la queue et de la relation. Comme les deux problèmes sont symétriques, nous décrivons ici uniquement l'inférence de la queue. Dans la suite, nous considérons donc e_i et r_k comme étant fixés (ils n'apparaissent pas dans les indices) et e_j comme étant variable. Notre approche de la prédiction de liens s'inspire du travail de Denœux (1995) que nous adaptons aux concepts de plus proches voisins (CNN). Decœux définit une règle de classification à partir des k-plus proches voisins en se basant sur la théorie de Dempster-Shafer (D-S). Chaque k-plus proche voisin x_l d'une instance à classer x est utilisé comme support de l'appartenance de x à la classe c_l de x_l. Le degré de support est fonction de la distance entre x et x_l, de telle sorte que le choix de k est bien moins sensible. La théorie D-S permet de fusionner les k supports en un support global, et ainsi de définir une mesure de *croyance* pour chaque classe.

Nous adaptons le travail de Denœux à l'inférence de e_j dans un triplet (e_i, r_k, e_j) comme suit. Étant donnée une partition des entités $\{(S_l,Q_l)$ avec $Q_l = x \leftarrow P_l\}_l$, comme approximation de $CNN(e_i,K)$, chaque cluster (S_l,Q_l) est utilisé comme support à l'inférence de la queue e_j relativement à la relation r_k. Le degré de support dépend de la *distance extensionnelle* d_l entre e_i et les entités de S_l, définie par

$$d_l = |ans(Q_l)|,$$

c-à-d. le nombre de réponses de la requête Q_l, et de la *confiance* $\phi_{l,j}$ de la règles d'association $(x,r_k,e_j) \leftarrow P_l$, qui est définie par

$$\phi_{l,j} = \frac{|ans(x \leftarrow P_l \cup \{(x,r_k,e_j)\})|}{|ans(Q_l)|},$$

c-à-d. la proportion des entités réponses de Q_l qui ont un r_k-lien vers l'entité e_j. Puisque dans un GC une entité-tête peut être liée à plusieurs entités-queues via la même relation, nous considérons un problème de classification distinct pour chaque entité-queue candidate $e_j \in E$, avec deux classes : c_j^1 (e_j est une entité-queue) et c_j^0 (e_j n'est pas une entité-queue). Pour chaque cluster S_l et chaque entité candidate $e_j \in E$, le degré de support peut ainsi être formalisée en définissant une *affectation de croyance de base* $m_{l,j}$ sur les ensembles de classes.

$$m_{l,j}(\{c_j^1\}) = \alpha_0 \phi_{l,j} e^{-d_l} \qquad m_{l,j}(\{c_j^0\}) = 0 \qquad m_{l,j}(\{c_j^1, c_j^0\}) = 1 - \alpha_0 \phi_{l,j} e^{-d_l}$$

$m_{l,j}(\{c_j^1\})$ représente le degré de croyance que e_j soit une entité-queue, tandis que $m_{l,j}(\{c_j^1, c_j^0\})$ représente le degré d'incertitude. $m_{l,j}(\{c_j^0\})$ est mis à 0 pour refléter l'hypothèse du monde ouvert des GC selon laquelle un fait manquant n'est pas considéré comme faux. La constante α_0 détermine le degré maximum de croyance, qui peut être inférieur à 1 pour refléter l'incertitude concernant la véracité des triplets existants. Le degré de croyance décroit de façon exponentielle avec la distance. Enfin, nous rendons ce degré proportionnel à la confiance de l'inférence de e_j par Q_l. Dans Denœux (1995), ce facteur confiance n'existe pas car il vaudrait 1 pour la classe du plus proche voisin et 0 pour toute autre classe.

En appliquant la thérie D-S pour fusionner les supports venant de tous les clusters de notre partition $\{(S_l, Q_l)\}_l$, nous arrivons à l'équation suivante pour la croyance en chacune des entités candidates e_j.

$$Bel_j = Bel_j(\{c_j^1\}) = 1 - \Pi_l(1 - m_{l,j}(\{c_j^1\}))$$

À partir de la croyance en chaque entité e_j, il est possible de classer les entités par croyance décroissante, ou bien de sélectionner un sous-ensemble d'entités en choisissant un seuil de croyance minimale. Ensuite, les classements d'entités peuvent être évalués par des mesures telles que Hits@N (la proportion des tâches d'inférences où l'entité correcte est parmi les N premières entités) et MRR (Mean Reciprocal Rank, la moyenne des inverses du rang de l'entité correcte). Les sélections d'entités peuvent être évaluées avec des mesures telles que précision, rappel et score F1.

On notera que la méthode ci-dessus se généralise facilement à l'inférence conjointe de la relation r_k et de l'entité-queue e_j. Il suffit d'utiliser les indices k,j partout où l'index j est utilisé : $\phi_{l,k,j}$ serait la confiance dans l'inférence de la relation r_k et de l'entité e_j par Q_l, $c_{k,j}^1$ serait la classe des entités liées à e_j via r_k et $Bel_{k,j}$ serait la croyance en l'inférence d'un tel lien.

5 Évaluation

5.1 Méthodologie

La page "expériences" [1] fournit des liens vers le code source, les jeux de données et les logs.

Données. Nous utilisons le jeu de données FB15k-237 introduit par Toutanova et Chen (2015) comme sous-ensemble exigeant du jeu de données FB15k, introduit par Bordes et al. (2013) pour l'évaluation en prédiction de liens. C'est un ensemble de triplets extraits de Freebase. Les triplets utilisent 237 relations et sont partagés en trois parties : *train* (272 115 triplets), *validation* (17 535) et *test* (20 466). FB15k-237 est plus exigeant que FB15k car les relations qui sont équivalentes à une autre relation ou à son inverse sont ôtées et parce que les paires d'entités de la partie *train* sont ôtées

1. http://www.irisa.fr/LIS/ferre/pub/link_prediction/index.html

des autres parties pour éviter d'éventuelles inférences triviales. Le jeu de données contient aussi des mentions textuelles mais nous les ignorons car nous ne considérons ici que les GC.

En complément, nous utilisons aussi le jeu de données Mondial (May, 1999) qui contient des informations géographiques mondiales. Comme la tâche considérée est d'inférer des relations entre entités, nous l'avons simplifié en enlevant les triplets contenant des nombres ou des dates, en enlevant les triplets de nommage des entités et en déréifiant les relations n-aires. Les triplets restant utilisent 20 relations et sont partagés en trois parties : *train* (7979 triplets), *validation* (778) et *test* (970).

Tâche. Nous suivons le même protocole qu'introduit par Bordes et al. (2013) et suivi par les travaux ultérieurs. La tâche consiste à inférer, pour chaque triplet test, l'entité-queue à partir de la tête et de la relation et aussi l'entité-tête à partir de la queue et de la relation. Nous appelons *entité test* l'entité connue et *entité manquante* l'entité à inférer. Nous évaluons la performance de notre approche en utilisant les mesures : MRR (Mean Reciprocal Rank), H@{1,3,10} (Hits@{1,3,10}). Comme dans les travaux existants nous utilisons les versions filtrées de ces mesures pour prendre en compte le fait qu'il peut y avoir plusieurs entités valides dans un classement d'entités. Par exemple, si l'entité correcte est au rang 7 mais que 2 des 6 premières entités forment des triplets valides (c-à-d. qui appartiennent au jeu de données), alors on considère que l'entité correcte est au rang 5.

Méthode. Comme notre approche n'a pas de phase d'entraînement, nous pouvons utiliser les deux parties *train* et *validation* comme exemples pour notre inférence. Notre approche a seulement deux paramètres pour le calcul des CNN : la profondeur de la description de l'entité test et le temps de calcul alloué (*timeout*). Nous étudions la sensibilité à ces deux paramètres. Pour l'inférence d'un classement d'entités, nous posons $\alpha_0 = 0.95$ et utilisons tous les CNN calculés (pas de sélection des k-plus proches CNN). L'implémentation de notre approche a été intégrée à SEWELIS comme amélioration d'une précédente contribution sur l'édition guidée de graphes RDF (Hermann et al., 2012). Nous avons exécuté nos expériences sur une machine Fedora 25, avec un CPU Intel(R) Core(TM) i7-6600U @ 2.60GHz et une mémoire 16GB DDR4.

Baselines. Nous avons choisi la même tâche et les mêmes mesures que Schlichtkrull et al. (2018) de façon à pouvoir récupérer les résultats de plusieurs approches existantes à base de traits latents (LinkFeat, DistMult, CP, TransE, HolE, ComplEx, R-GCN), et les comparer avec notre approche à base de CNN. Nous nous comparons également à une approche à base de règles, AMIE, que nous avons exécuté avec ses paramètres par défaut. Comme suggéré par les auteurs d'AMIE (équation 8, Galárraga et al. (2015)), nous classons les entités e_j en agrégeant les confiances $\phi_{l,j}$ des règles R_l permettant d'inférer le triplet (e_i, r_k, e_j) : $\phi_j = 1 - \Pi_l(1 - \phi_{l,j})$. Nous ajoutons également une méthode naïve, Freq, qui consiste simplement à classer les entités e_j selon leur fréquence décroissante d'usage dans les triplets de relation r_k, que l'on définit par $freq_j = |ans(x \leftarrow (x, r_k, y), y = e_j)|$. Le classement obtenu est indépendant de l'entité test e_i et agit donc comme un classement par défaut.

5.2 Résultats et discussion

Le tableau 1 compare les résultats de notre approche (CNN) aux autres approches. CNN est exécuté avec des timeouts compatibles avec une interaction utilisateur (0.01s,0.1s,1s) et une profondeur de description égale à 10, qui garantit de capturer à peu près tous les traits pertinents de l'entité test. CNN surpasse les performances des autres approches sur FB15k-237 dès 0.01s pour les mesures les plus fines (MRR, H@1, H@3) et dès 1s pour la mesure H@10. CNN-1s atteint un MRR de 0.286 comparé à 0.248 pour la meilleure autre approche, R-CGN, soit une marge de 3,8%. Cette marge est encore plus grande pour les mesures H@1 (6,2%) et H@3 (4,8%). Le choix systématique de la première entité dans les classements produits serait précis à 21,5% au lieu de 15,3% pour

	FB15k-237				Mondial			
Approche	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
Freq	*0.236*	*0.175*	*0.253*	*0.356*	*0.142*	*0.069*	*0.159*	*0.309*
TransE	0.233	0.147	0.263	0.398	-	-	-	-
DistMult	0.191	0.106	0.207	0.376	-	-	-	-
HolE	0.222	0.133	0.253	0.391	-	-	-	-
ComplEx	0.201	0.112	0.213	0.388	-	-	-	-
R-GCN	0.248	0.153	0.258	**0.414**	-	-	-	-
AMIE	0.143	0.096	0.155	0.241	0.179	0.127	0.208	0.281
CNN 0.01s	**0.250**	**0.186**	**0.268**	0.377	0.313	0.254	0.349	0.424
CNN 0.1s	**0.264**	**0.198**	**0.284**	0.395	0.327	0.271	0.355	0.433
CNN 1s	**0.286**	**0.215**	**0.311**	**0.428**	0.320	0.267	0.344	0.431

TAB. 1 – *Résultats sur FB15k-237 et Mondial pour la baseline Freq, plusieurs approches à traits latents (TransE, DistMult, HolE, ComplEx, R-GCN), une approche à règles (AMIE), et notre approche (CNN) avec trois timeouts (0.01s, 0.1s, 1s). Les résultats pour les approches latentes viennent de Schlichtkrull et al. (2018).*

	FB15k-237 (timeout=1s)				Mondial (timeout=0.1s)			
prédictions	MRR	H@1	H@3	H@10	MRR	H@1	H@3	H@10
toutes	0.286	0.215	0.311	0.428	0.327	0.271	0.355	0.433
queues	0.391	0.307	0.428	0.553	0.584	0.503	0.636	0.734
têtes	0.182	0.123	0.194	0.303	0.057	0.027	0.060	0.117

TAB. 2 – *Résultats de CNN (profondeur=10, timeout=1s/0.1s) pour toutes les prédictions, pour les prédictions des queues et pour les prédictions des têtes.*

R-CGN. Ces résultats semblent confirmés sur Mondial, mais nous n'avons malheureusement pas pu obtenir les résultats pour les approches latentes. CNN surpasse AMIE d'une marge allant de 13,4% à 14.8% pour la mesure MRR. On observe également que CNN surpasse Freq sur toutes les mesures et pour tous les timeouts, ce qui implique que les concept de voisins apprennent quelque chose d'utile au-delà de simple statistiques globales. On notera que ce n'est pas le cas des autres approches, en particulier pour les mesures à grain fin (MRR et H@1). Dans FB15k-237, aucune autre approche n'améliore H@1 par rapport à Freq.

Nous étudions maintenant l'impact de la profondeur de description sur les résultats. On s'attend à priori à ce qu'une plus grande profondeur fournisse plus d'information mais soit plus coûteuse en calcul. Cependant, en variant la profondeur de 1 à 20, nous observons de très faibles écarts-types sur les quatre mesures, tous entre 0.003 et 0.007. Cela montre qu'une grande part de l'information utile se trouve déjà à la profondeur 1. De plus, c'est une bonne propriété que l'accroissement de la profondeur ne dététiore pas les résultats car cela signifie que cette profondeur peut être fixée à une valeur élevée sans problème. L'explication de cette propriété est que l'algorithme de partitionnement itératif commence avec les triplets peu profonds et se poursuit avec des triplets de profondeur croissante.

Le tableau 2 détaille les résultats de notre approche, en distingant la prédiction des queues et la prédictions des têtes. Il montre clairement qu'il est nettement plus facile de prédire les queues que les têtes. Ce n'est pas surprenant sachant que dans les GC, les relations sont généralement orientées

relation	#têtes	#queues	MRR_{Freq}	MRR	H@1	H@3	H@10
profession	4245	150	*0.434*	0.601	0.455	0.694	0.874
gender	4094	2	*0.882*	0.899	0.798	1.000	1.000
nationality	4068	100	*0.720*	0.772	0.662	0.866	0.941
award	3386	406	*0.080*	0.270	0.154	0.296	0.511
type_of_union	3033	4	*0.971*	0.971	0.942	1.000	1.000
place_of_birth	2613	704	*0.155*	0.183	0.100	0.235	0.359
place_lived	2519	804	*0.172*	0.194	0.108	0.239	0.344
film/genre	1875	123	*0.315*	0.380	0.226	0.429	0.711
film/language	1735	59	*0.744*	0.759	0.688	0.790	0.911
film/country	1708	61	*0.685*	0.701	0.573	0.809	0.931

TAB. 3 – *Résultats des prédictions des queues pour certaines des relations les plus fréquentes dans FB15k-237. #têtes (resp. #queues) est le nombre de têtes uniques (resp. queues uniques) pour cette relation. Le MRR de Freq est aussi inclus pour comparaison.*

dans le sens le plus déterministe. Par exemple, la relation entre films et genres est orientée des films vers les genres parce que chaque film a seulement un ou quelques genres alors que pour chaque genre il y a de nombreux films.

Le tableau 3 détaille davantage les résultats pour une sélection de 10 des relations les plus fréquentes du jeu de données FB15k-237, en considérant seulement la prédiction des queues. Afin de donner une idée de la difficulté de ces prédictions pour chaque relation, nous donnons le nombre de têtes et queues uniques dans les parties *train+valid*, ainsi que le MRR de la baseline *Freq*. Les résultats montrent que le MRR augmente de façon significative avec notre approche pour toutes les relations, sauf *type_of_union* dont le MRR de la baseline est déjà très élevé (0.971). Par exemple, le MRR de *profession* augmente de 16,7% et celui de *award* augmente de 19%. Pour la moitié des relations Hits@1 est supérieur à 0.5, ce qui signifie que choisir la première entité du classement serait correct dans plus de 50% des cas. Cela inclut des relations avec un grand nombre de queues uniques et donc potentiellement difficiles à prédire, par exemple la relation *nationality* avec 100 queues uniques et H@1 = 0.668.

5.3 Exemples d'inférences et explications

Pour compléter cette évaluation, nous illustrons notre méthode d'inférence en inspectant quelques exemples en détail. Dans FB15k-237, la langue du film "Dragon Ball Z : Bojack Unbound" est correctement prédite comme étant "Japanese" avec MRR=1, comparé à MRR=0.2 pour Freq. CNN-1s génère 26 concepts, parmi lesquels la meilleure explication (en terme de croyance) est que le film est produit au Japon et a Toshiyuki Morikawa comme acteur. Le lieu de vie de "Tabu" est correctement prédit comme étant "Mumbai" avec MRR=1, comparé à MRR=0.059 pour Freq. CNN-1s génère 32 concepts, parmi lesquels la meilleure explication est que "Tabu" a reçu le prix "Filmfare Award for Best Actress", ce qui indique que un certain nombre de personnes ayant reçu ce prix vivent à Mumbai. Les prédictions suivantes dans le classement sont d'autres villes en Inde. Dans Mondial, le continent de la Suède est correctement prédit comme étant l'Europe car c'est un pays de type monarchie constitutionnelle (requête à 3 éléments), comme le Danemark. Matterhorn est correctement prédit comme étant localisé en Suisse car c'est une montagne des Alpes qui est aussi localisée en Italie

(requête à 5 éléments), comme le Monte Rosa. Lagen est correctement prédit comme étant localisé en Norvège car c'est l'estuaire d'une rivière localisée en Norvège (requête à 4 éléments).

De façon plus systématique, nous avons examiné toutes les inférences correctes et non-triviales de la nationalité des personnes dans FB15k-237, c-à-d. les inférences où l'entité correcte n'est pas une des trois nationalités les plus fréquentes ($\text{MRR}_{Freq} < 0.333$). Parmi ces 10 inférences, le nombre de concepts générés par CNN-1s est remarquablement stable et petit, entre 22 et 37. Les intensions de ces concepts (les explications) sont des requêtes à 2 ou 3 éléments, et les extensions ont 2 à 29 entités (les entités similaires servant d'exemples). Les meilleures explications disent que la nationalité des personnes peut être inféré soit par le lieu d'habitation (ex., Tokyo $\rightarrow$ Japan), le lieu du décès, la langue parlée, un film dans lequel la personne a joué ou un prix gagné.

Notre approche à base d'instances est capable de trouver des explications très spécifiques, comme illustré ci-dessus, que des approches à base de règles auraientt peu de chance de produire vu leur nombre gigantesque sur l'ensemble du jeu de données. Cependant, une limite de notre méthode d'inférence est qu'elle ne fournit pas encore d'explications généralisées telles que "si une personne X habite dans une ville située dans un pays Y, alors X a pour nationalité Y", alors que c'est le principal type d'explications sur lesquelles s'appuient les approches à base de règles ou de marches aléatoires.

6 Conclusion

Nous avons montré des résultats expérimentaux positifs et encourageants pour une approche symbolique au problème de la prédiction de liens dans les graphes de connaissances, en particulier pour les mesures à grain fin (MRR, Hits@1). Comparé aux approches à traits latents, nous sommes capable de fournir des explications intelligibles à chaque inférence sous forme de motifs de graphe. Comparé aux approches à base de règles, nous avons un meilleur contrôle du temps de calcul, et donc une meilleure capacité à passer à l'échelle, en évitant la phase d'apprentissage (fouille des règles) et en utilisant un algorithme *any-time* contrôlé par un timeout à l'inférence. Notre approche est analogue à la classification par les k-plus proches voisins mais nos distances sont définis par des motifs de graphe plutôt que par des métriques.

De nombreuses pistes de recherche restent ouvertes. Étendre les motifs de graphe avec des filtres plus riches sur les nombres, dates, etc. Optimiser le calcul des CNN en trouvant de bonnes stratégies pour guider le processus de partitionnement ou en le parallélisant. Étendre la procédure d'inférence pour imiter les règles non-instanciées d'AMIE (ex., *la nationalité est le pays du lieu d'habitation*). Évaluer notre approche sur d'autres jeux de données et d'autres tâches d'inférence.

Références

Bordes, A., N. Usunier, A. Garcia-Duran, J. Weston, et O. Yakhnenko (2013). Translating embeddings for modeling multi-relational data. In *Advances in neural information processing systems*, pp. 2787–2795.

Denœux, T. (1995). A k-nearest neighbor classification rule based on Dempster-Shafer theory. *IEEE Trans. Systems, Man, and Cybernetics 25*(5), 804–813.

Ferré, S. (2015). A proposal for extending formal concept analysis to knowledge graphs. In J. Baixeries, C. Sacarea, et M. Ojeda-Aciego (Eds.), *Int. Conf. Formal Concept Analysis (ICFCA)*, LNCS 9113, pp. 271–286. Springer.

Ferré, S. (2017). Concepts de plus proches voisins dans des graphes de connaissances. In *28es Journées francophones d'Ingénierie des Connaissances (IC)*, pp. 163–174.

Ferré, S. (2018). Answers partitioning and lazy joins for efficient query relaxation and application to similarity search. In A. Gangemi et al. (Eds.), *Int. Conf. The Semantic Web (ESWC)*, LNCS 10843, pp. 209–224. Springer.

Galárraga, L., C. Teflioudi, K. Hose, et F. Suchanek (2015). Fast rule mining in ontological knowledge bases with amie+. *Int. Journal Very Large Data Bases 24*(6), 707–730.

Hermann, A., S. Ferré, et M. Ducassé (2012). An interactive guidance process supporting consistent updates of RDFS graphs. In A. ten Teije et al. (Eds.), *Int. Conf. Knowledge Engineering and Knowledge Management (EKAW)*, LNAI 7603, pp. 185–199. Springer.

Lao, N., T. Mitchell, et W. W. Cohen (2011). Random walk inference and learning in a large scale knowledge base. In *Conf. Empirical Methods in Natural Language Processing*, pp. 529–539. Association for Computational Linguistics.

Liben-Nowell, D. et J. Kleinberg (2007). The link-prediction problem for social networks. *Journal of the American society for information science and technology 58*(7), 1019–1031.

May, W. (1999). Information extraction and integration with FLORID : The MONDIAL case study. Technical Report 131, Universität Freiburg, Institut für Informatik. Available from `http://dbis.informatik.uni-goettingen.de/Mondial`.

Muggleton, S. (1995). Inverse entailment and progol. *New Generation Computation 13*, 245–286.

Nickel, M., K. Murphy, V. Tresp, et E. Gabrilovich (2016). A review of relational machine learning for knowledge graphs. *Proc. IEEE 104*(1), 11–33.

Plotkin, G. (1971). *Automatic Methods of Inductive Inference*. Ph. D. thesis, Edinburgh University.

Schlichtkrull, M., T. N. Kipf, P. Bloem, R. van den Berg, I. Titov, et M. Welling (2018). Modeling relational data with graph convolutional networks. In *The Semantic Web Conf. (ESWC)*, pp. 593–607. Springer.

Toutanova, K. et D. Chen (2015). Observed versus latent features for knowledge base and text inference. In *Work. Continuous Vector Space Models and their Compositionality*, pp. 57–66.

Summary

The open nature of Knowledge Graphs (KG) often implies that they are incomplete. Link prediction consists in infering new links between entities based on existing links. Most existing approaches rely on the learning of latent feature vectors for the encoding of entities and relations. In general however, latent features cannot be easily interpreted. Rule-based approaches offer interpretability but a distinct ruleset must be learned for each relation. We propose a new approach that does not need a training phase, and that can provide interpretable explanations for each inference. It relies on the computation of Concepts of Nearest Neighbours (CNN) to identify similar entities based on common graph patterns. Dempster-Shafer theory is then used to draw inferences from CNNs. We evaluate our approach on FB15k-237, a challenging benchmark for link prediction, where it gets improved performance compared to existing approaches.