Découverte de règles causales dans les graphes de connaissances à l'aide de plongements dans les graphes

L. Simonne[1], N. Pernelle[2], F. Saïs[1], R. Thomopoulos[3]

[1] LISN, CNRS (UMR 9015), Université Paris-Saclay
[2] LIPN, CNRS (UMR 7030), Université Sorbonne Paris Nord
[3] INRAE (UMR IATE)

simonne@lisn.fr

Résumé

La découverte de relations causales est l'objectif de nombreuses expériences. Lorsque des données observationnelles sont disponibles, l'utilisation du cadre d'étude des résultats potentiels est un des standards pour découvrir de telles relations. Dans cet article, nous nous plaçons dans ce cadre afin de découvrir des règles causales au sein de graphes de connaissances (KGs). Ces règles expriment que des différences de traitements conduisent à des différences de valeur pour une caractéristique étudiée. Cependant, ce cadre repose sur la similarité entre instances, et sa quantification dans un KG n'est pas triviale, notamment parce que leurs descriptions peuvent être incomplètes et erronées. Nous proposons une nouvelle méthode de découverte de règles causales qui exploite un appariement basé sur les plongements de graphes de connaissances. Les expérimentations menées sur deux KG de domaines différents ont montré la capacité de notre approche à découvrir des règles qui expliquent plus de différences dans la caractéristique étudiée que les approches existantes, et qui est plus robuste en cas d'information incomplète.

Mots-clés

Règles causales, Plongements de graphes, Explicabilité, Graphes de connaissances

Abstract

Discovering causal relationships between different observations is the goal of many experiments in science. When observational data are available, the potential outcome framework is a well-used framework for discovering such relationships. In this paper, we place ourselves in this framework to discover causal rules in Knowledge Graphs (KGs) that express that differences in treatments lead to differences in a studied characteristic. However, quantifying the similarity between individuals represented in a knowledge graph is challenging, especially because their descriptions can be incomplete and erroneous. We propose a new approach based on knowledge graph embeddings to discover causal rules in KGs. The experiments that we conducted on two KGs, including a scientific knowledge graph, showed that our approach is able to discover rules that explain much more differences in the studied characteristic than existing state of the art approaches.

Keywords

Causal Rules, Graph Embeddings, Explainability, Knowledge Graphs

1 Introduction

L'adoption des technologies du web sémantique pour la représentation des données et des connaissances est en plein essor. Cela a conduit à la disponibilité de nombreux ensembles de données représentés sous forme de graphes de connaissances [14] où les données sont représentées en RDF et les connaissances du domaine sous forme d'ontologie exprimée en OWL ou RDFS. Ces graphes de connaissances peuvent contenir des données et des connaissances de différents domaines comme DBPedia et Wikidata ou plus spécifiques à un domaine comme MusicBrainz ou Bio2RDF. Ces KG peuvent être exploités pour découvrir de nouvelles connaissances telles que des règles d'associations (i.e. $hasChild(x, y) \wedge citizenOf(x, z) \Rightarrow citizenOf(y, z)$). Ces associations peuvent être utiles pour prédire de nouveaux faits, détecter des erreurs, mais elles expriment rarement des relations causales.

Une *relation causale* décrit la relation entre deux variables, où une variable nommée *traitement* a un *effet* sur une variable nommée *résultat*. Les relations causales sont intéressantes dans de nombreux domaines, comme par exemple en santé pour déterminer si un médicament traite ou non une maladie, ou en politiques publiques pour comprendre si une nouvelle loi a eu un impact attendu ou non. Il existe différents cadres pour étudier le problème de la découverte de relations causales à partir de données d'observation tabulaires, tels que le *modèle causal structurel* [15] ou le *cadre des résultats potentiels* [18]. Dans ce dernier cas, l'effet d'un traitement peut être estimé en comparant des instances similaires qui diffèrent sur le traitement. Bien que de nombreuses approches existent pour découvrir les relations causales dans les données tabulaires, seules quelques approches [11, 20] se concentrent sur la découverte des effets d'un traitement dans les KG. L'approche proposée dans [20], permet de découvrir des règles causales différentielles

qui expriment qu'une différence de valeurs sur le traitement explique une différence de valeurs sur le résultat. Cette approche, fondée sur le cadre des résultats potentiels, s'appuie sur une étape d'appariement tronqué fondé sur le regroupement de certains chemins de propriétés. Les règles découvertes sont très expressives, car elles représentent explicitement le sous-ensemble de classes d'instances pour lesquelles la règle est valide. Un processus de généralisation des règles est effectué, mais il conduit à très peu de règles générales, en raison de fortes contraintes sur la complétude des données. Ainsi, seule une faible partie des différences de résultats peut être expliquée à l'aide des règles.

Dans cet article, nous présentons une nouvelle approche hybride appelée Dicare-E combinant les plongements de graphes et les techniques de fouille de règles symboliques pour découvrir des règles différentielles causales dans les graphes de connaissances. Ce type de règle permet à la fois d'expliquer des différences de valeur sur une propriété étudiée comme la croissance de la taille d'une tumeur, ou le taux de réinsertion des chômeurs, et d'en tirer des conclusions pour effectuer des décisions. Nous adaptons le cadre des résultats potentiels pour pouvoir découvrir de telles règles. Pour apparier des instances, nous exploitons des plongements pré-entraînés et mesurons la similarité de deux vecteurs intégrés en observant la similarité des prédictions utilisant ces vecteurs. Ainsi, la méthode d'appariement est plus robuste aux données erronées et incomplètes. En outre, elle permet d'obtenir des règles plus générales qui peuvent être appliquées à un plus grand nombre d'instances. Nos contributions sont (i) la définition d'une nouvelle méthode d'appariement basée sur les plongements de graphes, (ii) un algorithme s'appuyant sur ces appariements pour découvrir des règles exprimant des effets causaux de traitements et (iii) une évaluation expérimentale sur un jeu de données réel qui montre l'efficacité de l'approche développée en comparaison avec l'état de l'art.

Dans la section 2, nous présentons les travaux antérieurs sur la découverte de la causalité dans les KG et introduisons le problème fondamental de l'inférence causale et la définition de l'appariement d'instances. Ensuite, dans la section 3, nous présentons l'énoncé du problème que nous abordons dans ce travail. La section 4 explique en détail comment nous utilisons les vecteurs issus des plongements pour calculer la similarité entre deux instances. L'algorithme est présenté dans la section 5. Enfin, dans la section 6, nous présentons les expériences et les résultats obtenus sur un graphe de connaissances scientifiques.

2 Travaux antérieurs

Dans cette section, nous présentons les travaux antérieurs qui traitent de la découverte causale dans les données tabulaires et dans les graphes de connaissances, ainsi que les principales méthodes de comparaison d'instances.

Causalité dans les données tabulaires. La découverte de relations causales est étudiée depuis des décennies et de nombreuses approches ont été définies. Certaines approches sont basées sur le modèle causal structurel, introduit par Pearl [15], où les modèles visent à décrire les systèmes avec des modèles graphiques. D'autres utilisent les réseaux bayésiens [12] qui représentent des ensembles de covariables avec des modèles probabilistes graphiques, qui peuvent montrer des liens causaux sous certaines hypothèses. Comme les relations causales peuvent être difficiles à extraire dans des systèmes décrivant de nombreuses variables [2], certains travaux visent à déterminer les structures causales locales.

L'étude de la découverte de relations causales peut également être introduite par le biais du problème fondamental de l'inférence causale [18]. Étant donné $Y_i(T)$ le résultat de l'individu i lors de l'étude du traitement T, l'effet du traitement individuel est défini comme $TE_i = Y_i(1) - Y_i(0)$, c'est-à-dire la différence entre le résultat de i avec le traitement et le résultat de i sans le traitement. Cependant, le problème de l'inférence causale réside dans le fait que TE_i ne peut pas être calculé, car pour un individu donné i, si $Y_i(1)$ est observé, le contrefactuel $Y_i(0)$ ne peut pas être observé. Le standard pour trouver les effets du traitement est de planifier une expérience avec une attribution aléatoire du traitement. De telles expériences sont difficiles à planifier pour des raisons d'éthique et de coût. Il n'est par exemple pas possible de forcer les gens à commencer à fumer. Par conséquent, la plupart des études causales sont menées en exploitant des données observationnelles, où l'attribution du traitement ne se fait pas de manière aléatoire. Dans ces approches, deux ensembles d'individus sont créés : l'ensemble *traité* des individus $(T = 1)$, et l'ensemble *contrôle* des individus non traités $(T = 0)$. Comparer naïvement les résultats des deux ensembles introduit un biais de sélection, car la distribution des covariables n'est pas la même dans les deux groupes. Par exemple, en étudiant comme traitement l'âge et en résultat la probabilité qu'une personne ait un cancer, l'ensemble de contrôle pourrait être composé d'une majorité d'hommes, et l'ensemble traité d'une majorité de femmes, ce qui introduirait un biais.

Le cadre d'étude des *résultats potentiels* estime un effet en trouvant un contrefactuel pour les individus traités [18]. Le contrefactuel peut être estimé ou déterminé parmi les individus du *contrôle* [21]. Il permet d'équilibrer la distribution des covariables dans les deux ensembles, de sorte qu'ils partagent une probabilité égale de traitement, et supprime le biais de sélection. Une technique est l'appariement, qui vise à sous-échantillonner l'ensemble de données pour équilibrer la distribution des covariables. Elle consiste, pour un individu traité, à le comparer à un individu contrôle similaire. Dans l'exemple précédent, l'appariement consiste à comparer des paires de personnes qui ont le même sexe, la même taille et le même poids (et autres), mais qui diffèrent sur l'âge de traitement. Bien que cette technique soit populaire, l'appariement devient de plus en plus difficile avec la complexité de la description des individus [1]. Plus le nombre d'attributs décrivant les individus est élevé, plus il est difficile de trouver des individus qui partagent les mêmes valeurs sur un ensemble d'attributs. Des alternatives ont donc été proposées. L'appariement tronqué consiste à relâcher la contrainte d'appariement exact. Elle facilite non

seulement le processus d'appariement, mais minimise également le biais dans les effets découverts [7]. L'appariement par score de propension [17] est une autre technique d'appariement reconnue qui repose sur l'utilisation d'un modèle de classification. Ce modèle produit pour chaque instance un score représentant la probabilité que l'instance soit traitée, et les instances ayant un score similaire sont appariées.

Dans le cadre des graphes de connaissances, l'estimation du contrefactuel doit être adaptée pour prendre en compte les descriptions RDF des individus qui peuvent être encore plus complexes que les données tabulaires.

Causalité dans les graphes de connaissances. À notre connaissance, seules trois approches se sont concentrées sur l'exploration des relations causales dans les KG. Dans [11], les auteurs proposent de transformer les données du KG en un schéma relationnel en utilisant des connaissances expertes et d'apprendre un modèle relationnel probabiliste. Cependant, les résultats représentent la distribution conjointe entre les variables et n'indiquent pas nécessairement les relations causales. De plus, comme elle repose sur des réseaux bayésiens, l'approche ne peut pas prendre en compte de grands graphes de connaissances et elle a été évaluée sur un petit ensemble de données ($\approx$ 7k triplets). [20] découvre des règles différentielles causales expressives, qui expriment un effet de traitement pour un sous-ensemble d'instances décrites par un motif de graphe nommé strate. De telles règles sont intéressantes car elles peuvent montrer des effets locaux. Cependant, une règle définie sur une strate donnée est sélectionnée si sa validité est vérifiée sur toutes les strates les plus spécifiques. Une telle contrainte stricte empêche de découvrir des règles génériques dans le cas de strates spécifiques avec trop peu d'instances correspondantes. Enfin, [6] exploite les plongements de graphes afin de découvrir de nouvelles hypothèses scientifiques à partir d'hypothèses recueillies sur un ensemble de papiers scientifiques. Bien qu'il ne soit pas fait mention d'un cadre d'étude de la causalité, cette approche repose sur l'utilisation de plongements d'hypothèses et de papiers pour découvrir de nouvelles hypothèses, et peut s'apparenter au cadre d'étude des résultats potentiels.

Appariement d'instances dans les graphes de connaissances. Il existe de nombreuses approches de liaison de données visant à découvrir les liens d'identité représentés par le prédicat *owl :sameAs* dans les graphes de connaissances (voir [13] pour un aperçu). Cependant, nous ne sommes pas intéressés par les liens d'identité puisque notre objectif est de comparer des entités distinctes. Ces entités devraient différer sur le traitement et sur le résultat, tout en étant très similaires sur le reste de la description RDF. Des prédicats moins stricts comme *identiConTo* [16] ont été définis pour exprimer la relation d'identité entre des entités limitées à un contexte conceptuel donné (c'est-à-dire une sous-partie de l'ontologie), mais cette approche ne fournit pas de similarité quantifiée et n'est pas adaptée aux données incomplètes. Dans des approches récentes, [19] fait de l'appariement strict. Cette approche est utilisable sur des graphes ayant des schémas simples mais n'est pas applicable lorsque le schéma est complexe, l'appariement devenant rare voire impossible. [20] propose un appariement tronqué pour notamment traiter l'incomplétude des données en utilisant des propriétés abstraites qui sont dérivées de clusters de propriétés obtenus grâce à leur co-occurrence. Cependant, le processus de clustering doit être guidé par un expert du domaine lorsque le nombre de propriétés est élevé.

La recherche par similarité a également été étudiée pour la relaxation de requêtes [4]. L'utilisation de telles approches dépend cependant de la définition d'un espace de recherche de requêtes, ce qui n'est pas trivial. La similarité entre les instances peut également être quantifiée comme dans [3], où les auteurs proposent une similarité entre les clauses de Horn qui est basée sur des prédicats et des arguments (non-)partagés. Cette approche repose sur des descriptions de logique de premier ordre complètes et n'est donc pas conçue pour traiter des données incomplètes.

Les modèles de plongements présentent des caractéristiques intéressantes pour notre approche. En effet, si ces techniques capturent la sémantique des entités (i) des instances RDF similaires ont des vecteurs similaires dans l'espace de plongements [22] et (ii) les vecteurs similaires dans l'espace de plongements représenteront des instances RDF similaires [10, 8]. Cependant, pour utiliser ces approches afin de déterminer des instances similaires mais non identiques, il est nécessaire de fournir un grand ensemble d'instances similaires et dissemblables, et de fixer un seuil qui peut être utilisé pour décider que deux instances sont suffisamment similaires.

Fouille de règles d'association. La fouille de règles d'association est un élément important de la recherche au sein des bases de connaissances. Ces règles sont composées d'un corps $\vec{B}$ et d'une tête $\vec{H}$, où $\vec{B}$ peut contenir un ou plusieurs atomes et $\vec{H}$ un atome, et expriment le lien $\vec{B} \Rightarrow \vec{H}$. Les méthodes de fouille de règles d'association, telles que [5], sont couramment utilisées afin d'obtenir de nouvelles connaissances ou encore de supprimer des triplets erronés. Bien qu'une association soit représentée dans ces règles, elle n'indique pas nécessairement un lien de causalité. De plus, une règle d'association n'indique pas clairement l'effet d'un traitement, et les algorithmes utilisés pour les déterminer ne prennent pas en compte la similarité des instances, *i.e.* il n'y a pas d'étape de contrôle réalisée. Par exemple, un effet présent dans la tête d'une règle pourrait être du à un traitement non indiqué dans le corps. Ainsi, une telle règle ne peut être utilisée pour expliquer l'effet d'un traitement.

3 Préliminaires et Définitions

Nous cherchons à découvrir des règles causales exprimées en logique du premier ordre sous la forme $\vec{B} \Rightarrow \vec{H}$ où des différences de valeurs dans $\vec{B}$, représentant le *traitement* expliquent des différences de valeurs dans $\vec{H}$ qui représente le *résultat*. Dans ce qui suit, nous présentons les définitions formelles permettant la définition des règles différentielles causales qui nous intéressent ainsi que la définition de la mesure permettant d'évaluer leur qualité.

3.1 Définitions

Graphes de connaissances. Nous considérons un graphe de connaissances KG défini par une paire $(\mathcal{O}, \mathcal{F})$ où $\mathcal{O}$ est une ontologie représentée en OWL composée d'un ensemble de classes et de propriétés. $\mathcal{F}$ est un ensemble de triplets RDF décrivant des instances de classes de $\mathcal{O}$.

Traitement. Dans le cadre des graphes de connaissances, nous considérons des chemins de propriétés P_t sous la forme $P_t : p_1(X, Y_1) \wedge p_2(Y_1, Y_2) \wedge \ldots \wedge p_n(Y_{n-1}, Y_n)$ de longueur maximale l_{tmax}. Les propriétés se trouvant à l'extrémité de ces chemins peuvent avoir des valeurs catégorielles ou numériques. Ces propriétés peuvent être monovaluées (i.e, fonctionnelles) ou multi-valuées. Par abus de langage, nous considérons la (les) valeur(s) d'un chemin de propriété comme faisant référence à la (les) valeur(s) de la propriété à l'extrémité du chemin. Nous considérons que les règles s'appliquent à deux instances (X_1, X_2) d'une classe de $\mathcal{O}$, et qu'un traitement T représente une différence de valeurs de P_t entre X_1 et X_2.
Nous distinguons deux types de traitements : un *traitement catégoriel* T_c impliquant un chemin de propriétés dont l'extrémité est une propriété catégorielle (e.g, littéral, date, valeurs hiérarchisées) et un *traitement numérique* T_n impliquant un chemin de propriétés dont l'extrémité est une propriété numérique (e.g, entier, réel).

Traitement catégoriel. Soient X_1 et X_2 deux instances d'une classe, un chemin de propriétés P_t et deux ensembles de valeurs V_1 et V_2 du chemin P_t pour X_1 et X_2 respectivement. Un traitement catégoriel T_c est défini par :

$$T_c(X_1, X_2) : P_t(X_1, V_1) \wedge P_t(X_2, V_2) \wedge belongs(v_1, V_1)$$

$$\wedge belongs(v_2, V_2) \wedge \neg belongs(v_1, V_2) \wedge \neg belongs(v_2, V_1)$$

où $belongs(v, V)$ est une fonction qui vérifie que v appartient à l'ensemble des valeurs V.

Traitement numérique. Soient X_1 et X_2 deux instances d'une classe, un chemin de propriétés P_t et deux ensembles de valeurs V_1 et V_2 du chemin P_t pour X_1 et X_2 respectivement. Un traitement numérique T_n est défini par :

$$T_n(X_1, X_2) : P_t(X_1, V_1) \wedge P_t(X_2, V_2) \wedge$$

$$compare_{Tn}(s(V_1), s(V_2))$$

où $compare_{Tn}$ est une fonction de comparaison de valeurs numériques pouvant être implémentée par $lessThan$ ou $greaterThan$ et s est une fonction d'agrégation qui peut être par exemple max, min, sum, etc.
Par exemple, un traitement T_c peut être que deux athlètes ont des manualités différentes : l'un est droitier et le second est gaucher. Un traitement T_n exprime une différence sur une valeur numérique, par exemple que le budget du club d'un athlète est plus élevé que celui d'un autre athlète.

Résultat. Nous considérons un chemin de propriétés P_o sous la forme $P_o : p_1(X, Z_1) \wedge p_2(Z_1, Z_2) \wedge \ldots \wedge p_m(Z_{m-1}, Z_m)$. Pour les résultats, nous considérons seulement les chemins de propriétés menant à des valeurs numériques. Soient X_1 et X_2 deux instances d'une classe, un chemin de propriétés P_o et deux ensembles de valeurs numériques V_1 et V_2 du chemin P_o pour X_1 et X_2 respectivement. Le résultat O est défini par :

$$O(X_1, X_2) : P_o(X_1, V_1) \wedge P_o(X_2, V_2) \wedge$$

$$lessThan(s(V_1), s(V_2))$$

où s est une fonction d'agrégation.

Règle Différentielle Causale. Une règle causale différentielle RDC_T représente la relation de causalité entre le traitement T et son résultat. Elle exprime que le traitement, i.e. une différence de valeurs sur un chemin de propriétés P_t, explique un résultat, i.e. une différence de valeurs sur un chemin de propriétés P_o tel que $lessThan(P_o(s(V_1)), P_o(s(V_2)))$.

Définition 1. (Règle Différentielle Causale). Étant données X_1 et X_2 deux instances d'une classe cible de l'ontologie, le chemin de propriétés menant au résultat P_o, un traitement $T \in \{T_n(X_1, X_2), T_c(X_1, X_2)\}$ défini par le chemin de propriété P_t, et s une fonction d'agrégation, une règle différentielle causale RDC_T est définie comme suit :

$$RDC_T : T \wedge P_o(X_1, V_1) \wedge P_o(X_2, V_2) \Rightarrow lessThan(s(V_1), s(V_2)) \quad (1)$$

Il est à noter que le résultat O est exprimé en partie dans le corps de la règle et dans sa conclusion. Une règle impliquant un traitement numérique est appelée une *règle différentielle causale numérique* et de manière analogue une règle impliquant un traitement catégoriel est appelée une *règle différentielle causale catégorielle*.

Exemple. Étant donné un graphe de connaissances décrivant les athlètes, leurs pays et leur sport, un résultat à étudier pourrait être le classement des athlètes, *i.e.* l'on va chercher à expliquer pourquoi des athlètes ont des performances différentes. Une règle différentielle causale numérique RDC_{age} peut exprimer qu'être plus jeune qu'un autre athlète peut expliquer un meilleur classement, *i.e.* un rang plus bas : $age(X_1, Y_1) \wedge age(X_2, Y_2) \wedge lessThan(Y_1, Y_2) \wedge rank(X_1, Z_1) \wedge rank(X_2, Z_2) \Rightarrow lessThan(Z_1, Z_2)$. Une règle différentielle causale catégorielle $RDC_{manualite}$ peut indiquer qu'être gaucher plutôt que droitier pourrait être une autre explication d'une meilleure performance.

3.2 Effet d'un Traitement

L'effet d'un traitement d'une règle est quantifié en calculant un score par la mesure $causal_T$ inspirée de [9]. Cette mesure correspond à un odds ratio OR qui compare les chances qu'un résultat se produise en fonction d'une exposition particulière, et les chances que le résultat se produise en l'absence de cette exposition. Alors que l'OR considère un ensemble d'instances, $causal_T$ considère seulement un ensemble de paires d'instances similaires et vérifiant le traitement.

Compte tenu des définitions des règles différentielles causales, nous définissons d'abord les deux supports utilisés pour calculer $causal_T$. Soit T un traitement tel que $T \in \{T_n(X_1, X_2), T_c(X_1, X_2)\}$, et O un résultat, $supp_{TO}$ représente le nombre de paires d'instances telles que le traitement et le résultat sont tous deux vérifiés :

$$supp_{TO} = \#(X_1, X_2) : \exists\{Y_1, .., Z_m\}\ tq\ T \wedge O(X_1, X_2) \quad (2)$$

Le support $supp_{T\overline{O}}$ représente le nombre de paires d'instances où le traitement et l'inverse du résultat sont vérifiés, *i.e.* avec le chemin de propriété P_o instancié pour les deux instances mais des valeurs numériques qui vérifient le prédicat $greaterThan$ au lieu de $lessThan$. Il est calculé de manière analogue à celui de $supp_{TO}$, sauf que le résultat O est remplacé par $\overline{O}$, avec $\overline{O} \equiv P_o(X_1, V_1) \wedge P_o(X_2, V_2) \wedge greaterThan(s(V_1), s(V_2))$.

Étant donné un ensemble de paires, la mesure de l'effet d'un traitement est défini dans l'équation 3 suivante :

$$causal_T = \frac{supp_{TO}}{supp_{T\overline{O}}} \quad (3)$$

$causal_T$ retourne un score dans $[0, +\infty[$ et mesure la force de la relation entre le traitement et le résultat. S'il est égal à 1, le traitement et le résultat sont considérés comme indépendants, et plus il est différent de 1, plus cette relation est forte. Une valeur supérieure à 1 montre que le traitement et le résultat sont positivement associés.

Nous utilisons cette mesure pour sélectionner les règles pertinentes et ordonner les explications fournies. Un intervalle de confiance CI est construit pour tester si $causal_T$ est significativement supérieur à 1 : $CI_\alpha(causal_T) = exp(ln(causal_T) \pm u_{1-\alpha/2}\sqrt{\frac{1}{supp_{TO}} + \frac{1}{supp_{T\overline{O}}}})$ avec $u_{1-\alpha/2}$ le $(1-\alpha/2)$ quantile de la loi normale $\mathcal{N}(0, 1)$.

4 Appariement d'instances fondé sur les plongements de graphes

L'effet d'une règle est calculé en considérant des paires d'instances similaires. Comme il a été mentionné dans la section 2, les méthodes classiques d'appariement symbolique peuvent échouer lorsque les descriptions des instances sont hétérogènes, erronées ou incomplètes. Pour notre problème, nous avons besoin d'une mesure de similarité capable de déterminer des instances similaires même lorsque le graphe de connaissances est imparfait. Pour ce faire, nous avons défini une nouvelle mesure de similarité qui exploite les plongements de graphes dans un espace vectoriel à faible dimensions. Plus précisément, nous entraînons un modèle de plongements pour fournir un vecteur à chaque instance. Ensuite, pour mesurer la similarité de deux vecteurs, nous analysons la similarité des prédictions obtenues en utilisant ces vecteurs.

Dans la figure 1, nous présentons le déroulement général de notre approche en cinq étapes. Tout d'abord, (a) nous considérons un graphe de connaissances KG et entraînons un modèle de plongements de graphe qui fournit une représentation vectorielle de chaque instance et relation du KG dans un espace de faible dimension. Ensuite, dans (b), un ensemble de paires est sélectionné aléatoirement, puis la distance de leur vecteurs d et leur similarité sim_e (c.f. équation 4 en section 4.1) sont calculées. Dans (c), un modèle g, en considérant d et sim_e est appris pour définir le seuil de distance d_{tr} étant donné un paramètre sim_{tr}. À (d), un ensemble d'instances appariées, *i.e.* avec une distance d telle que $d < d_{tr}$, est alors créé. Enfin, à (e), nous calculons l'effet de traitement sur le résultat analysé.

4.1 Mesures de similarité sur plongements

Distance basée sur des vecteurs de plongements. Étant donné un KG, nous entraînons un modèle de plongements pour obtenir des vecteurs qui représentent les entités et les relations du KG dans un espace de faible dimension.

Une distance d entre deux vecteurs v_{i_1} et v_{i_2} peut être calculée à l'aide de diverses fonctions telles que la distance euclidienne. Une telle distance est une valeur dans $[0, +\infty[$. Cette valeur peut être utilisée pour décider que deux instances sont similaires en utilisant un seuil donné d_{tr} (*i.e.* i_1 et i_2 considérés similaires si $d(v_{i_1}, v_{i_2}) < d_{tr}$).

Cependant, la définition d'un tel seuil d_{tr} est difficile sans connaissance de ce que les distances représentent. Sans seuil défini, l'étape d'appariement pourrait sélectionner, pour une instance i_1, i_2 tel que $\min_{\forall i_x \in I} d(v_{i_1}, v_{i_x})$, mais i_1 et i_2 pourraient être très différents. Par exemple, un KG pourrait comporter des descriptions de pays comme la France, les États-Unis et l'ensemble des pays d'Asie. En supposant que l'on cherche un pays similaire à la France, l'on pourrait trouver les États-Unis. Cependant, ces 2 pays restent très différents, et ne pourraient peut-être pas être considérés similaires pour notre problème.

Ainsi, un seuil d_{tr} est à définir pour (i) sélectionner des paires d'instances suffisamment similaires pour analyser un traitement, et (ii) élaguer les paires trop différentes.

Mesures de similarité basées sur des prédictions de plongements. Nous proposons une nouvelle mesure de similarité, sim_e, qui repose sur les plongements des instances. sim_e mesure la similarité de deux vecteurs appris en observant la similarité des prédictions utilisant ces vecteurs. Nous supposons que deux instances ayant deux descriptions RDF similaires conduiront à des vecteurs similaires dans l'espace de plongement [22], et que ces vecteurs produiront des prédictions similaires de la part du modèle de plongement. Nous rappelons qu'un modèle de plongement f reçoit en entrée un triplet (h, r, t), et retourne un score $f_r(h, t)$ qui sera élevé si le modèle considère le triplet correct, et bas avec un triplet considéré faux. La principale motivation de cette mesure est que la comparaison des vecteurs permettra à l'approche d'être moins sensible aux données incomplètes ou erronées.

Plus précisément, étant donné un sujet et une propriété p_k, un modèle de plongements f prédit un score pour chaque objet possible de p_k et peut trier les objets par score décroissant. Ainsi, deux instances similaires devraient avoir des prédictions similaires sur les propriétés qu'elles instancient.

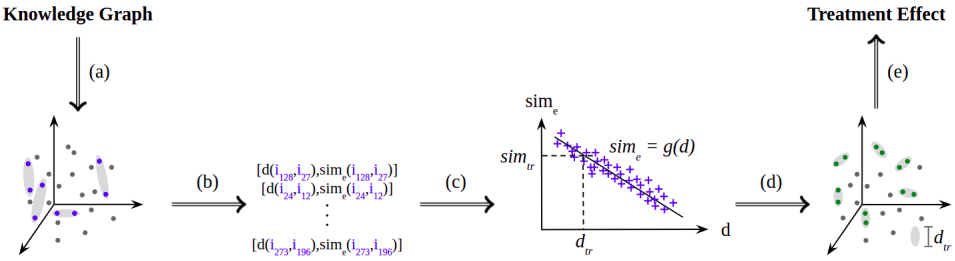

FIGURE 1 – Vision globale de l'approche

Pour calculer sim_e, les prédictions sur chaque propriété p_k instanciant les instances sont étudiées. Pour une propriété p_k d'un chemin P de longueur $l_P < l_{max}$, un degré de fonctionnalité $degre(p_k)$ est défini et représente le nombre moyen d'objets distincts qui instancient p_k. l_{max} est un paramètre qui définit la longueur maximale d'étude d'un chemin de propriétés. Avec $n = degre(p_k)$, la similarité entre deux instances sur la propriété p_k est étudiée en analysant les n premiers objets prédits par f pour chaque instance. Par exemple, pour les pays visités par une personne, nous pouvons poser le degré de fonctionnalité $n = 3$. Ainsi, les 3 pays les mieux classés pour chaque personne sont sélectionnés et utilisés. La similarité sim_{p_k} de deux instances i et j sur une propriété p_k de la classe cible est calculée récursivement comme suit :

$$sim_{p_k}(i,j) = \begin{cases} 1,\ si\ p_k(i) = p_k(j) \\ 0,\ si\ p_k(i) \neq p_k(j),\ i \in L,\ j \in L \\ \\ \frac{\sum_{o_i \in p_k(i)} Max_{o_j \in p_k(j)} sim_e(o_i, o_j)}{degre(p_k)} \\ si\ p_k(i) \neq p_k(j),\ i \in I,\ j \in I \end{cases}$$

où L (resp. I) est l'ensemble des littéraux (resp. des IRIs), $p_k(i)$ est l'ensemble des objets liés à i par p_k.
La similarité entre i et j, $sim_e(i,j)$, est obtenue en faisant la moyenne de la similarité obtenue pour chaque propriété p_k qui appartient à l'ensemble P des propriétés instanciées pour i et j :

$$sim_e(i,j) = \frac{\sum_{p_k \in P} sim_{p_k}(i,j)}{\mid P \mid} \quad (4)$$

Il convient de noter qu'un poids relatif w_i représentant l'importance d'une propriété p_i dans le calcul de similarité peut être introduit et défini par des experts du domaine (par exemple, w_i peut être fixé à 0 pour le nom d'une personne). Nous proposons un exemple pour illustrer sim_e basé sur la Fig. 2. Premièrement, $sim_{registered}(\#person1, \#person2) = \frac{sim_e(\#master1, \#master2)}{1} = \frac{sim_{hasTopic}(\#master1, \#master2)}{2} = \frac{(max(sim_e(IT,IT), sim_e(IT,Physics)) + max(sim_e(Maths,IT), sim_e(Maths,Physics)))}{2} = \frac{1}{2}$.
Ensuite, $sim_{citizenship}(\#person1, \#person2) = \frac{sim_e(\#greece1, \#greece1)}{1} = 1$. Enfin, $sim_e(\#person1, \#person2) = \frac{1/2}{2} + \frac{1}{2} = \frac{3}{4}$.

TABLE 1 – Distance entre vecteurs de plongement de 8 instances

	i_1	i_2	i_3	i_4	i_5	i_6	i_7	i_8
i_1	*0,0*	2,7	1,6	1,9	**0,5**	2,1	2,9	2,7
i_2	2,7	*0,0*	1,6	2,0	3,1	**0,3**	2,5	1,4
i_3	1,6	1,6	*0,0*	1,4	2,2	2,4	3,2	2,3
i_4	1,9	2,0	1,4	*0,0*	2,5	1,8	2,4	2,5
i_5	**0,5**	3,1	2,2	2,5	*0,0*	1,3	3,6	0,9
i_6	2,1	**0,3**	2,4	1,8	1,3	*0,0*	0,8	2,9
i_7	2,9	2,5	3,2	2,4	3,6	0,8	*0,0*	**0,5**
i_8	2,7	1,4	2,3	2,5	0,9	2,9	**0,5**	*0,0*

4.2 Définition du seuil de similarité d_{tr}

d_{tr} est défini en analysant la distribution entre sim_e et d (parties (b) et (c) de la figure 1). Un modèle g entre d et sim_e, tel que $g(d) = sim_e$, est appris. Etant donné qu'aucune hypothèse n'est avancée sur la relation entre d et sim_e, g peut être un modèle linéaire ou non linéaire. Étant donné sim_{tr} le seuil défini sur la similarité sim_e fixé par l'utilisateur, le seuil d_{tr} est défini tel que $g(d_{tr}) = sim_{tr}$.
Le calcul de sim_e étant complexe en temps, la distribution entre sim_e et d est étudiée sur un échantillon de paires.
Dans la table 1, la distance entre 8 vecteurs de plongements d'instances d'une classe cible est affichée. Nous supposons que sim_{tr} a été fixé à 0,8, que le modèle $sim_e = -0,13 * d + 1$ a été appris et que d_{tr} est fixé à 1,3. Ainsi, 3 paires seraient sélectionnées.

5 Algorithme

L'algorithme Dicare-E est composé de 2 parties : (i) la première consiste à déterminer d_{tr}, et (ii) la seconde crée des appariements d'instances similaires en utilisant d_{tr} afin d'évaluer la règle différentielle causale pour un traitement.

5.1 Définition de d_{tr}

Dans la première partie de l'algorithme, d_{tr} est déterminé en analysant la relation entre la distance d et la métrique de similarité sim_e. À cette fin, l'algorithme prend en entrée le KG, la classe cible t_c, les chemins de propriétés liés au résultat et au traitement étudiés P_o et P_t, un modèle de plongement f et le paramètre sim_{tr} pour définir d_{tr}. Le graphe de connaissances KG_{tr}, utilisé pour entraîner f, est obtenu en retirant les derniers prédicats de P_o et P_t de KG. d_{tr} est défini de la façon suivante. Étant donné

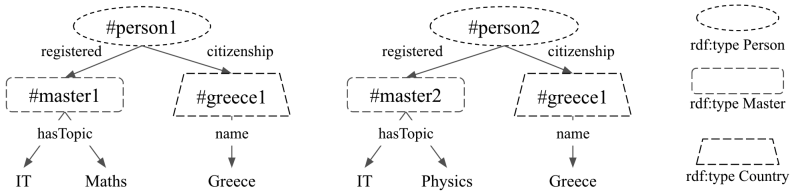

FIGURE 2 – Descriptions prédites de 2 instances pour illustrer sim_e

$\mathcal{I}_{tc}$ l'ensemble des instances de t_c, un échantillon de paires de t_c, $\{(i_i, i_j) \in \mathcal{I}_{tc} \times \mathcal{I}_{tc}\} \subset \mathcal{I}_{tc} \times \mathcal{I}_{tc}$, est tiré, et les distances d entre les vecteurs correspondants $d(v_{i_i}, v_{i_j})$ et les sim_e sont calculées pour chaque paire grâce au modèle appris f. Un second modèle, $g(d) = sim_e$, est entraîné pour obtenir les paramètres de la distribution entre d et sim_e. g peut être une régression linéaire ou polynomiale en fonction de la distribution, pouvant être linéaire ou non. Les paramètres estimés du modèle sont utilisés pour obtenir d_{tr} tel que $g(d_{tr}) = sim_{tr}$.

5.2 Création des paires et découverte de règles RDC_T

La deuxième partie de l'algorithme consiste à déterminer des règles causales différentielles en utilisant d_{tr} précédemment défini. En entrée, l'algorithme considère la classe cible t_c, les chemins P_o et P_t, d_{tr} et α qui est un paramètre statistique utilisé pour calculer un intervalle de confiance pour les effets du traitement.

L'extraction des règles se fait en deux étapes. La première étape consiste à construire les paires d'instances similaires vérifiant T grâce à d_{tr}. Premièrement, pour chaque paire $(i_i, i_j) \in \mathcal{I}_{tc} \times \mathcal{I}_{tc}$, la distance euclidienne entre leurs vecteurs $d(v_{i_i}, v_{i_j})$ est calculée. Les distances sont stockées dans une matrice de distance $\mathcal{D}$. Ensuite, l'ensemble d'instances appariées $\mathcal{M}$ est créé en sélectionnant à chaque itération la paire présentant la plus petite distance et vérifiant le traitement, $\min_{\forall (i_i,i_j) \in \mathcal{I}_{tc} \times \mathcal{I}_{tc}} \mathcal{D}$ et en l'enlevant de $\mathcal{D}$ ensuite. Ce processus est appliqué jusqu'à ce qu'il ne reste aucune paire (i_i, i_j) dans $\mathcal{D}$ telle que $d(v_{i_i}, v_{i_j}) < d_{tr}$.

La deuxième étape consiste à calculer l'effet du traitement. Pour cela, l'algorithme prend en entrée l'ensemble des paires d'instances $\mathcal{M}$, les chemins P_o et P_t, et α, et est initialisé en fixant à 0 les deux supports décrits dans l'équation 3. Ensuite, pour chaque paire, le traitement et le résultat sont obtenus en utilisant P_o et P_t, et les supports sont modifiés en conséquence. Une fois que toutes les paires ont été traitées, l'effet du traitement peut être calculé et son intervalle de confiance construit.

Cet algorithme permet de déterminer l'effet d'un traitement. Afin de déterminer l'effet d'un autre traitement, KG_{tr} doit être mis à jour en enlevant le nouveau traitement et en ajoutant l'ancien, et f réentrainé en conséquence.

TABLE 2 – Description des Données

	DBPediaW	***Vitamin***
# Triplets	6908	86006
# Classes	4	19
# Instances t_c	185	1714
# Propriétés	8	22

6 Expériences

6.1 Données Utilisées

Il n'existe pas de référence pour la découverte de causalité dans les KGs. Nous avons exploité deux KGs déjà utilisés : *Vitamin* [20], pour lequel un expert du domaine est disponible pour une évaluation qualitative, et un extrait relativement simple de *DBPedia*, que l'on nomme *DBPediaW*, utilisé dans [11] et [20]. Les informations de ces graphes sont présentés dans la table 2.

Vitamin décrit des personnes et leurs caractéristiques socio-économiques telles que leur âge, leur lieu de vie, leur travail, leur sexe, leur régime alimentaire actuel et idéal, leurs opinions sur des faits liés au bien-être animal et au changement climatique. Ce graphe a une profondeur de 2. La classe cible $Person$ a 1714 instances, et nous souhaitons expliquer la différence entre le régime actuel et idéal d'une personne, *i.e.* sa volonté de réduire sa consommation de viande, indiquée par le prédicat *reduceMeat* ayant des valeurs $\in \mathbb{N}$ Nous nous concentrons sur la recherche de traitements qui pourraient expliquer la volonté d'une personne de changer ses habitudes alimentaires.

Le lecteur est invité à consulter [11] pour visualiser le schéma de *DBPediaW*. Ce graphe a une profondeur de 2 et décrit des auteurs, leur parcours académique et des livres qu'ils ont écrits. La classe $Writer$ est composée de 185 instances. Nous cherchons à expliquer pourquoi certains auteurs publient leur premier livre plus jeunes que d'autres.

6.2 Évaluation et Résultats

L'objectif est de montrer l'efficacité et la robustesse de l'utilisation de modèles de plongements de graphes pour la découverte de règles causales différentielles dans les KGs.

Tout d'abord, nous avons entraîné et évalué les modèles suivants sur les KGs : *TransE*, *DistMult*, *ComplEx*, *HolE*, *ConvE* et *ConvKB*. Ensuite, nous avons étudié la distribution entre la distance euclidienne d et la métrique de similarité sim_e afin de montrer que cette distribution peut être

TABLE 3 – Résultats

	DBPediaW	***Vitamin***
# Règles ([19])	12	0
# Règles ([20])	12	77
# Règles Dicare-E	3	48
% Paires expliquées ([19])	21,2	0
% Paires expliquées ([20])	21,2	50.1
% Paires expliquées Dicare-E	**78,1**	**92,8**
% Règles pertinentes ([20])	*NA*	66,6
% Règles pertinentes Dicare-E	*NA*	**76,6**

TABLE 4 – Performance des modèles entrainés sur *Vitamin*

Model	***MRR***	***Hits@1***	***Hits@3***	***Hits@10***
ConvE	**0,3384**	**0,2498**	**0,3969**	**0,4850**
DistMult	0,2167	0,1341	0,2451	0,3675
ComplEx	0,1732	0,1011	0,1867	0,3113
TransE	0,1470	0,1341	0,1563	0,2474
Baseline	*0,0057*	*0,0009*	*0,0023*	*0,0081*

estimée par un modèle. Enfin, nous avons appliqué notre algorithme pour découvrir un ensemble de règles causales différentielles qui représentent des effets de traitements, et comparons les résultats obtenus aux règles différentielles causales obtenues par [19] - appariement strict - et [20] - appariement par communautés. Plus précisément, l'objectif est de comparer l'interprétabilité et la pertinence des règles obtenues, le nombre de paires d'instances qu'elles peuvent expliquer, et la robustesse des deux approches en cas de données incomplètes.

Entraînement des modèles de plongements. Le tableau 4 montre que, parmi tous les modèles testés à l'aide de la librairie AmpliGraph sur *Vitamin*, $ConvE$ est le plus performant, *i.e.* avec les *MRR* et *Hits@n* les plus élevés, et est donc utilisé par la suite. *DistMult* est utilisé pour représenter les vecteurs de *DBPediaW* car il obtient les meilleurs performances. Par soucis de représentation des résultats, l'ensemble des tables n'est pas présenté mais peut être trouvé à ce lien [1].

Association entre distance d et similarité sim_e. Pour chaque KG, un ensemble de paires d'instances est tiré aléatoirement et les valeurs d et sim_e sont calculées pour chaque paire. La distribution entre d et sim_e a été modélisée par un modèle linéaire dans les 2 cas. Pour *Vitamin*, la distribution entre d et sim_e est présentée dans la Fig. 3 (à gauche). Le même processus est effectué avec la baseline (à droite). Comme prévu par [22], plus la distance d entre les vecteurs de deux instances est élevée, moins les instances sont similaires, car sim_e diminue lorsque d augmente. La comparaison avec la baseline montre que $ConvE$ est capable de capturer correctement les représentations des entités de *Vitamin* en rapprochant les entités similaires dans l'espace de plongements, et que l'utilisation d'un modèle avec de bonnes performances est nécessaire pour cette approche. La distribution entre d et sim_e de la Fig. 3 est modélisée par un modèle linéaire avec $(r^2 = 0,78)$. En utilisant ce modèle, le seuil de distance d_{tr} est fixé à $1,1$ pour obtenir une similarité sim_{tr} de 0,75.

Règles découvertes et effet de d_{tr}. Deux traitements différents sont analysés pour illustrer l'importance de la détermination de d_{tr} dans la Fig. 4. : le sexe et le lieu de vie de personnes. Une règle est considérée valable si la barre d'erreur ne croise pas la barre horizontale placée à 1. Cette figure montre que plus d_{tr} est faible, plus les barres d'erreurs sont larges. Les paires d'instances des ensembles obtenus sont très similaires, mais la taille de ces ensembles diminuent avec d_{tr}, résultant en une variance élevée dans l'effet estimé. Inversement, plus d_{tr} est élevé, plus les ensembles sont grands mais avec des paires moins similaires. En conséquence, cela mène à une plus faible variance de l'effet et à un biais plus élevé [21]. Le fait que d_{tr} soit fixé avant le calcul des effets de traitement évite d'introduire un biais dans les règles de la part de l'utilisateur. La Fig. 4 nous indique que, pour d_{tr} fixé à 1.1, la volonté de réduire sa consommation de viande est indépendante du genre car la barre d'erreur correspondante croise la barre horizontale. Il semble en revanche y avoir un effet du lieu d'habitation, les barres ne se croisant pas, habiter en campagne plutôt qu'en aire urbaine pourrait expliquer une volonté de réduire sa consommation de viande.

$$RDC_{hasDiet} : hasDiet(X_1, omnivorous) \wedge hasDiet(X_2, vegetarian) \wedge reducing(X_1, Z_1) \wedge reducing(X_2, Z_2) \Rightarrow lessThan(Z_2, Z_1) \quad (5)$$

$$RDC_{bornIn} : bornIn(X_1, Y_1) \wedge bornIn(X_2, Y_2) \wedge publishedIn(X_1, Z_1) \wedge publishedIn(X_2, Z_2) \wedge greaterThan(Y_2, Y_1) \Rightarrow lessThan(Z_2, Z_1) \quad (6)$$

Évaluation qualitative et quantitative et comparaison. Pour évaluer notre approche et la comparer à l'état de l'art [19, 20], trois critères ont été évalués. Qualitativement, les règles des deux approches ont été évaluées. Ensuite, nous avons étudié le pourcentage de paires pour lesquelles les approches peuvent fournir une explication concernant une différence de régimes. Enfin, nous avons testé la robustesse des deux approches aux données incomplètes.
Notre approche découvre 48 règles différentielles causales pour *Vitamin*, et 3 pour *DBPediaW*. Nous présentons 2 règles dans les équations 5 et 6 [2] et invitons le lecteur à visiter le GitHub pour obtenir la liste exhaustive des règles. La règle de l'équation 5 exprime qu'être omnivore par rapport à être végétarien explique une volonté plus forte de réduire sa consommation de viande. Cette règle fait sens car, pour A et B deux personnes similaires, si A consomme déjà moins de produits animaux que B, alors B est plus susceptible de réduire sa consommation de produits animaux.

1. https ://github.com/IC2022RuleEmbeddings/Soumission

2. La représentation des chemins a été simplifiée, les $belong$ sont omis, et la fonction s n'est pas définie car les propriétés numériques sont monovaluées dans $Vitamin$ et $DBPediaW$.

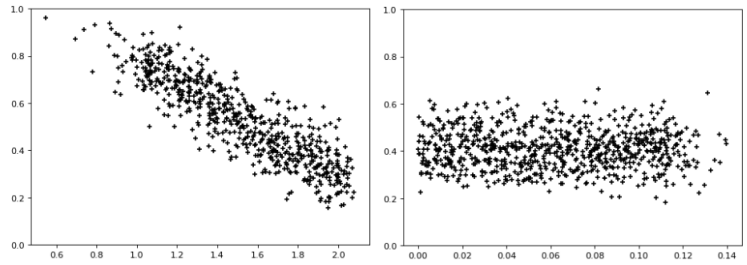

FIGURE 3 – Distance (axe x) et similarité (axe y) d'un ensemble de paires pour $ConvE$ (gauche) et $Baseline$ (droite)

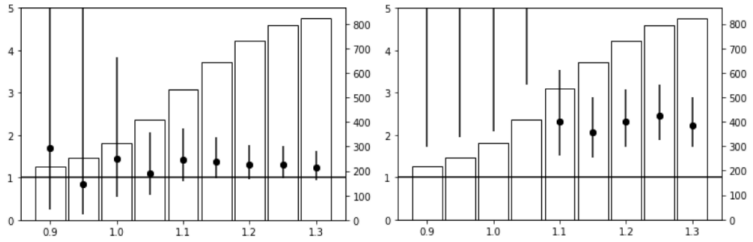

FIGURE 4 – $causal_{OR_T}$ (axe y gauche, points) et nombre de paires créées (axe y droite, histogramme) selon d_{tr} (axe x). Traitements : (gauche) genre (X_1 : femme, X_2 : homme) ; (droite) lieu d'habitation (X_1 : campagne, X_2 : aire urbaine)

La règle de l'équation 6 exprime qu'être né plus tard qu'un autre auteur explique une publication à un âge plus jeune de son premier livre. Ceci peut être expliqué par une accessibilité grandissante de la publication due à un plus grand nombre d'éditeur ou encore aux réductions de coûts de publication.

Vitamin. Les règles découvertes par [20] et Dicare-E ont été évaluées par un expert du domaine, en comportement humain et nutrition. Aucune instance n'a pu être appariée avec [19] qui nécessite que leurs représentations soit isomorphes au traitement et au résultat près. En conséquence, aucune règle n'a pu être extraite (voir table 6.1). Pour faciliter le processus de comparaison des approches, les 30 meilleures règles de [20] et Dicare-E ont été évaluées.

Pour rappel, les règles générées dans [20] peuvent utiliser un motif de graphe plus ou moins spécifique, nommé strate, qui définit l'ensemble des instances auxquelles la règle est appliquée. Dans les deux approches, toutes les règles sont interprétables par l'expert. L'expert a souligné que plus la strate est spécifique, plus elles sont difficiles à interpréter, alors que la plupart des règles de Dicare-E sont faciles à interpréter.

Pour chaque règle, quatre possibilités ont été données à l'expert : la règle (1) *semble être pertinente*, (2) *pourrait être pertinente*, (3) l'expert *ne sait pas si la règle est pertinente* et (4) *semble être fausse*. Cependant, elles diffèrent dans la distribution : pour Dicare-E, 11 règles ont été classées dans (1), 12 dans (2), 6 dans (3) et 1 dans (4). Pour les règles de [20], la distribution est la suivante : 12 règles en (1), 8 en (2), 6 en (3) et 4 en (4). Par conséquent, 66,6% des règles de [20] semblent être pertinentes ou pourraient l'être, ce qui est inférieur à notre approche (76,6%). De plus, Dicare-E n'a qu'une seule règle qui semble fausse, alors que [20] en a 4. La justification de l'expert sur la règle semblant être fausse est que les valeurs de traitement sont trop similaires pour que la règle ait un sens : les deux font référence à l'importance de l'entourage d'une personne dans sa prise de décision, l'une reposant davantage que l'autre sur son entourage. Par ailleurs, notre approche fournit un ensemble plus large de traitements : des chemins de propriété supplémentaires par rapport à [20] ont été déterminés.

Pour une évaluation quantitative, le pourcentage de paires pour lesquelles au moins une explication du résultat peut être fournie a été calculé pour les deux approches. Notre approche peut fournir une explication pour 92,8% de ces paires alors que [20] obtient un nombre de 50,1%, elle explique donc beaucoup plus de différences. Nous avons calculé la même métrique en supprimant 15% des triplets de *Vitamin* pour tester la robustesse des deux approches. Le nombre de règles extraites et le pourcentage de paires expliquées ont été réduits dans les deux approches, mais notre approche est plus robuste : le pourcentage de paires expliquées est passé de 92,8% à 89,5% pour notre approche, et de 50,1% à 24,7% pour [20]. Ceci s'explique du fait qu'une instance avec une description incomplète ne sera pas utilisée pour la détermination de règles dans [20] alors qu'elle le sera dans notre approche.

DBPediaW. Les 3 règles obtenues par Dicare-E sont compréhensibles et semblent pertinentes. De plus, elles comportent des traitements également obtenus dans [20]. Sur les 12 règles obtenues par [20], 9 semblent pertinentes. Les résultats de [19] sont les mêmes que [20] car, le schéma étant simple, aucune détection de communauté n'a été réalisée. Il est intéressant de voir que par opposition au jeu de données précédent, pour un KG comme *DBPediaW* dont le schéma est relativement simple, les règles déterminées par [20] ont des strates avec peu d'éléments qui sont donc facilement compréhensibles et qui permettent de générer des

règles plus expressives que Dicare-E (*e.g.* règle concernant la date de naissance valide pour les auteurs ayant étudié aux États-Unis dans une université bien classée).
Quantitativement, notre approche permet d'expliquer 78,1% des paires contre 21,2% pour [20]. En enlevant 15% des triplets, les règles obtenues avec notre approche sont les mêmes. Le nombre de paires expliquées ne change donc pas. Avec [20], ce nombre passe à 6,0%. Comme sur *Vitamin*, notre approche permet d'expliquer plus de paires ayant une différence de résultat et est plus robuste aux données manquantes.

7 Conclusion

Dans cet article, nous avons proposé une approche qui combine des modèles de plongements de graphes et des techniques de fouille de règles symboliques pour la découverte de règles différentielles causales dans les graphes de connaissances. Une telle approche hybride est capable de traiter efficacement des données incomplètes tout en fournissant des règles interprétables, qui peuvent expliquer les différences dans une caractéristique numérique étudiée. Nous avons montré qu'elle peut être utilisée pour apparier des instances similaires grâce à leur représentation dans l'espace des plongements appris, permettant ainsi l'application du cadre des résultats potentiels. La métrique de similarité proposée, basée sur les prédictions du modèle de plongements, garantit la création de paires similaires uniquement. Notre expérience et collaboration avec un expert montre que notre approche peut être utilisée sur des domaines variés ainsi que sur des KGs complexes. Dans de futurs travaux, nous souhaitons analyser les règles plus en profondeur, notamment le nombre de règles pouvant expliquer le résultat d'une paire et ses conséquences.

Références

[1] Althauser, R.P., Rubin, D. : The Computerized Construction of a Matched Sample. American Journal of Sociology 76(2), 325–346 (1970)

[2] Chickering, D.M., Heckerman, D., Meek, C. : Large-sample learning of Bayesian networks is NP-hard. JMLR 5, 1287–1330 (2004)

[3] Ferilli, S., Basile, T.M., Biba, M., Di Mauro, N., Esposito, F. : A general similarity framework for horn clause logic. Fundamenta Informaticae 90(1-2), 43–66 (2009)

[4] Ferré, S. : Answers Partitioning and Lazy Joins for Efficient Query Relaxation and Application to Similarity Search. Lecture Notes in Computer Science 10843 LNCS, 209–224 (2018)

[5] Galárraga, Luis Teflioudi, Christina Hose, Katja Suchanek, Fabian. (2013). AMIE : Association rule mining under incomplete evidence in ontological knowledge bases. WWW 2013. 413-422.

[6] Haan, Rosaline Tiddi, Ilaria Beek, Wouter. Discovering Research Hypotheses in Social Science Using Knowledge Graph Embeddings. The Semantic Web 477-494 (2021)

[7] Iacus, S.M., King, G., Porro, G. : Causal inference without balance checking : Coarsened exact matching. Political Analysis 20(1), 1–24 (2012)

[8] Jain, N., Kalo, J.C., Balke, W.T., Krestel, R. : Do embeddings actually capture knowledge graph semantics ? In : Verborgh, R., Hose, K., Paulheim, H., Champin, P.A., Maleshkova, M., Corcho, O., Ristoski, P., Alam, M. (eds.) The Semantic Web. pp. 143–159.

[9] Li, Jiuyong le, Thuc Liu, Lin Liu, Jixue Jin, Zhou Sun, Bingyu. (2013). Mining Causal Association Rules. ICDMW 2013. 114-123.

[10] Moon, C., Jones, P., Samatova, N.F. : Learning entity type embeddings for knowledge graph completion. CIKM '17 p. 2215–2218. ACM, NY, USA (2017)

[11] Munch, M., Dibie, J., Wuillemin, P., Manfredotti, C.E. : Towards interactive causal relation discovery driven by an ontology. In : International Florida Artificial Intelligence Research Society Conference (2019)

[12] Neapolitan, R.E. : Learning Bayesian Networks. Pearson Prentice Hall. (2003)

[13] Nentwig, M., Hartung, M., Ngomo, A.N., Rahm, E. : A survey of current link discovery frameworks. Semantic Web 8(3), 419–436 (2017)

[14] Paulheim, H. : Knowledge graph refinement : A survey of approaches and evaluation methods. Semantic Web 8(3), 489–508 (2017).

[15] Pearl, J. : Causality. Cambridge University Press (2009)

[16] Raad J., Pernelle N., Saïs F. Detection of Contextual Identity Links in a Knowledge Base. In Proceedings of the Knowledge Capture Conference (K-CAP 2017). Association for Computing Machinery

[17] Rosenbaum, P.R., Rubin, D.B. : Reducing bias in observational studies using subclassification on the propensity score. Journal of the American Statistical Association 79(387), 516–524 (1984)

[18] Rubin D. B : Estimating causal effects of treatment in randomized and nonrandomized studies. Journal of Educational Psychology 66(5), 688–701 (1974)

[19] Simonne, L., Pernelle, N., Sais, F. : Fouille de règles différentielles causales dans les graphes de connaissances, EGC 2021, pp.293-300

[20] Simonne L., Pernelle N, Saïs F., Thomopoulos R. Differential Causal Rules Mining in Knowledge Graphs. In Proceedings of the 11th on Knowledge Capture Conference (K-CAP '21). Association for Computing Machinery, 105–112.

[21] Stuart, E.A. : Matching methods for causal inference : A review and a look forward. Statistical science : a review journal of the Institute of Mathematical Statistics 25(1), 1–21 (2010)

[22] Wang, C., Pan, S., Hu, R., Long, G., Jiang, J., Zhang, C. : Attributed graph clustering : A deep attentional embedding approach. IJCAI 2019 (2019), 3670-3676.