Graphes de connaissances et intelligence artificielle explicable : application au repositionnement de médicaments

Martin Drancé

université de BORDEAUX

THÈSE PRÉSENTÉE

POUR OBTENIR LE GRADE DE

DOCTEUR DE L'UNIVERSITÉ DE BORDEAUX

ECOLE DOCTORALE SOCIETES, POLITIQUE, SANTE PUBLIQUE

OPTION INFORMATIQUE ET SANTÉ

Par Martin DRANCÉ

Graphe de Connaissances et Intelligence Artificielle Explicable : Application au Repositionnement de Médicaments.

Sous la direction de : Gayo DIALLO
Co-direction : Akka ZEMMARI
Co-encadrement : Fleur MOUGIN

Soutenue le 04/12/2024

Membres du jury :

M. Zied BOURAOUI *Professeur des universités, Université d'Artois*	Rapporteur
Mme. Sandra BRINGAY *Professeure des universités, Université Paul-Valéry Montpellier*	Rapporteure
M. Adrien COULET *Chargé de recherche HDR, Inria Paris*	Examinateur
M. Antoine PARIENTE *Professeur des universités - Praticien hospitalier, Université de Bordeaux*	Président du jury
M. Gayo DIALLO *Professeur des universités, Université de Bordeaux*	Directeur de thèse
M. Akka ZEMMARI *Professeur des universités, Université de Bordeaux*	Co-Directeur de thèse

Membre invitée :

Mme. Fleur MOUGIN *Professeure des universités, Université de Bordeaux*	Co-encadrante

TITRE : Graphe de Connaissances et Intelligence Artificielle Explicable : Application au Repositionnement de Médicaments.

Résumé : Le repositionnement de médicaments consiste à trouver de nouvelles utilisations thérapeutiques pour des médicaments existants qui sont déjà approuvés pour traiter d'autres pathologies. Cette approche profite des connaissances déjà existantes sur ces molécules, permettant ainsi un développement plus rapide et moins coûteux par rapport à la création de nouveaux médicaments. Le repositionnement est particulièrement utile pour répondre à des besoins médicaux non satisfaits, comme par exemple pour les maladies rares ou émergentes. Ces dernières années, le développement de graphes de connaissances a permis de concentrer toutes ces informations biomédicales autour du médicament issues de grandes bases de données ou de connaissances. Un graphe de connaissances est une représentation structurée d'informations provenant de différentes sources, qui relie ces informations les unes aux autres par l'utilisation de relations. Cette représentation est particulièrement utile pour mieux comprendre les relations complexes qui structurent nos connaissances sur un médicament. Elle est utilisée de nos jours pour la tâche de repositionnement en particulier. Une façon efficace de repositionner des médicaments à partir de ces graphes est d'utiliser des méthodes d'intelligence artificielle qui prédisent de nouveaux liens entre les objets du graphe. De cette manière, un modèle correctement entraîné sera capable de proposer une nouvelle connexion entre un médicament et une maladie, indiquant une potentielle opportunité de repositionnement. Cette méthodologie présente cependant un gros désavantage : les modèles pour la prédiction de liens fournissent souvent des résultats opaques, qui ne peuvent pas être interprétée par l'utilisateur final des prédictions. Cette thèse propose d'étudier l'utilisation de méthodes d'intelligence artificielle explicables dans le but de repositionner des médicaments à partir de données biomédicales représentées dans des graphes de connaissances. Dans un premier temps, nous analysons l'impact du pré-entraînement sur les modèles de multihop reasoning pour la prédiction de liens. Nous montrons que la construction des représentations des entités du graphe avant l'entraînement du modèle permet une amélioration des performances prédictives, ainsi que de la quantité et la diversité des explications. Dans un second temps, nous étudions comment l'ajout de relations dans un graphe de connaissances affecte les résultats de prédiction de liens. Nous montrons que l'ajout de liens dans trois graphes biomédicaux permet une amélioration des performances prédictives du modèle SQUIRE, et ce sur différents types de relations lien avec le repositionnement de médicaments. Une analyse de l'impact sur l'explicabilité du modèle est aussi menée à la suite de l'ajout de ces relations. Enfin, nous proposons une nouvelle méthodologie pour la tâche de classification de liens dans un graphe de connaissances, basée sur l'utilisation de forêts aléatoires. À partir des informations concernant le voisinage de chaque nœud dans le graphe, nous montrons qu'un modèle de forêts aléatoires est capable de prédire correctement l'existence ou non d'un lien entre deux nœuds. Ces résultats permettent une visualisation des nœuds utilisés pour réaliser la prédiction. Enfin, nous appliquons cette méthode au repositionnement de médicaments pour la sclérose latérale amyotrophique (SLA).

Mots-clés : Repositionnement de médicaments, Machine learning, Graphes de connaissances

TITLE : Knowledge Graphs and Explainable Artificial Intelligence : Application to Drug Repositioning.

Abstract : Drug repositioning involves finding new therapeutic uses for existing medications that are already approved to treat other conditions. This approach takes advantage of the existing knowledge about these molecules, enabling faster and less costly development compared to creating new drugs. Repositioning is particularly useful for addressing unmet medical needs, such as rare or emerging diseases. In recent years, the development of knowledge graphs has enabled the consolidation of all this biomedical information around drugs, coming from large data sources or knowledge repositories. A knowledge graph is a structured representation of information integrated from different sources, linking these pieces of information together using relationships. This representation is especially useful for understanding the complex relationships that structure knowledge about drugs. Nowadays, it is widely used for the task of drug repositioning. An effective way to reposition drugs using these graphs is to employ artificial intelligence (AI) methods that predict new links between objects in the graph. In this way, a well-trained model can suggest a new connection between a drug and a disease, indicating a potential opportunity for repositioning. However, this methodology has a significant disadvantage : link prediction models often provide opaque results that cannot be easily interpreted by the end users. This thesis proposes to explore the use of explainable AI methods for the purpose of repositioning drugs based on biomedical data represented in knowledge graphs. First, we analyze the impact of pre-training on multihop reasoning models for link prediction. We demonstrate that building representations of the graph entities before model training improves the predictive performance, as well as the quantity and diversity of explanations. Secondly, we examine how the addition of relationships in a knowledge graph affects link prediction results. We show that adding links in three biomedical knowledge graphs improves the predictive performance of the SQUIRE model across different types of relationships related to drug repositioning. An analysis of the impact on model explainability is also conducted, following the addition of these relationships. Finally, we propose a new methodology for the task of link classification in a knowledge graph, based on the use of random forests. Using information about the neighborhood of each node in the graph, we show that a random forest model can accurately predict the existence or absence of a link between two nodes. These results allow for a visualization of the nodes used to make the predictions. Lastly, we apply this method to drug repositioning for amyotrophic lateral sclerosis (ALS).

Keywords : Drug repurposing, Machine learning, Knowledge graphs

---

Centre de recherche Bordeaux Population Health (BPH)
Equipe AHeaD, Inserm U1219, 126 rue Léo Saignat, 33076 Bordeaux CEDEX

Table des matières

Chapitre 1

Introduction

Ces dernières années, la santé publique fait face à des défis majeurs, notamment l'augmentation des maladies chroniques, l'émergence régulière de nouvelles pandémies et les inégalités d'accès aux soins. Entre autres, le vieillissement de la population [Bloom et al., 2011], les résistances aux antibiotiques [Larsson and Flach, 2022] et l'exposition à un environnement de plus en plus pollué [Briggs, 2003] aggravent ces problématiques. Dans ce contexte, l'innovation thérapeutique peut jouer un rôle clé pour répondre aux besoins médicaux non satisfaits, en proposant des traitements plus efficaces, plus personnalisés, plus accessibles et plus rapidement.

C'est dans ce contexte que le numérique a profondément transformé le secteur de la santé publique, offrant de nombreux avantages pour les patients, les professionnels de santé et les systèmes de soins [Imison et al., 2016]. Grâce aux technologies numériques, telles que les dossiers médicaux électroniques [Bates et al., 2003] ou l'intelligence artificielle (IA), il est désormais possible de personnaliser les traitements, d'améliorer le suivi des patients ou encore de faciliter la coordination des soins. Récemment et en suivant l'évolution d'autres domaines du numérique, les données relatives à la santé, à la biologie ou à la pharmacologie sont collectées, stockées et utilisées massivement. Cette approche orientée "big data" des données de santé [Lee and Yoon, 2017] contribue à une meilleure compréhension des maladies et à des avancées en matière de recherche, tout en optimisant les prises de décision médicale et la gestion des ressources.

Cette prolifération de données et l'essor du numérique ont favorisé le développement d'un procédé déjà connu : le repositionnement de médicaments [Pushpakom et al., 2019a]. Cette approche consiste à utiliser des médicaments déjà approuvés pour traiter de nouvelles maladies, réduisant ainsi considérablement les coûts et le temps nécessaires à leur mise sur le marché. Grâce aux avancées en IA et à la construction de grandes bases de données médicales, il est possible d'identifier rapidement de nouvelles indications pour des molécules existantes. Cela permet non seulement d'accélérer la disponibilité des traitements pour des pathologies rares ou émergentes, mais aussi de capitaliser sur les connaissances déjà acquises sur ces médicaments.

Toutefois, il subsiste un frein majeur à l'adoption des méthodes numériques récentes à base d'IA : leur manque d'explicabilité. En effet, ces dernières années, l'effort s'est porté sur le développement de modèles d'IA toujours plus efficaces dans leurs prédictions, au détriment de la capacité à comprendre comment ces prédictions ont été générées. Ces modèles, très performants mais au fonctionnement indéchiffrable, sont qualifiés de "boîtes noires". Ce manque d'explicabilité rend leur utilisation problématique dans des domaines sensibles tels que la santé publique, où la technologie doit éclairer la décision du professionnel et non la remplacer.

Cette thèse vise à définir quelles méthodes d'IA explicables sont adaptées au repositionnement de médicaments. En particulier, nous nous intéressons aux modèle d'IA qui opèrent sur des graphes de connaissances biomédicales, une structure de données adaptée à la fois aux données volumineuses et à l'utilisation de modèles d'IA explicables.

Afin d'apporter les éléments de contexte du travail réalisé, nous abordons tout d'abord la notion de graphes, les méthodes d'apprentissage appliquées aux graphes, l'explicabilité de ces méthodes et le repositionnement de médicaments.

1.1 Les graphes

Les **graphes** se définissent par des nœuds (ou sommets) et des arêtes qui relient ces nœuds. Nous verrons par la suite qu'il existe une multitude de types de graphes, avec parfois un vocabulaire différent pour désigner les nœud et les arêtes. Nous utiliserons parfois le terme entités pour parler des nœuds du graphe, comme nous utiliserons les termes liens ou relations pour parler des arêtes.

Une des traces les plus anciennes de problèmes résolus en utilisant un graphe est le problème du cavalier devant visiter chaque case d'un échiquier. C'est le théoricien d'échecs arabe Al-Adli qui l'a résolu au IXe siècle [Ádlí ar Rúmí, 840]. Ce problème est en fait un cas particulier du problème des sept ponts de Königsberg, proposé au XVIIIe siècle par le mathématicien suisse Leonhard Euler [Euler, 1741]. Ce dernier a cherché à déterminer s'il était possible de faire une promenade partant d'un point donné et revenant à ce point après avoir traversé chacun des sept ponts de la ville une seule fois. Euler est considéré comme le père de la théorie des graphes, car il fut le premier à proposer un traitement mathématique de ces questions. Aujourd'hui, la théorie des graphes propose un classement en trois familles, dépendantes des propriétés que l'on peut retrouver ou non au sein du graphe : les graphes structurés, les graphes quelconques et les graphes multipolaires. Les **graphes structurés** regroupent tous les graphes qui présentent des propriétés remarquables, au sens où leur organisation présente des motifs identifiables. On retrouve dans cette famille :

— Les **graphes homogènes** : tous les sommets et les arêtes produisent un schéma régulier. La représentation la plus simple de ces graphes est un carré ou un rectangle quadrillé.

— Les **graphes hiérarchiques** : les sommets s'organisent de manière pyramidale, formant des couches souvent représentées du haut vers le bas. On peut aussi dans ce cas parler d'arbres, avec un sommet qualifié de **racine** et les nœuds finaux qualifiés de **feuilles**.
— Les **graphes cycliques** : ils contiennent des cycles, c'est-à-dire des suites d'arêtes consécutives qui permettent de revenir sur le nœud de départ après déplacement, sans repasser deux fois par la même arête.
— Les **graphes polaires** : tous les nœuds sont connectés à un seul et même nœud central, appelé **pôle**.

À l'inverse des graphes structurés, on trouve la famille des **graphes quelconques**, qui ne présentent aucune propriété remarquable. Ces graphes ne semblent pas montrer de patterns récurrents, ni de hiérarchie. Ils peuvent par ailleurs contenir des cycles, mais pas de manière systématique, et certains nœuds possèdent plus d'arêtes que les autres sans pour autant créer un pôle.

Enfin, la famille des **graphes multipolaires** rassemble des graphes qui associent graphes polaires et graphes quelconques. On y retrouve donc des graphes constitués de plusieurs pôles, reliés entre eux par des nœuds et des arêtes qui servent de pont ou de liaison, comme illustré sur la Figure 1.1. Cette dernière famille de graphes est aujourd'hui celle qui est la plus étudiée, car ce sont les graphes multipolaires qui décrivent au mieux la réalité et l'organisation de nombreux domaines. Ils sont utilisés notamment en neurosciences [Bullmore and Sporns, 2009] pour modéliser les connexions neuronales, dans la détection de fraudes financières [Pourhabibi et al., 2020] pour modéliser les transactions, dans la représentation des voies biologiques [Zhang and Wiemann, 2009], ou encore pour optimiser le transport au travers d'un réseau [Petric Maretic et al., 2019].

1.1.1 Propriétés importantes des graphes

Quel que soit le graphe considéré, les données qu'il contient ou encore la finalité pour laquelle il est construit, il existe des propriétés communes à tous les graphes qui peuvent être mesurées, afin de mieux comprendre leur organisation.

Premièrement, le concept de **distance** est très souvent utilisé quand on souhaite étudier la position des nœuds d'un graphe les uns par rapport aux autres [Pollack and Wiebenson, 1960]. La distance correspond le plus souvent au nombre d'arêtes qu'il faut parcourir pour connecter deux nœuds entre eux. De cette notion de distance découlent plusieurs propriétés du graphe [Bondy et al., 1976] :

— Chemin le plus court : longueur du chemin le plus court entre deux nœuds ;
— Diamètre du graphe : nombre d'arêtes contenues dans le plus long des plus courts chemins ;
— Longueur caractéristique : moyenne des plus courts chemins entre tous les nœuds

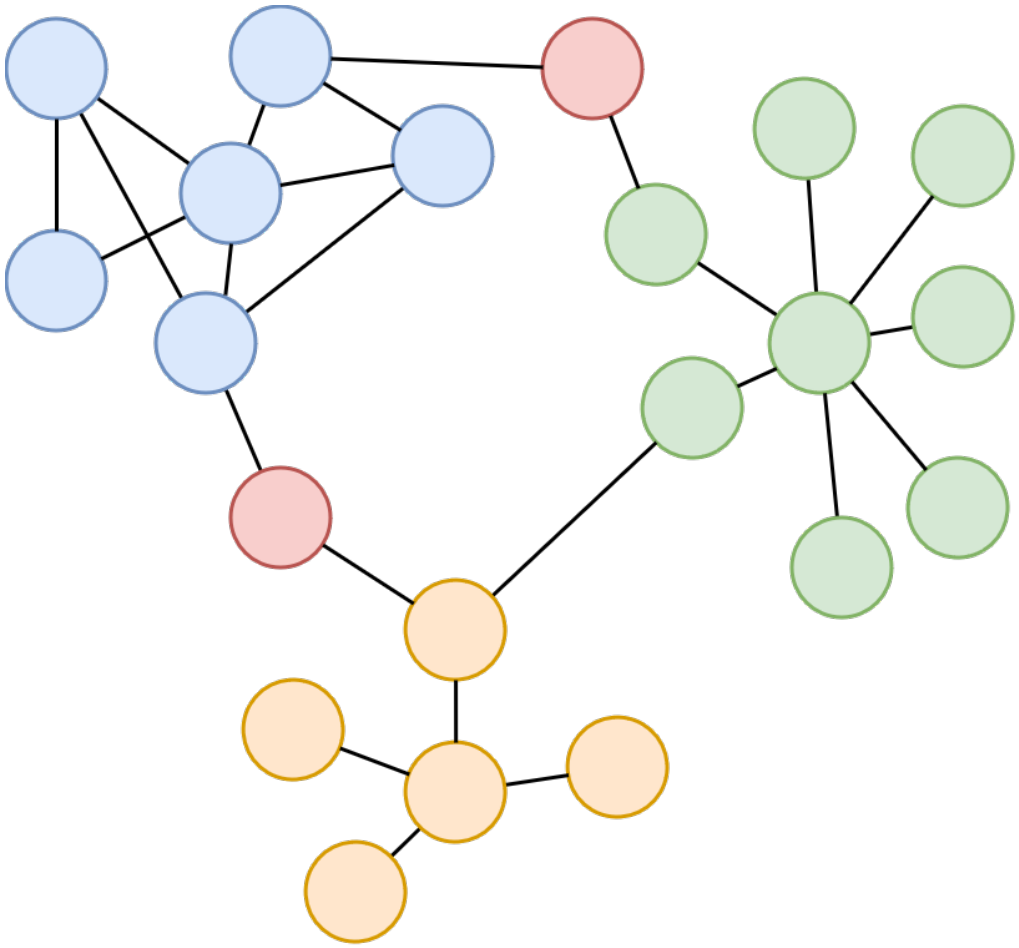

FIGURE 1.1 – Illustration d'un graphe multipolaire. En jaune et vert, deux graphes polaires reliés entre eux par une liaison "faible" (arête unique). En bleu, un graphe quelconque relié aux graphes polaires par des nœuds solitaires, en rouge.

du graphe.

Les graphes sont très rarement homogènes dans la répartition des nœuds et des arêtes, avec parfois des régions où les nœuds sont plus intensément connectés entre eux. Pour caractériser ce phénomène, des métriques de **clustering** (groupement) peuvent être utilisées [Schaeffer, 2007]. On notera en particulier :

- Coefficient de clustering : probabilité que les voisins directs d'un nœud soient aussi connectés entre eux ;
- Modularité : découpage du graphe en groupes de nœuds très connectés entre eux.

La troisième notion importante relative à l'analyse d'un graphe et ses nœuds est la **centralité** [Ghosh and Lerman, 2011]. La centralité sert à évaluer à quel point un nœud est important pour les autres nœuds du graphe et pour ses voisins directs. Cette importance se définit par exemple par la capacité du nœud à transmettre de l'information entre deux pôles du graphe, ou à servir lui-même de pôle pour un grand nombre de voisins.

On retrouve trois mesures principales de la centralité [Kang et al., 2011] :

— Degré : nombre de voisins d'un nœud ;

— Centralité de proximité : distance d'un nœud donné avec tous les autres nœuds du graphe pour déterminer à quel point ce nœud est central dans le graphe ;

— Centralité entre-deux : est la fréquence à laquelle un nœud réside sur le chemin le plus court entre d'autres paires de nœuds. Cela sert à définir à quel point un nœud peut servir de pont entre les autres nœuds. Par exemple, cela permet d'évaluer un nœud ayant peu d'arêtes, mais permettant de connecter deux parties du graphe qui ne le seraient pas sinon.

En dehors de l'analyse pure d'un graphe, de sa structure, ou de nœuds en particulier, ces mesures de propriétés peuvent être utilisées pour l'entraînement d'algorithmes d'apprentissage (*machine learning*) appliqués aux graphes. Un exemple d'utilisation de ces propriétés est la détection de comptes factices au sein d'un réseau social [Fakhraei et al., 2015]. Après avoir représenté les utilisateurs d'un réseau social sous forme d'un graphe (les nœuds étant les utilisateurs et les arêtes les interactions entre eux), les auteurs ont effectué plusieurs mesures sur le graphe obtenu (liste non exhaustive) :

— L'indicateur PageRank [Page et al., 1999], qui est une mesure de la centralité d'un nœud,

— Le degré de chaque nœud, qui est aussi un indicateur de centralité,

— La taille du *"weakly connected component"* [Pemmaraju and Skiena, 2003] auquel appartient chaque nœud, qui est une mesure de clustering.

Après avoir obtenu ces informations à propos de chacun des nœuds du graphe, les auteurs montrent que ces données sont suffisantes pour entraîner un algorithme de machine learning et à classer si un utilisateur est un compte factice ou non.

1.2 Le machine learning sur les graphes

Comme dans d'autres domaines tels que l'analyse d'images ou de textes, le machine learning sur les graphes s'est énormément développé ces dernières années. Il existe aujourd'hui trois grandes tâches réalisables par des modèles de machine learning à partir de données disponibles sous forme de graphes :

— Au niveau des **nœuds** : le modèle tente de classer le nœud dans la bonne classe et/ou de lui attribuer le bon label [Bhagat et al., 2011], c'est-à-dire le ranger dans la bonne catégorie (par exemple définir si un nœud doit être classé comme une protéine ou un gène) ;

— Au niveau des **relations** : le modèle tente de trouver le label ou la classe d'un lien qui relie deux nœuds, voire de prédire l'existence d'un lien entre deux nœuds qui

ne sont pas encore connectés [Pandey et al., 2019] ;
— Au niveau du **graphe** : le modèle est entraîné à labéliser des graphes entiers, souvent de petite taille, en leur attribuant la bonne classe, par exemple pour classer des molécules comme toxiques ou non [Zhang et al., 2018].

La première étape quand on désire utiliser un modèle de machine learning est de construire des ***features*** décrivant les données, qui servent ensuite de caractéristiques observables par le modèle lors de son entraînement. Par exemple, Liben-Nowell et Kleinberg [Liben-Nowell and Kleinberg, 2003] sont parmi les premiers à voir montré que l'utilisation de mesures de centralité ou de distance entre deux nœuds permet de prédire si un lien devrait exister entre ces deux nœuds. À partir de quatre métriques différentes liées à la centralité et trois mesures de distance, les auteurs ont montré que la prédiction de liens était systématiquement améliorée par rapport à une prédiction aléatoire, et ce, sur les cinq graphes différents qu'ils ont utilisés.

Cependant, ces méthodes ont rapidement été remplacées par des modèles qui sont capables de construire eux-mêmes les features à attribuer à chaque nœud et chaque lien dans le graphe. Ces méthodes sont basées sur la construction de représentations vectorielles de chaque nœud du graphe : les embeddings (ou plongements, comme préciser précédemment). Le but de ce type de modèles est alors d'apprendre une fonction de correspondance qui permet de transformer un nœud en une représentation vectorielle $f : V \to \mathbb{R}^n$. Cette fonction f essaie de construire un espace vectoriel qui conserve l'information topologique du graphe d'origine : les nœuds et arêtes qui sont proches dans le graphe auront des distances euclidiennes faibles, et inversement. Une des méthodes les plus utilisées pour construire l'embedding d'un graphe est l'algorithme Node2Vec [Grover and Leskovec, 2016], qui se base sur des déplacements aléatoires dans le graphe pour construire petit à petit les embeddings des nœuds. Ces embeddings peuvent ensuite être utilisés pour des tâches secondaires. De la même manière, Edge2Vec [Wang et al., 2020] et Graph2Vec [Narayanan et al., 2017] sont utilisés pour créer les représentations des arêtes ou d'un graphe tout entier, respectivement.

Ces méthodes ont largement prouvé leur efficacité pour construire des représentations vectorielles fidèles à la structure initiale du graphe. Elles présentent cependant un désavantage de taille : **le manque de transparence**. En effet, comme toutes les méthodes d'embeddings, les représentations vectorielles sont latentes et ne portent donc plus aucun sens interprétable pour les humains. Par exemple, imaginons que nous souhaitions utiliser un algorithme de clustering pour détecter des communautés dans un graphe. Pour construire les features de chaque nœud, nous construisons des embeddings. L'algorithme peut ensuite créer les meilleurs groupes possibles en se basant sur ces embeddings. Mais avec cette approche, il est impossible pour l'utilisateur de comprendre pourquoi deux nœuds sont proches. Est-ce parce qu'ils ont des voisins en commun ? Est-ce parce qu'ils

décrivent des concepts sémantiquement proches ? Est-ce parce qu'ils sont tous les deux des nœuds centraux dans le graphe ? Ce manque de transparence est problématique quand on souhaite appliquer des méthodes de machine learning dans des domaines dits "sensibles", comme la défense, la finance ou la santé. Dans ces domaines, il est important de pouvoir contrôler et valider le fonctionnement et les prédictions des modèles [Rudin et al., 2022].

1.3 L'explicabilité dans le machine learning

L'explicabilité dans le machine learning est cruciale lorsque l'on veut appliquer ces méthodes dans des domaines sensibles. C'est pour cette raison que, dans ces domaines, une volonté de transparence émerge depuis quelques années. Ce besoin est exprimé par les institutions, comme l'UNESCO [1] ou l'Union Européenne [2], mais aussi par les utilisateurs des méthodes d'IA. L'importance de l'explicabilité est devenue évidente dans le but d'éviter des problèmes sociétaux graves affectant la santé, la liberté ou la sécurité des citoyens. Dans le domaine de la justice pénale, Glenn Rodríguez a été condamné à une année supplémentaire de prison à cause du modèle Compas [Wexler, 2017]. En effet, une erreur du modèle attribuait à G. Rodríguez un score trop haut de risque de récidive. Dans un tout autre domaine, un modèle privé (dont le code source n'est pas disponible) évaluant la qualité de l'air a eu des conséquences graves pour la sécurité publique lors d'incendies de forêts à Sacramento, aux États-Unis [Schmidt, 2020] : le modèle indiquait une qualité de l'air optimale dans des zones où l'incendie était en cours.

Ces modèles, qualifiés de "boîtes noires" (*black-box*), sont soit des modèles dont le mécanisme est trop compliqué pour être analysé par les humains, soit des modèles dont les paramètres et le fonctionnement ne sont pas rendus publics à la communauté scientifique. Cette opacité les rend particulièrement difficiles, voire impossibles, à déboguer. En outre, les prédictions faites par ces modèles sont souvent de très bonne qualité, ce qui accentue la difficulté de s'en séparer. Cependant, ces résultats peuvent être obtenus en se basant sur de mauvais critères [Schramowski et al., 2020, Hamamoto et al., 2020]. En effet, on retrouve souvent le phénomène *"Hans le malin"* dans le fonctionnement des réseaux de neurones profonds : plutôt que d'apprendre à faire une prédiction en se basant sur les données connues, le modèle apprend à déceler des corrélations cachées (donc qui ne sont pas censées être exploitées) au sein des données et base ses prédictions là-dessus. Dans la majorité des cas, ces modèles ne seront pas capables de faire des prédictions correctes une fois utilisés sur des données réelles, car ces corrélations peuvent ne plus être présentes à ce moment-là.

Dans le domaine de la santé, les modèles de type "boîte noire" transforment des

1. Recommandation sur l'éthique de l'intelligence artificielle, document publié en 2021 et disponible en ligne à l'adresse suivante : `https://unesdoc.unesco.org/ark:/48223/pf0000381137_fre`
2. Le Règlement européen visant à encadrer l'usage de l'IA, l'*AI Act*, a été proposé en 2021, publié en 2024 et sera appliqué en 2026

décisions qui étaient précédemment assistées par ordinateur en des décisions entièrement automatiques. Ceci est dû au fait que les professionnels de santé ne peuvent pas comprendre les processus de raisonnement de ces modèles.

1.4 Le repositionnement de médicaments dans ce contexte

Dans le domaine de la santé justement, une tâche qui bénéficie grandement des nouvelles techniques d'IA est le repositionnement de médicaments : le principe consiste à étudier des médicaments déjà commercialisés, ou qui ont passé les premières phases d'essais cliniques, afin de voir s'ils pourraient être utilisés pour d'autres pathologies que celles pour lesquelles ils ont été conçus. Le coût toujours plus important pour développer de nouvelles molécules [DiMasi et al., 2016] a rendu le repositionnement de médicaments de plus en plus attrayant, avec la promesse de diminuer les coûts de développement et le délai de mise sur le marché. Le repositionnement de médicaments peut se produire de manière fortuite, comme c'est le cas pour le sildenafil (connu sous le nom de Viagra) [Kolata, 1998], qui était initialement destiné à traiter l'hypertension pulmonaire, mais qui a révélé avoir un effet non négligeable sur l'érection lors des essais cliniques. Le repositionnement de médicaments peut aussi se faire en approfondissant l'étude de molécules déjà sur le marché, comme c'est le cas pour la duloxetine. Initialement approuvée comme antidépresseur, la molécule a ensuite été homologuée pour traiter la fibromyalgie aux États-Unis [Arnold, 2007] et l'incontinence urinaire d'effort en Europe [Thor and Katofiasc, 1995]. On observe que depuis le début des années 2000, le repositionnement de médicaments bénéficie grandement des nouvelles méthodes informatiques, permettant de traiter rapidement un nombre important de données, qu'elles soient génétiques, protéomiques, chimiques ou encore phénotypiques [Ostrov, 2004].

Le développement de ces méthodes permet d'obtenir des résultats de plus en plus nombreux, à partir de données de plus en plus fournies. Ces améliorations s'accompagnent malheureusement d'un accroissement de la complexité de ces modèles, rendant les résultats difficiles, voire impossibles, à interpréter. Initialement pensées pour réduire la complexité des données biologiques et médicales, pouvoir trier, classer, séparer toutes ces informations pour isoler l'essentiel et les rendre plus facilement compréhensibles pour l'utilisateur, les méthodes computationnelles modernes produisent des résultats opaques et en quantité trop élevée. À partir de données biologiques contenues dans des graphes de connaissances, cette thèse s'intéresse à des méthodes transparentes pour le repositionnement de médicaments, qui peuvent fournir une explication des résultats aux utilisateurs finaux. Ainsi, l'amélioration des résultats ne repose plus uniquement sur la mesure de la performance d'un modèle, mais aussi sur la mesure de son explicabilité.

Chapitre 2

Background

Sommaire

Ce chapitre introduit les données, les notions et les méthodes utilisées dans le cadre de cette thèse. Dans un premier temps, nous introduisons formellement ce qu'est un graphe de connaissances et présentons les différents KGs avec lesquels nous avons travaillé (section 2.1). Ensuite, nous présentons un état de l'art des méthodes d'apprentissage automatique (*machine learning*) en lien avec celles que nous utilisons pour réaliser des prédictions à partir des KGs présentés (sections 2.2 et 2.3). Enfin, la section 2.4 se concentre sur des méthodes et des travaux importants qui ont été réalisés dans le domaine du repositionnement de médicaments.

2.1 Graphes de connaissances

2.1.1 Définition et propriétés

Une des premières utilisations des **graphes de connaissances**, alors désigné simplement de graphes pour des raisons de simplification, s'est faite dans le domaine du développement des systèmes d'instruction. La recherche dans ce domaine cherche à créer un instructeur informatique le plus efficace possible dans sa manière de distiller du savoir à un élève, ainsi que dans sa compréhension du feedback donné par cet élève. C'est ainsi qu'en 1973, Ryder et Redding [Schneider, 1973] ont utilisé un graphe de connaissances pour organiser les connaissances des instructeurs, ensemble de connaissances qui devront être transmises à l'élève. Chaque nœud représente un concept et chaque arête une relation typée entre deux concepts. C'est ensuite dans le début des années 1980 que le projet WordNet [Miller, 1995] est lancé dans le but de créer une base de connaissances lexicale pour l'anglais. Tous les mots du lexique sous-représentés sous forme de nœuds, reliés entre eux par des arêtes correspondant à des relations sémantiques (par exemple de la synonymie entre deux termes). En 1989, les universités de Trento (Italie) et de Groningen (Pays-Bas) lancent le projet Knowledge Graph [De Vries, 1989] pour continuer à travailler sur la création de graphes sémantiques. L'utilisation des graphes de connaissances a ensuite explosé avec leur exploitation par des moteurs de recherche tels que celui de Google en 2012, ou encore la construction de gigantesques bases de connaissances comme DBPeda [1], la version data de Wikipédia. Aujourd'hui, la plupart des grandes entreprises du Web (Google, Microsoft, Amazon, Linkedin, Airbnb, etc...) utilisent des graphes de connaissances pour représenter et faciliter l'accès à leurs données.

Les graphes de connaissances (KGs) sont un type particulier de graphes. Ils permettent de représenter une base de connaissances sous forme graphique. Chaque nœud représente alors une entité du monde réel et chaque arête matérialise la relation qui existe entre deux entités. Ces liens sont nommés (via un label) et orientés.

Plus formellement, les KGs peuvent être définis comme suit [Paulheim, 2017] : un KG (i) décrit des entités du monde réel et leurs interactions au sein d'un graphe, (ii) définit des entités et des relations entre ces entités sous forme de schéma, (iii) permet de mettre en relation des entités entre elles de manière arbitraire et (iv) peut regrouper des informations issues de domaines différents. Cette mise en relation des informations se fait au travers de triplets reliant deux entités, le sujet et l'objet, par une relation, qualifiée de prédicat, sous la forme $(sujet, prédicat, objet)$. En 2013, Min *et al.* ont souligné que la plupart des KGs sont incomplets [Min et al., 2013]. Par exemple, dans le KG Freebase [Bollacker et al., 2008], 93,8% et 78,5% des nœuds $Person$ ne sont pas liés à un nœud de type $PlaceOfBirth$ et $Nationality$, respectivement. C'est à partir de cette observation que

1. https ://www.dbpedia.org/

deux tâches principales ont émergé pour découvrir les connaissances manquantes dans un KG : (i) la **complétion de KG**, qui vise à récupérer les faits réels qui sont absents d'un KG, et (ii) la **prédiction de liens**, qui vise à découvrir des liens inconnus entre les entités.

Pour revenir à la définition formelle, étant donné un ensemble d'entités $\mathcal{E}$ et un ensemble de relations binaires $\mathcal{R}$, un KG $\mathcal{G} \subset \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ représente la collection de faits exprimés sous forme de triplets (h, r, t) avec $h, t \in \mathcal{E}$ et $r \in \mathcal{R}$. Toutes les relations r sont dirigées et définissent le rôle des entités dans chaque triplet, avec le nœud h (head) étant le sujet (ou source) et le nœud t (tail) étant l'objet (ou cible) du triplet. Toutes les entités $e \in \mathcal{E}$ et les relations $r \in \mathcal{R}$ appartiennent à des catégories ou types spécifiques, définissant précisément les types de triplets qui peuvent être trouvés dans le KG. Par exemple, si le KG définit que la relation $worksFor$ doit lier un nœud de type $Person$ à un nœud de type $Organization$, on peut alors déduire du triplet $(Tim, worksFor, OxfordUniversity)$ que Tim est une personne et $OxfordUniversity$ est une organisation.

Les KGs sont de nature flexible et peuvent être utilisés pour représenter des connaissances générales, comme celles de Wikidata [van Veen, 2019], ou des connaissances spécifiques à un domaine, comme Hetionet [Himmelstein et al., 2017] qui organise des données biomédicales ou FinKG [Cheng et al., 2020] pour des données financières.
Dans le cadre de cette thèse, nous travaillons avec trois KGs, dits "benchmarks", qui sont principalement utilisés dans des travaux de recherche pour développer et comparer les modèles de prédiction de liens entre eux. Par ailleurs, nous étudions trois KGs biomédicaux sur lesquels évaluer ce type de modèles pour le repositionnement de médicaments. Les caractéristiques de ces graphes peuvent être trouvées dans le Tableau 2.1 et une présentation de ceux-ci est donnée dans les deux sous-sections suivantes.

TABLE 2.1 – Caractéristiques des KGs utilisés dans la suite des travaux. La partie supérieure présente les KGs benchmarks, la partie inférieure les KGs biomédicaux.

Graphe	#Nœuds	#Relations	#Triplets
WN18RR	40 945	11	86 835
NELL-995	75 492	200	154 213
FB15K-237	14 505	237	272 115
Oregano	361 889	17	780 775
Hetionet	45 158	24	2 249 925
BioKG	105 524	13	2 043 846

2.1.2 Graphes de connaissances benchmarks

FB15K-237 FB15K-237 [Toutanova et al., 2015] est un KG construit à partir de la base de connaissances Freebase [Bollacker et al., 2008]. Freebase est un projet collaboratif qui

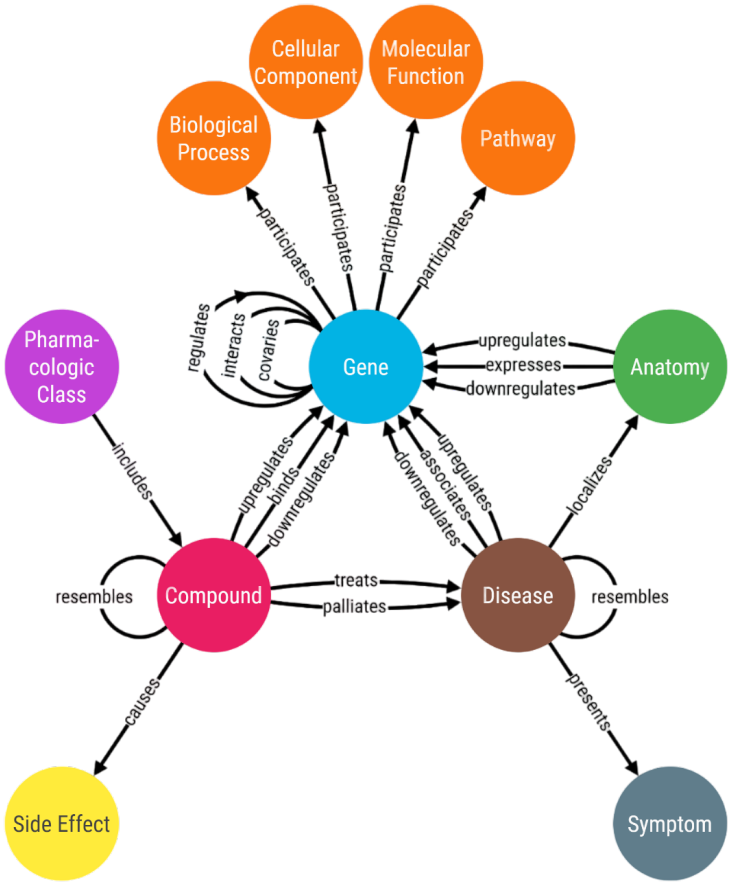

FIGURE 2.1 – Schéma illustrant l'organisation du graphe de connaissances Hetionet : il contient 11 types de nœuds et 24 types de relations orientées. Source : [Himmelstein et al., 2017]

avait pour ambition de rassembler et connecter entre elles toutes les données disponibles sur le Web. Le projet a pris fin en 2015, Freebase contenait alors 1,9 milliard d'éléments.

WN18RR WN18RR [Dettmers et al., 2018] est un KG construit à partir de WordNet [Miller, 1995]. WordNet étant une base de données lexicales qui répertorie, classe et met en relation le contenu lexical de la langue anglaise. Par exemple, WordNet connecte entre eux tous les mots ou locutions qui sont interchangeables, et se propose de classer chaque mot dans un ensemble de concepts (dits *synsets*).

NELL-995 NELL-995 [Xiong et al., 2017] est un KG qui reprend les données de NELL (Never-Ending Language Learner) [Carlson et al., 2010]. NELL a été construit à partir de textes libres issus de pages Web. À partir de ces textes, des règles de Horn ont été extraites [Horn, 1951] pour organiser et structurer les informations s'y trouvant.

2.1.3 Graphes de connaissances biomédicaux

Les KGs biomédicaux intègrent généralement différentes bases de connaissances médicales et biologiques disponibles en ligne, comme UniProt [Consortium, 2022], Entrez Gene [Maglott et al., 2005] ou encore PharmGKB [Whirl-Carrillo et al., 2021]. Nous présentons ici les trois KGs auxquels nous nous sommes intéressés dans le cadre de cette thèse.

Hetionet Hetionet [Himmelstein et al., 2017] combine les informations de 29 bases de données publiques, contenant des informations médicales, biologiques ou pharmacologiques. Comme illustré dans la Figure 2.1, plusieurs types de relations peuvent relier les mêmes types de nœuds. Hetionet propose de définir 11 types de nœuds et 24 types de relations différentes.

Oregano Oregano propose de créer un KG biomédical intégrant 10 bases de connaissances, y compris des données sur des composés naturels. En effet, depuis 1981, plus de 60% des médicaments développés l'ont été à partir de composés naturels (remèdes à base de plantes) [Boudin et al., 2023]. En utilisant la base de connaissances NPASS [Zeng et al., 2018], 96 000 composés naturels ont ainsi été inclus dans le graphe, qui se compose lui aussi de 11 types de nœuds différents. La Figure 2.2 correspond au schéma d'Oregano.

BioKG BioKG [Walsh et al., 2020] intègre 18 bases de connaissances dans la construction du graphe. La différence principale avec Oregano et Hetionet est le nombre limité de types de nœuds et de relations différentes. Bien que le graphe contienne plus de 2 millions de triplets, il est uniquement composé de six types de nœuds différents. La Figure 2.3 présente le schéma de BioKG.

2.2 Prédiction de liens

Nous présentons ici les trois types de méthodes principalement utilisées pour la tâche de prédiction de liens dans des KGs. D'abord, nous expliquons les méthodes d'embeddings, qui ont été les premières à être développées spécifiquement pour cette tâche. Ensuite, nous nous focalisons sur les méthodes symboliques, qui sont basées sur l'utilisation de règles logiques. Enfin, nous détaillons les méthodes de multi-hop reasoning, qui tentent de combiner performances et explicabilité. La dernière partie de cette section est consacrée à la définition des métriques relatives à la tâche de prédiction de liens.

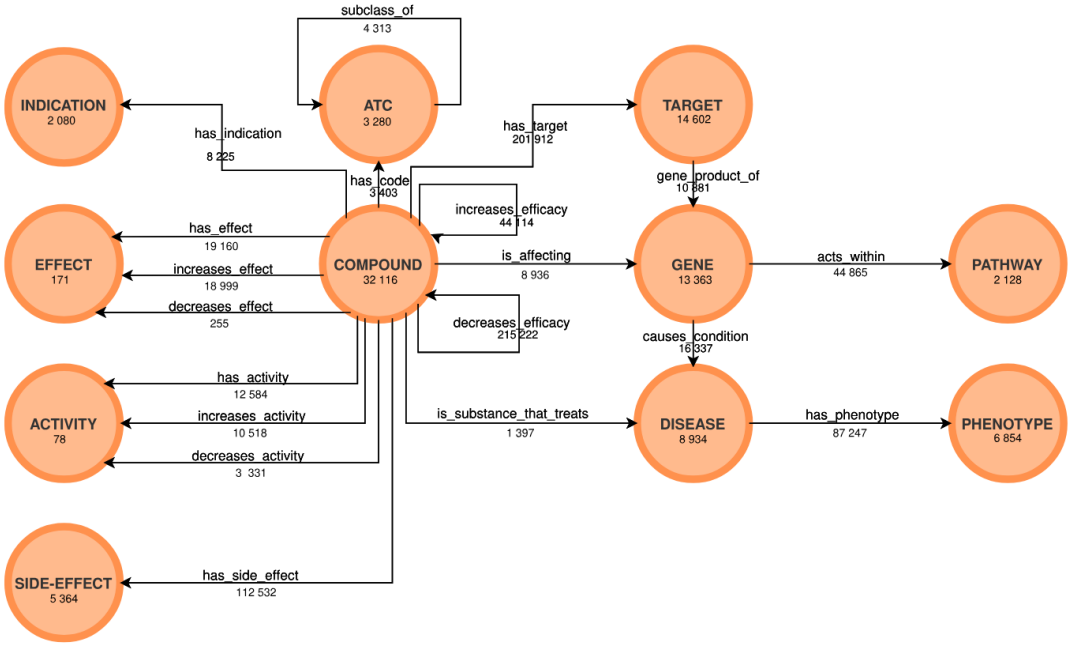

FIGURE 2.2 – Schéma illustrant l'organisation du graphe de connaissances Oregano : il contient 11 types de nœuds et 17 types de relations orientées. Source : [Boudin et al., 2023]

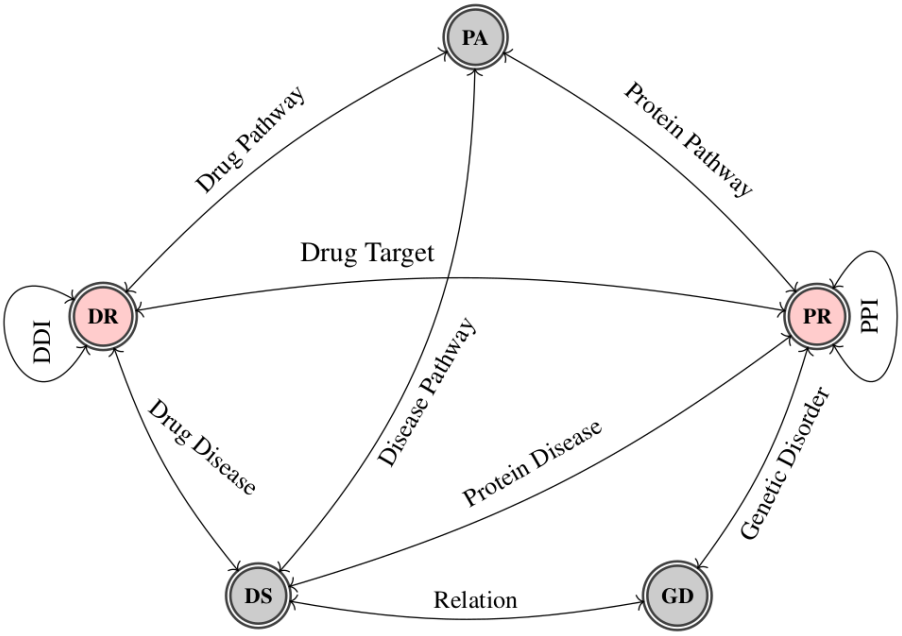

FIGURE 2.3 – Schéma illustrant l'organisation du graphe de connaissances BioKG : il contient 6 types de nœuds et 13 types de relations orientées. Notons que dans ce schéma, les auteurs ont regroupé sous le même nœud "PR" les protéines et les complexes protéiques. Source : [Walsh et al., 2020]

2.2.1 Méthodes d'embedding de graphes de connaissances

La première famille de **méthodes d'embedding de graphes** (*knowledge graph embedding* ou KGE) est la famille des **embeddings par translation**. Pour un triplet (h, r, t), ces méthodes construisent l'embedding de t en reportant les valeurs de h et r. La méthode la plus connue utilisant cette technique repose sur l'algorithme TransE [Bordes et al., 2013]. Il définit l'embedding de t comme $h + r \approx t$ et utilise une distance $d(h + r, t)$ comme fonction de coût. TransE est considéré comme l'algorithme le plus basique de cette famille, puisqu'il ne permet pas la représentation de relations de type 1-N, telles que $(Paracétamol, soigne, Cephalées)$ et $(Paracétamol, soigne, Fièvre)$. Pour pallier cette limite, plusieurs autres algorithmes qui se basent sur le calcul de la distance $d(h + r, t)$ ont été développés, parmi lesquels TransH [Yang et al., 2014b], TransR [Lin et al., 2015] ou encore RotatE [Sun et al., 2019]. Les différences résident dans la manière de calculer la distance $d(h + r, t)$ ou encore la capacité ou non à représenter des relations 1-N et N-N. Ces modèles par transfert ont l'avantage d'être simples, robustes et de produire de très bons résultats quand les données ne sont pas trop complexes.

Il est aussi possible d'utiliser des techniques d'embedding plus expressives que les méthodes par transfert, à savoir les **méthodes multiplicatives** (ou bilinéaires). Elles se basent sur la création de matrices de poids pour modéliser les relations entre les entités. L'algorithme le plus connu de ce type est RESCAL [Nickel et al., 2011], qui calcule le score d'un triplet (h, r, t) grâce à la fonction $h^T \times W_r \times t$ où W_r est une matrice contenant les poids décrivant les interactions entre les nœuds du graphe et h et t les embeddings de ces nœuds. Alors que ces algorithmes permettent de mieux capturer les relations entre les objets composant le graphe, ils sont aussi plus sensibles au surapprentissage (overfitting) [Nickel et al., 2011], c'est-à-dire l'incapacité du modèle à généraliser les prédictions à des données qu'il n'a pas vues lors de la phase d'entraînement. Pour pallier ce problème, plusieurs alternatives ont été proposées, comme DistMult [Yang et al., 2014a] ou ComplEx [Trouillon et al., 2016], qui minimisent le phénomène d'overfitting en diminuant le nombre de paramètres du modèle.

Le dernier groupe de méthodes permettant l'embedding de graphes correspond aux approches fonctionnant à la manière des **réseaux de neurones**. Ces méthodes s'inspirent du fonctionnement des réseaux de neurones profonds déjà utilisés dans d'autres domaines, tels que le traitement d'images [Russakovsky et al., 2015] ou le traitement automatique des langues [Hochreiter and Schmidhuber, 1997]. Il existe deux grandes familles de réseaux de neurones : les réseaux de neurones convolutifs, permettant une extraction des features principales d'une image, et les réseaux de neurones récurrents, permettant de traiter les séquences de caractères. C'est le cas par exemple de ConvE [Dettmers et al., 2018] ou

RGCN [Schlichtkrull et al., 2018] qui s'inspirent des réseaux de neurones convolutifs pour réaliser des tâches de prédiction de liens sur un graphe en sélectionnant les features des nœuds voisins.

Le point commun de toutes ces méthodes est qu'elles fonctionnent comme des boîtes noires. Basée exclusivement sur des calculs mathématiques et statistiques, la création de l'embedding ne fait pas intervenir de logique ou de règles symboliques dans son fonctionnement, perdant ainsi une partie de l'information contenue dans les données, mais aussi la possibilité de fournir une explication autre que mathématique sur son fonctionnement.

2.2.2 Méthodes symboliques

En opposition aux méthodes statistiques d'embedding, il est possible d'appliquer des **algorithmes d'IA dite symbolique** aux KG. Ces méthodes se basent sur la logique présente dans les données et les liens entre les entités pour en dégager de nouveaux nœuds ou de nouvelles relations. Le but est souvent d'extrapoler à partir des données existantes, afin de créer des règles de Horn (ou clauses de Horn) [Horn, 1951]. Une règle de Horn est constituée d'une suite d'atomes composant le corps de la règle de Horn, et d'un atome final constituant la tête de la règle de Horn. En logique, un atome est une formule qui ne contient pas de sous-formules. Dans notre cas, un atome correspond à un triplet dont une des entités est une variable, par exemple *soigne(X, Paracetamol)*. Un exemple de règle de Horn pourrait être $cause(gène, maladie) \wedge affecte(médicament, gène) \rightarrow traite(medicament, maladie)$, avec son corps à gauche de la flèche et sa tête à droite. Cette règle s'interprète comme suit : si un *gène* cause une *maladie* et qu'un *médicament* affecte ce *gène* alors ce *médicament* pourrait être mis en lien avec la *maladie* via une relation *traite*. Ce type d'IA est nommée programmation logique inductive (*inductive logic programming* ou ILP), dont le but est de comprendre les règles logiques qui sous-tendent un jeu de données. L'avantage de ces méthodes est qu'elles sont extrêmement robustes pour résoudre des problèmes simples et fonctionnent de manière totalement transparente. En revanche, elles peinent à égaler les performances des méthodes statistiques pour les problèmes très complexes et nécessitent un temps d'exécution particulièrement élevé quand les données sont très nombreuses [Barredo Arrieta et al., 2020]. Le modèle le plus couramment utilisé de cette famille est AnyBURL [Meilicke et al., 2019]. Ce modèle utilise une approche dite "de bas en haut" (bottom-up), qui consiste à considérer qu'un simple exemple trouvé dans les données est une représentation d'une règle spécifique qui pourrait être généralisée pour retrouver tous les exemples positifs existant dans ces données. AnyBURL est un algorithme itératif qui se répète pendant un certain nombre de périodes courtes *ts*. Pendant chaque intervalle *ts*, l'algorithme découvre un maximum de règles en parcourant le KG. Ces règles peuvent être stockées dans trois ensembles distincts : R contient toutes les règles

trouvées aux itérations précédentes, R_s contient toutes les règles trouvées pendant l'itération actuelle et $R^{'}_s$ contient les règles trouvées dans cette itération qui appartiennent aussi à R. L'algorithme commence par chercher les règles de longueur 2, c'est-à-dire les règles qui impliquent deux relations dans leur corps. Une fois un certain degré de saturation atteint, l'algorithme cherche des règles de longueur 3, et ce, jusqu'à une longueur n prédéfinie. La saturation permet de mesurer la redondance dans les règles qui sont trouvées par le modèle. En calculant $|R^{'}_s|/|R_s|$, on peut calculer la proportion de règles qui avaient déjà été trouvées dans les itérations précédentes. Par exemple, une saturation de 99% indique que 99% des règles trouvées pendant la durée ts avaient déjà été trouvées avant, on peut donc passer à l'apprentissage de règles plus longues.

2.2.3 Multi-hop reasoning et apprentissage par renforcement

La dernière classe de méthodes de prédiction de liens que nous évoquerons ici est le **multi-hop reasoning** (MHR). Ces méthodes sont dites neuro-symboliques, car elles associent réseaux de neurones et règles logiques pour fonctionner. Le but de ces méthodes est de tirer pleinement profit de la puissance statistique des réseaux de neurones en compensant leur manque de transparence par l'intégration de modules symboliques (basés sur des règles logiques, et donc transparents) qui permettent une explication des prédictions [Broda et al., 2002]. Plus précisément, les modèles de MHR se basent sur l'apprentissage par renforcement pour prédire de nouveaux liens dans le graphe.

Apprentissage par renforcement

L'apprentissage par renforcement est un paradigme d'IA se basant sur le principe que l'apprentissage se fait par l'interaction avec l'environnement. Le but pour l'apprenant, appelé agent, est d'apprendre à choisir la meilleure action à faire dans chacune des situations qu'il peut rencontrer. L'agent ne sait pas ce qu'il doit faire, mais il reçoit ou non des récompenses en fonction des actions qu'il choisit dans chaque situation. Le but de l'agent sur le long terme est de maximiser la valeur globale de ses actions, et non les récompenses instantanées. Pour reproduire cette manière d'apprendre de manière computationnelle, quatre éléments sont nécessaires :

— la **politique**, qui définit la manière dont l'agent doit agir à chaque instant. C'est une association entre ce qui est perçu de l'environnement à un instant donné et l'action il faut prendre en conséquence. La politique peut être définie à l'avance, mais peut aussi être apprise au fil du temps.

— la **récompense**, qui sert de signal entre l'environnement et l'agent. Pour chaque action que l'agent choisit d'effectuer, l'environnement envoie à l'agent un signal positif, négatif ou neutre qui lui permet d'apprendre à distinguer les bonnes actions des mauvaises.

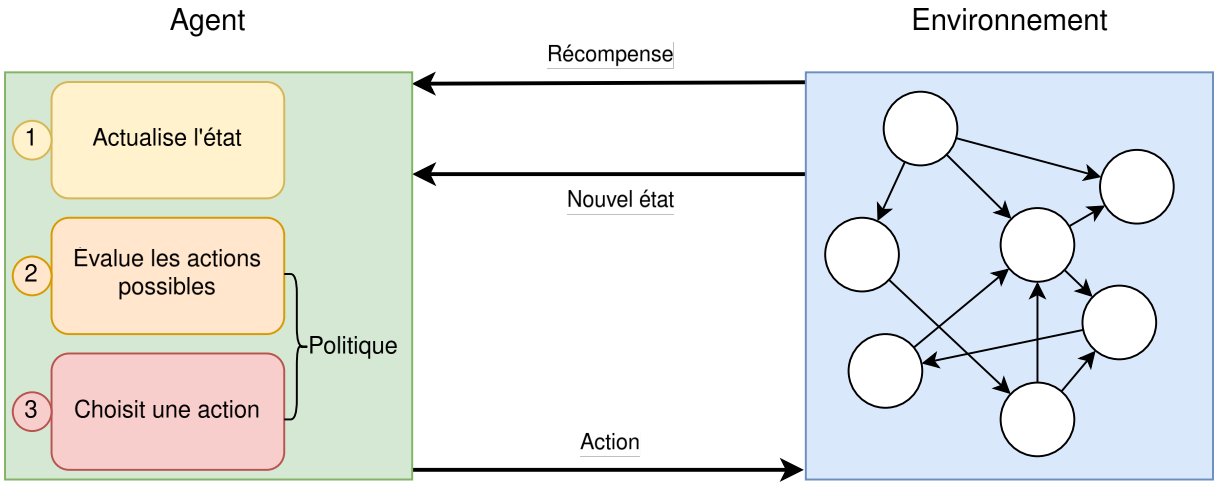

FIGURE 2.4 – Schéma représentant le fonctionnement d'un modèle de multi-hop reasoning. À chaque déplacement dans le graphe, le modèle doit évaluer quelle sera la prochaine meilleure action en fonction du chemin déjà parcouru.

— la **valeur**, qui représente ce que l'agent peut espérer gagner dans le futur s'il choisit une action donnée, là où la récompense représente ce que l'agent gagnera immédiatement à chaque action. La valeur d'un état dans lequel se trouve l'agent représente à quel point il est souhaitable de se trouver dans cet état sur le long terme.
— un **modèle de l'environnement**, qui sert à déterminer à l'avance comment l'environnement réagira aux actions de l'agent. Il peut être présent ou non. Modéliser l'environnement sert à planifier et savoir à l'avance ce qui se passera avant même d'avoir décidé quelle action était la meilleure.

Multi-hop reasoning

Dans le cas des modèles de MHR, le KG représente l'environnement dans lequel l'agent doit apprendre à se déplacer, en passant de nœud en nœud, comme schématisé dans la Figure 2.4. Lorsque appliqué au repositionnement de médicaments, l'idée générale est que l'agent peut apprendre à partir d'un nœud de type médicament et atteindre un nœud de type maladie en effectuant plusieurs déplacements. Si l'agent atteint une maladie qui n'était pas directement connectée au médicament de départ, alors on peut considérer que le modèle prédit que ces deux entités devraient être reliées. Dans ce cas, le chemin parcouru par l'agent pour faire cette prédiction sera utilisé comme l'explication justifiant la possibilité de créer un lien direct entre le médicament et la maladie.

DeepPath [Xiong et al., 2017] est la première tentative de modélisation du problème de recherche de chemins dans un KG en utilisant l'algorithme d'apprentissage par renforcement REINFORCE [Williams, 1992]. La particularité de DeepPath est que les entités source et

cible doivent être connues, c'est-à-dire que DeepPath apprend à trouver les bons chemins de raisonnement entre les deux entités e_{source} et e_{target}, mais pas à prédire un nouveau lien entre des nœuds non connectés dans le graphe.

MINERVA [Das et al., 2017] est le premier modèle de MHR à réellement aborder le problème de prédiction de liens en utilisant l'apprentissage par renforcement. En effet, au lieu de se concentrer sur la recherche de chemins réels entre deux entités, MINERVA apprend à atteindre le nœud correct qui complète une requête $(e_{source}, r_{query}, ?)$, en traversant la meilleure séquence de relations et d'entités soutenant le choix du nœud prédit.

PoLo [Liu et al., 2021] est la première approche de MHR qui tente de réduire les récompenses injustement reçues par l'agent lorsque des chemins erronés sont utilisés mais mènent à une prédiction correcte. Comme il existe de nombreux chemins pour relier deux nœuds, certains d'entre eux peuvent ne pas être valides pour soutenir la prédiction finale. Cependant, dans MINERVA, la récompense est seulement binaire $\{0, 1\}$, ce qui permet à l'agent d'utiliser des chemins erronés pour des prédictions valides, c'est-à-dire des chemins qui sont sans signification par rapport au nouveau lien prédit. En utilisant MINERVA et pour guider l'agent en éliminant ces récompenses liées à des explications invalides, PoLo utilise un ensemble de règles logiques connues d'avance comme mécanisme de validation de la récompense : si une prédiction utilise l'une de ces règles, la récompense est augmentée.

MultiHopKG [Lin et al., 2018] va plus loin et propose deux avancées de modélisation pour les méthodes basées sur le MHR. Premièrement, il aborde le problème des faits manquants dans le KG en adoptant un nouveau mécanisme d'attribution de la récompense. Étant donné que les KGs sont incomplets par nature [Min et al., 2013], l'agent peut arriver à une réponse correcte qui n'est pas présente dans les données d'entraînement, et donc ne pas recevoir de récompense pour cette prédiction. Au lieu d'utiliser une récompense binaire $\{0, 1\}$, un modèle d'embedding de graphes est utilisé pour estimer une récompense intermédiaire pour les prédictions n'existant pas dans les données d'entraînement. Pour chaque nouveau triplet (h, r, t) prédit par le modèle de MHR, si le triplet n'est pas présent dans les données d'entraînement, la fonction de score du modèle de KGE $f(h, r, t)$ est utilisée pour évaluer la plausibilité de la prédiction et attribuer la récompense en conséquence. Deuxièmement, le modèle améliore la capacité d'exploration de l'agent en ajoutant un masque sur les actions possibles pour chaque étape d'entraînement. Comme REINFORCE est un algorithme d'apprentissage basé sur la politique, l'agent peut être enclin à utiliser des chemins erronés mais gratifiants rencontrés précédemment dans la phase d'entraînement. Le masque déconnecte aléatoirement certaines relations sortantes à chaque étape, forçant l'exploration de chemins plus diversifiés.

RuleGuider [Lei et al., 2020a] propose une autre stratégie pour améliorer le mécanisme de mise en forme de la récompense de MultiHopKG en utilisant lui aussi des règles logiques. Comme avec PoLo, une méthode symbolique extrait d'abord les règles logiques, qui sont ensuite utilisées pour modifier la récompense de l'agent en fonction de la confiance dans le

type de règles logiques utilisées pour faire la prédiction.

Tous les modèles présentés précedemment (KGE, symboliques, MHR) prédisent les nouveaux liens sous forme de liste : pour une requête $(h, r, ?)$, ils proposent plusieurs nœuds t qui pourraient compléter ce triplet et les classent du plus au moins probable. Pour évaluer ces modèles, nous avons donc besoins de métriques particulier basés sur le rang.

2.2.4 Définition des métriques utilisées pour la tâche de prédiction de liens

La tâche de prédiction de liens requiert des **métriques pour mesurer la performance des modèles**. Ces métriques sont basées sur le rang auquel le modèle parvient à faire une prédiction correcte. En effet, lorsque l'on entraîne le modèle à compléter une requête $(h, r, ?)$, celui-ci propose une liste d'entités t qui pourraient correspondre au triplet d'origine (h, r, t). À partir de cette liste de réponses possibles, deux métriques basées sur le rang sont communément utilisées : le MRR et le Hit@k.

Le Mean Reciprocal Rank (MRR)

Le rang réciproque d'une réponse t à une requête $(h, r, ?)$ est l'inverse du rang de la première réponse correcte, soit $1/rang$. Si la bonne réponse est classée première de la liste, le rang réciproque vaut 1 ; si la bonne réponse est classé deuxième de la liste, le rang réciproque vaut $1/2$, etc. Le MRR correspondant à la moyenne des rangs réciproques est calculé pour un ensemble de requêtes Q comme suit :

$$MRR = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{rang_i} \tag{2.1}$$

avec $rang_i$ le rang de la réponse t correcte dans la liste.

Le Hit@k

Le Hit@k sert à mesurer la probabilité que la bonne réponse soit donnée à une position inférieure à k dans la liste des réponses t proposées par le modèle. En d'autres termes, un Hit@5 égal à 0,5 indique que le modèle a une chance sur deux de classer la bonne entité t dans les cinq premiers éléments de la liste de réponses possibles. Le Hit@k ne prend pas en compte la position de la bonne réponse si celle-ci est supérieure à k : l'erreur est considérée comme identique si le modèle classe la bonne réponse au rang $k + 1$ ou au rang $k + d$ avec $d >> 1$. C'est pour cette raison que certains préfèrent comparer les modèles en se basant sur le MRR. Pourtant, ces deux métriques sont complémentaires. Par exemple, si un modèle réalise deux prédictions et qu'il obtient un MRR de 0,5, il n'est pas possible

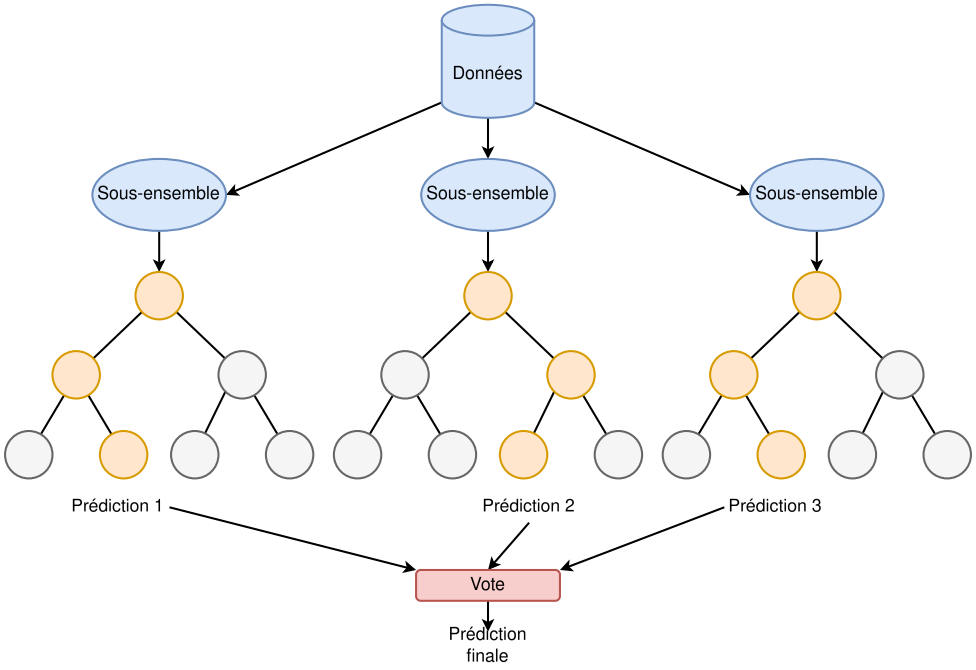

FIGURE 2.5 – Schéma représentant la façon dont les forêts aléatoires utilisent plusieurs sous-ensembles des données initiales pour entraîner les différents arbres de décision. La prédiction majoritaire parmi l'ensemble des arbres constitue la prédiction finale.

de déterminer le rang des bonnes réponses. D'après la formule 2.1, le modèle peut avoir classé les deux bonnes réponses au rang 2, ou une au premier rang et l'autre au dernier rang. Dans ce cas, regarder le Hit@5 permet de mieux comprendre : dans le premier cas, le Hit@5 sera égal à 1 et dans le second cas, il sera égal à 0,5.

2.3 Forêts aléatoires

Le terme **forêt aléatoire** (*Random forest*) désigne un algorithme d'apprentissage automatisé introduit en 2001 par Breiman [Breiman, 2001]. Ce type de modèles sont souvent considérés comme parmi les plus efficaces et les plus performants de la dernière décennie [Howard and Bowles, 2012, Varian, 2014]. Ces modèles, qui utilisent un ensemble d'arbres de décision, suivent une logique simple : diviser pour mieux régner. À partir des données initiales, des sous-ensembles sont constitués. Un arbre de décision est alors entraîné à partir de chacun de ces sous-ensembles puis chaque résultat est considéré pour former le résultat final, comme illustré en Figure 2.5. Les forêts aléatoires ont été utilisées dans presque tous les domaines liés à la biologie où l'IA a un intérêt. On peut citer l'écologie [Prasad et al., 2006], la chimie [Svetnik et al., 2003] ou encore la génomique [Díaz-Uriarte and Alvarez de Andrés, 2006]. De par son utilisation de nombreux arbres qui voteront pour fournir la prédiction finale, cette méthode dite "ensembliste" permet de limiter l'impact de potentiels biais dans les données ainsi que le phénomène d'overfitting [Tang et al., 2018].

Plus formellement, on définit **M** comme le nombre d'arbres dans la forêt et **N** le nombre d'exemples dans les données d'entraînement, chacune possédant p attributs. Pour chaque arbre :

- On tire aléatoirement un sous-ensemble d'exemples n_M appartenant à N,
- On tire $q < p$ attributs,
- On entraîne l'arbre de décision avec les données n_M et leurs attributs q en choisissant à chaque étape l'attribut de q qui réalise le meilleur partage des données selon un critère de segmentation qui diminue l'impureté des données, c'est-à-dire l'hétérogénéité des données à un nœud (peuvent être utilisées l'impureté de Gini [Weisstein, 2000] ou l'entropie de Shannon [Shannon, 1948]),
- Une fois un choix fait par chaque arbre, la classe majoritairement prédite est choisie comme réponse finale.

L'utilisation de forêts aléatoires permet un certain degré d'interprétation des résultats, grâce à des méthodes de quantification de l'importance des attributs. Rappelons qu'à chaque fois qu'un nœud d'un arbre est divisé en sous-ensembles, l'impureté de ces sous-ensembles tend à être inférieure à celle du nœud initial. La première mesure qui calcule l'importance des attributs est appelée *Mean Decrease Impurity (MDI)* [Breiman, 2002]. Pour chaque attribut dans l'abre de décision, on mesure à quel point elle a contribué à réduire l'impureté totale quand elle est utilisée pour séparer un nœud. En faisant la moyenne sur l'ensemble des arbres, on peut estimer dans quelle mesure chaque attribut a contribué à réduire l'impureté lorsqu'il est utilisé, et donc déterminer son importance pour réaliser la prédiction. La seconde mesure, *Mean Decrease Accuracy (MDA)* [Breiman, 2001], quantifie les différences de précision du modèle quand on altère les valeurs de chaque attribut. Après l'entraînement, on sélectionne une partie des données que le modèle n'a pas encore vues, on permute aléatoirement les valeurs d'un des attributs et on observe la différence en terme de précision du modèle avant et après permutation. Si la précision diminue fortement, alors cet attribut est important pour le modèle. On répète les perturbations pour chaque attribut, on a alors un classement des attributs par ordre d'importance pour le modèle.

2.4 Repositionnement de médicaments

Le développement de nouveaux médicaments est réputé pour être une entreprise coûteuse et chronophage. Bien que le sujet soit débattu, on estime que le coût de mise sur le marché d'un seul médicament est passé de 179 millions de dollars en 1970 à un montant stupéfiant de 2,558 millions de dollars en 2010, et le coût moyen de développement d'une nouvelle molécule a récemment été estimé entre 314 millions et 2,8 milliards de dollars [DiMasi et al., 2016, Wouters et al., 2020]. Exacerbant encore plus ce défi, une proportion significative (environ 80%) des médicaments n'atteint pas la phase cruciale des

essais cliniques de phase III [Morgan et al., 2018]. Dans ce contexte, pouvoir identifier un nouvel usage pour un médicament déjà existant et déjà testé en essais cliniques est particulièrement pertinent.

Le plus souvent, le repositionnement de médicaments se fait en plusieurs étapes. D'abord, il faut générer l'hypothèse de repositionnement, il faut ensuite évaluer le potentiel de la molécule lors d'essais pré-cliniques et enfin l'évaluer lors des essais cliniques. Un avantage du repositionnement est que, souvent, les molécules ont déjà validé la phase I des essais cliniques, qui évalue la toxicité du composé. C'est pour la phase de génération d'hypothèses que les méthodes computationnelles peuvent se montrer particulièrement efficaces. Dans leur revue de la littérature sur le sujet, S.Pushpakom [Pushpakom et al., 2019b] distingue six catégories d'approches computationnelles pour le repositionnement de médicaments :

- Signature Matching [Zhang and Gant, 2009] : l'idée est de comparer les effets d'une molécule avec ceux d'autres molécules dont on connaît l'indication,
- Molecular Docking [Dakshanamurthy et al., 2012] : le principe est d'analyser la structure d'une molécule pour prédire son effet sur une cible (souvent une protéine),
- Genetic Association [Grover et al., 2015] : il s'agit de rechercher les gènes responsables d'une maladie et de définir les moyens de les cibler,
- Pathway Mapping [Greene and Voight, 2016] : l'idée est d'étudier le fonctionnement de voies moléculaires liées à la maladie pour identifier des opportunités de repositionnement,
- Retrospective Clinical Analysis [Jensen et al., 2012] : le but est de réaliser une analyse poussée des données contenues dans les essais cliniques, les dossiers patients ou les données de surveillance médicale pour y trouver des possibilités de repositionnement,
- Novel Data Sources [Wei and Denny, 2015] : le principe est d'analyser les données génomiques, protéiques ou pharmacologiques contenues dans les sources de données et de connaissances disponibles en ligne, telles que DrugBank [Knox et al., 2010], Entrez Gene [Maglott et al., 2005], Reactome [Milacic et al., 2023], SIDER [Kuhn et al., 2016] et UniProt [Consortium, 2022].

Se plaçant dans la catégorie "Novel Data Sources", les KGs biomédicaux sont de plus en plus présents dans le domaine du repositionnement de médicaments depuis quelques années, notamment depuis 2017 avec la création de Hetionet [Himmelstein et al., 2017]. Se basant sur de nombreuses sources de données en ligne, les concepteurs d'Hetionet l'ont utilisé pour proposer des molécules à repositionner afin de traiter le tabagisme et l'épilepsie. Sur la base de l'analyse des chemins reliant les médicaments et les maladies dans le graphe, un modèle de classification a proposé le bubropion, initialement connu comme antidépresseur, comme une aide potentielle pour arrêter de fumer.

Suivant le même principe d'intégration de données disponibles en ligne, d'autres KGs ont été publiés dans les années suivantes. Parmi les plus utilisés, on peut citer :

— DRKG [Zhang et al., 2021], qui a été construit en ajoutant des données au graphe existant Hetionet, dans le but de trouver des opportunités de repositionnement contre le COVID-19,

— BioKG [Walsh et al., 2020], qui utilise en plus des données extraites à partir des résumés d'articles présents dans MEDLINE pour construire le graphe. Les auteurs ont montré son intérêt pour le repositionnement de molécules contre le COVID-19 et plusieurs formes de cancer,

— PharmKG [Whirl-Carrillo et al., 2021], qui a pour but de ne garder que des informations de la meilleure qualité possible, le rendant bien plus compact que les graphes cités précédemment. Les auteurs ont illustré son efficacité pour le repositionnement dans le cadre des maladies d'Alzheimer et de Parkinson.

Dans cette thèse, nous étudions comment ces méthodes d'IA peuvent être utilisées pour la tâche de prédiction de liens dans un KG, avec pour but final le repositionnement de médicaments. Ainsi, dans le chapitre 3, nous exposons la manière d'améliorer les résultats de modèles de multihop reasoning en pré-entraînant les embeddings représentant les relations et les entités du graphe.

Les résultats de ces travaux ont fait l'objet d'une publication au *Workshop on Knowledge-Based Compositional Generalization* de la conférence *International Joint Conference on Artificial Intelligence (IJCAI)* en 2023 [Drance et al., 2023]. Dans le chapitre 4, nous proposons d'augmenter les données de KGs biomédicaux via l'ajout de nouveaux types de relations. Nous analysons comment ces changements impactent les résultats de prédiction de liens d'un modèle de multihop reasoning. Enfin, dans le chapitre 5, nous présentons une nouvelle méthodologie basée sur les forêts aléatoires pour le repositionnement de médicaments. À partir des connexions existantes entre les médicaments, les maladies et leurs voisins, nous montrons que les forêts aléatoires sont un outil efficace pour la tâche de classification de liens appliquée au repositionnement de médicaments. Ce travail, actuellement en cours de relecture par le journal *Scientific Reports* est disponible sous la forme d'une pré-publication à l'adresse suivante : `https://dx.doi.org/10.2139/ssrn.4867527`.

Chapitre 3

Utilisation d'embeddings pré-entraînés pour l'entraînement de modèle de multi-hop reasoning

Sommaire

Dans ce chapitre, nous proposons de modifier le fonctionnement des modèles de MHR pour y inclure des embeddings pré-entraînés. Le pré-entraînement est une méthode qui a

fait ses preuves dans le domaine de l'IA [Devlin et al., 2019, Russakovsky et al., 2015], notamment dans le domaine du traitement automatique des langues où des embeddings sont construits pour représenter les mots du lexique. Malgré un fonctionnement similaire pour les modèles de MHR, où les embeddings sont utilisés pour représenter les nœuds et les relations, à notre connaissance, aucun travail n'avait été mené jusque-là sur la possibilité de pré-entraîner ces embeddings. En adoptant cette stratégie, nous proposons de tester les performances de deux modèles de MHR lors de l'utilisation d'embeddings générés par différents modèles de KGE, comme décrit dans la Figure 3.1.

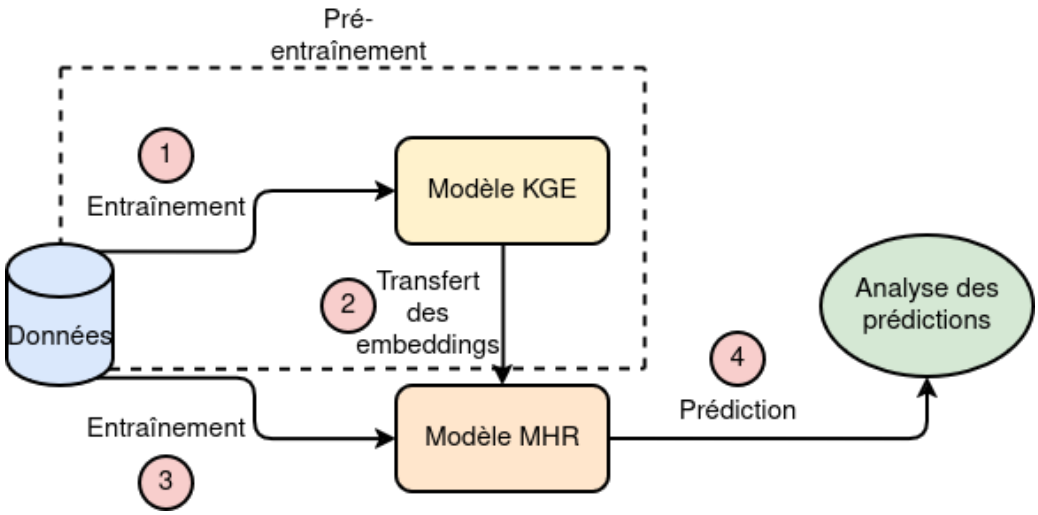

FIGURE 3.1 – Schéma explicatif de l'ajout d'embeddings pré-entraînés à un modèle de MHR. À partir du même KG, un modèle de KGE est d'abord entraîné pour transmettre les embeddings des nœuds et des relations au modèle de MHR.

3.1 Introduction

Les modèles de MHR ont démontré de bonnes performances prédictives et la capacité à générer des décisions explicables [Lv et al., 2021a]. Un modèle de raisonnement multihop comporte généralement deux étapes : 1) la construction d'une représentation précise des entités et des relations du KG ; et 2) l'utilisation de ces représentations pour explorer les chemins de raisonnement dans le KG qui justifient les liens nouvellement prédits.

Comme expliqué en section 2.2.3, les modèles de MHR peuvent utiliser des chemins erronés, mais qui mènent à des prédictions valides, comme illustré en Figure 3.2.

De plus, comme les graphes sont incomplets, certaines prédictions valides faites par ce type de modèles sont considérées comme fausses et ne sont donc pas récompensées. Par exemple, durant la phase d'entraînement dans un KG biomédical, le modèle prédit un lien entre un médicament et une maladie qui n'existe pas dans le KG. Cette donnée n'existe pas dans le graphe, elle est donc considérée fausse et indiquée comme fausse au modèle. En réalité, cette information est vraie, mais n'a pas encore été ajoutée au graphe, car l'essai

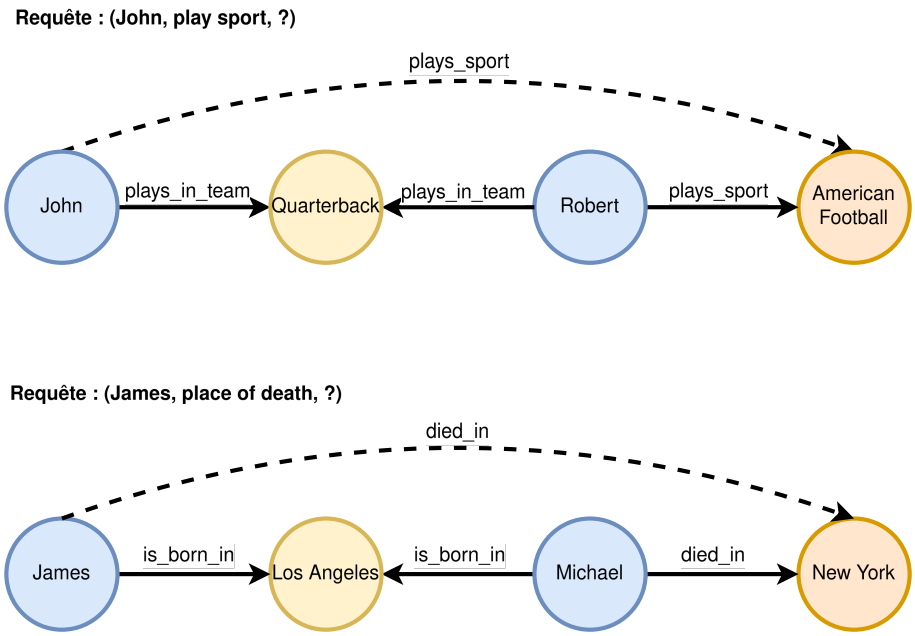

FIGURE 3.2 – Illustration d'un chemin erroné que le modèle peut utiliser pour faire une prédiction correcte. Dans le premier cas, il est possible de déduire que si John et Robert sont tous les deux *quarterbacks*, ils jouent tous les deux au football américain. Dans le second cas en revanche, se servir de la ville de naissance commune entre James et Michael pour déduire le triplet (James, place of death, New York) n'est pas un raisonnement valable, même si ce triplet existe bel et bien.

clinique relatif à cette molécule n'était pas terminé au moment de la construction du KG.

Dans le premier cas, une récompense est attribuée, mais ne devrait pas l'être, tandis que dans le second cas, elle n'est pas attribuée, mais devrait l'être. À cause de ces dysfonctionnements, certaines informations erronées sont propagées dans les embeddings des nœuds et des relations.

À notre connaissance, les modèles d'embeddings de KG (comme ConvE utilisé ici) sont les méthodes les plus efficaces pour la tâche de prédiction de liens. Les résultats à l'état de l'art sont obtenus en utilisant ces modèles ou en améliorant les modèles de KGE existants [Lu et al., 2022, Zhang et al., 2019c, Chen et al., 2021, Pan and Wang, 2021]. Dans ce chapitre, nous examinons comment l'utilisation d'embeddings pré-entraînés pour les entités et les relations du KG impacte les performances d'un modèle de MHR. L'intuition qui sous-tend l'utilisation d'embeddings pré-entraînés est que l'information contenue dans ces embeddings sera moins, voire pas du tout, modifiée pendant l'entraînement du modèle.

Dans la section 3.2, nous introduisons la notion de pré-entraînement et comment elle est déjà utilisée dans d'autres domaines du machine learning. Nous introduisons ensuite le modèle que nous avons utilisé en section 3.3 puis détaillons notre méthodologie dans la section 3.4. Les performances du modèle, analysées en termes de scores de prédiction de liens et de la qualité des chemins générés utilisés comme explications, sont présentées en section 3.5. Nos expérimentations ont été menées sur trois KGs généralistes de référence :

WN18RR [Dettmers et al., 2018], NELL-995 [Das et al., 2017] et FB15K-237 [Toutanova et al., 2015].

3.2 Utilisation de modèles pré-entraînés

Le pré-entraînement de modèles est récemment devenu une étape cruciale dans le développement de modèles prédictifs. Elle a été initialement utilisée pour la classification d'images, popularisée notamment par AlexNet [Krizhevsky et al., 2012]. L'idée sous-jacente au pré-entraînement est que l'apprentissage préalable d'une représentation générique permet au modèle d'acquérir des connaissances qui peuvent être transférées à d'autres tâches. Très souvent, le modèle est entraîné sur un dataset de grande taille et générique, puis affiné pour une tâche plus précise avec un dataset adapté à cette tâche, sur la base des poids acquis lors du pré-entraînement. De cette manière, les meilleurs résultats d'AlexNet ont été atteints sur le jeu de données ImageNet [Russakovsky et al., 2015] de 2012 en utilisant les poids d'un pré-entraînement sur les données de 2011. Cette méthode est très largement utilisée depuis lors par les modèles de classification d'images les plus performants, comme VGGNet [Simonyan and Zisserman, 2014], ResNet [He et al., 2016] ou encore DenseNet [Huang et al., 2017]. Aujourd'hui, certains modèles sont publiés avec des poids issus d'un pré-entraînement sur des images génériques (tel ImageNet), et seule une petite portion de leurs paramètres est modifiée pendant la phase d'affinement, correspondant souvent aux paramètres des dernières couches de neurones [Rebuffi et al., 2017].

Plus récemment, le pré-entraînement a été systématiquement utilisé pour développer de grands modèles de langues (Large Language Models ou LLMs) tels que BERT [Devlin et al., 2019], XLNet [Yang et al., 2019] ou encore les modèles GPT [Brown et al., 2020]. Dans le cas des LLMs, plusieurs poids coexistent lors de la phase d'entraînement : les poids des représentations vectorielles des mots du lexique (embeddings) et les paramètres du modèle. L'ensemble de ces poids est appris lors de la phase de pré-entraînement et donne lieu à deux approches pour affiner le modèle. La première consiste, comme pour la classification d'images, à entraîner à nouveau le modèle sur un jeu de données spécialisé afin de modifier légèrement les paramètres du modèle pour que ceux-ci soient plus adaptés à une tâche spécifique. La deuxième option consiste à ne pas toucher aux paramètres du modèle, mais à modifier quelque peu les embeddings des mots du lexique afin que leur représentation soit plus adaptée à un domaine particulier. C'est cette deuxième approche qui peut être directement mise en relation avec la prédiction de liens dans un KG, où les modèles de MHR possèdent leurs propres paramètres, en plus de construire des embeddings pour chaque nœud et chaque relation.

3.3 Fonctionnement de MultiHopKG

3.3.1 Environnement et états

Comme évoqué précédemment, le KG $\mathcal{G}$ représente l'environnement, tel que $\mathcal{G} \subset \mathcal{E} \times \mathcal{R} \times \mathcal{E}$ où $\mathcal{E}$ représente l'ensemble des entités dans $\mathcal{G}$ et $\mathcal{R}$ l'ensemble des relations dans $\mathcal{G}$. La tâche de prédiction de liens consiste à trouver l'ensemble des réponses possibles $e_o \in E_o$ pour une requête $(e_s, r_q, ?)$, où e_s est l'entité source et r_q est la relation de la requête, de telle sorte que chaque (e_s, r_q, e_o) soit un triplet manquant dans $\mathcal{G}$. Ces réponses sont sélectionnées après que l'agent se soit déplacé successivement d'un nœud à un autre, jusqu'à atteindre une cible plausible. L'état s_t de l'agent à l'étape t doit encoder l'entité source, la relation de requête et la position de l'agent e_t à cette étape. Ainsi, l'état courant s_t est défini par $s_t = (e_s, r_q, e_t) \in S$.

3.3.2 Espace d'actions

L'espace d'actions A_t à l'étape t comprend toutes les relations sortantes depuis le nœud courant e_t, tel que $A_t = \{(r_{t'}, e_{t'}) | e_s, r_q, e_t\}$, avec $r_{t'} \in \mathcal{R}$ et $e_{t'} \in \mathcal{E}$. À chaque étape, la sélection de l'action se fait en choisissant une relation sortante, en connaissant le type de $r_{t'}$ et le nœud suivant $e_{t'}$. Pour chaque nœud de départ e_s, la recherche est limitée à un certain nombre d'étapes T. Pour permettre à l'agent de rester à sa position courante s'il atteint une réponse plausible à l'étape $t < T$, une action supplémentaire "NO_OP" est ajoutée, correspondant à une relation qui boucle sur le nœud courant e_t.

3.3.3 Récompenses

La fonction de récompense R de base définit une récompense de 1 si l'agent atteint une entité cible correcte, et 0 sinon : $R = \mathbb{1}\{(e_s, r_q, e_o) \in \mathcal{G}\}$. Pour MultiHopKG, la récompense est modulée en utilisant la fonction de score $f(e_s, r_q, e_o)$ d'un modèle de KGE (ConvE, dans notre cas). L'idée est de donner une récompense de 1 si le triplet $(e_s, r_q, e_o) \in \mathcal{G}$, sinon la récompense est définie uniquement par la fonction de score du modèle de KGE comme suit : $R' = R + (1 - R) f(e_s, r_q, e_o)$.

3.3.4 Politique

Comme défini dans [Das et al., 2017], la politique utilise trois informations pour choisir l'action appropriée : (i) la position courante de l'agent e_t, (ii) la relation de requête r_q, et (iii) l'historique de toutes les actions précédentes de l'agent. Chaque entité et relation dans $\mathcal{G}$ se voit attribuer respectivement un embedding $e \in \mathbb{R}^d$ et $r \in \mathbb{R}^d$ et toutes les actions A_t à l'étape t sont représentées par $a_t = [r_{t'}; e_{t'}]$ où $[;]$ est la concaténation vectorielle de la relation choisie et de son nœud cible correspondant. L'historique $h_t \in \mathbb{R}^{2d}$ représente la

séquence des observations et actions passées effectuées jusqu'à l'étape t. Comme l'historique représente une séquence évoluant à chaque étape t, elle est encodée à l'aide d'un réseau de neurones récurrents type LSTM [Hochreiter and Schmidhuber, 1997]. Cette séquence de déplacements est définie comme suit :

$$h_0 = LSTM(0, [r_0; e_s]) \tag{3.1}$$

$$h_t = LSTM(h_{t-1}, a_{t-1}) \tag{3.2}$$

où l'équation 3.1 est utilisée à l'étape t_0 pour encoder l'action initiale dans l'historique, utilisant r_0 qui correspond à la représentation vectorielle de la position de départ de l'agent. L'équation 3.2 est utilisée aux étapes suivantes. En fonction de cet historique h_t, de l'état courant s_t et de la relation de requête r_q, la politique π est définie par un réseau de neurones de type perceptron (*feed-forward*) [Gallant et al., 1990] à deux couches, qui produit les distributions de probabilité de toutes les actions possibles A_t à l'étape t comme suit :

$$\pi_\theta(a_t|s_t) = \sigma(A_t \times W_2 ReLU(W_1[h_t; e_t; r_q])) \tag{3.3}$$

avec σ la fonction *softmax* qui transforme le vecteur de la couche de sortie du perceptron en probabilités.

Les équations 3.2 et 3.3 sont répétées pour chaque étape de transition jusqu'à ce que le nombre maximum d'étapes T soit atteint. Les paramètres que le modèle peut apprendre sont les paramètres du LSTM, les paramètres du réseau feed-forward W_1, W_2 et les embeddings des entités e et des relations r.

3.3.5 Entraînement

Pour entraîner la politique et trouver les meilleurs paramètres θ, l'algorithme REINFORCE est utilisé pour maximiser la récompense attendue $\mathbb{E}$ comme suit :

$$J(\theta) = \mathbb{E}_{(e_s, r_q, e_o) \in \mathcal{G}}[\mathbb{E}_{a_1, a_2, \ldots, a_T \sim \pi_\theta}[R(S_T|e_s, r_q)]] \tag{3.4}$$

en utilisant le gradient stochastique suivant :

$$\nabla_\theta J(\theta) \approx \nabla_\theta \sum_t R(s_T|e_s, r_q) log \pi_\theta(a_t|s_t) \tag{3.5}$$

Pendant l'entraînement, MultiHopKG propose d'ajouter un mécanisme de masque des actions pour masquer aléatoirement certaines relations sortantes à l'étape d'échantillonnage de REINFORCE. L'objectif de cette stratégie est d'encourager l'agent à rechercher des chemins plus diversifiés, car l'exploration dans l'équation 3.4 peut être biaisée vers des chemins erronés conduisant à des réponses correctes, comme illustré en Figure 3.2.

3.3.6 Embeddings pré-entraînés

L'entraînement d'un modèle de MHR peut être divisé en deux parties distinctes :

— L'apprentissage des embeddings des entités e et des relations r en tant que représentations vectorielles, décrivant l'information portée par chaque nœud et chaque arête dans le KG. Cette tâche est également effectuée dans les modèles de KGE.

— L'apprentissage des paramètres du LSTM dans l'équation (3.2) et des paramètres feed-forward W_1 et W_2 dans l'équation (3.3), étant donné l'information de l'historique de l'agent h_t, de l'entité courante e_t et de la relation de requête r_q. Cette tâche est spécifique au fonctionnement des modèles de MHR.

Pendant le processus d'entraînement, les représentations des entités e et des relations r sont mises à jour en utilisant les équations (3.4) et (3.5), qui dépendent des récompenses obtenues aux étapes T. Cette procédure d'entraînement implique que, dans certains cas, les représentations sont mises à jour en utilisant de fausses informations. Premièrement, lorsqu'un chemin erroné est utilisé pour atteindre un bon nœud cible, les embeddings des nœuds et des relations sont mis à jour en utilisant une récompense positive basée sur un chemin totalement illogique. Deuxièmement, pour les prédictions correctes qui ne sont pas présentes dans les données d'entraînement en raison de l'incomplétude des KGs, les représentations sont mises à jour en utilisant un signal de récompense erroné causé par un faux négatif.

3.4 Méthodes

Dans cette section, nous présentons dans un premier temps les résultats de l'ajout d'embeddings pré-entraînés au modèle MINERVA [Das et al., 2017]. Ensuite, les datasets et la configuration expérimentale du modèle MultiHopKG [Lin et al., 2018] sont présentés. Enfin, nous introduisons les différentes mesures qui ont été utilisées pour rendre compte de l'impact des modifications apportées sur les explications fournies par le modèle.

3.4.1 Expérimentations avec MINERVA

Les premiers résultats expérimentaux que nous ayons obtenus en utilisant des embeddings pré-entraînés sont ceux générés avec le modèle MINERVA [Das et al., 2017]. Comme expliqué en section 2.2.3, MINERVA est le premier modèle de MHR qui a été conçu pour prédire des liens dans un KG. Le fonctionnement est exactement le même que celui de MultiHopKG, mais sans ajustement de la récompense par un modèle de KGE. Pour tester l'intuition selon laquelle l'utilisation d'embeddings pré-entraînés permettrait d'améliorer les performances du modèle de MHR, nous créons les embeddings des entités et des relations en utilisant le modèle ConvE. Ces embeddings ont ensuite été utilisés par MINERVA. Le Tableau 3.1 présente les résultats de MINERVA sur les métriques habituelles avec et

TABLE 3.1 – Résultats sur le modèle de MHR MINERVA de l'utilisation d'embeddings pré-entraînés. Les meilleurs résultats apparaissent en gras.

Méthode / Dataset	**WN18RR**			**NELL-995**			**FB15K-237**		
	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR
MINERVA	0,38	0,52	0,43	0,43	0,56	0,48	0,20	0,37	0,26
MINERVA (ConvE)	**0,41**	**0,53**	**0,45**	**0,47**	**0,59**	**0,51**	**0,24**	**0,41**	**0,29**

sans l'utilisation des embeddings entraînés par ConvE. On constate que pour les trois graphes et pour les trois métriques présentées, l'ajout d'une étape de pré-entraînement des embeddings des entités et des relations permet une augmentation de la qualité des résultats. Cela confirme l'intuition que la stratégie basée sur le pré-entraînement des embeddings peut améliorer les résultats de modèles de MHR. Toutefois, malgré ces observations positives concernant l'utilisation de MINERVA, nous nous sommes orientés vers une investigation plus détaillée du pré-entraînement appliqué au modèle MultiHopKG. En effet, MINERVA n'est plus un modèle à l'état de l'art.

3.4.2 Adaptation des modèles de MHR

Pour pallier les problèmes identifiés dans la phase d'entraînement, nous proposons d'utiliser des embeddings pré-entraînés pour les entités et les relations dans le KG, générés par le modèle de KGE ConvE. Nous utilisons le modèle ConvE pour sa facilité d'utilisation : il est le meilleur modèle de KGE par rapport à son nombre de paramètres et le temps nécessaire à son entraînement [Ali et al., 2021a]. Ce choix est discuté en section 3.6.

Par conséquent, les embeddings e, r dans les équations (3.1), (3.3), (3.4) et (3.5) sont remplacés par les représentations pré-entraînées correspondantes e^+, r^+ correspondant aux paramètres pré-entraînés apprenables, ou bien par e^-, r^- correspondant aux paramètres pré-entraînés non-apprenables. En effet, une fois les embeddings appris, il est possible de les figer en les excluant des poids du modèle, ou de les optimiser en les laissant comme poids entraînables. L'objectif de cette approche est d'atténuer l'impact des fausses informations sur les embeddings des entités et des relations pendant le processus d'entraînement.

Jeux de données

Pour comparer notre approche à d'autres méthodes basées sur le MHR, nous évaluons ses performances sur les trois KGs de référence suivants : WN18RR [Dettmers et al., 2018], Nell-995 [Das et al., 2017] et FB15K-237 [Toutanova et al., 2015]. Leurs caractéristiques sont données dans le Tableau 2.1. Pour chaque triplet existant (h, r, t) dans $\mathcal{G}$, le triplet inverse (t, r^{-1}, h) a été ajouté afin de permettre à l'agent de se déplacer dans les deux sens entre h et t. À chaque étape, le nombre maximum de relations sortantes est limité à un nombre n. Les n voisins les mieux classés ont été sélectionnés en utilisant leurs scores PageRank [Page et al., 1999]. Cette méthode de sélection des voisins est utilisée pour

éviter de saturer la mémoire graphique disponible pendant l'entraînement du modèle avec l'utilisation de l'équation (3.3). En effet, les KGs ne sont pas homogènes, certains nœuds étant connectés à un grand nombre de voisins (pouvant causer cette saturation) et d'autres à très peu. Sélectionner un nombre fixe n de voisins permet une utilisation maîtrisée de la mémoire du GPU. Pour MINERVA comme pour MultiHopKG, cette sélection se fait grâce aux scores PageRank, qui permettent de calculer l'importance de chaque nœud dans un graphe. PageRank attribue à chaque nœud un score d'importance reflétant la facilité avec laquelle l'information peut circuler vers ou depuis ce nœud. Intuitivement, conserver les n voisins ayant les scores PageRank les plus élevés revient à sélectionner des voisins à partir desquels il sera simple de trouver une action pertinente.

Hyperparamètres

Comme indiqué précédemment, nous utilisons les embeddings pré-entraînés générés par ConvE comme défini dans [Lin et al., 2018]. Nous conservons une taille de 200 pour les embeddings des nœuds et des relations. Nous avons effectué une optimisation des hyperparamètres du type *grid-search* pour le *dropout* des embeddings [0 ; 0,5], des couches feed-forward du perceptron [0 ; 0,5], le taux de dropout des actions [0,1 ; 0,9] et le taux d'apprentissage [0,001 ; 0,003]. Le grid-search permet une recherche exhaustive des meilleurs hyperparamètres, en testant toutes les combinaisons possibles définies par les hyperparamètres sélectionnés et leur gamme de valeurs. Le dropout est un mécanisme qui va masquer certaines informations avec une probabilité p, il est utilisé ici de trois manières différentes :

- Dropout des embeddings $p \in [0; 0,5]$: masque certains éléments des embeddings des nœuds et des relations avec une probabilité p. Il sert à éviter que les modèles reposent uniquement sur certaines features à l'intérieur de ces embeddings.
- Dropout du perceptron $p \in [0; 0,5]$: masque (désactive) certains des neurones dans le perceptron avec une probabilité p. Il sert alors à limiter l'overfitting en désactivant aléatoirement certains des neurones du réseau.
- Dropout des actions $p \in [0; 0,9]$: masque certaines actions à chaque déplacement de l'agent. Dans ce cas, il sert à inciter l'agent à avoir un comportement plus exploratoire.

Nous avons mené l'expérimentation en utilisant les embeddings pré-entraînés comme paramètres apprenables e^+, ou bien en les maintenant figés e^-.

Modèles de KGE

Dans un premier temps, nous testons la validité de l'implémentation des modèles de KGE utilisés pour le calcul de la récompense (ComplEx [Trouillon et al., 2016] et ConvE [Dettmers et al., 2018]). En effet, les résultats peuvent légèrement varier entre

différentes implémentations des modèles de KGE [Ali et al., 2019]. Dans notre cas, l'intuition est que des embeddings générés par un modèle de KGE performant permettront au modèle de MHR d'obtenir de meilleurs résultats. Il faut donc s'assurer que nous utilisons le meilleur modèle de KGE possible. En utilisant la librairie Python Pykeen [Ali et al., 2021b], nous entraînons ces deux modèles pour la tâche de prédiction de liens sur les trois KGs WN18RR, Nell-995 et FB15K-237. Les résultats avec Pykeen sont exactement les mêmes que ceux obtenus par les versions de ComplEx et ConvE qui servent à calculer la récompense dans MultiHopKG. Les expériences sont donc menées en utilisant les embeddings générés par ces modèles, n'augmentant pas la taille finale de MutlihopKG. Le processus d'entraînement et de test des modèles de KGE a été réalisé sur les mêmes ensembles de données utilisés pour l'entraînement et le test du modèle de MHR (entraînement, validation, test).

3.4.3 Analyse des chemins

Une caractéristique clé des modèles de MHR est leur capacité à fournir des chemins de "raisonnement" pour expliquer chaque prédiction. On s'attend donc à ce que les méthodes basées sur le MHR soient plus fiables que les méthodes basées sur le KGE, car l'explication de chaque prédiction peut être facilement analysée. Cependant, il a été démontré que ces chemins sont souvent irrationnels, c'est-à-dire que certains chemins n'ont pas de sens alors qu'ils mènent à une réponse correcte, ou qu'ils sont incomplets [Lv et al., 2021b]. Nous avons donc choisi d'analyser la manière dont l'ajout d'embeddings pré-entraînés sur MultiHopKG modifie les chemins explicatifs associés à chaque prédiction. Nous avons ainsi mesuré : (i) le nombre de chemins uniques utilisés comme explications, (ii) la diversité de ces chemins, et (iii) le nombre de triplets de l'ensemble test pour lesquels le modèle a trouvé une réponse.

Chemins uniques Les modèles de MHR donnent souvent plus d'un chemin unique explicatif pour justifier chaque prédiction, ce qui peut conduire à de trop nombreuses explications diminuant l'interprétabilité des résultats. Pour chaque chemin $p = (h, r, t) \leftarrow (h, r_1, e_1) \wedge (e_1, r_2, e_2) \wedge (e_2, r_3, t)$, la règle logique correspondante est $l = (r_1 \wedge r_2 \wedge r_3)$. Nous avons alors calculé le nombre de règles logiques uniques utilisées sur l'ensemble des prédictions, ce qui représente la capacité du modèle à généraliser les chemins rencontrés lors de l'entraînement sur les entités de l'ensemble de test.

Diversité des règles Le nombre de règles logiques uniques fournit une mesure de la variété des explications, mais il ne tient pas compte de la variété de chaque chemin de raisonnement. Certains chemins peuvent être très redondants parce qu'ils utilisent le même type de relation pour plusieurs étapes. La diversité dans le nombre de types de relations employés par chaque règle logique se traduit par des explications plus informatives, car elle correspond à des règles avec une plus grande variété de types de nœuds et de relations

TABLE 3.2 – Comparaison des performances de MultiHopKG avec et sans pré-entraînement des embeddings. PT-MultiHopKG$^-$ correspond aux embeddings figés pendant l'entraînement du modèle de MHR, PT-MultiHopKG$^+$ correspond aux embeddings non figés pendant l'entraînement du modèle MHR. Les meilleurs résultats sont en gras.

Méthode / Dataset	**WN18RR**			**NELL-995**			**FB15K-237**		
	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR
MultiHopKG (ConvE)	41,4	51,7	44,8	65,6	84,4	72,7	32,7	56,4	40,7
MultiHopKG (ComplEx)	42,5	52,6	46,1	64,4	81,6	71,2	**32,9**	54,4	39,3
PT-MultiHopKG$^+$	43,0	52,4	46,0	67,3	84,6	74,0	32,7	**58,1**	**41,2**
PT-MultiHopKG$^-$	**44,1**	**52,8**	**46,8**	**68,1**	**85,6**	**74,9**	31,1	57,2	39,9

sémantiquement distinctes. Nous quantifions cette diversité comme suit :

$$d = \frac{\sum_l Unique(l)}{|l|} \tag{3.6}$$

où $Unique$ représente le nombre de relations uniques $r \in \mathcal{R}$ trouvées dans les règles l.

Rappel Nous utilisons également le score de rappel des explications tel que défini dans [Lv et al., 2021b] pour quantifier le nombre de triplets dans l'ensemble de test pouvant être prédits par le modèle. Un rappel élevé indique que le modèle prédit et explique avec précision une plus grande proportion des triplets de test. Le rappel est défini comme suit :

$$PR = \frac{\sum_{(h,r,t)\in T^{test}} Cnt(h,r,t)}{|T^{test}|} \tag{3.7}$$

où $Cnt(h,r,t) = 1$ si le modèle trouve au moins un chemin de h à t, 0 sinon.

3.5 Résultats

3.5.1 Comparaison des modèles

Le Tableau 3.2 présente la comparaison des performances entre MultiHopKG et notre modèle avec embeddings pré-entraînés, PT-MultiHopKG. Pour les trois KGs, l'ajout d'embeddings pré-entraînés améliore les performances prédictives du modèle. Pour WN18RR et NELL-995, l'utilisation d'embeddings figés fournit les meilleurs résultats (PT-MultiHopKG$^-$), indiquant que les représentations construites par ConvE sont plus efficaces que l'utilisation d'embeddings en tant que paramètres du modèle. Pour NELL-995, l'utilisation d'embeddings pré-entraînés améliore les performances dans les deux configurations. Pour FB15K-237, les meilleures performances sont obtenues en affinant les embeddings (PT-MultiHopKG$^+$). Bien que WN18RR et NELL-995 contiennent plus de nœuds uniques, FB15K-237 est un graphe plus complexe car il contient plus de triplets et chaque nœud est beaucoup plus connecté à ses voisins (degré médian de 14). C'est l'une

TABLE 3.3 – Analyse des chemins explicatifs donnés par MultiHohKG avec et sans pré-entraînement des embeddings.

	Chemins uniques		**Diversité**		**Rappel**	
Datasets	MultiHopKG	PT-MultiHopKG	MultiHopKG	PT-MultiHopKG	MultiHopKG	PT-MultiHopKG
WN18RR	**1 918**	1 790	2,76	**2,78**	0,62	**0,63**
NELL-995	29 091	**29 396**	2,94	**2,96**	0,66	**0,67**
FB15K-237	82 426	**82 992**	2,94	**2,95**	0,74	**0,76**

des raisons possibles des résultats différents obtenus pour ce KG par rapport à WN18RR et NELL-995.

Le Tableau 3.3 montre les résultats de l'analyse des chemins obtenus pour MultiHopKG avec et sans embeddings pré-entraînés. Sauf pour WN18RR, l'ajout d'embeddings pré-entraînés augmente le nombre de chemins uniques qui servent d'explications pour le modèle. Cela suggère que l'amélioration des performances n'est pas uniquement attribuable à la capacité du modèle à prédire plus de faits en utilisant les mêmes règles, mais plutôt à l'utilisation de règles différentes. En ce qui concerne la diversité et le rappel des chemins, les embeddings pré-entraînés améliorent les résultats pour tous les KGs. Premièrement, cela indique que les relations dans les explications utilisées sont généralement plus diversifiées, c'est-à-dire que le modèle est capable de réutiliser un plus grand nombre de chemins différents observés lors de l'entraînement lorsqu'il prédit de nouveaux liens. Deuxièmement, les valeurs de rappel indiquent que le modèle est capable d'identifier au moins un chemin pour un plus grand nombre de triplets dans l'ensemble de test. Dans le cas de FB15K-237 par exemple, notre modèle a été en mesure de fournir une prédiction et un chemin de raisonnement pour 409 triplets tests supplémentaires (correspondant au passage d'un rappel de 0,74 à 0,76 dans le Tableau 3.3).

3.5.2 Étude d'ablation et variations du modèle

Utilisation de ComplEx au lieu de ConvE

Tous les résultats présentés concernent l'utilisation de ConvE pour générer les embeddings pré-entraînés. Nous avons cependant réalisé les mêmes expériences en remplaçant ConvE par ComplEx pour générer ces embeddings. Nos résultats ont montré que l'utilisation de ConvE est systématiquement la meilleure option et conduit aux meilleurs résultats. Quels que soient les hyperparamètres ou la configuration, c'est-à-dire avec ou sans ajustement de la récompense et avec ou sans embeddings figés, utiliser ComplEx à la place de ConvE dégrade les résultats de MultiHopKG.

Ajustement de la récompense

L'objectif dans l'utilisation d'embeddings pré-entraînés est d'avoir des informations sur chaque entité et relation contenues dans les embeddings avant l'entraînement du modèle

TABLE 3.4 – Comparaison des performances (MRR) de l'ajustement de la récompense et de l'impact des embeddings pré-entraînés sur le modèle. MultiHopKG -RS correspond au modèle sans ajustement de la récompense, MultiHopKG +RS correspond au modèle avec ajustement de la récompense. PT-MultiHopKG$^+$ -RS et PT-MultiHopKG$^-$ -RS correspondent respectivement aux embeddings pré-entraînés non figés et aux embeddings pré-entraînés figés sans ajustement de la récompense. Le pourcentage d'amélioration pour chaque méthode est indiqué entre parenthèses.

Méthode / Dataset	WN18RR	NELL-995	FB15K-237
MultiHopKG -RS	46,2	72,2	32,4
MultiHopKG +RS	44,8 (-3%)	72,7 (+0.5%)	40,7 (+25%)
PT-MultiHopKG$^+$ -RS	48,1 (+4%)	73,3 (+1.5%)	36.2 (+12%)
PT-MultiHopKG$^-$ -RS	49,0 (+6%)	71,4(-1%)	35,2 (+9%)

de MHR. Les mécanismes d'ajustement de la récompense (*reward shaping*) sont également une façon d'utiliser des connaissances préalables ou externes pour aider le modèle de MHR lors de l'entraînement. Comme indiqué dans [Lin et al., 2018], l'ajustement de la récompense améliore les performances sur FB15K-237 et NELL-995, mais les diminue pour WN18RR. Nous comparons l'efficacité de l'ajustement de la récompense avec celui du pré-entraînement en excluant le mécanisme d'ajustement de la récompense. Le Tableau 3.4 montre les résultats obtenus pour MultiHopKG avec et sans mécanisme d'ajustement de la récompense en utilisant ou non des embeddings pré-entraînés. Premièrement, l'utilisation d'embeddings pré-entraînés améliore les performances du modèle de base pour les trois KGs, mettant en évidence que ces embeddings remplissent leur rôle en fournissant des connaissances préalables lors du processus d'entraînement du modèle de MHR. Deuxièmement, l'ajustement de la récompense a un impact plus fort sur FB15K-237 mais un impact négatif sur WN18RR, là où l'utilisation d'embeddings pré-entraînés a systématiquement un impact positif sur le MRR.

Comparaison avec l'état de l'art

Un résumé de tous les résultats et une comparaison avec d'autres méthodes symboliques et de MHR sont présentés dans le Tableau 3.5. Nous comparons les résultats de notre travail avec trois autres modèles de prédictions de liens explicables : RuleGuider et MINERVA, qui sont deux modèles de MHR, et AnyBURL. Dans un premier temps, on constate que le modèle AnyBURL est celui qui performe le moins bien des modèles explicables pour les datasets WN18RR et NELL-995, MINERVA étant celui proposant les moins bons résultats pour FB15k-237. RuleGuider et MultiHopKG sont les meilleurs modèles de MHR actuels pour la tâche de prédiction de liens, RuleGuider étant légèrement plus performant que MultiHopKG. Cependant, l'ajout de l'étape de pré-entraînement des embeddings permet à MultiHopKG d'obtenir un meilleur MRR que RuleGuider sur les trois datasets. RuleGuider est une amélioration du modèle MultiHopKG, introduisant un mécanisme de

TABLE 3.5 – Résumé de nos résultats et comparaison avec d'autres méthodes de MHR et symbolique. Les meilleurs résultats sont en gras. †Indique les résultats obtenus par [Lei et al., 2020a].

Méthode / Dataset	**WN18RR**			**NELL-995**			**FB15K-237**		
	H@1	H@10	MRR	H@1	H@10	MRR	H@1	H@10	MRR
AnyBURL†	42,9	53,7	-	44,0	57,0	-	26,9	52,0	-
MINERVA†	41,3	51,3	44,8	66,3	83,1	72,5	21,7	45,6	29,3
MultiHopKG (ConvE)	41,4	51,7	44,8	65,6	84,4	72,7	32,7	56,4	40,7
MultiHopKG (ComplEx)	42,5	52,6	46,1	64,4	81,6	71,2	**32,9**	54,4	39,3
RuleGuider (ConvE)†	42,2	53,6	46,0	66,0	85,1	73,1	31,6	57,4	40,8
RuleGuider (ComplEx)†	44,3	55,5	48,0	66,4	**85,9**	73,6	31,3	56,4	39,5
PT-MultiHopKG$^+$	43,0	52,4	46,0	67,3	84,6	74,0	32,7	**58,1**	**41,2**
PT-MultiHopKG$^-$	44,1	52,8	46,8	**68,1**	85,6	**74,9**	31,1	57,2	39,9
PT-MultiHopKG$^+$ -RS	44,8	54,4	48,1	66,8	83,5	73,3	28,5	51,6	36,2
PT-MultiHopKG$^-$ -RS	**45,5**	**55,8**	**49,0**	63,9	84,6	71,4	27,2	51,2	35,2

modification de la récompense en fonction de règles logiques pré-établies. C'est encore une fois une tentative d'apporter de la connaissance supplémentaire au modèle lors de son apprentissage. Nos résultats suggèrent que le pré-entraînement des embeddings est une meilleure manière d'améliorer les performances d'un modèle de MHR.

3.6 Discussion

Ces dernières années, le pré-entraînement a été largement étudié comme moyen d'améliorer les performances prédictives d'un modèle. De la classification d'images, où le pré-entraînement du modèle est utilisé pour améliorer la détection des caractéristiques [Russakovsky et al., 2015], au traitement automatique des langues, où les embeddings de mots pré-entraînés sont affinés pour correspondre à un domaine spécifique [Lee et al., 2019], le pré-entraînement permet d'utiliser des connaissances préalables ou d'adapter le modèle à une tâche spécifique. Les résultats expérimentaux présentés dans la section 3.5.1 montrent que les embeddings pré-entraînés ont un impact positif sur MINERVA et MultiHopKG. Comme dans d'autres domaines de l'apprentissage automatique, les paramètres du modèle doivent être affinés lorsqu'ils sont appliqués à des tâches ou datasets plus complexes. Par conséquent, bien que les embeddings puissent être utilisés tels quels pour les graphes WN18RR et NELL-995, ils doivent être inclus dans les paramètres du modèle pour FB15K-237, qui est significativement plus complexe.

Par rapport à l'ajustement de la récompense, la sous-section 3.5.2 a montré que les embeddings pré-entraînés amélioraient systématiquement les performances du modèle, tandis que le mécanisme d'ajustement de la récompense nuisait aux performances pour le KG WN18RR sans que l'on puisse en connaître la cause. Pour FB15K-237, l'ajustement de la récompense a un impact plus important que les embeddings pré-entraînés, confirmant

la complexité de la tâche de prédiction de liens sur ce dataset. C'est le KG qui bénéficie le plus de l'ajout de connaissances préalables, mais les meilleurs résultats ont été obtenus en utilisant les deux méthodes simultanément.

Enfin, lors des tests avec le modèle de KGE ComplEx pour construire les embeddings, nous observons des résultats inférieurs à ceux obtenus avec ConvE. Il est important de noter que pour la prédiction de liens, ConvE produit systématiquement de meilleurs résultats que ComplEx, comme l'ont montré Ali *et al.* [Ali et al., 2021a].

Étant donné que l'atout des modèles de MHR réside dans leur explicabilité, l'analyse des chemins explicatifs a montré un intérêt à l'utilisation d'embeddings pré-entraînés en ce qui concerne la qualité de ces explications. Le fait de disposer d'un plus grand nombre de triplets de l'ensemble de test pour lesquels une prédiction peut être faite est une caractéristique importante. Cependant, les modèles de MHR ne sont pas toujours capables de faire une prédiction s'ils ne trouvent pas de chemins jugés pertinents pour la réaliser. Dans le cas d'une application concrète de ces méthodes, cela peut laisser une grande partie des requêtes sans réponse. Par exemple pour FB15K-237, un maximum de 76% des triplets tests ont obtenu une prédiction, laissant un quart des données sans aucun résultat.

De plus, chaque prédiction peut être étayée par des centaines de chemins de raisonnement alternatifs. Le problème est que ces chemins sont souvent redondants dans le type de relations qu'ils emploient, traversant diverses entités tout en utilisant exactement le même type de relation. Des chemins de raisonnement diversifiés permettent des explications plus sophistiquées et exhaustives. Dans notre cas, nous avons montré que le pré-entraînement des embeddings avait un impact positif sur la diversité des explications, réduisant donc la redondance des chemins utilisés pour faire les prédictions.

Il serait cependant intéressant de tester d'autres approches pour construire ces embeddings. En effet, les modèles de KGE ne sont pas les seuls à pouvoir construire des représentations pour les nœuds et les arêtes d'un graphe. Récemment, les méthodes de *Graph Neural Network* (GNN, réseaux de neurones en graphe) [Zhang et al., 2019a] sont de plus en plus utilisés, avec des performances qui peuvent dépasser les modèles de KGE suivant l'architecture utilisée.

Par ailleurs, nos résultats permettent de mettre en évidence deux points faibles des modèles de MHR. Nous les abordons ci-dessous.

La quantité d'explications

Le nombre d'explications différentes est incroyablement élevé, ce qui rend particulièrement compliquée l'interprétation des résultats. S'il est avantageux d'avoir une explication pour chaque prédiction, cela peut s'avérer problématique lorsque l'on constate que pour une prédiction, il existe plusieurs centaines d'explications possibles. Le nombre de chemins

uniques est donc une métrique qui reflète un phénomène à double tranchant : en théorie, les embeddings pré-entraînés permettent d'avoir des explications plus variées mais en pratique, cela peut rendre la tâche d'interprétation des résultats encore plus délicate.

Ressources nécessaires à leur utilisation

La deuxième problématique que nous avons rencontrée en travaillant avec des modèles de MHR est l'impossibilité d'utiliser ces modèles sur des graphes de grande taille. Nous ne sommes pas parvenus à entraîner MultiHopKG sur un KG biomédical, et cela malgré l'utilisation de GPUs A100 dotés de 80GB de mémoire vidéo. Cette limitation est inhérente à l'architecture de ces modèles qui utilisent l'apprentissage par renforcement. Chaque arête dans le graphe est potentiellement une action qu'il faut évaluer à chaque étape. Même en appliquant des restrictions sur le nombre d'actions à considérer, il nous a été impossible de les utiliser sur des graphes aussi volumineux que BioKG ou Hetionet.

3.7 Conclusion

Ce chapitre s'est intéressé à l'impact des embeddings pré-entraînés sur le modèle de MHR MultiHopKG pour la tâche de prédiction de liens. Nous avons établi que l'ajout d'embeddings pré-entraînés améliorait les performances de ce modèle sur les trois KGs WN18RR, NELL-995 et FB15K-237. En plus d'améliorer les résultats de la tâche de prédiction de liens, les expérimentations ont montré que les embeddings pré-entraînés permettaient à notre modèle de prédire de nouveaux liens pour un plus grand nombre de requêtes, augmentant donc la quantité de prédictions pouvant être faites par le modèle. Nous avons également observé que l'usage des embeddings pré-entraînés est une méthode valide pour fournir des connaissances préalables lors du processus d'entraînement d'un modèle de MHR. De plus, ces résultats suggèrent que les méthodes basées sur le KGE sont appropriées pour construire ces embeddings pré-entraînés.

Nous avons travaillé dans ce chapitre à la modification d'un modèle de MHR dans le but d'améliorer ces résultats. Dans le chapitre suivant, nous proposons au contraire d'analyser comment la modification des données a un impact sur la tâche de prédiction de liens par un modèle de MHR.

Chapitre 4

Augmentation des données dans un graphe de connaissances

Sommaire

Les KGs sont construits dans le but de refléter notre savoir dans un domaine particulier, ou de regrouper le savoir de plusieurs domaines voisins. Quel que soit le domaine, ils sont construits en utilisant des bases de connaissances ou des évènements dont la véracité a été validée. De par leur vocation à représenter les informations comme elles existent réellement, il est difficile de manipuler les arêtes d'un KG sans altérer les connaissances qu'elles représentent. C'est pour cette raison qu'il y a assez peu de recherches qui sont conduites sur la modification de ces graphes. Ajouter un lien entre deux nœuds peut conduire à ajouter une information qui est fausse dans la réalité. C'est particulièrement vrai pour les KGs qui représentent des connaissances dites générales, comme FB15K-237 :

les nœuds y représentent des personnes, des villes, des films, des professions, des équipes sportives, des morceaux de musique, des langues, etc. Dans ce graphe, il existe 1345 types de relations différentes. Il n'est donc pas évident de savoir comment et entre quelles entités ajouter de nouvelles relations. Dans ce cas, il est plus sûr d'ajouter l'information au cas par cas pour s'assurer qu'aucun lien erroné ne serait ajouté au graphe. Cependant, l'utilisation de modèles de multihop reasoning a besoin, par définition, d'un grand nombre de liens dans le graphe pour pouvoir fonctionner : cela représente plus de données d'entraînement et plus de possibilité d'explications. Ajouter des données sous forme de nouvelles relations dans un KG pourrait donc être une solution pour améliorer le fonctionnement des modèles de MHR. Comme illustré en Figure 4.1 qui reprend les types de nœuds et de relations de Hetionet, si deux molécules présentent les mêmes effets secondaires, cela ne garantit en aucun cas que ce sont des molécules similaires. Au contraire, si une molécule agit sur un gêne qui intervient dans une voie de signalisation métabolique, on peut affirmer que la molécule interagit avec cette voie métabolique.

Dans ce chapitre, nous étudions la possibilité d'ajouter de nouveaux liens dans un KG biomédical et mesurons l'impact de ces ajouts sur la capacité d'un modèle de MHR à prédire de nouveaux liens entre médicaments et cibles biologiques.

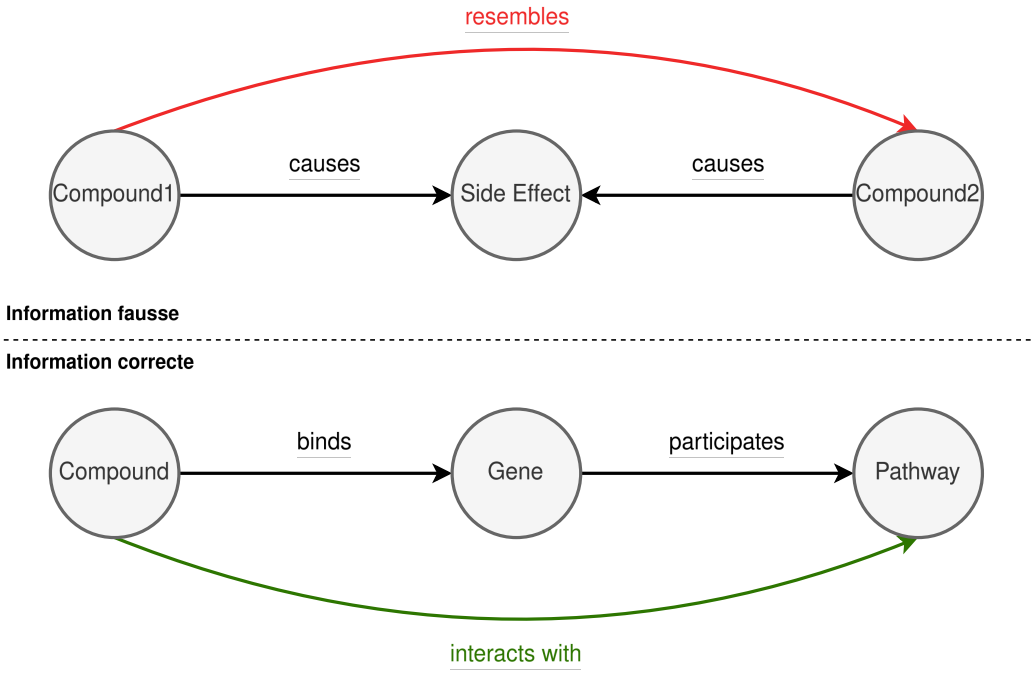

FIGURE 4.1 – Exemple d'ajout de liens dans un graphe en se basant sur une suite de faits connus. Dans le premier cas, l'ajout du lien pourrait avoir du sens dans certains cas mais pas dans d'autres, nous savons par exemple que plusieurs molécules très différentes peuvent impacter négativement le foie. Dans le second cas, l'ajout du nouveau lien ne peut pas introduire de fausses informations.

4.1 L'augmentation de données appliquée aux graphes

Pour améliorer la quantité et la qualité des données d'entraînement, l'augmentation des données est proposée comme une approche efficace pour accroître les données d'entrée fournies en modifiant légèrement les instances de données existantes ou en générant des instances fictives, dites synthétiques, à partir des instances existantes. L'importance de l'augmentation des données a été largement reconnue ces dernières années, notamment dans les domaines de la vision par ordinateur [Shorten and Khoshgoftaar, 2019] et du traitement automatique des langues [Devlin et al., 2019].

Outre les données conventionnelles d'images ou de textes, les données structurées en graphe sont connues pour être bien plus complexes [Bronstein et al., 2021], avec des modalités d'information hétérogènes (différents types de nœuds et d'arêtes) et des propriétés multiples, ce qui entraîne un espace de conception plus vaste ainsi que des défis supplémentaires pour l'augmentation des données.

Le but de l'augmentation des données au sein d'un graphe (*graph data augmentation* ou GraphDA) est de trouver une fonction de transformation pour générer des graphes augmentés, afin d'enrichir ou améliorer les informations des graphes initiaux. En général, l'objectif ultime du GraphDA est d'améliorer les performances de modèles sur les tâches d'apprentissage en aval. Étant donné que les graphes se composent généralement de multiples modalités d'information, les techniques GraphDA peuvent être naturellement divisées en trois catégories basées sur la modalité d'augmentation [Zhou et al., 2022], qui sont : (i) les techniques modifiant la structure, (ii) les techniques modifiant les features, et (iii) les techniques modifiant les labels. Les techniques modifiant les features concernent les graphes pour lesquels les nœuds sont porteurs d'informations autres que simplement leur label, ce qui n'est pas le cas dans les KGs utilisés ici. De même, les techniques concernant les labels sont utilisées pour les tâches de classification de graphes, donc sans intérêt pour la prédiction de liens.

Dans le cadre de la prédiction de liens dans un KG, nous nous intéressons ici aux méthodes modifiant la structure du KG afin d'ajouter, modifier et/ou supprimer de l'information en passant par la manipulation des arêtes ou des nœuds.

Il existe plusieurs moyens de modifier la structure du KG [Ding et al., 2022] :

— La **perturbation des arêtes**, qui permet d'ajouter ou de supprimer aléatoirement des arêtes [Veličković et al., 2019] ;

— Le **recâblage du graphe**, qui consiste aussi à ajouter ou supprimer des arêtes dans le KG, mais sans que le processus soit aléatoire. En passant par l'optimisation de la fonction de perte du modèle, une partie du modèle apprend les arêtes qui peuvent être supprimées ou ajoutées ;

— Les **méthodes de diffusion**, qui permettent d'ajouter des liens en se basant sur la topologie du KG [Topping et al., 2021]. Ces méthodes sont utilisées pour ajouter

des arêtes entre des nœuds qui ne sont pas directement connectés, en utilisant des liens existants intermédiaires ;

— L'**échantillonnage de graphe**, qui permet de sélectionner des sous-ensembles d'un KG à partir d'un échantillonneur qui sélectionne un sous-graphe en se basant sur des informations sur les arêtes ou les nœuds que l'on désire garder dans le sous-graphe [Hamilton et al., 2017]. Comme pour la perturbation des arêtes, il est aussi possible d'ajouter [Gilmer et al., 2017] ou de supprimer des nœuds, dans le but d'augmenter ou de diminuer le nombre de voisins de certains nœuds du KG.

L'ensemble de ces méthodes sont pensées pour augmenter les données de graphes simples non hétérogènes. La tâche d'augmentation des données sur des graphes hétérogènes est particulièrement complexe car il faut préserver l'information sémantique inhérente au KG. Un nombre limité de travaux [Wang et al., 2021, Yu et al., 2024a] se sont intéressés à l'augmentation des données pour les KGs, et uniquement dans le but d'utiliser des modèles de type HGNN (Heterogeneous Graph Neural Network) [Zhang et al., 2019b], qui sont une famille de GNN et donc des modèles basés uniquement sur des réseaux neuronaux fonctionnant comme des boîtes noires. À notre connaissance, à date, il n'existe pas de travaux s'intéressant à l'augmentation des donnés d'un graphe hétérogène pour l'utilisation de modèles de MHR.

4.1.1 Intérêt de l'augmentation de données pour le repositionnement de médicaments

Notre hypothèse est que l'augmentation de données appliquée aux KGs biomédicaux pourrait présenter trois avantages majeurs pour l'entraînement de modèles de MHR : (i) augmenter le nombre de triplets pouvant servir à entraîner le modèle, (ii) relier entre eux des nœuds qui n'appartiennent pas aux mêmes bases de connaissances, et enfin (iii) augmenter le nombre de chemins existant entre les nœuds du graphe.

Augmentation de la taille du set d'entraînement

Les modèles de MHR, comme la plupart des modèles de machine learning, ont besoin d'une quantité importante de données d'entraînement pour obtenir de bonnes performances. Comme dit précédemment, le jeu de données d'entraînement d'un modèle de MHR consiste en une collections de triplets (h, r, t) pour lesquels on va masquer l'entité t dans le but d'entraîner le modèle à retrouver le triplet d'origine. Dans ce contexte, l'ajout de liens dans un KG est synonyme d'ajout de nouveaux triplets, c'est-à-dire de données pouvant servir à l'entraînement des modèles. Dans le cas du graphe BioKG par exemple, il existe approximativement 67 000 liens entre des entités de type Drug et Disease et seulement 28 000 liens entre des entités de type Drug et Protein. En théorie, le modèle

pourra donc apprendre plus facilement à partir des triplets $(Drug, treats, Disease)$ qu'à partir des triplets $(Drug, interacts, Protein)$, car ceux-ci sont plus nombreux dans le dataset d'entraînement. Le problème est que le repositionnement de médicaments n'est pas seulement basé sur des liens entre des médicaments et des maladies, il peut également se faire en prédisant des interactions entre des médicaments et des protéines ou entre des médicaments et des gènes [Olayan et al., 2017]. Dans ce cas, l'augmentation du nombre d'arêtes entre les entités Drug et Protein peut être bénéfique pour améliorer les prédictions du modèle pour ce type de triplets.

Connecter entre eux des nœuds provenant de différentes sources de données

Le deuxième avantage à l'ajout de nouveaux liens dans un KG biomédical est de connecter entre elles des entités qui ne proviennent pas des mêmes sources de données. Comme souligné dans le cadre de la construction du graphe BioKG [Walsh et al., 2020], chaque base de connaissances utilisée pour construire les KGs contient des informations spécifiques, c'est-à-dire que chacune de ces bases contient une partie des informations nécessaires à la construction d'un KG. Prenons par exemple les bases de connaissances REACTOME [Milacic et al., 2023] et KEGG [Kanehisa et al., 2016] : REACTOME contient des informations sur les liens entre des protéines et les gènes dont elles sont issues mais pas d'informations sur les médicaments. KEGG contient des liens entre des médicaments et les protéines qu'ils ciblent mais pas d'informations sur les gènes. Dans ce cas, lors de l'intégration de ces bases de connaissances dans un KG biomédical, il est possible de former des triplets $(Médicament, relation1, Protéine)$ à partir de KEGG et des triplets $(Protéine, relation2, Gène)$ à partir de REACTOME. En utilisant les informations disponibles, il n'est pas possible de connecter dans le graphe un médicament à un gène, alors que cette interaction existe par le biais de la protéine exprimée par ce gène et ciblée par le médicament, comme cela est décrit dans le graphe Oregano via le lien $is\_affecting$ ou dans Hetionet via le lien *binds*. Bien que la nature de l'interaction entre le médicament et le gène exprimant la protéine ne soit pas connue, elle existe (via la protéine) et pourrait être ajoutée dans le cadre de l'augmentation des données.

Augmentation du nombre de chemins utilisables par le modèle

Un autre avantage à l'augmentation des données dans le contexte de la prédiction de liens à l'aide de modèles de MHR est la possibilité de faciliter la constitution des chemins de raisonnement utilisés pour faire des prédictions. En effet, ces modèles se basent sur les chemins qui existent entre deux entités dans le graphe pour justifier d'un potentiel nouveau lien. De ce fait, on peut considérer que pour un triplet donné (h, r, t), chaque chemin constitué d'une suite de plusieurs relations reliant h et t est un exemple supplémentaire qui peut être utilisé pour entraîner le modèle. De la même manière que l'augmentation du

nombre de triplets dans le graphe accroît la taille du dataset d'entraînement, l'augmentation du nombre de chemins possibles entre deux entités dans le graphe accroît le nombre de possibilités de découvrir des chemins pouvant former un nouveau triplet.

4.2 Méthode

Dans le cadre de l'utilisation de modèles de MHR pour la prédiction de liens, le but est d'ajouter des arêtes aux KGs de manière à améliorer le MRR du modèle sur le lien d'intérêt (généralement le lien entre les nœuds Drug et Disease dans le cadre du repositionnement de médicaments) ainsi que la quantité ou la qualité des explications fournies par le modèle. Comme défini dans le chapitre sur les embeddings pré-entrainés en section 3.4.3, nous mesurons la diversité d des explications fournies par le modèle d'après l'équation (3.6). Ici, nous travaillons avec le modèle SQUIRE [Bai et al., 2022], qui était le modèle de MHR obtenant les meilleurs résultats pour la tâche de prédiction de liens au moment de la réalisation de ce travail, en plus d'être le modèle fournissant les explications les plus complètes. Par ailleurs, SQUIRE est un modèle beaucoup plus facile à entraîner que les autres modèles de MHR sur des KGs volumineux, comme c'est le cas pour les graphes biomédicaux. Comme démontré dans [Bai et al., 2022], dans le cas des modèles de MHR basés sur l'apprentissage par renforcement, le temps d'entraînement et les ressources nécessaires augmentent proportionnellement à la taille du KG, ce qui n'est pas le cas pour SQUIRE. Il constitue donc une solution aux problèmes de ressources évoqués au chapitre précédent. L'intuition est que de nouveaux liens pourraient être ajoutés aux KGs afin de générer plus facilement des chemins entre les nœuds Drug et Disease, qui sont les explications justifiant les prédictions du modèle. Un des problèmes qui empêchent les modèles de MHR d'utiliser toute l'information présente dans un KG est le nombre de déplacements qu'ils peuvent effectuer. Pour tous les modèles de MHR [Bai et al., 2022, Das et al., 2017, Liu et al., 2021, Lin et al., 2018], le nombre de déplacements que l'agent peut faire est limité à trois, créant des explications qui contiennent quatre nœuds au maximum : le nœud source, deux nœuds intermédiaires et le nœud cible.

Dans le cas du repositionnement de médicaments, le nœud source est toujours un nœud de type Drug et le nœud cible de type Disease. Ces deux contraintes, le nombre de déplacements et les nœuds de départ et d'arrivée fixes, contraignent le modèle à ne passer que par certaines relations dans les KGs. Prenons l'exemple d'Hetionet [Himmelstein et al., 2017], d'après les contraintes énoncées ci-avant, quatre types de nœuds ne peuvent pas être utilisés par le modèle : Pathway, Molecular Function, Biological Process et Cellular Component. En effet, si la prédiction part d'un nœud Compound, le modèle va utiliser un premier déplacement pour atteindre un nœud Gene, un deuxième pour atteindre un nœud Pathway et le troisième déplacement reviendra sur un nœud de type Gene ; les trois déplacements sont ainsi utilisés, sans que le modèle puisse atteindre une maladie.

De la même manière, ces contraintes restreignent le type de chemins qui peuvent être utilisés si l'on veut passer par certains types de nœuds. Par exemple dans Oregano [Boudin et al., 2023], les nœuds ATC Code, Activity, Effect, Indications et Side Effects ne peuvent être atteints que si l'on repasse par un nœud Drug pour finir sur un nœud Disease. Ces contraintes sur le fonctionnement des modèles de MHR limitent donc la quantité d'informations que le modèle peut utiliser, tant pour apprendre que pour générer des explications. Dans ce cas, l'augmentation des données dans les graphes a pour but de faciliter le déplacement de l'agent entre tous les types de nœuds afin que chaque type de nœud soit connecté à tous les autres types de nœuds. Pour ce faire, il faut s'assurer de ne pas modifier le graphe d'une manière qui pourrait ajouter de fausses informations. Basé sur des échanges avec un praticien hospitalier du service de biochimie et biologie moléculaire du CHU de Bordeaux (Dr. Joris Guyon), nous avons pu déterminer les liens qui pouvaient être ajoutés dans les graphes Oregano, Hetionet et BioKG.

4.2.1 Présentation du modèle de prédiction de liens SQUIRE

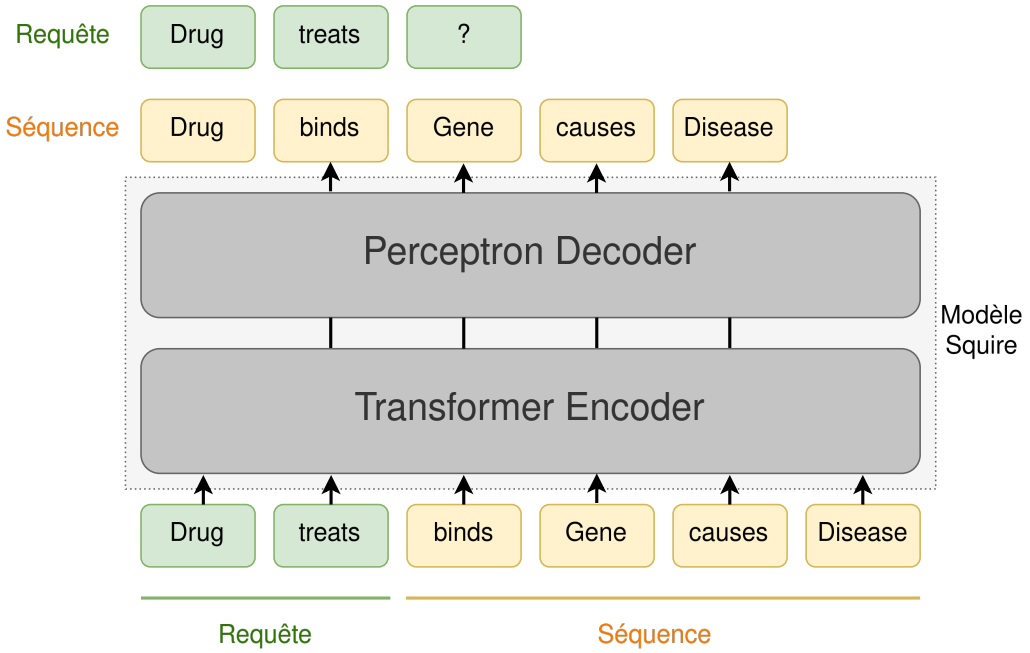

FIGURE 4.2 – Schéma illustrant le fonctionnement du modèle SQUIRE. Contrairement aux modèles de MHR utilisés précédemment qui se basent sur l'apprentissage par renforcement, la requête est traitée ici comme le début d'une séquence qu'il faut compléter en se basant sur les données du graphe.

SQUIRE est un modèle de MHR qui ne se base pas sur l'apprentissage par renforcement, contrairement aux modèles de MHR utilisés précédemment. SQUIRE se base sur l'utilisation d'un transformer [Vaswani, 2017] pour effectuer la tâche de prédiction de liens. La requête $(h, r, ?)$ est encodée par un transformer qui traduit les nœuds et les relations en embeddings, puis décodée par un perceptron multi-couches [Gallant et al., 1990] utilisé pour prédire la

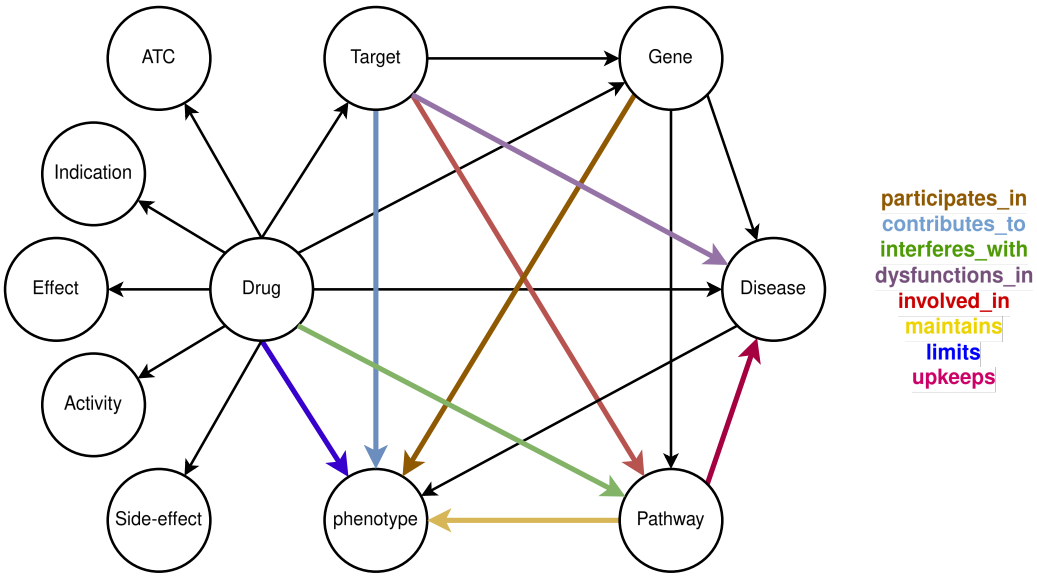

FIGURE 4.3 – Schéma illustrant les types de relations qui ont été ajoutées au graphe Oregano. Les nouvelles relations sont représentées en couleur, tandis que les relations en noir sont celles qui existaient dans le KG initialement.

prochaine action, c'est-à-dire le prochain nœud à placer dans la séquence. Cette opération se répète jusqu'à obtenir une séquence qui contient le nœud de départ, le nœud d'arrivée et un ou des nœuds intermédiaires. Cette séquence est utilisée pour justifier la prédiction de liens, à la manière des autres modèles de MHR. La Figure 4.2 illustre le fonctionnement de SQUIRE. Cette approche du problème de prédiction de liens peut donc être assimilée à une méthode de traitement automatique des langues, où le modèle apprendrait à compléter une séquence de mots. SQUIRE a donc besoin de séquences lors de sa phase d'entraînement et celles-ci sont générées selon les deux manières suivantes :

— AnyBURL est utilisé pour découvrir des règles de Horn qui ont un score de confiance élevé, comme décrit dans la section 2.2.2. Ces règles sont ensuite instanciées : chaque triplet du dataset d'entraînement (h, r, t) est associé à plusieurs séquences de nœuds et de relations en suivant les règles de Horn trouvées précédemment ;

— Si aucune des règles de Horn trouvées par AnyBURL ne peut générer de séquences pour un triplet du dataset d'entraînement, des séquences sont quand même générées en utilisant un algorithme de parcours aléatoire [Bundy and Wallen, 1984].

À la fin de ce processus, chaque triplet du dataset d'entraînement est associé à différentes séquences, l'ensemble constituant les données d'entraînement. Un avantage significatif de SQUIRE comparativement aux autres modèles de MHR est son faible temps d'entraînement : il est 4 à 7 fois plus rapide que MINERVA et MultiHopKG sur les datasets WN18RR et FB15K-237. Cette amélioration est particulièrement importante dans notre cas, où nous utilisons des KGs biomédicaux qui sont bien plus grands que ces deux graphes avant même l'augmentation de données.

4.2.2 Augmentation des données dans Oregano

Dans le cas du KG Oregano, huit nouveaux types de relations ont été ajoutées dans le KG comme synthétisé dans le Tableau 4.1, qui indique également le nombre de triplets créés grâce à ces nouvelles relations. L'idée principale est de favoriser les transitions entre les nœuds Drug, Protein, Gene, Pathway, Phenotype et Disease, comme illustré sur la Figure 4.3. Le but est de déterminer quels liens déjà présents dans le KG peuvent être utilisés pour en créer de nouveaux. Il faut que ces combinaisons de liens existants ait un sens biologique, mais aussi que les labels donnés aux nouveaux liens ne soient pas sources de mauvaise interprétation au moment de l'analyse des résultats. Chaque nouveau type de relation est décrit comme suit :

Limits C'est la nouvelle relation entre les nœuds Drug et Phenotype. Elle représente la combinaison des relations *treats* et *has_phenotype*. Si un nœud médicament est lié à une maladie et que cette maladie exprime un phénotype particulier, alors on peut dire que le but de ce médicament est de limiter ce phénotype résultant de la maladie.

Interferes_with C'est la nouvelle relation entre les nœuds Drug et Pathway. Elle est créée si une combinaison de relations *is_affecting* et *acts_within* existe. Si un médicament est connu pour avoir un effet sur un gène et que ce gène est une composante d'une voie biologique, alors on peut créer ce lien qui décrit l'interaction du médicament avec une partie de cette voie biologique.

Participates_in Cette relation entre un gène et un phénotype est créée s'il existe une combinaison des relations *causes_condition* et *has_phenotype*. Elle exprime le fait que si un gène est connu comme responsable d'une maladie et que cette maladie présente un phénotype particulier, alors ce gène participe à l'expression de ce phénotype.

Involved_in Cette nouvelle relation est construite entre les nœuds Protein et Pathway. S'il existe une combinaison des relations *gene_product_of* et *acts_within* alors un tel lien est ajouté. Il décrit qu'une protéine exprimée par un gène appartenant à une voie biologique peut être classifiée comme intervenant dans cette voie biologique.

Dysfunctions_in Cette relation est ajoutée entre un nœud Protein et un nœud Disease grâce à la combinaison des relations *gene_product_of* et *causes_condition*. Elle représente le fait qu'une protéine n'est plus fonctionnelle en présence de la maladie à laquelle elle est liée. Le lien est ajouté si la protéine est reliée à son gène codant et que ce gène est responsable d'une maladie.

TABLE 4.1 – Relations et nombre de triplets correspondants ajoutés au KG Oregano grâce à l'augmentation de données.

Nouvelle relation	Composition	#Triplets
Limits	treats + has_phenotype	198 061
Interferes_with	is_affecting + acts_within	14 751
Participates_in	causes_condition + has_phenotype	155 589
Involved_in	gene_product_of + acts_within	405 866
Dysfunctions_in	gene_product_of + causes_condition	165 349
Upkeeps	causes_condition + acts_within	10 288
Maintains	upkeeps + has_phenotype	217 803
Contributes_to	involved_in + maintains	100 000
Relations ajoutées : 8		Triplets ajoutés : 1 267 707

Upkeeps Cette relation est créée entre des nœuds Pathway et Disease, s'il existe une relation *causes_condition* entre un gène et une maladie et que ce gène est aussi lié à une voie biologique par la relation *acts_within*. Dans ce cas, on peut dire que cette voie biologique, via le gène problématique qu'il contient, participe au développement de la maladie.

Maintains Cette relation est créée si une relation *upkeeps* et une relation *has_phenotype* existent entre des nœuds Pathway et Phenotype. Elle décrit le fait que si une voie biologique, par un gène qu'elle contient, entretient une maladie et que cette maladie est caractérisée par un phénotype particulier, alors cette voie biologique maintient l'expression de ce phénotype.

Contributes_to Cette relation connecte un nœud Protein et un nœud Phenotype. Elle correspond à la combinaison des relations *involved_in* et *maintains*. Si une protéine fait partie d'une voie biologique qui participe à l'expression du phénotype d'une maladie, alors on peut dire que cette protéine contribue à ce phénotype. Cette relation est un peu particulière car elle implique deux des nouvelles relations pré-citées. Comme plus de 3 millions de liens auraient pu être construits pour cette relation, nous avons décidé de fixer une limite de 100 000 liens. Ce choix est discuté en section 4.4.

4.2.3 Augmentation des données dans Hetionet

Dans le cas d'Hetionet (Tableau 4.2), dix nouveaux types de relations ont été ajoutés au KG. Le but pour ce graphe est principalement de connecter les nœuds Compound et Disease aux nœuds Pathway, Molecular Function, Biological Process et Cellular Component. Comme évoqué précédemment, ces quatre types de nœuds ne pouvaient pas être utilisés par les modèles de MHR pour la tâche de prédiction de liens car ils sont trop éloignés des nœuds Compound et Disease pour être parcourus lors des trois déplacements de l'agent du modèle. Ici, les nouvelles relations sont nommées uniquement en fonction de la composition

TABLE 4.2 – Relations et nombre de triplets correspondants ajoutés au KG Hetionet grâce à l'augmentation de données.

Nouvelle relation	Composition	#Triplets
CbGpPW	CbG + GpPW	71 653
CbGpMF	CbG + GpMF	75 511
CbGpBP	CbG + GpBP	68 523
CbGpCC	CbG + GpCC	38 458
CtDlA	CtD + DlA	23 953
DaGDpS	DaG + DpS	71 191
DaGpPW	DaG + GpPW	57 273
DaGpMF	DaG + GpMF	43 871
DaGpBP	DaG + GpBP	83 112
DaGpCC	DaG + GpCC	26 213
Relations ajoutées : 10		Triplets ajoutés : 559 758

de relations utilisées pour les former. Elles n'ont donc pas de sens biologique dans leur appellation, mais sont ajoutées car il est pertinent de créer ces nouvelles connexions au sein du graphe. L'explication de chaque nouvelle relation est fournie ci-après.

CbGpPW, CbGpMF, CbGpBP, CbGpCC Ces relations connectent un nœud de type Compound à, respectivement, un nœud Pathway, Molecular Function, Biological Process et Cellular Component. Dans tous les cas, ces nouvelles relations sont construites en passant par un nœud Gene.

DaGpPW, DaGpMF, DaGpBP, DaGpCC Ces relations associent un nœud de type Disease à, respectivement, un nœud Pathway, Molecular Function, Biological Process et Cellular Component. Là encore, ces nouvelles relations sont établies via un nœud Gene.

CtDlA En passant par les relations CtD connectant un médicament (Compound) et une maladie (Disease), et DlA, connectant une maladie et sa localisation anatomique (Anatomy), on construit la nouvelle relation CtDlA qui associe un nœud Compound et un nœud Anatomy.

DaGDpS Cette relation exprime une connexion entre un gène et un symptôme d'une maladie. Si une maladie est causée par un gène et que cette maladie présente un symptôme, alors ce gène peut être relié au symptôme correspondant.

4.2.4 Augmentation des données dans BioKG

BioKG comporte le plus petit nombre de types de nœuds différents, avec seulement six types. Globalement, il est facile pour l'agent de se déplacer dans ce KG puisque les nœuds Pathway, Protein et Disease sont déjà en lien avec quatre des autres types de nœuds, et

TABLE 4.3 – Relations et nombre de triplets correspondants ajoutés au KG BioKG grâce à l'augmentation de données.

Nouvelle relation	Composition	#Triplets
DT_MC	drug_target + member_of_complex	152 025
DDA_DGD	drug_disease_association + disease_genetic_disorder	16 120
MC_PDA	member_of_complex + protein_disease_association	305 850
RGD_MC	related_genetic_disorder + member_of_complex	31 141
RGD_PPA	related_genetic_disorder + protein_pathway_association	25 099
Relations ajoutées : 5		Triplets ajoutés : 530 235

Drug avec trois d'entre eux. Cependant, les nœuds de type Complex et Genetic Disorder ne sont reliés qu'à deux autres types de nœuds. Le but de l'augmentation de données pour BioKG est donc majoritairement de mieux connecter les nœuds de types Complex et Genetic Disorder au reste du KG. Au total, cinq nouveaux types de relations ont été ajoutées (Tableau 4.3).

DT_MC Ce nouveau lien relie un médicament à un complexe protéique et est créé en combinant les liens *drug_target* entre un médicament et une protéine ainsi que *member_of_complex* entre une protéine et un complexe.

DDA_DGD En combinant les liens *drug_disease_association* entre les nœuds Drug et Disease et *disease_genetic_disorder* entre les nœuds Disease et Genetic Disorder, on crée un nouveau lien entre des médicaments et des troubles génétiques.

MC_PDA Ce lien relie un nœud Complex à un nœud Disease dans le cas où une protéine appartenant au complexe protéique (*member_of_complex*) est connue comme liée à la maladie (*protein_disease_association*).

RGD_MC Ce lien est la composition des liens *related_genetic_disorder* et *member_of_complex* pour connecter les nœuds Genetic Disorder et Complex par l'intermédiaire d'un nœud Protein commun.

RGD_PPA Cette relation relie un nœud Genetic Disorder et un nœud Pathway par l'intermédiaire d'une protéine en commun.

4.3 Résultats

Pour quantifier l'impact de l'augmentation des données sur les résultats du modèle de MHR SQUIRE, nous mesurons deux types de performances : la capacité du modèle à prédire correctement un triplet par le biais du MRR et la capacité du modèle à proposer des explications riches par le biais de leur diversité. Il est important de regarder ces métriques

TABLE 4.4 – MRR du modèle SQUIRE avant et après augmentation des données, sur chacun des KGs biomédicaux considérés. Chaque colonne présente le MRR sur l'ensemble du KG concerné, Dr-Di pour le lien Drug-Drug, Dr-P pour le lien Drug-Protein, Dr-G pour le lien Drug-Gene. NA signifie qu'aucun lien n'existe entre les deux types de nœuds présentés. Les meilleurs résultats apparaissent en gras.

KG	**Oregano**				**Hetionet**				**BioKG**			
	Global	Dr-Di	Dr-P	Dr-G	Global	Dr-Di	Dr-P	Dr-G	Global	Dr-Di	Dr-P	Dr-G
Avant	0,42	0,60	**0,33**	0,25	**0,18**	0,30	NA	0,31	0,18	0,065	0,17	NA
Après	**0,54**	**0,63**	0,27	**0,32**	0,16	**0,42**	NA	**0,35**	**0,27**	**0,093**	**0,26**	NA

TABLE 4.5 – Diversité des explications du modèle SQUIRE avant et après augmentation des données, sur chacun des KGs considérés. Les colonnes "Global" présentent la diversité sur l'ensemble du graphe, Dr-Di pour le lien Drug-Drug, Dr-P pour le lien Drug-Protein, Dr-G pour le lien Drug-Gene. NA signifie que le lien entre les deux types de nœuds n'existe pas. Les meilleurs résultats sont en gras.

KG	**Oregano**				**Hetionet**				**BioKG**			
	Global	Dr-Di	Dr-P	Dr-G	Global	Dr-Di	Dr-P	Dr-G	Global	Dr-Di	Dr-P	Dr-G
Avant	2	2	1,3	0	**2,5**	0	NA	**3**	1	0	0	NA
Après	**2,9**	**3**	**2**	**3**	2,1	0	NA	0	**1,4**	0	0	NA

spécifiquement pour des liens qui étaient déjà présents dans le KG avant augmentation des données. En effet, les nouveaux liens, qui ont été ajoutés pour faciliter l'entraînement et les prédictions, ne sont pas présents dans les bases de connaissances biomédicales d'origine et ne correspondent donc pas à des connaissances vérifiées par la communauté scientifique. Par conséquent, notre objectif n'est pas d'obtenir de bons résultats prédictifs pour ces liens. De plus, le MRR global pour la tâche de prédiction de liens sur l'ensemble du KG peut être biaisé si le modèle obtient un MRR particulièrement haut sur un de ces nouveaux liens. Nous avons donc choisi de calculer le MRR et la diversité des explications sur l'ensemble du graphe, mais surtout sur des liens spécifiques : le lien entre les médicaments et les maladies et le lien entre les médicaments et les protéines/gènes. Ce choix s'explique par le fait que la tâche de repositionnement de médicaments passe systématiquement par un de ces deux types de liens. Ce sont donc les performances du modèle sur ces liens que nous souhaitons maximiser.

4.3.1 Résultats sur le MRR

Le Tableau 4.4 présente les résultats du MRR pour la tâche de prédiction de liens sur les trois KGs biomédicaux, avant et après augmentation des données. Concernant le MRR global, on observe que l'ajout de nouveaux liens génère de meilleurs résultats pour Oregano et BioKG. Cette augmentation ne reflète cependant pas forcément un meilleur MRR pour les liens déjà présents dans le graphe avant augmentation. En ce qui concernent les liens Drug-Disease, qui sont les liens les plus intéressants pour la tâche de repositionnement de médicaments, on constate que le MRR est systématiquement plus élevé après augmentation

du KG : plus 3 points de pourcentage pour Oregano, 12 pour Hetionet et 3 pour BioKG. Ces résultat montrent également que BioKG est beaucoup moins adapté que les deux autres graphes pour la tâche de repositionnement de médicaments lorsque l'on utilise le modèle SQUIRE. Concernant le lien Drug-Protein pour lequel seuls Oregano et BioKG sont concernés, l'augmentation de données permet une amélioration du MRR uniquement pour BioKG. Enfin, pour le lien Drug-Gene, on observe une augmentation du MRR pour les deux graphes Oregano et Hetionet.

Globalement, l'augmentation des données permet ainsi une amélioration des performances du modèle dans huit des dix cas considérés.

4.3.2 Résultats sur l'explicabilité

Les modèles de MHR sont conçus pour fournir des explications accompagnant chaque prédiction. Travailler avec ces modèles implique donc d'améliorer la qualité des explications en plus des métriques liées aux performances du modèle. Dans le Tableau 4.5, on constate que les résultats sur la diversité des explications sont moins positifs que pour le MRR. Pour Oregano, la diversité est améliorée pour tous les liens considérés après augmentation des données. Pour Hetionet en revanche, l'ajout de nouveaux liens fait baisser la diversité des prédictions globales et pour le lien reliant les nœuds Drug et Gene. Enfin pour BioKG, la diversité augmente pour la prédiction de liens sur l'ensemble du graphe et reste inchangée pour les deux liens reliant les médicaments aux maladies/gènes. Le premier constat est que, avec ou sans augmentation de données, SQUIRE ne permet pas de générer des explications constituées de plusieurs relations dans quatre cas sur dix. Pour les relations Drug-Drug dans Hetionet et BioKG et la relation Drug-Protein dans BioKG, le modèle ne propose que des résultats de prédiction de liens "directs", sans passer par plusieurs nœuds avant de faire la prédiction. Le second constat est que l'augmentation des données peut avoir un effet délétère sur cette diversité des explications, comme par exemple pour le lien Drug-Gene sur Hetionet, où le modèle utilisait trois relations différentes initialement et aucune après augmentation des données. Ces résultats semblent indiquer qu'Oregano bénéficie plus fortement de l'augmentation des données quand il s'agit d'améliorer les déplacements de l'agent dans le KG, là où l'effet est inexistant - voire délétère - sur Hetionet et BioKG. En revanche, certaines prédictions ne pourraient pas être faites sans l'ajout de nouveaux liens. Sur Oregano par exemple, comme illustré en Figure 4.4, l'ajout des relations *limits* et *participates_in* permet au modèle de prédire une relation possible entre l'acide nalixidique et la malaria, observation par ailleurs faite en laboratoire [Dube et al., 2023]. Sans l'ajout de ces nouveaux liens, l'organisation initiale du graphe n'aurait pas permis au modèle de faire cette prédiction. Comme illustré dans la Figure 4.3, les relations *limits* et *participates_in* connectent respectivement les nœuds Drug et Gene à des nœuds Phenotype, ce qui a permis au modèle de recourir à un nœud Phenotype dans l'explication de la prédiction.

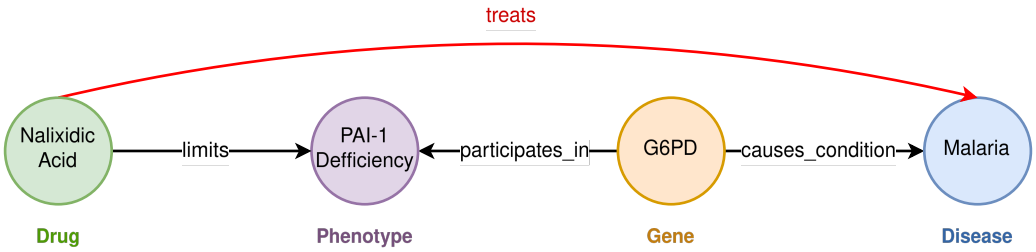

FIGURE 4.4 – Schéma d'une prédiction faite sur Oregano en utilisant deux des relations ajoutées au graphe. Sans l'ajout de ces relations, le modèle aurait été incapable de générer des explications passant à la fois par un nœud Gene et un nœud Phenotype. Le lien prédit apparaît en rouge.

4.4 Discussion

L'analyse des résultats du MRR pour la tâche de prédiction de liens sur les KGs Oregano, Hetionet et BioKG montre que l'ajout de nouveaux liens améliore systématiquement les performances pour les liens Drug-Disease, avec une amélioration très significative du MRR pour Hetionet. BioKG semble moins adapté que les deux autres graphes pour la tâche de repositionnement de médicaments avec le modèle SQUIRE. Bien que ces résultats soient intéressants du point de vue de la performance du modèle, ils sont à nuancer vis-à-vis de leur explicabilité. L'ajout de nouveaux liens permet d'améliorer le nombre de relations différentes utilisées dans les explications pour le KG Oregano, mais n'ont absolument aucun impact sur les explications pour Hetionet et BioKG. Sur ces deux graphes, le modèle n'arrive pas à parcourir le KG pour faire une prédiction ; les résultats sont donc systématiquement des prédictions directes : un lien est prédit entre un médicament et une maladie sans passer par des nœuds intermédiaires.

Pour les liens Drug-Protein, l'augmentation des données n'améliore le MRR que pour BioKG, tandis que pour les liens Drug-Gene, une amélioration est observée pour Oregano et Hetionet. Ces résultats sur le MRR sont donc plus nuancés que pour le lien entre les médicaments et les maladies. De plus, comme pour les résultats précédents, l'augmentation des données permet d'améliorer l'explicabilité uniquement pour Oregano.

Il est aussi important de remarquer que dans certains cas, il est préférable de ne pas ajouter tous les liens qui pourraient être créés à partir de leur combinaison. Pour la nouvelle relation *contributes_to* dans Oregano par exemple, résultant de la combinaison de deux relations qui n'existaient pas dans le KG d'origine, la génération de tous les liens *contributes_to* aurait signifié l'ajout de plus de 3 millions de nouveaux liens. Il a été prouvé qu'un déséquilibre dans la quantité d'information disponible dans un KG avait un effet délétère sur la qualité des résultats pour la tâche de prédiction de liens [Bonner et al., 2022b, Wu et al., 2022]. En effet, si un lien est sur-représenté ou bien que des nœuds sont connectés à un nombre trop important de voisins, le modèle a tendance à toujours attribuer des scores élevés à ces nœuds/liens et des scores plus faibles aux autres nœuds/liens du

graphe.

Ces résultats montrent à quel point la relation de cause à effet entre la structure du graphe et la performance du modèle peut être incertaine. Bien que dans la grande majorité des cas l'ajout de liens améliore le MRR, il n'y a que sur Oregano que l'augmentation des données permet d'avoir des explications plus riches et diversifiées. De plus, on a pu observer pour ce KG une amélioration quasi systématique du MRR pour tous les cas considérés (excepté pour le lien entre médicaments et protéines). Comme expliqué dans [You et al., 2021], l'augmentation des données sur des graphes n'est pas aussi facilement applicable que pour les images ou le texte. Il faut souvent procéder "à tâtons" et de manière différente pour chaque KG. Cette incertitude est ce qui a mené à créer des méthodes apprenant au fur et à mesure les liens à ajouter, essayant de minimiser une fonction de perte liée à la qualité des résultats en fonction des données ajoutées aux KG [Velickovic et al., 2019, Zhu et al., 2021]. Ces méthodes restent cependant très liées à l'utilisation de modèles de la famille des GNN. Il pourrait être intéressant de considérer une approche qui ajoute ou enlève des liens en fonction d'un score d'explicabilité pré-défini, plutôt que de baser ces méthodes uniquement sur les performances prédictives.

Une conclusion importante émerge également de ce travail, qui fait écho à celle du chapitre précédent : que l'on modifie le modèle ou les données, la promesse d'un modèle explicable n'est pas toujours tenue. Comme avec MultiHopKG, SQUIRE ne fournit pas d'explications satisfaisantes, malgré un ajout substantiel de chemins possibles à emprunter pour réaliser les prédictions. On observe que sur Hetionet et BioKG pour la prédiction de liens entre Drug et Disease ou entre Drug et Protein, le MRR s'est amélioré tandis que la diversité d'explication est nulle. Cela indique que pour ces relations, le modèle réalise des prédictions sans fournir de chemin explicatif. Nous nous retrouvons donc dans le même cas qu'avec MultiHopKG : il n'y a pas de garantie d'obtenir une explication pour chaque prédiction. Ce manque de consistance freine grandement l'utilisation de ces modèles pour le repositionnement de médicaments. En effet, nous pourrions nous retrouver dans le cas où, après entraînement, aucun de ces deux modèles ne pourraient fournir d'explication pour une prédiction concernant la ou les maladies d'intérêt de l'étude.

4.5 Conclusion

Ce chapitre porte sur l'augmentation des données au sein d'un KG biomédical. Nous avons montré que l'ajout de nouveaux liens dans ces KGs peut améliorer les performances prédictives du modèle de MHR SQUIRE, et cela sur les trois graphes Oregano, Hetionet et BioKG. Cependant, nous avons constaté qu'il n'est pas systématiquement possible d'améliorer la qualité des explications en augmentant la quantité de données dans un KG. Ces résultats suggèrent que l'augmentation des données dans un KG est une piste intéressante mais qu'il est difficile de généraliser une méthode à plusieurs graphes pour

obtenir des résultats similaires.

Nous avons travaillé dans ce chapitre à la modification des données dans le but d'améliorer le fonctionnement d'un modèle de MHR. Dans le chapitre précédent, nous avions modifié le fonctionnement du modèle lui-même. Ces deux chapitres ont montré à quel point l'aspect explicatif des modèles de MHR peut être insuffisant pour le repositionnement de médicaments. Dans le chapitre suivant, nous proposons une nouvelle méthodologie pour cette tâche basée sur la construction de features explicables et pouvant systématiquement fournir une prédiction.

Chapitre 5

Forêts aléatoires pour le repositionnement de médicaments

Sommaire

S'il existe un domaine pour lequel le repositionnement peut se montrer particulièrement utile, c'est celui des maladies rares. En effet, ces maladies sont par définition mal comprises par la communauté scientifique, avec seulement 5% des patients qui bénéficient d'un traitement thérapeutique [Pushpakom et al., 2019b]. De plus, les retombées économiques pour les laboratoires et les entreprises sont limitées dans le cas des maladies rares, peu de ventes étant possibles pour un travail en amont qui est long et coûteux. Dans ce contexte, le repositionnement de médicaments a déjà été appliqué avec succès pour certaines maladies

rares, comme la progéria [Gordon et al., 2012] ou le syndrome de Muckle-Wells [Kuemmerle-Deschner et al., 2013]. Dans ces cas, le fait de passer par le repositionnement de médicaments a permis de diviser par 5 le risque d'échec des essais cliniques [Roessler et al., 2021].

Dans ce chapitre, nous présentons une nouvelle méthodologie de repositionnement de médicaments basée sur l'utilisation du voisinage des nœuds pour entraîner un modèle de forêts aléatoires. Nous exploitons les trois KGs biomédicaux Oregano, Hetionet et BioKG [Boudin et al., 2023, Himmelstein et al., 2017, Walsh et al., 2020] pour construire des features explicables basées sur le concept des voisins communs entre les médicaments et les maladies. Ces features sont ensuite utilisées pour entraîner une forêt aléatoire qui prédit les relations potentielles entre les médicaments et les maladies. Pour évaluer notre approche, nous cherchons à identifier des médicaments candidats au repositionnement pour traiter une maladie rare, la sclérose latérale amyotrophique (SLA).

5.1 Introduction

Comme vu dans les chapitres précédents, les méthodes basées sur le MHR sont celles qui sont les plus utilisées pour le repositionnement de médicaments explicable, à travers la tâche de prédiction de liens. Néanmoins, nous avons montré dans les chapitres précédents que ces modèles ne permettaient pas d'atteindre un niveau d'explicabilité acceptable pour la tâche de repositionnement de médicaments. Il existe une problématique très proche de la prédiction de liens : la **classification de triplets**. Plutôt que de suggérer des nouveaux liens, les modèles de classification de triplets évalue si un triplet est vrai ou faux, transformant la prédiction de liens en classification binaire. Dans le cadre de la classification de triplets, les forêts aléatoires ont déjà été largement utilisées afin de prédire les interactions entre des nœuds présents dans un graphe [Wu et al., 2018, Han and Xu, 2016, Wang and Sukthankar, 2013]. Les forêts aléatoires peuvent être entraînées en utilisant le principe de culpabilité par association (*guilt by association*), par exemple lorsque des médicaments semblables sont connus pour interagir avec des protéines similaires [Olayan et al., 2017], ou en exploitant des embeddings pour représenter les nœuds et les arêtes dans le KG [Djeddi et al., 2023].

Bien que les modèles de MHR offrent des explications, nos résultats ainsi que d'autres études suggèrent que la qualité de l'explication peut ne pas être optimale [Bai et al., 2022, Lv et al., 2021a]. Les explications peuvent en effet manquer de sens, ou utiliser seulement une partie limitée des informations du KG. Parmi les méthodes ayant utilisé les forêts aléatoires pour le repositionnement de médicaments, beaucoup s'appuient sur l'usage d'embeddings plutôt pour représenter les nœuds, ce qui donne des features qui ne sont pas facilement interprétables [Djeddi et al., 2023, Zhao et al., 2023]. Alternativement, les features peuvent être construites en utilisant des ressources externes aux KGs, telles que des concepts sémantiques qui ne sont pas présents dans les grands KGs biomédicaux [Sousa

et al., 2024]. Par exemple, le travail réalisé dans [Malas et al., 2019] se base sur un KG biomédical auquel sont ajoutées les propriétés sémantiques auxquelles chaque nœud appartient en se basant sur l'Unified Medical Language System (UMLS) [Bodenreider, 2004]. L'UMLS est un système qui intègre plus de 150 terminologies biologiques ou médicales. Il permet d'interpréter, partager et analyser les données médicales codées avec différentes terminologies. Ainsi, chaque médicament du graphe est associé à plusieurs types sémantiques, tels que "Organic Chemical", "Pharmacologic Substance" ou "Biologically Active Substance". Ces informations ne sont pas présentes dans les grands KG biomédicaux tels que ceux utilisés dans cette thèse.

Dans ce chapitre, nous proposons une approche innovante pour construire des features transparentes visant à entraîner un modèle de forêts aléatoires pour le repositionnement de médicaments via la classification de triplets. À notre connaissance, il s'agit de la première tentative d'utiliser des forêts aléatoires pour le repositionnement de médicaments sur des KGs biomédicaux de cette taille, en se concentrant sur des features transparentes dérivées uniquement des informations disponibles dans les KGs. Nous évaluons la méthode proposée à travers une étude de cas visant à identifier des médicaments candidats au repositionnement pour le traitement de la SLA.

5.2 Méthodes

Cette section présente dans un premier temps la manière dont nous avons sélectionné les features qui seront utilisées par le modèle lors de l'entraînement (sous-section 5.2.2). Nous détaillons ensuite la manière dont le modèle a été entraîné et optimisé, ainsi que les métriques utilisées qui sont différentes des métriques relatives à la prédiction de liens puisque il s'agit ici d'une tâche de classification (sous-section 5.2.3). Enfin, nous présentons la méthodologie suivie pour le repositionnement de médicaments appliqué à la SLA (sous-section 5.2.4).

5.2.1 Sélection des ensembles de données et des échantillons

Pour les expériences, nous utilisons les trois mêmes KGs biomédicaux que dans le chapitre 4, à savoir Oregano [Boudin et al., 2023], Hetionet [Himmelstein et al., 2017] et BioKG [Walsh et al., 2020]. Les caractéristiques de ces KGs sont décrites dans le Tableau 2.1. Pour rappel, les trois KGs utilisés présentent des tailles et des propriétés différentes. Tout d'abord, Oregano comporte un nombre très élevé de nœuds pour un petit nombre de triplets, ce qui en fait un graphe peu connecté, avec très peu de relations entrantes ou sortantes des nœuds. Ensuite, Hetionet possède un petit nombre de nœuds pour un grand nombre de triplets et dispose également d'un grand nombre de types de relations différents, indiquant potentiellement un grand nombre de voisins de différents

types pour chaque nœud. Enfin, BioKG ressemble à Hetionet en ce qui concerne le ratio entre le nombre de nœuds et le nombre de triplets, mais il ne contient que quelques types de relations différents. Dans l'ensemble, pour les deux derniers KGs, le nombre élevé de triplets par rapport au nombre de nœuds indique des KGs avec une connectivité élevée.

Dans chaque KG, nous sélectionnons tous les triplets correspondant à un médicament lié à une maladie via les relations *treats* pour Oregano, *CtD* pour Hetionet et *drug_disease_association* pour BioKG. Ce sont les triplets qui décrivent les traitements connus pour une maladie, que nous qualifions donc de triplets positifs. Pour chaque triplet positif, nous corrompons le sujet (c'est-à-dire le médicament) ou l'objet (c'est-à-dire la maladie) du triplet en le remplaçant par un autre nœud du même type, avec une probabilité de 50% chacun. Nous nous assurons que les triplets corrompus ne sont pas déjà présents dans le KG et si tel est le cas, ils ne sont pas conservés. Inspirés par la construction de triplets négatifs dans les modèles d'embedding de KG [Qian et al., 2021], nous répétons l'étape de corruption N fois pour créer N triplets négatifs pour chaque triplet positif, en testant avec les valeurs de N égales à 1, 5, 10 et 20.

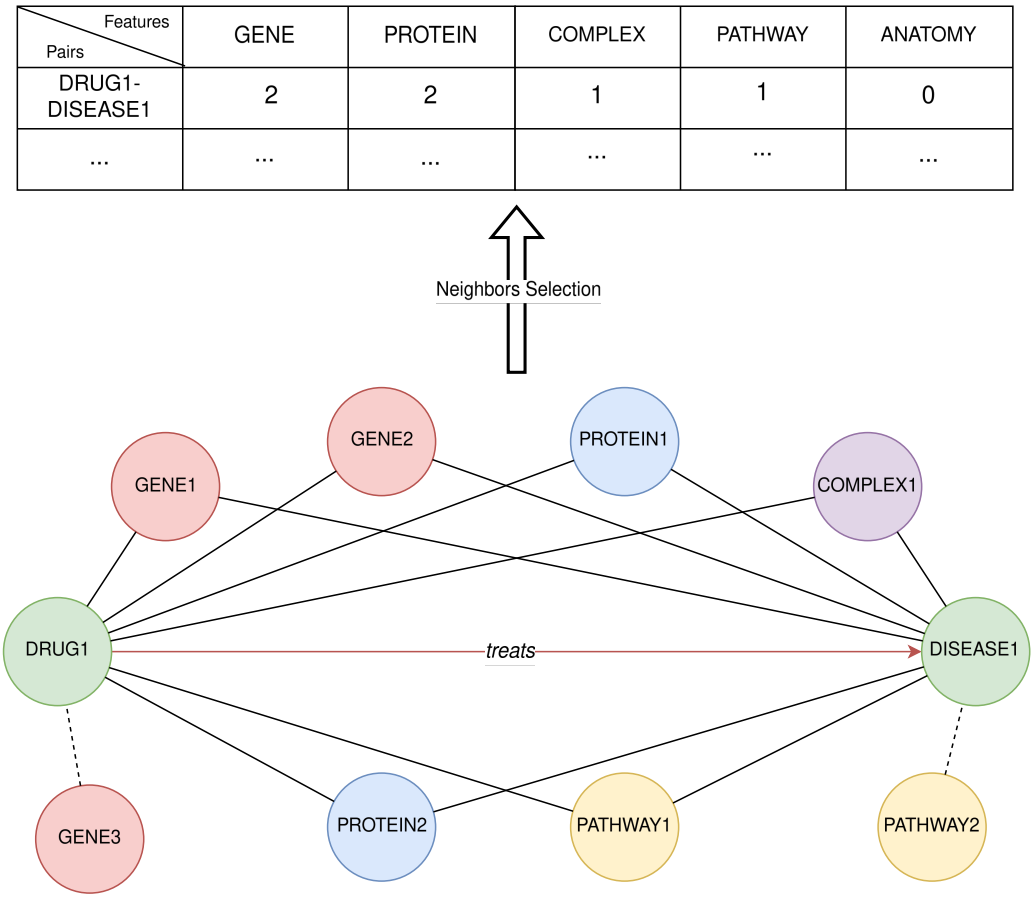

FIGURE 5.1 – Illustration de la méthode utilisée pour créer les features en se basant sur le voisinage des triplets d'intérêt, ici DRUG1-DISEASE1 en ne sélectionnant que les voisins de rang 1. Les lignes pointillées représentent les voisins qui ne sont pas communs aux deux nœuds du triplet d'intérêt, et qui ne sont donc pas utilisés pour construire les features.

5.2.2 Construction des features

Pour chaque triplet positif et négatif, nous construisons ses features en fonction des voisins des deux nœuds qui constituent le triplet (c'est-à-dire le médicament et la maladie) et du nombre de types de nœuds dans ce voisinage, comme illustré sur la Figure 5.1. Tout d'abord, les features sont construites en utilisant les voisins communs de rang 1 du médicament et de la maladie, c'est-à-dire les voisins communs directement liés à la fois au médicament et à la maladie. Cette méthode produit un nombre raisonnable de features pour BioKG mais presque aucune pour Oregano et Hetionet. Dans un second temps, et pour atténuer ce problème, nous essayons de construire les features en utilisant les voisins communs de rang 2 du médicament et de la maladie, c'est-à-dire en sélectionnant les voisins communs qui sont au plus à deux nœuds de distance du médicament et de la maladie. La sélection de tous les voisins de rang 2 produit un très grand nombre de voisins pour Hetionet et BioKG. Ce résultat est indésirable car le pouvoir explicatif de notre modèle réside dans la capacité d'observer les voisins directement dans le KG après obtention des résultats de prédiction. Pour un résultat de classification de triplets donné, nous souhaitons analyser son voisinage, c'est-à-dire examiner les nœuds qui sont liés à la fois au médicament et à la maladie. Avoir un très grand nombre de voisins rendrait cette tâche trop complexe. Par conséquent, nous décidons de sélectionner uniquement les nœuds les plus importants à conserver lors de la construction des features.

Comme dans d'autres travaux [Lin et al., 2018, Lei et al., 2020b], nous utilisons les scores PageRank pour identifier les nœuds les plus importants dans le KG [Page et al., 1999]. En utilisant l'algorithme PageRank [Page et al., 1999], nous pouvons filtrer le nombre de nœuds en fonction de leur importance. Nous utilisons ainsi la fonction PageRank de la bibliothèque Python Networkx [1]. Pour chaque triplet (positif ou négatif), nous sélectionnons tous les voisins communs de rang 1 et le voisin A avec le score PageRank le plus élevé. Nous sélectionnons ensuite un nombre n de voisins de A ayant le score PageRank le plus élevé. Cela nous permet de collecter les nœuds de rang 2 du médicament et de la maladie en conservant uniquement les plus importants, et ainsi de contrôler le nombre de nœuds utilisés pour construire les features. Lors de nos expérimentations, nous avons constaté que sélectionner 50 voisins constituait un bon compromis entre la conservation de l'information et le nombre de nœuds à analyser pour obtenir des résultats explicables. Pour les trois méthodes de sélection des voisins pour construire les features, le nombre médian de nœuds finalement obtenus est donné en Tableau 5.1

1. `https://networkx.org/documentation/stable/reference/algorithms/generated/networkx.algorithms.link_analysis.pagerank_alg.pagerank.html`

TABLE 5.1 – Nombre médian de nœuds sélectionnés en utilisant les trois méthodes différentes de sélection des voisins.

Méthode/Dataset	**Oregano**	**Hetionet**	**BioKG**
Voisins de rang 1	1	2	27
Voisins de rang 2	574	6 043	3 375
Voisins selon le score PageRank	50	22	48

5.2.3 Entraînement et test du modèle

À partir de ces données, nous effectuons un entraînement par validation croisée en cinq strates en utilisant 80% des données, les 20% restants correspondant à l'ensemble de test. Pendant l'entraînement, nous optimisons trois hyperparamètres : le nombre d'arbres dans la forêt, le nombre de features à utiliser dans chaque sous-ensemble de données et les critères de mesure de la qualité d'une division (Gini, Shannon) comme définis dans la section 2.3.

Nous évaluons notre modèle sur la tâche de classification des triplets, en testant si le modèle est capable de prédire si un triplet existe dans le KG ou s'il est faux. Nous choisissons d'utiliser l'exactitude et l'AUROC comme métriques principales, mais aussi les courbes de précision-rappel, car nous avons créé un déséquilibre de classes en générant des triplets négatifs.

Métriques de classification

Une courbe ROC, abréviation de *Receiver Operating Characteristic curve*, sert à représenter visuellement les performances d'un modèle de classification binaire. Cette courbe trace le taux de vrais positifs (TPR ou sensibilité ou rappel) en fonction du taux de faux positifs (FPR) à travers divers réglages de seuil, offrant un aperçu de la capacité de discrimination du modèle. L'AUROC (*Area Under Receiver Operating Characteristic*) correspond à l'aire sous la courbe ROC et illustre les compromis relatifs entre vrais positifs (bénéfices) et faux positifs (coûts). La méthode de prédiction optimale conduirait à un point positionné aux coordonnées (0,1) sur le graphe, indiquant une sensibilité de 100% (pas de faux négatifs) et une spécificité de 100% (pas de faux positifs). En revanche, un résultat aléatoire générerait un point le long de la ligne de non-discrimination (diagonale de (0,0) à (1,1)).

Les courbes de précision-rappel représentent la précision (ratio de vrais positifs sur l'ensemble des prédictions positives) par rapport au rappel (ratio de vrais positifs sur l'ensemble des observations positives) pour tous les seuils. Une précision élevée indique peu de prédictions faussement positives, tandis qu'un rappel élevé implique l'identification d'une grande proportion de toutes les instances positives.

Les courbes ROC et de précision-rappel évaluent les performances de la classification binaire, mais les courbes précision-rappel offrent une meilleure compréhension des change-

ments de performance lorsque la classe la plus courante est la classe négative. En de telles circonstances, il est possible d'atteindre des valeurs élevées d'AUROC simplement en prédisant toujours la classe négative. Dans de tels cas, une analyse de la courbe précision-rappel peut fournir plus d'informations que l'AUROC.

Mathématiquement, les métriques sont définies comme suit :

$$Exactitude = \frac{VraiPositif+VraiNégatif}{VraiPositif+VraiNégatif+FauxPositif+FauxNégatif}$$

$$TPR = Rappel = \frac{VraiPositif}{VraiPositif+FauxNégatif}$$

$$FPR = \frac{FauxPositif}{FauxPositif+VraiNégatif}$$

$$Précision = \frac{VraiPositif}{VraiPositif+FauxPositif}$$

TABLE 5.2 – Hyperparamètres considérés et valeurs testées pour leur optimisation.

Hyperparamètre	**Valeurs**
#Arbres	[100, 200, 300, 400, 500]
#Features	[0,2 ; 0,3 ; 0,4 ; 0,5 ; 0,6 ; 0,7 ; 0,8 ; 0,9 ; 1,0]
Critère	[“gini”, “entropy”]
#Triplets négatifs	[1, 5, 10, 20]

Nous avons décidé d'optimiser quatre hyperparamètres : trois hyperparamètres du modèle et le nombre de triplets négatifs à créer lors de la construction des features (Tableau 5.2). Nous entraînons le modèle et optimisons nos hyperparamètres en utilisant le modèle *RandomForestClassifier* de la bibliothèque Python *scikit-learn*[2]. Les trois hyperparamètres de ce modèle que nous avons optimisés sont les suivants :

— le nombre d'arbres utilisés dans la forêt (*n_estimators*),
— le nombre de features (*max_features*) à considérer pour chaque division,
— la fonction utilisée pour mesurer la qualité d'une division (*criterion*) à savoir l'impureté de Gini ou l'entropie de Shannon.

5.2.4 Utilisation des forêts aléatoires pour le repositionnement de médicaments

Pour évaluer l'efficacité de notre méthode, une étude de cas sur la SLA [3] a été réalisée. La SLA est un trouble neurodégénératif caractérisé par la mort des neurones moteurs

2. `https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html`
3. `https://www.omim.org/entry/105400`

dans le cerveau et la moelle épinière, entraînant une atrophie des muscles squelettiques et finalement une paralysie. Avec une prévalence de 5,2 pour 100 000 individus et une hétérogénéité génétique avec plusieurs gènes responsables, c'est une maladie rare hautement complexe qui fait l'objet de nombreuses recherches [Kiernan et al., 2011].

Pour trouver des médicaments pouvant être repositionnés pour la SLA, nous avons sélectionné tous les médicaments qui ne sont pas déjà liés à la SLA dans nos KGs. Ensuite, pour chaque médicament, nous l'avons relié à la SLA, formant ainsi un nouveau triplet. Nous avons finalement créé les features comme décrit précédemment et utilisé le modèle déjà entraîné pour classifier ces triplets comme étant vrais ou faux.

5.3 Résultats

5.3.1 Classification des triplets

Après avoir créé les features, nous avons obtenu 60 990 triplets positifs et 609 900 triplets négatifs pour Oregano, 483 triplets positifs et 4 830 triplets négatifs pour Hetionet, 66 868 triplets positifs et 668 680 triplets négatifs pour BioKG ; le choix de dix triplets négatifs pour un triplet positif ayant donné les meilleurs résultats. Le nombre de features dépend du nombre de types de nœuds différents pour chaque KG : 12 pour Oregano, 11 pour Hetionet et 6 pour BioKG (section 2.1).

D'après le Tableau 5.3, nous observons une excellente exactitude pour les trois méthodes de sélection des features. Ce premier résultat valide l'intuition que l'information des voisins est suffisante pour entraîner un classificateur de type forêts aléatoires pour la tâche de classification de triplets dans des KGs de grande taille. Comme nous nous concentrons sur des prédictions explicables, nous nous intéressons ensuite à l'interprétation des résultats du modèle entraîné en utilisant les voisins sélectionnés grâce à leur score PageRank. Cela nous permet de contrôler la quantité d'informations contenues dans les features (les meilleurs hyperparamètres de ce modèle pour les différents KGs sont présentés dans le Tableau 5.4).

TABLE 5.3 – Résultats (exactitude) du modèle pour les trois méthodes différentes de sélection des voisins.

Méthode/Dataset	**Oregano**	**Hetionet**	**BioKG**
Voisins de rang 1	0,973	0,910	0,984
Voisins de rang 2	0,999	0,975	0,999
Voisins selon le score PageRank	0,974	0,950	0,985

Dans cette configuration, nous calculons l'AUROC pour les trois KGs (Figure 5.2, courbes de gauche). Nous obtenons des valeurs de 0,945 pour Oregano, 0,939 pour Hetionet et 0,925 pour BioKG.

Comme l'AUROC peut être biaisée pour les classificateurs entraînés sur des ensembles de données déséquilibrés, nous observons ensuite les courbes de précision-rappel (Figure 5.2,

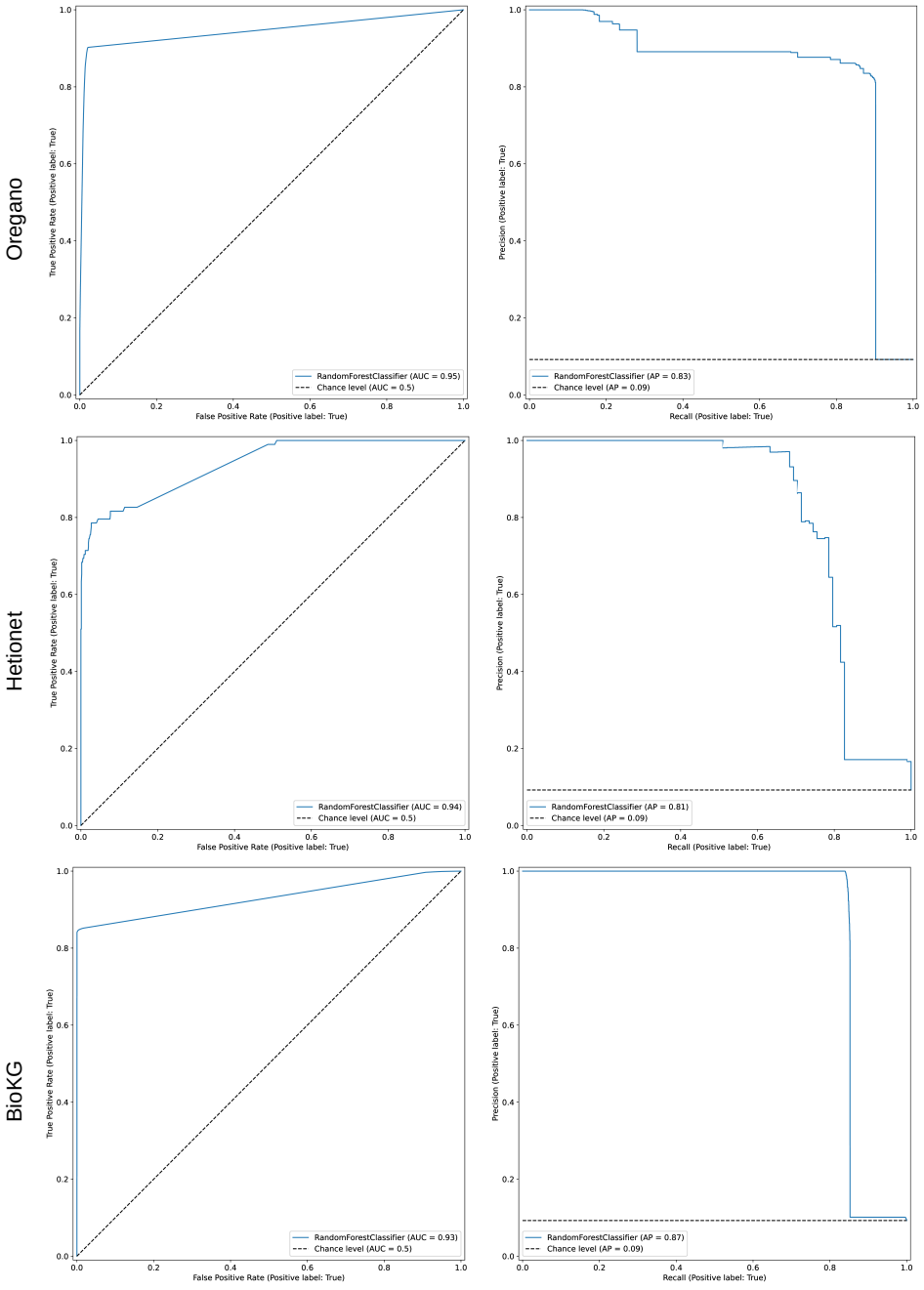

FIGURE 5.2 – AUROC (à gauche) et courbe de précision-rappel (à droite) pour la tâche de classification de triplets sur les trois KGs biomédicaux considérés. La première rangée présente les résultats pour Oregano, la rangée du milieu pour Hetionet, et la dernière rangée pour BioKG.

TABLE 5.4 – Meilleures valeurs des hyperparamètres pour chaque dataset.

Hyperparamètre	**Oregano**	**Hetionet**	**BioKG**
#Arbres	200	200	100
#Features	0,2	0,3	0,2
Critère	gini	gini	gini
#Triplets négatifs	10	10	10

courbes de droite). Pour les trois KGs, un seul seuil de probabilité avec une bonne précision et un bon rappel peut être identifié. Pour Oregano, le modèle atteint une précision de 0,832 et un rappel de 0,886. Pour Hetionet une précision de 0,923 et un rappel de 0,743 et pour BioKG une précision de 0,999 et un rappel de 0,833. Une haute valeur de précision indique que notre modèle classe très rarement un faux triplet comme étant vrai. Les scores de précision pour les trois datasets sont très bons, en particulier pour Hetionet (0,923) et BioKG (0,999). En revanche, nous observons des valeurs de rappel supérieures à 0,8 pour Oregano et BioKG et légèrement inférieures pour Hetionet (0,743). Globalement, nos expériences montrent une meilleure performance en termes de précision que de rappel. Dans le cas du repositionnement de médicaments, un score de précision élevé est une propriété plus importante car il indique que si notre modèle classe un triplet inexistant comme vrai, ce nouveau triplet est très probablement d'intérêt.

TABLE 5.5 – Pourcentage de prédictions correctes avec une explication pour les trois modèles de MHR

Modèle de MHR / Dataset	**Oregano**	**Hetionet**	**BioKG**
AnyBURL	100	100	100
SQUIRE	8,4	0	0
MultiHopKG	99	75	0

5.3.2 Explicabilité des résultats

L'un des avantages de notre méthode est la possibilité d'expliquer systématiquement les résultats. Nous avons testé trois modèles (MultiHopKG [Lin et al., 2018], SQUIRE [Bai et al., 2022] et AnyBURL [Meilicke et al., 2019]) pour la tâche de repositionnement de médicaments. Pour chaque modèle et chaque KG, nous observons le nombre de prédictions correctes accompagnées d'une explication. Les résultats sont décrits dans le Tableau 5.5. Il apparaît que l'utilisation des modèles de MHR ne garantit pas que les résultats soient explicables. AnyBURL est le seul modèle à fournir une explication pour chacune de ses prédictions. Dans le cas de MultiHopKG et de SQUIRE, la quantité d'explications varie entre 0% et 99%. Comme mentionné par Bai *et al.* et Lv *et al.* [Bai et al., 2022, Lv et al., 2021a], lorsque des explications sont fournies, celles-ci peuvent être de très mauvaise qualité. Ici, le modèle MultiHopKG appliqué sur Oregano fournit le plus grand nombre

d'explications, mais nous avons remarqué que 95% d'entre elles n'utilisent qu'une seule relation dans le KG, créant ainsi des explications simplistes qui n'utilisent pas toutes les informations que le KG contient.

La force de notre méthode réside dans le fait que pour chaque résultat, nous pouvons visualiser directement les features utilisées pour faire la prédiction, c'est-à-dire les voisins les plus importants des nœuds formant le triplet d'intérêt. Ces voisins peuvent être trouvés en interrogeant simplement un système de gestion de bases de données graphes contenant le KG, tel que Neo4j [4]. Nous illustrons cette fonctionnalité sur la SLA dans la sous-section suivante.

En plus de cette observation directe des nœuds ayant servi à l'entraînement, les forêts aléatoires permettent d'analyser l'importance de chaque feature, c'est-à-dire le nombre de nœuds voisins de chaque type dans notre cas. Le Tableau 5.6 montre, pour chaque KG utilisé, les types de nœuds ayant un coefficient de Gini d'au moins 0,1. Pour rappel, le coefficient de Gini est utilisé dans la construction des arbres de décision pour mesurer l'importance de la séparation des données lors de l'utilisation d'une feature donnée. Pour les trois KGs, seule une petite partie des nœuds est responsable de la grande majorité des décisions du modèle. Pour BioKG, la moitié (3 sur 6) des différents nœuds existants ont une influence de 96% dans les prédictions. Pour Hetionet, 5 types de nœuds sur 11 influencent à 82% les décisions et pour Oregano c'est seulement 3 types de nœuds sur 11 qui influencent à 96% les prédictions.

TABLE 5.6 – Pour chaque KG, présentation des types de voisins les plus importants pour la tâche de classification de liens. Les types de voisins reportés sont ceux ayant un coefficient de Gini d'au moins 0,1. Le total indiqué correspond à la somme des impacts des types de voisins sur les prédictions finales.

Dataset	Type de voisins	Total
Oregano	Gene, Disease, Drug	0,955
Hetionet	Gene, Compound, Disease, Anatomy, Biological Process	0,821
BioKG	Protein, Disease, Drug	0,961

5.3.3 Médicaments proposés pour la SLA

Après avoir évalué tous les triplets possibles qui impliquent la SLA, nous avons identifié 12 médicaments candidats pour un repositionnement à partir d'Oregano et BioKG. Comme la SLA n'est pas présente dans Hetionet, aucun résultat n'a été trouvé en utilisant ce KG. Les 12 composés sont les suivants : metformine, rosiglitazone, atorvastatine, clopidogrel, épirubicine, fluorouracile, nicotine, aspirine, oxaliplatine, simvastatine, bévacizumab et palipéridone. Pour chacun de ces résultats, le graphe contenant les voisins servant de

4. `https://neo4j.com/`

features fournit une explication visuelle, comme illustré dans la Figure 5.3 pour le cas de la metformine et du rosiglitazone.

Pour évaluer la pertinence biologique de nos résultats, nous avons recherché des publications reliant chacun de ces médicaments à la SLA en interrogeant PubMed. Parmi les 12 médicaments proposés, nous avons trouvé que six faisaient déjà l'objet de recherches en lien avec la SLA (Tableau 5.7), avec un essai clinique en cours pour la metformine [5]. En ce qui concerne le rosiglitazone, il a été montré que les patients atteints de SLA présentaient une réduction de l'ARN messager (ARNm) et de la protéine Peroxisome proliferator-activated receptor gamma coactivator 1 alpha dans les muscles squelettiques [Russell et al., 2013]. Par ailleurs, des auteurs ont montré que la metformine et le rosiglitazone augmentaient l'expression du gène correspondant (PPARGC1A - PPAR gamma coactivator 1 alpha) [Muhlhausler et al., 2009, Aatsinki et al., 2014]. Pour l'atorvastatine, Merwin *et al.* ont souligné le lien entre la SLA et une faible activité du gène Paraoxonase 1 (PON1) [Merwin et al., 2017], tandis que Sardo *et al.* ont montré que l'atorvastatine créait une augmentation de l'expression de l'ARNm et de la protéine de PON1 [Sardo et al., 2005]. De manière similaire pour la simvastatine, Deakin *et al.* ont prouvé que ce composé pouvait moduler l'expression in vitro du gène PON1 et qu'il est associé à une concentration et une activité accrues de PON1 [Deakin et al., 2003]. Concernant le fluorouracile, Rando *et al.* ont publié une étude confirmant que ce médicament anticancéreux était capable d'augmenter la durée de vie, de retarder l'apparition de la maladie et d'améliorer les performances motrices chez les souris atteintes de SLA [Rando et al., 2019]. Enfin, la publication d'Aouti *et al.* suggère que la palipéridone pourrait prévenir l'agrégation de l'enzyme superoxide dismutase (SOD1) responsable de la SLA [Aouti et al., 2023], ce qui en fait un candidat pour le développement de médicaments contre la SLA.

5.4 Discussion

Cette étude s'est intéressée à l'évaluation des méthodes de classification de triplets pour le repositionnement de médicaments en utilisant uniquement les informations de voisinage des nœuds dans un KG. Les expériences ont été menées sur trois KGs, deux d'entre eux étant couramment utilisés pour le repositionnement de médicaments [Walsh et al., 2020, Himmelstein et al., 2017, Boudin et al., 2023, Bonner et al., 2022a]. Notre méthode est la première à tester les performances d'un modèle de forêts aléatoires sur ces KGs pour le repositionnement, sans utiliser d'informations externes ou de méthodes d'embeddings de KG, se concentrant ainsi sur des résultats explicables. Bien que tous nos résultats ne puissent être reliés à des publications scientifiques, le fait que la moitié d'entre eux puissent l'être (avec un médicament en essai clinique) est encourageant quant à l'intérêt de nos résultats.

5. `https://clinicaltrials.gov/study/NCT04220021`

TABLE 5.7 – Propositions de repositionnement potentiel trouvées par notre méthode pour la SLA, accompagnées le cas échéant des publications scientifiques étayant les résultats (NA indique qu'aucune publication n'a été trouvée).

Drug	**Literature**
Metformine	[Russell et al., 2013, Aatsinki et al., 2014]
Rosiglitazone	[Russell et al., 2013, Muhlhausler et al., 2009]
Atorvastatine	[Sardo et al., 2005, Merwin et al., 2017]
Simvastatine	[Merwin et al., 2017, Deakin et al., 2003]
Fluorouracile	[Rando et al., 2019]
Palipéridone	[Yu et al., 2024b, Aouti et al., 2023]
Oxaliplatine	NA
Clopidogrel	NA
Épirubicine	NA
Nicotine	NA
Bévacizumab	NA
Aspirine	NA

5.4.1 Points forts

Le premier avantage de notre méthode est qu'elle offre de bonnes performances comparativement à l'état de l'art pour la tâche de classification de triplets sur des KGs biomédicaux. En effet, d'autres études ont montré l'intérêt des forêts aléatoires pour la classification de triplets, mais seulement sur des KGs construits pour une tâche spécifique [Djeddi et al., 2023, Olayan et al., 2017, Aisopos and Paliouras, 2023]. Notre méthode est adaptée aux KGs de grande taille qui n'ont pas été spécifiquement construits pour un repositionnement en particulier. Ces KGs sont construits à partir d'informations provenant de différentes sources de connaissances disponibles en ligne gratuitement, telles que UniProt, REACTOME, DrugBank, SIDER ou encore PharmGKB. Travailler avec ces graphes offre donc des résultats potentiels de repositionnement de médicaments pour de nombreuses maladies. Le choix d'une maladie ou d'un médicament particulier dépend de l'intérêt des cliniciens, mais la plupart des médicaments et maladies connus peuvent être trouvés dans ces KGs, car ils sont conçus pour être exhaustifs. Les maladies rares peuvent cependant être un cas particulier, puisque nous avons constaté que la SLA n'était pas présente dans Hetionet.

Un autre avantage de notre méthode réside dans ses excellents résultats en termes de précision. Pour les datasets, le modèle a été capable de distinguer clairement les triplets positifs des triplets négatifs. Malgré la construction d'un plus grand nombre de triplets négatifs que de triplets positifs existant dans les KGs, notre méthode ne souffre pas d'une surprédiction de la classe négative. Ce comportement est important car l'objectif du repositionnement est de fournir aux chercheurs une liste de médicaments potentiels pour des essais cliniques, et de préférence uniquement des médicaments prometteurs. Notre modèle présente un taux de faux positifs très faible pour les trois KGs (0,023 pour Oregano, 0,055 pour Hetionet et 0,004 pour BioKG), indiquant sa capacité à fournir uniquement

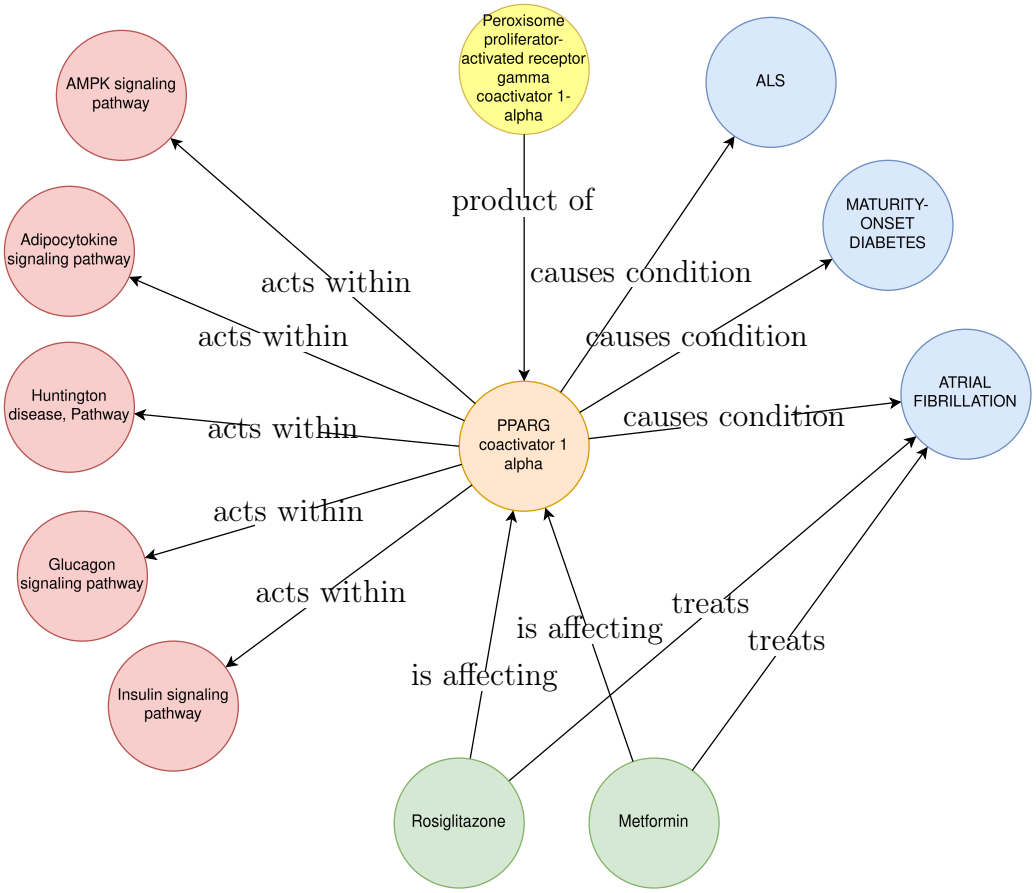

FIGURE 5.3 – Exemple de visualisation visant à expliquer les conclusions selon lesquelles la metformine et la rosiglitazone sont des candidats potentiels pour le traitement de la SLA. Les nœuds bleus représentent des maladies, les rouges des voies métaboliques, les verts des médicaments, les bruns des gènes et les jaunes des protéines.

des résultats pertinents et très peu de faux positifs.

Le troisième avantage de notre méthode est qu'elle fournit des résultats entièrement transparents et explicables. Des travaux sur l'utilisation des forêts aléatoires sur des KGs ont été réalisés en utilisant des embeddings de nœuds et de relations pour créer les features du modèle [Zhao et al., 2023, Djeddi et al., 2023, Sousa et al., 2024]. En procédant ainsi, l'interprétation des features les plus importantes pour une prédiction est impossible. Dans notre cas, nous nous concentrons sur la création de features qui peuvent être retrouvées ultérieurement dans les KGs et comprises par les cliniciens. Pour chaque triplet prédit, qu'il soit vrai ou faux, ses voisins les plus importants pour la prédiction peuvent être facilement collectés et/ou visualisés de manière à fournir des informations complémentaires sur les protéines associées, les gènes ou les voies biologiques, permettant ainsi aux cliniciens de valider ou rejeter la prédiction en fonction de leurs connaissances.

Un dernier avantage de notre méthode est sa rapidité à obtenir des résultats. Dans une étude précédente, un modèle de forêts aléatoires pour la prédiction de triplets avec des features explicables a été testé en utilisant des chemins sémantiques entre des paires médicament-gène [Aisopos and Paliouras, 2023]. Cette méthode a fourni des features

explicables sous forme de chemins existants dans le KG et les a utilisées pour entraîner le modèle. Le KG construit pour ce travail est un graphe de médicaments et de gènes, représentant environ 620 000 triplets. Pour construire les features, les auteurs ont rapporté un temps d'exécution de 19,45 heures en utilisant un serveur Ubuntu avec 12 Go de RAM et 8 cœurs CPU. Dans notre cas, la création des features de BioKG (qui est le KG le plus large avec plus de deux millions de triplets) a pris environ 25 minutes avec la même configuration matérielle.

5.4.2 Limites

Ce travail présente cependant un certain nombre de limites. Tout d'abord, contrairement aux modèles de prédiction de liens, notre méthode ne permet pas de classer facilement et rapidement de nouveaux triplets. Les modèles de prédiction de liens utilisent les embeddings de chaque nœud et relation ainsi qu'une fonction de score pour déterminer la vraisemblance d'un triplet, ce qui permet de classer rapidement des données qui n'ont jamais été vues auparavant. Dans notre cas, après l'entraînement du modèle, toutes les combinaisons potentielles de triplets médicament-maladie doivent être préalablement construites. Seulement alors, nous pouvons récupérer les features pour chacun de ces triplets et utiliser le modèle entraîné pour les classer.

Deuxièmement, lorsqu'elle est utilisée sur un KG avec une connectivité faible, notre méthode peut créer des explications peu informatives. En effet, les voisins de différents triplets peuvent être les mêmes, ce qui crée des explications similaires pour des triplets différents, comme illustré sur la Figure 5.3 avec la metformine et le rosiglitazone. À l'inverse, pour les KGs dont les nœuds sont extrêmement connectés, certaines explications visuelles peuvent manquer de lisibilité car les voisins communs sélectionnés pour construire les features peuvent être trop nombreux.

En termes d'explications fournies, il manque un cadre de référence pour évaluer la qualité des explications. Même si dans notre cas, les explications sont basées sur des données réelles contenues dans les KGs (à savoir les voisins du triplet prédit), il serait intéressant de voir comment ces explications sont évaluées par les cliniciens.

Enfin, il est étonnant de voir que certains de nos résultats sont étayés par des publications scientifiques dont le contenu n'était pas représenté dans les KGs utilisés dans cette étude. La plupart des articles scientifiques soutenant nos résultats ont été publiés il y a plusieurs années, ce qui exclut l'hypothèse que les résultats étaient trop récents pour être intégrés dans les graphes. De plus, nous avons remarqué que certains liens entre les médicaments proposés et la SLA se trouvaient dans la base de données CTD [6], qui n'est pas incluse dans les bases de connaissances utilisées pour construire les KGs. Cela montre la difficulté de constituer des KGs complets, incluant toutes les connaissances

6. `https://ctdbase.org/`

disponibles sur un sujet donné. En outre, il est très compliqué de maintenir à jour les KGs une fois construits. En effet, l'évolution rapide des connaissances est difficile à répercuter systématiquement et en temps réel dans un graphe. Un exemple éloquent sur cet aspect est celui de DBPedia, un KG dérivé de Wikipédia, qui a dû être mis à jour 36 340 fois entre octobre 2015 et avril 2016, soit en moyenne 200 fois par jour [Shi and Weninger, 2018].

5.5 Conclusion

Notre méthodologie démontre de solides performances dans la tâche de classification des liens entre les médicaments et les maladies, notamment grâce à la construction de features explicables qui favorisent l'interprétation des résultats. Contrairement aux modèle de MHR utilisés dans les deux chapitres précédents, le fait de baser os prédictions sur des features explicables garantit qu'il y a toujours une explication pendant la phase de prédiction. Dans le cas de la SLA, notre modèle a identifié douze médicaments candidats pour le repositionnement. Plus particulièrement, six de ces médicaments ont déjà été décrits dans la littérature scientifique comme étant associés à la SLA, l'un d'eux faisant même l'objet d'un essai clinique actuellement en phase 2. Ce travail confirme le potentiel d'utiliser les forêts aléatoires pour la tâche de prédiction de liens sur des KGs, en particulier pour le repositionnement de médicaments appliqué aux maladies rares. Nous démontrons que la priorité donnée à l'explicabilité ne compromet pas la qualité des prédictions, même avec un modèle d'apprentissage automatique relativement simple.

Chapitre 6

Conclusion et discussion

Cette thèse a pour but de mieux comprendre les opportunités offertes par l'IA pour le repositionnement de médicaments, avec comme contrainte principale le besoin croissant d'explicabilité dans le domaine de la santé. À travers ce travail, nous avons exploré comment des modèles d'IA explicables pouvaient apprendre à partir de KGs à prédire de nouveaux liens entre les entités de ces graphes.

Dans le chapitre 3, nous avons proposé d'ajouter à l'entraînement de modèles de MHR une étape visant à pré-entrainer les embeddings représentant les entités et les relations du graphe. Les résultats ont démontré que ce pré-entraînement était une approche efficace pour améliorer les performances prédictives des modèles de MHR. Dans le chapitre 4, nous avons cherché à augmenter la quantité de données dans trois KGs biomédicaux au travers de l'ajout de nouvelles relations. Cette augmentation de données appliquées aux KGs a montré une amélioration des performances du modèle de MHR SQUIRE pour la tâche de prédiction de liens appliquée au repositionnement de médicaments. Enfin dans le chapitre 5, nous avons utilisé un algorithme de forêts aléatoires pour l'appliquer au repositionnement de médicaments à partir de KGs biomédicaux. En construisant des features basées sur le voisinage des traitements connus, nous avons établi que cette méthode était adaptée à la tâche de repositionnement de médicaments et permettait d'obtenir des propositions de repositionnement explicables pour la SLA.

Ces résultats permettent de mieux appréhender les possibilités et les difficultés qui entourent le repositionnement de médicaments via l'utilisation de modèle d'IA explicables.

Les modèles de MHR par renforcement Les modèles de MHR basés sur l'apprentissage par renforcement sont les premiers modèles qui ont proposé d'utiliser le *deep learning* couplé à un mécanisme d'explicabilité pour la prédiction de liens. Nous avons montré que, comme dans d'autres domaines du machine learning, une étape de pré-entraînement

améliorait les performances de ces modèles. Cette étape, bien que répandue dans le domaine de l'analyse d'images ou du traitement automatique des langues, n'avait pas encore été mise en œuvre dans des modèles apprenant sur des graphes. Cependant, ces méthodes ne sont pas adaptées aux KGs de grande taille, comme les graphes biomédicaux, en raison de leurs exigences élevées en matière de ressources. Cette limite matérielle est le premier frein à leur utilisation pour le repositionnement de médicaments.

Les modèles de MHR et l'explicabilité En se basant sur l'apprentissage par renforcement ou sur l'utilisation de transformers comme SQUIRE, les modèles de MHR sont conçus pour fournir un mécanisme de prédiction explicable, par le biais de chemins de raisonnement. Cependant, ces chemins peuvent être redondants, peu informatifs, voire complètement faux. Pour pallier ces problèmes, nous avons augmenté le nombre de relations existantes dans trois KGs biomédicaux. La tâche d'augmentation des données dans un KG étant difficile à réaliser sans compromettre la qualité des informations qu'il contient, nous avons privilégié l'expertise d'un médecin pour nous y aider. Nous avons démontré que l'ajout de nouvelles relations au sein de KGs biomédicaux permettait à SQUIRE d'obtenir de meilleurs résultats de prédictions pour la tâche de repositionnement de médicaments. Cependant, la qualité des explications n'a pas été améliorée par cet enrichissement des graphes. De plus, quel que soit le modèle de MHR et le jeu de données utilisé, nous avons observé que les prédictions n'étaient pas systématiquement accompagnées d'explications, voire que ces modèles ne parvenaient pas à réaliser de prédiction pour une partie des données. Ce manque d'homogénéité dans les prédictions de ces modèles représente le deuxième frein à leur utilisation pour le repositionnement de médicaments. Il n'y a donc aucune garantie que le modèle sera capable de faire une prédiction sur un médicament ou une maladie d'intérêt, et aucune garantie que cette prédiction puisse être expliquée.

La construction de features transparentes pour le repositionnement de médicaments Compte tenu de ces limites, nous avons écarté ces modèles au profit d'un modèle plus simple, les forêts aléatoires, qui baserait l'apprentissage sur des features susceptibles d'être interprétées ultérieurement. Plutôt que d'expliquer les prédictions, nous avons gardé une transparence totale sur les données qui ont servi de base à leur élaboration. L'avantage avec les KGs est qu'il est facile d'obtenir une représentation visuelle claire des données en n'observant qu'un sous-ensemble du graphe initial. Bien que les forêt aléatoires aient déjà été utilisées pour le repositionnement de médicaments, nous avons montré qu'il était possible de le faire sans aucune transformation des données, telle que l'embedding. Nos résultats ont permis de proposer 12 médicaments candidats au repositionnement pour la SLA, avec systématiquement une explication visuelle par le biais du sous-graphe qui contient ces nouveaux liens entre la maladie et les médicaments. Il convient toutefois de noter que l'analyse de l'importance des features a fait apparaître que seule une petite

partie de ce sous-graphe était réellement utilisée pour la prédiction.

Direction de la recherche L'un des principaux obstacles au repositionnement des médicaments est le manque de compréhension des utilisateurs à l'égard des résultats des modèles d'IA. Malheureusement, il n'existe actuellement pas de métriques permettant d'évaluer correctement les explications d'un modèle sans intervention humaine. Ce processus devient particulièrement long et fastidieux, et peut dépendre de l'expertise de quelques personnes seulement. Ce manque d'outils pour l'évaluation de l'explicabilité signifie qu'il est impossible de savoir comment agir sur les modèles ou sur les données pour augmenter leur potentiel d'explicabilité. Le problème principal est que chaque KG possède ses propres particularités : nombre de nœuds, de relations, leurs types, le domaine des connaissances qu'il contient. Dans ce cas, l'établissement d'une métrique d'explicabilité pour un modèle appliqué sur un graphe peut ne pas être transférable à un autre graphe. Le sens que portent les nœuds et les relations dans un KG biomédical n'est pas du tout le même que dans un graphe de données financières. Orienter la recherche vers une utilisation commune des métriques de performance et d'explicabilité permettrait de mieux apprécier les capacités d'un modèle en fonction du domaine dans lequel il doit être utilisé.

Une autre piste intéressante est celle du traitement des données avant entraînement de ces modèles. Les KGs, par définition, sont construits dans le but d'être exhaustifs. Cela ne veut pas dire pour autant qu'ils doivent être utilisés tels quels dans des tâches d'apprentissage automatique. Nous avons constaté dans nos résultats que, avec les modèles de MHR comme avec les modèles de forêts aléatoires, aucun modèle n'utilise toute l'information disponible dans les graphes pour faire des prédictions. Les modèles de MHR exploitent généralement une petite partie des différents types de relations pour réaliser la majorité de leurs prédictions. Les modèles de forêts aléatoires ne s'appuient que sur la moitié des voisins (voire moins) pour produire plus de 80% de leurs résultats. Dans les deux cas, une partie des données semble superflue. Sélectionner des données d'entraînement de qualité, c'est s'assurer de meilleurs résultats après entraînement du modèle. Dans notre cas, sélectionner des données de meilleure qualité pourrait également signifier des résultats plus explicables.

Bibliographie*(n=168)*

[Aatsinki et al., 2014] Aatsinki, S.-M., Buler, M., Salomäki, H., Koulu, M., Pavek, P., and Hakkola, J. (2014). Metformin induces pgc-1α expression and selectively affects hepatic pgc-1α functions. *British Journal of Pharmacology*, 171(9) :2351–2363.

[Aisopos and Paliouras, 2023] Aisopos, F. and Paliouras, G. (2023). Comparing methods for drug–gene interaction prediction on the biomedical literature knowledge graph : performance versus explainability. *BMC Bioinformatics*, 24(1) :272.

[Ali et al., 2021a] Ali, M., Berrendorf, M., Hoyt, C., Vermue, L., Galkin, M., Sharifzadeh, S., Fischer, A., Tresp, V., and Lehmann, J. (2021a). Bringing light into the dark : A large-scale evaluation of knowledge graph embedding models under a unified framework. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PP :1–1.

[Ali et al., 2021b] Ali, M., Berrendorf, M., Hoyt, C. T., Vermue, L., Sharifzadeh, S., Tresp, V., and Lehmann, J. (2021b). PyKEEN 1.0 : A Python Library for Training and Evaluating Knowledge Graph Embeddings. *Journal of Machine Learning Research*, 22(82) :1–6.

[Ali et al., 2019] Ali, M., Jabeen, H., Hoyt, C. T., and Lehmann, J. (2019). The keen universe : An ecosystem for knowledge graph embeddings with a focus on reproducibility and transferability. In *The Semantic Web–ISWC 2019 : 18th International Semantic Web Conference, Auckland, New Zealand, October 26–30, 2019, Proceedings, Part II 18*, pages 3–18. Springer.

[Aouti et al., 2023] Aouti, S., Padavattan, S., and Padmanabhan, B. (2023). Structure-based discovery of an antipsychotic drug, paliperidone, as a modulator of human superoxide dismutase 1 : a potential therapeutic target in amyotrophic lateral sclerosis. *Acta Crystallographica Section D*, 79(6) :531–544.

[Arnold, 2007] Arnold, L. M. (2007). Duloxetine and other antidepressants in the treatment of patients with fibromyalgia. *Pain medicine*, 8(suppl_2) :S63–S74.

[Bai et al., 2022] Bai, Y., Lv, X., Li, J., Hou, L., Qu, Y., Dai, Z., and Xiong, F. (2022). SQUIRE : A sequence-to-sequence framework for multi-hop knowledge graph reasoning. In Goldberg, Y., Kozareva, Z., and Zhang, Y., editors, *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 1649–1662, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

[Barredo Arrieta et al., 2020] Barredo Arrieta, A., Díaz-Rodríguez, N., Del Ser, J., Bennetot, A., Tabik, S., Barbado, A., Garcia, S., Gil-Lopez, S., Molina, D., Benjamins, R., Chatila, R., and Herrera, F. (2020). Explainable artificial intelligence (xai) : Concepts, taxonomies, opportunities and challenges toward responsible ai. *Information Fusion*, 58 :82–115.

[Bates et al., 2003] Bates, D. W., Ebell, M., Gotlieb, E., Zapp, J., and Mullins, H. (2003). A proposal for electronic medical records in us primary care. *Journal of the American Medical Informatics Association*, 10(1) :1–10.

[Bhagat et al., 2011] Bhagat, S., Cormode, G., and Muthukrishnan, S. (2011). Node classification in social networks. *Social network data analytics*, pages 115–148.

[Bloom et al., 2011] Bloom, D. E., Boersch-Supan, A., McGee, P., Seike, A., et al. (2011). Population aging : facts, challenges, and responses. *Benefits and compensation International*, 41(1) :22.

[Bodenreider, 2004] Bodenreider, O. (2004). The unified medical language system (umls) : integrating biomedical terminology. *Nucleic acids research*, 32(suppl_1) :D267–D270.

[Bollacker et al., 2008] Bollacker, K., Evans, C., Paritosh, P., Sturge, T., and Taylor, J. (2008). Freebase : A collaboratively created graph database for structuring human knowledge. In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data*, SIGMOD '08, page 1247–1250, New York, NY, USA. Association for Computing Machinery.

[Bondy et al., 1976] Bondy, J. A., Murty, U. S. R., et al. (1976). *Graph theory with applications*, volume 290. Macmillan London.

[Bonner et al., 2022a] Bonner, S., Barrett, I. P., Ye, C., Swiers, R., Engkvist, O., Hoyt, C. T., and Hamilton, W. L. (2022a). Understanding the performance of knowledge graph embeddings in drug discovery. *Artificial Intelligence in the Life Sciences*, 2 :100036.

[Bonner et al., 2022b] Bonner, S., Kirik, U., Engkvist, O., Tang, J., and Barrett, I. P. (2022b). Implications of topological imbalance for representation learning on biomedical knowledge graphs. *Briefings in bioinformatics*, 23(5) :bbac279.

[Bordes et al., 2013] Bordes, A., Usunier, N., Garcia-Duran, A., Weston, J., and Yakhnenko, O. (2013). Translating embeddings for modeling multi-relational data. In Burges, C., Bottou, L., Welling, M., Ghahramani, Z., and Weinberger, K., editors, *Advances in Neural Information Processing Systems*, volume 26. Curran Associates, Inc.

[Boudin et al., 2023] Boudin, M., Diallo, G., Drancé, M., and Mougin, F. (2023). The oregano knowledge graph for computational drug repurposing. *Scientific Data*, 10(1) :871.

[Breiman, 2001] Breiman, L. (2001). Random forests. *Machine learning*, 45 :5–32.

[Breiman, 2002] Breiman, L. (2002). Manual on setting up, using, and understanding random forests v3. 1. *Statistics Department University of California Berkeley, CA, USA*, 1(58) :3–42.

[Briggs, 2003] Briggs, D. (2003). Environmental pollution and the global burden of disease. *British medical bulletin*, 68(1) :1–24.

[Broda et al., 2002] Broda, K. B., d'Avila Garcez, A., and Gabbay, D. (2002). Neural-symbolic learning system : foundations and applications.

[Bronstein et al., 2021] Bronstein, M. M., Bruna, J., Cohen, T., and Veličković, P. (2021). Geometric deep learning : Grids, groups, graphs, geodesics, and gauges. *arXiv preprint arXiv :2104.13478*.

[Brown et al., 2020] Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess, B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

[Bullmore and Sporns, 2009] Bullmore, E. and Sporns, O. (2009). Complex brain networks : graph theoretical analysis of structural and functional systems. *Nature reviews neuroscience*, 10(3) :186–198.

[Bundy and Wallen, 1984] Bundy, A. and Wallen, L. (1984). Breadth-first search. *Catalogue of artificial intelligence tools*, pages 13–13.

[Carlson et al., 2010] Carlson, A., Betteridge, J., Kisiel, B., Settles, B., Hruschka, E., and Mitchell, T. (2010). Toward an architecture for never-ending language learning. In *Proceedings of the AAAI conference on artificial intelligence*, volume 24, pages 1306–1313.

[Chen et al., 2021] Chen, Y., Minervini, P., Riedel, S., and Stenetorp, P. (2021). Relation prediction as an auxiliary training objective for improving multi-relational graph representations. In *Proceedings of the International Conference on Automated Knowledge Base Construction*.

[Cheng et al., 2020] Cheng, D., Yang, F., Wang, X., Zhang, Y., and 0001, L. Z. (2020). Knowledge graph-based event embedding framework for financial quantitative investments. In *Proceedings of the 43rd International ACM SIGIR conference on research and development in Information Retrieval*, pages 2221–2230. ACM.

[Consortium, 2022] Consortium, T. U. (2022). UniProt : the Universal Protein Knowledgebase in 2023. *Nucleic Acids Research*, 51(D1) :D523–D531.

[Dakshanamurthy et al., 2012] Dakshanamurthy, S., Issa, N. T., Assefnia, S., Seshasayee, A., Peters, O. J., Madhavan, S., Uren, A., Brown, M. L., and Byers, S. W. (2012). Predicting new indications for approved drugs using a proteochemometric method. *Journal of medicinal chemistry*, 55(15) :6832–6848.

[Das et al., 2017] Das, R., Dhuliawala, S., Zaheer, M., Vilnis, L., Durugkar, I., Krishnamurthy, A., Smola, A., and McCallum, A. (2017). Go for a walk and arrive at the answer : Reasoning over paths in knowledge bases using reinforcement learning. *ArXiv*, abs/1711.05851.

[De Vries, 1989] De Vries, P. H. (1989). Representation of scientific texts in knowledge graphs.

[Deakin et al., 2003] Deakin, S., Leviev, I., Guernier, S., and James, R. W. (2003). Simvastatin modulates expression of the pon1 gene and increases serum paraoxonase. *Arteriosclerosis, Thrombosis, and Vascular Biology*, 23(11) :2083–2089.

[Dettmers et al., 2018] Dettmers, T., Minervini, P., Stenetorp, P., and Riedel, S. (2018). Convolutional 2d knowledge graph embeddings. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*. AAAI Press.

[Devlin et al., 2019] Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). BERT : Pre-training of deep bidirectional transformers for language understanding. In Burstein, J., Doran, C., and Solorio, T., editors, *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

[Díaz-Uriarte and Alvarez de Andrés, 2006] Díaz-Uriarte, R. and Alvarez de Andrés, S. (2006). Gene selection and classification of microarray data using random forest. *BMC bioinformatics*, 7 :1–13.

[DiMasi et al., 2016] DiMasi, J. A., Grabowski, H. G., and Hansen, R. W. (2016). Innovation in the pharmaceutical industry : New estimates of r&d costs. *Journal of Health Economics*, 47 :20–33.

[Ding et al., 2022] Ding, K., Xu, Z., Tong, H., and Liu, H. (2022). Data augmentation for deep graph learning : A survey. *ACM SIGKDD Explorations Newsletter*, 24(2) :61–77.

[Djeddi et al., 2023] Djeddi, W. E., Hermi, K., Ben Yahia, S., and Diallo, G. (2023). Advancing drug–target interaction prediction : a comprehensive graph-based approach integrating knowledge graph embedding and protbert pretraining. *BMC Bioinformatics*, 24(1) :488.

[Drance et al., 2023] Drance, M., Mougin, F., Zemmari, A., and Diallo, G. (2023). Pretrained embeddings for enhancing multi-hop reasoning. In *International Joint Conference on Artificial Intelligence 2023 Workshop on Knowledge-Based Compositional Generalization*.

[Dube et al., 2023] Dube, P. S., Legoabe, L. J., and Beteck, R. M. (2023). Quinolone : a versatile therapeutic compound class. *Molecular Diversity*, 27(3) :1501–1526.

[Euler, 1741] Euler, L. (1741). Solutio problematis ad geometriam situs pertinentis. *Commentarii academiae scientiarum Petropolitanae*, Volume 8, pp. 128-140.

[Fakhraei et al., 2015] Fakhraei, S., Foulds, J., Shashanka, M., and Getoor, L. (2015). Collective spammer detection in evolving multi-relational social networks. In *Proceedings of the 21th acm sigkdd international conference on knowledge discovery and data mining*, pages 1769–1778.

[Gallant et al., 1990] Gallant, S. I. et al. (1990). Perceptron-based learning algorithms. *IEEE Transactions on neural networks*, 1(2) :179–191.

[Ghosh and Lerman, 2011] Ghosh, R. and Lerman, K. (2011). Parameterized centrality metric for network analysis. *Physical Review E—Statistical, Nonlinear, and Soft Matter Physics*, 83(6) :066118.

[Gilmer et al., 2017] Gilmer, J., Schoenholz, S. S., Riley, P. F., Vinyals, O., and Dahl, G. E. (2017). Neural message passing for quantum chemistry. pages 1263–1272.

[Gordon et al., 2012] Gordon, L. B., Kleinman, M. E., Miller, D. T., Neuberg, D. S., Giobbie-Hurder, A., Gerhard-Herman, M., Smoot, L. B., Gordon, C. M., Cleveland, R., Snyder, B. D., et al. (2012). Clinical trial of a farnesyltransferase inhibitor in children with hutchinson–gilford progeria syndrome. *Proceedings of the National Academy of Sciences*, 109(41) :16666–16671.

[Greene and Voight, 2016] Greene, C. S. and Voight, B. F. (2016). Pathway and network-based strategies to translate genetic discoveries into effective therapies. *Human Molecular Genetics*, 25(R2) :R94–R98.

[Grover and Leskovec, 2016] Grover, A. and Leskovec, J. (2016). node2vec : Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 855–864.

[Grover et al., 2015] Grover, M. P., Ballouz, S., Mohanasundaram, K. A., George, R. A., Goscinski, A., Crowley, T. M., Sherman, C. D., and Wouters, M. A. (2015). Novel therapeutics for coronary artery disease from genome-wide association study data. *BMC medical genomics*, 8 :1–11.

[Hamamoto et al., 2020] Hamamoto, R., Suvarna, K., Yamada, M., Kobayashi, K., Shinkai, N., Miyake, M., Takahashi, M., Jinnai, S., Shimoyama, R., Sakai, A., et al. (2020). Application of artificial intelligence technology in oncology : Towards the establishment of precision medicine. *Cancers*, 12(12) :3532.

[Hamilton et al., 2017] Hamilton, W., Ying, Z., and Leskovec, J. (2017). Inductive representation learning on large graphs. *Advances in neural information processing systems*, 30.

[Han and Xu, 2016] Han, S. and Xu, Y. (2016). Link prediction in microblog network using supervised learning with multiple features. *J. Comput.*, 11(1) :72–82.

[He et al., 2016] He, K., Zhang, X., Ren, S., and Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778.

[Himmelstein et al., 2017] Himmelstein, D. S., Lizee, A., Hessler, C., Brueggeman, L., Chen, S. L., Hadley, D., Green, A., Khankhanian, P., and Baranzini, S. E. (2017). Systematic integration of biomedical knowledge prioritizes drugs for repurposing. *eLife*, 6 :e26726.

[Hochreiter and Schmidhuber, 1997] Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8) :1735–1780.

[Horn, 1951] Horn, A. (1951). On sentences which are true of direct unions of algebras1. *The Journal of Symbolic Logic*, 16(1) :14–21.

[Howard and Bowles, 2012] Howard, J. and Bowles, M. (2012). The two most important algorithms in predictive modeling today. In *Strata Conference presentation, February*, volume 28.

[Huang et al., 2017] Huang, G., Liu, Z., Van Der Maaten, L., and Weinberger, K. Q. (2017). Densely connected convolutional networks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 4700–4708.

[Imison et al., 2016] Imison, C., Castle-Clarke, S., Watson, R., and Edwards, N. (2016). *Delivering the benefits of digital health care*. Nuffield Trust London.

[Jensen et al., 2012] Jensen, P. B., Jensen, L. J., and Brunak, S. (2012). Mining electronic health records : towards better research applications and clinical care. *Nature Reviews Genetics*, 13(6) :395–405.

[Kanehisa et al., 2016] Kanehisa, M., Sato, Y., Kawashima, M., Furumichi, M., and Tanabe, M. (2016). Kegg as a reference resource for gene and protein annotation. *Nucleic acids research*, 44(D1) :D457–D462.

[Kang et al., 2011] Kang, U., Papadimitriou, S., Sun, J., and Tong, H. (2011). Centralities in large networks : Algorithms and observations. In *Proceedings of the 2011 SIAM international conference on data mining*, pages 119–130. SIAM.

[Kiernan et al., 2011] Kiernan, M. C., Vucic, S., Cheah, B. C., Turner, M. R., Eisen, A., Hardiman, O., Burrell, J. R., and Zoing, M. C. (2011). Amyotrophic lateral sclerosis. *The lancet*, 377(9769) :942–955.

[Knox et al., 2010] Knox, C., Law, V., Jewison, T., Liu, P., Ly, S., Frolkis, A., Pon, A., Banco, K., Mak, C., Neveu, V., Djoumbou, Y., Eisner, R., Guo, A., and Wishart, D. S. (2010). Drugbank 3.0 : a comprehensive resource for 'omics' research on drugs. *Nucleic Acids Research*, 39 :D1035 – D1041.

[Kolata, 1998] Kolata, G. (1998). Us approves sale of impotence pill ; huge market seen. *The New York Times A*, 1.

[Krizhevsky et al., 2012] Krizhevsky, A., Sutskever, I., and Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, 25.

[Kuemmerle-Deschner et al., 2013] Kuemmerle-Deschner, J. B., Wittkowski, H., Tyrrell, P. N., Koetter, I., Lohse, P., Ummenhofer, K., Reess, F., Hansmann, S., Koitschev, A., Deuter, C., et al. (2013). Treatment of muckle-wells syndrome : analysis of two il-1-blocking regimens. *Arthritis research & therapy*, 15 :1–8.

[Kuhn et al., 2016] Kuhn, M., Letunic, I., Jensen, L. J., and Bork, P. (2016). The SIDER database of drugs and side effects. *Nucleic Acids Res.*, 44(D1) :D1075–9.

[Larsson and Flach, 2022] Larsson, D. and Flach, C.-F. (2022). Antibiotic resistance in the environment. *Nature Reviews Microbiology*, 20(5) :257–269.

[Lee and Yoon, 2017] Lee, C. H. and Yoon, H.-J. (2017). Medical big data : promise and challenges. *Kidney research and clinical practice*, 36(1) :3.

[Lee et al., 2019] Lee, J., Yoon, W., Kim, S., Kim, D., Kim, S., So, C. H., and Kang, J. (2019). BioBERT : a pre-trained biomedical language representation model for biomedical text mining. *Bioinformatics*, 36(4) :1234–1240.

[Lei et al., 2020a] Lei, D., Jiang, G., Gu, X., Mao, Y., and Ren, X. (2020a). Learning collaborative agents with rule guidance for knowledge graph reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing*, pages 8541–8547. Association for Computational Linguistics.

[Lei et al., 2020b] Lei, D., Jiang, G., Gu, X., Sun, K., Mao, Y., and Ren, X. (2020b). Learning collaborative agents with rule guidance for knowledge graph reasoning. pages 8541–8547.

[Liben-Nowell and Kleinberg, 2003] Liben-Nowell, D. and Kleinberg, J. (2003). The link prediction problem for social networks. In *Proceedings of the Twelfth International Conference on Information and Knowledge Management*, CIKM '03, page 556–559, New York, NY, USA. Association for Computing Machinery.

[Lin et al., 2018] Lin, X. V., Socher, R., and Xiong, C. (2018). Multi-hop knowledge graph reasoning with reward shaping. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3243–3253. Association for Computational Linguistics.

[Lin et al., 2015] Lin, Y., Liu, Z., Sun, M., Liu, Y., and Zhu, X. (2015). Learning entity and relation embeddings for knowledge graph completion. In *Proceedings of the AAAI conference on artificial intelligence*, volume 29.

[Liu et al., 2021] Liu, Y., Hildebrandt, M., Joblin, M., Ringsquandl, M., Raissouni, R., and Tresp, V. (2021). Neural multi-hop reasoning with logical rules on biomedical knowledge graphs. In *Eighteenth Extended Semantic Web Conference - Research Track*.

[Lu et al., 2022] Lu, H., Hu, H., and Lin, X. (2022). DensE : An enhanced non-commutative representation for knowledge graph embedding with adaptive semantic hierarchy. *Neurocomputing*, 476 :115–125.

[Lv et al., 2021a] Lv, X., Cao, Y., Hou, L., Li, J., Liu, Z., Zhang, Y., and Dai, Z. (2021a). Is multi-hop reasoning really explainable ? towards benchmarking reasoning interpretability. In Moens, M.-F., Huang, X., Specia, L., and Yih, S. W.-t., editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8899–8911, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

[Lv et al., 2021b] Lv, X., Cao, Y., Hou, L., Li, J., Liu, Z., Zhang, Y., and Dai, Z. (2021b). Is multi-hop reasoning really explainable ? Towards benchmarking reasoning interpretability. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8899–8911. Association for Computational Linguistics.

[Maglott et al., 2005] Maglott, D., Ostell, J., Pruitt, K. D., and Tatusova, T. (2005). Entrez gene : gene-centered information at ncbi. *Nucleic acids research*, 33(suppl_1) :D54–D58.

[Malas et al., 2019] Malas, T. B., Vlietstra, W. J., Kudrin, R., Starikov, S., Charrout, M., Roos, M., Peters, D. J. M., Kors, J. A., Vos, R., 't Hoen, P. A. C., van Mulligen, E. M., and Hettne, K. M. (2019). Drug prioritization using the semantic properties of a knowledge graph. *Scientific Reports*, 9(1) :6281.

[Meilicke et al., 2019] Meilicke, C., Chekol, M. W., Ruffinelli, D., and Stuckenschmidt, H. (2019). Anytime bottom-up rule learning for knowledge graph completion. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 3137–3143. International Joint Conferences on Artificial Intelligence Organization.

[Merwin et al., 2017] Merwin, S. J., Obis, T., Nunez, Y., and Re, D. B. (2017). Organophosphate neurotoxicity to the voluntary motor system on the trail of environment-caused amyotrophic lateral sclerosis : the known, the misknown, and the unknown. *Archives of Toxicology*, 91(8) :2939–2952.

[Milacic et al., 2023] Milacic, M., Beavers, D., Conley, P., Gong, C., Gillespie, M., Griss, J., Haw, R., Jassal, B., Matthews, L., May, B., Petryszak, R., Ragueneau, E., Rothfels, K., Sevilla, C., Shamovsky, V., Stephan, R., Tiwari, K., Varusai, T., Weiser, J., Wright, A., Wu, G., Stein, L., Hermjakob, H., and D'Eustachio, P. (2023). The Reactome Pathway Knowledgebase 2024. *Nucleic Acids Research*, 52(D1) :D672–D678.

[Miller, 1995] Miller, G. A. (1995). Wordnet : a lexical database for english. *Commun. ACM*, 38(11) :39–41.

[Min et al., 2013] Min, B., Grishman, R., Wan, L., Wang, C., and Gondek, D. (2013). Distant supervision for relation extraction with an incomplete knowledge base. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for*

Computational Linguistics : Human Language Technologies, pages 777–782. Association for Computational Linguistics.

[Morgan et al., 2018] Morgan, P., Brown, D. G., Lennard, S., Anderton, M. J., Barrett, J. C., Eriksson, U., Fidock, M., Hamrén, B., Johnson, A., March, R. E., Matcham, J., Mettetal, J., Nicholls, D. J., Platz, S., Rees, S., Snowden, M. A., and Pangalos, M. N. (2018). Impact of a five-dimensional framework on r&d productivity at astrazeneca. *Nature reviews. Drug discovery*, 17(3) :167—181.

[Muhlhausler et al., 2009] Muhlhausler, B. S., Morrison, J. L., and McMillen, I. C. (2009). Rosiglitazone Increases the Expression of Peroxisome Proliferator-Activated Receptor-γ Target Genes in Adipose Tissue, Liver, and Skeletal Muscle in the Sheep Fetus in Late Gestation. *Endocrinology*, 150(9) :4287–4294.

[Narayanan et al., 2017] Narayanan, A., Chandramohan, M., Venkatesan, R., Chen, L., Liu, Y., and Jaiswal, S. (2017). graph2vec : Learning distributed representations of graphs. *arXiv preprint arXiv :1707.05005*.

[Nickel et al., 2011] Nickel, M., Tresp, V., and Kriegel, H.-P. (2011). A three-way model for collective learning on multi-relational data. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, page 809–816. Omnipress.

[Olayan et al., 2017] Olayan, R. S., Ashoor, H., and Bajic, V. B. (2017). DDR : efficient computational method to predict drug–target interactions using graph mining and machine learning approaches. *Bioinformatics*, 34(7) :1164–1173.

[Ostrov, 2004] Ostrov, B. (2004). Renewed life for old drugs. *San Jose Mercury News E*, 1.

[Page et al., 1999] Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The PageRank Citation Ranking : Bringing order to the web. Technical Report 1999-66, Stanford InfoLab.

[Pan and Wang, 2021] Pan, Z. and Wang, P. (2021). Hyperbolic hierarchy-aware knowledge graph embedding for link prediction. In *Findings of the Association for Computational Linguistics : EMNLP 2021*, pages 2941–2948. Association for Computational Linguistics.

[Pandey et al., 2019] Pandey, B., Bhanodia, P. K., Khamparia, A., and Pandey, D. K. (2019). A comprehensive survey of edge prediction in social networks : Techniques, parameters and challenges. *Expert Systems with Applications*, 124 :164–181.

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

[Pemmaraju and Skiena, 2003] Pemmaraju, S. and Skiena, S. (2003). *Computational discrete mathematics : Combinatorics and graph theory with mathematica®*. Cambridge university press.

[Petric Maretic et al., 2019] Petric Maretic, H., El Gheche, M., Chierchia, G., and Frossard, P. (2019). Got : an optimal transport framework for graph comparison. *Advances in Neural Information Processing Systems*, 32.

[Pollack and Wiebenson, 1960] Pollack, M. and Wiebenson, W. (1960). Solutions of the shortest-route problem—a review. *Operations Research*, 8(2) :224–230.

[Pourhabibi et al., 2020] Pourhabibi, T., Ong, K.-L., Kam, B. H., and Boo, Y. L. (2020). Fraud detection : A systematic literature review of graph-based anomaly detection approaches. *Decision Support Systems*, 133 :113303.

[Prasad et al., 2006] Prasad, A. M., Iverson, L. R., and Liaw, A. (2006). Newer classification and regression tree techniques : bagging and random forests for ecological prediction. *Ecosystems*, 9 :181–199.

[Pushpakom et al., 2019a] Pushpakom, S., Iorio, F., Eyers, P. A., Escott, K. J., Hopper, S., Wells, A., Doig, A., Guilliams, T., Latimer, J., McNamee, C., et al. (2019a). Drug repurposing : progress, challenges and recommendations. *Nature reviews Drug discovery*, 18(1) :41–58.

[Pushpakom et al., 2019b] Pushpakom, S., Iorio, F., Eyers, P. A., Escott, K. J., Hopper, S., Wells, A., Doig, A., Guilliams, T., Latimer, J., McNamee, C., et al. (2019b). Drug repurposing : progress, challenges and recommendations. *Nature reviews Drug discovery*, 18(1) :41–58.

[Qian et al., 2021] Qian, J., Li, G., Atkinson, K., and Yue, Y. (2021). Understanding negative sampling in knowledge graph embedding. *International Journal of Artificial Intelligence & Applications*, 12 :71–81.

[Rando et al., 2019] Rando, A., De La Torre, M., Martinez-Muriana, A., Zaragoza, P., Musaro, A., Hernandez, S., Navarro, X., Toivonen, J. M., and Osta, R. (2019). Chemotherapeutic agent 5-fluorouracil increases survival of sod1 mouse model of als. *PLoS One*, 14(1).

[Rebuffi et al., 2017] Rebuffi, S.-A., Bilen, H., and Vedaldi, A. (2017). Learning multiple visual domains with residual adapters. *Advances in neural information processing systems*, 30.

[Roessler et al., 2021] Roessler, H. I., Knoers, N. V., van Haelst, M. M., and van Haaften, G. (2021). Drug repurposing for rare diseases. *Trends in pharmacological sciences*, 42(4) :255–267.

[Rudin et al., 2022] Rudin, C., Chen, C., Chen, Z., Huang, H., Semenova, L., and Zhong, C. (2022). Interpretable machine learning : Fundamental principles and 10 grand challenges. *Statistics Surveys*, 16(none) :1 – 85.

[Russakovsky et al., 2015] Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L.

(2015). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)*, 115(3) :211–252.

[Russell et al., 2013] Russell, A. P., Wada, S., Vergani, L., Hock, M. B., Lamon, S., Léger, B., Ushida, T., Cartoni, R., Wadley, G. D., Hespel, P., Kralli, A., Soraru, G., Angelini, C., and Akimoto, T. (2013). Disruption of skeletal muscle mitochondrial network genes and mirnas in amyotrophic lateral sclerosis. *Neurobiology of Disease*, 49 :107–117.

[Sardo et al., 2005] Sardo, M. A., Campo, S., Bonaiuto, M., Bonaiuto, A., Saitta, C., Trimarchi, G., Castaldo, M., Bitto, A., Cinquegrani, M., and Saitta, A. (2005). Antioxidant effect of atorvastatin is independent of pon1 gene t(–107)c, q192r and l55m polymorphisms in hypercholesterolaemic patients. *Current Medical Research and Opinion*, 21(5) :777–784. PMID : 15969877.

[Schaeffer, 2007] Schaeffer, S. E. (2007). Graph clustering. *Computer science review*, 1(1) :27–64.

[Schlichtkrull et al., 2018] Schlichtkrull, M., Kipf, T. N., Bloem, P., Van Den Berg, R., Titov, I., and Welling, M. (2018). Modeling relational data with graph convolutional networks. In *The semantic web : 15th international conference, ESWC 2018, Heraklion, Crete, Greece, June 3–7, 2018, proceedings 15*, pages 593–607. Springer.

[Schmidt, 2020] Schmidt, C. W. (2020). Into the black box : What can machine learning offer environmental health research ? *Environmental Health Perspectives*, 128(2) :022001.

[Schneider, 1973] Schneider, E. W. (1973). Course modularization applied : The interface system and its implications for sequence control and data analysis.

[Schramowski et al., 2020] Schramowski, P., Stammer, W., Teso, S., Brugger, A., Herbert, F., Shao, X., Luigs, H.-G., Mahlein, A.-K., and Kersting, K. (2020). Making deep neural networks right for the right scientific reasons by interacting with their explanations. *Nature Machine Intelligence*, 2(8) :476–486.

[Shannon, 1948] Shannon, C. E. (1948). A mathematical theory of communication. *The Bell system technical journal*, 27(3) :379–423.

[Shi and Weninger, 2018] Shi, B. and Weninger, T. (2018). Open-world knowledge graph completion. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

[Shorten and Khoshgoftaar, 2019] Shorten, C. and Khoshgoftaar, T. (2019). A survey on image data augmentation for deep learning. *Journal of Big Data*, 6.

[Simonyan and Zisserman, 2014] Simonyan, K. and Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv :1409.1556*.

[Sousa et al., 2024] Sousa, R. T., Silva, S., and Pesquita, C. (2024). Explaining protein–protein interactions with knowledge graph-based semantic similarity. *Computers in Biology and Medicine*, 170 :108076.

[Sun et al., 2019] Sun, Z., Deng, Z.-H., Nie, J.-Y., and Tang, J. (2019). RotatE : Knowledge graph embedding by relational rotation in complex space. In *Proceedings of the International Conference on Learning Representations*.

[Svetnik et al., 2003] Svetnik, V., Liaw, A., Tong, C., Culberson, J. C., Sheridan, R. P., and Feuston, B. P. (2003). Random forest : a classification and regression tool for compound classification and qsar modeling. *Journal of chemical information and computer sciences*, 43(6) :1947–1958.

[Tang et al., 2018] Tang, C., Garreau, D., and von Luxburg, U. (2018). When do random forests fail ? *Advances in neural information processing systems*, 31.

[Thor and Katofiasc, 1995] Thor, K. B. and Katofiasc, M. A. (1995). Effects of duloxetine, a combined serotonin and norepinephrine reuptake inhibitor, on central neural control of lower urinary tract function in the chloralose-anesthetized female cat. *Journal of Pharmacology and Experimental Therapeutics*, 274(2) :1014–1024.

[Topping et al., 2021] Topping, J., Di Giovanni, F., Chamberlain, B. P., Dong, X., and Bronstein, M. M. (2021). Understanding over-squashing and bottlenecks on graphs via curvature. *arXiv preprint arXiv :2111.14522*.

[Toutanova et al., 2015] Toutanova, K., Chen, D., Pantel, P., Poon, H., Choudhury, P., and Gamon, M. (2015). Representing text for joint embedding of text and knowledge bases. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1499–1509. Association for Computational Linguistics.

[Trouillon et al., 2016] Trouillon, T., Welbl, J., Riedel, S., Gaussier, E., and Bouchard, G. (2016). Complex embeddings for simple link prediction. In *Proceedings of The 33rd International Conference on Machine Learning*, volume 48 of *Proceedings of Machine Learning Research*, pages 2071–2080. PMLR.

[van Veen, 2019] van Veen, T. (2019). Wikidata. *Information Technology and Libraries*, 38(2) :72–81.

[Varian, 2014] Varian, H. R. (2014). Big data : New tricks for econometrics. *Journal of Economic Perspectives*, 28(2) :3–28.

[Vaswani, 2017] Vaswani, A. (2017). Attention is all you need. *Advances in Neural Information Processing Systems*.

[Veličković et al., 2019] Veličković, P., Fedus, W., Hamilton, W. L., Liò, P., Bengio, Y., and Hjelm, R. D. (2019). Deep Graph Infomax.

[Velickovic et al., 2019] Velickovic, P., Fedus, W., Hamilton, W. L., Liò, P., Bengio, Y., and Hjelm, R. D. (2019). Deep graph infomax. *ICLR (Poster)*, 2(3) :4.

[Walsh et al., 2020] Walsh, B., Mohamed, S. K., and Nováček, V. (2020). Biokg : A knowledge graph for relational learning on biological data. In *Proceedings of the 29th*

ACM International Conference on Information & Knowledge Management, CIKM '20, page 3173–3180, New York, NY, USA. Association for Computing Machinery.

[Wang et al., 2020] Wang, C., Wang, C., Wang, Z., Ye, X., and Yu, P. S. (2020). Edge2vec : Edge-based social network embedding. *ACM Trans. Knowl. Discov. Data*, 14(4).

[Wang et al., 2021] Wang, X., Liu, N., Han, H., and Shi, C. (2021). Self-supervised heterogeneous graph neural network with co-contrastive learning. In *Proceedings of the 27th ACM SIGKDD Conference on Knowledge Discovery & Data Mining*, KDD '21, page 1726–1736, New York, NY, USA. Association for Computing Machinery.

[Wang and Sukthankar, 2013] Wang, X. and Sukthankar, G. (2013). Link prediction in multi-relational collaboration networks. In *Proceedings of the 2013 IEEE/ACM international conference on advances in social networks analysis and mining*, pages 1445–1447.

[Wei and Denny, 2015] Wei, W.-Q. and Denny, J. C. (2015). Extracting research-quality phenotypes from electronic health records to support precision medicine. *Genome medicine*, 7 :1–14.

[Weisstein, 2000] Weisstein, E. W. (2000). Gini coefficient. *https ://mathworld. wolfram. com/*.

[Wexler, 2017] Wexler, R. (2017). Computers are harming justice. *The New York Times*, page A27(L). A27(L).

[Whirl-Carrillo et al., 2021] Whirl-Carrillo, M., Huddart, R., Gong, L., Sangkuhl, K., Thorn, C. F., Whaley, R., and Klein, T. E. (2021). An evidence-based framework for evaluating pharmacogenomics knowledge for personalized medicine. *Clin. Pharmacol. Ther.*, 110(3) :563–572.

[Williams, 1992] Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8(3–4) :229–256.

[Wouters et al., 2020] Wouters, O. J., McKee, M., and Luyten, J. (2020). Estimated Research and Development Investment Needed to Bring a New Medicine to Market, 2009-2018. *JAMA*, 323(9) :844–853.

[Wu et al., 2022] Wu, L., Lin, H., Huang, Y., and Li, S. Z. (2022). Knowledge distillation improves graph structure augmentation for graph neural networks. In Koyejo, S., Mohamed, S., Agarwal, A., Belgrave, D., Cho, K., and Oh, A., editors, *Advances in Neural Information Processing Systems*, volume 35, pages 11815–11827. Curran Associates, Inc.

[Wu et al., 2018] Wu, M., Huang, Y., Zhao, L., and He, Y. (2018). Link prediction based on random forest in signed social networks. In *2018 10th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, volume 2, pages 251–256. IEEE.

[Xiong et al., 2017] Xiong, W., Hoang, T., and Wang, W. Y. (2017). DeepPath : A reinforcement learning method for knowledge graph reasoning. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 564–573. Association for Computational Linguistics.

[Yang et al., 2014a] Yang, B., tau Yih, W., He, X., Gao, J., and Deng, L. (2014a). Embedding entities and relations for learning and inference in knowledge bases. *CoRR*, abs/1412.6575.

[Yang et al., 2014b] Yang, B., Yih, W.-t., He, X., Gao, J., and Deng, L. (2014b). Embedding entities and relations for learning and inference in knowledge bases. *arXiv preprint arXiv :1412.6575*.

[Yang et al., 2019] Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R., and Le, Q. V. (2019). *XLNet : generalized autoregressive pretraining for language understanding*. Curran Associates Inc., Red Hook, NY, USA.

[You et al., 2021] You, Y., Chen, T., Shen, Y., and Wang, Z. (2021). Graph contrastive learning automated. In *International Conference on Machine Learning*, pages 12121–12132. PMLR.

[Yu et al., 2024a] Yu, J., Ge, Q., Li, X., and Zhou, A. (2024a). Heterogeneous graph contrastive learning with meta-path contexts and adaptively weighted negative samples. *IEEE Transactions on Knowledge and Data Engineering*, page 1–13.

[Yu et al., 2024b] Yu, M., Xu, J., Dutta, R., Trapp, B., Pieper, A. A., and Cheng, F. (2024b). Network medicine informed multi-omics integration identifies drug targets and repurposable medicines for amyotrophic lateral sclerosis. *bioRxiv*.

[Zeng et al., 2018] Zeng, X., Zhang, P., He, W., Qin, C., Chen, S., Tao, L., Wang, Y., Tan, Y., Gao, D., Wang, B., et al. (2018). Npass : natural product activity and species source database for natural product research, discovery and tool development. *Nucleic acids research*, 46(D1) :D1217–D1222.

[Zhang et al., 2019a] Zhang, C., Song, D., Huang, C., Swami, A., and Chawla, N. V. (2019a). Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining*, pages 793–803.

[Zhang et al., 2019b] Zhang, C., Song, D., Huang, C., Swami, A., and Chawla, N. V. (2019b). Heterogeneous graph neural network. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '19, page 793–803, New York, NY, USA. Association for Computing Machinery.

[Zhang and Wiemann, 2009] Zhang, J. D. and Wiemann, S. (2009). Kegggraph : a graph approach to kegg pathway in r and bioconductor. *Bioinformatics*, 25(11) :1470–1471.

[Zhang et al., 2018] Zhang, M., Cui, Z., Neumann, M., and Chen, Y. (2018). An end-to-end deep learning architecture for graph classification. In *Proceedings of the AAAI conference on artificial intelligence*, volume 32.

[Zhang et al., 2021] Zhang, R., Hristovski, D., Schutte, D., Kastrin, A., Fiszman, M., and Kilicoglu, H. (2021). Drug repurposing for covid-19 via knowledge graph completion. *Journal of Biomedical Informatics*, 115 :103696.

[Zhang et al., 2019c] Zhang, S., Tay, Y., Yao, L., and Liu, Q. (2019c). Quaternion knowledge graph embeddings. In *Proceedings of the 33rd International Conference on Neural Information Processing Systems*. Curran Associates Inc.

[Zhang and Gant, 2009] Zhang, S.-D. and Gant, T. W. (2009). sscmap : an extensible java application for connecting small-molecule drugs using gene-expression signatures. *BMC bioinformatics*, 10 :1–4.

[Zhao et al., 2023] Zhao, B.-W., Su, X.-R., Hu, P.-W., Huang, Y.-A., You, Z.-H., and Hu, L. (2023). iGRLDTI : an improved graph representation learning method for predicting drug–target interactions over heterogeneous biological information network. *Bioinformatics*, 39(8) :btad451.

[Zhou et al., 2022] Zhou, J., Xie, C., Wen, Z., Zhao, X., and Xuan, Q. (2022). Data augmentation on graphs : a technical survey. *arXiv preprint arXiv :2212.09970*.

[Zhu et al., 2021] Zhu, Y., Xu, Y., Yu, F., Liu, Q., Wu, S., and Wang, L. (2021). Graph contrastive learning with adaptive augmentation. In *Proceedings of the Web Conference 2021*, WWW '21, page 2069–2080, New York, NY, USA. Association for Computing Machinery.

[Ádlí ar Rúmí, 840] Ádlí ar Rúmí, A. (840). Kitab ash-shatranj.