UNIVERSIDADE DE LISBOA

FACULDADE DE CIÊNCIAS

DEPARTAMENTO DE INFORMÁTICA

Comparing Link Prediction and Classification for Gene-Disease Association Discovery

Catarina Salema Canastra

Mestrado em Ciência de Dados

Dissertação orientada por:

Prof.ª Doutora Cátia Luísa Santana Calisto Pesquita

2024

”In life, there is nothing to be afraid of, but to be understood.”

— Marie Curie

Acknowledgements

This work was funded by Fundação para a Ciência e a Tecnologia, Portugal, through the LASIGE Research Unit under agreement No. UIDB/00408/2020 and UIDP/00408/2020. The KATY project also supported this work, which has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No. 101017453.

First, I would like to thank my supervisor, Prof. Cátia Pesquita, for guiding my work and always supporting me. Without her dedication, this dissertation would not have been possible. She is an authentic role model for her work methodology and dedication to Science, the scientific community and her students.

I am also grateful to my parents and grandmother for their unconditional support. To my boyfriend and friend Rita, I must appreciate their availability to review this work and provide insights about it. Finally, I am also thankful to my colleagues at LASIGE and LISEDA for their companionship through our faculty.

Abstract

The discovery of gene-disease links is an important challenge in biological and biomedical domains, as it presents opportunities in tasks such as disease detection and drug repurposing. *Machine Learning* approaches that predict gene-disease associations significantly accelerate this process by leveraging biological knowledge represented in *ontologies* and the structure of *knowledge graphs* to organize data.

State-of-the-art approaches for gene-disease association typically use Knowledge Graph Embeddings and other Machine Learning algorithms, modeling the problem as a pair binary classification task. Although this is generally the logic behind a Machine Learning approach, the effectiveness of *link classification* approaches is limited by the need to generate negative examples, the absence of relationships between genes and diseases, and because only some Knowledge Graph Embeddings are able to directly predict gene-disease associations.

This dissertation explores the differences between addressing the gene-disease association problem as a link classification task and a *link prediction* task. We compare means of combining vectors and classification algorithms for the link classification approach. We also analyzed the influence of considering several knowledge graph embeddings in both the link classification and link prediction approaches. The methods were evaluated using biomedical data sources such as DisGeNET and popular ontologies.

Our results show that enriching the semantic representation of disease does not support better performance of link classification methods and the performance of link prediction methods in predicting disease-linked genes. However, it does support better performance of link prediction methods in predicting gene-linked diseases. The results also suggest that link prediction methods better explore the semantic richness encoded in knowledge graphs through various ontologies and additional links between ontology classes.

Employing link prediction over link classification provides advantages across design aspects and techniques. For instance, link prediction leverages relationships between target entities within knowledge graphs and does not require the synthetic generation of negative examples. While link prediction methods offer an end-to-end approach that directly generates predictions from the learned embeddings, link classification methods require integrating various Machine Learning methods with strategies to combine the embeddings, leading to increased complexity and potential loss of information.

Keywords: Ontologies, Knowledge Graphs, Machine Learning, Link Classification, Link Prediction

Resumo Alargado

A descoberta de ligações gene-doença é um desafio importante nos domínios biológico e biomédico, pois apresenta oportunidades em tarefas como a prevenção de doenças, a sua rápida deteção, diagnóstico e reorientação de medicamentos. Recentemente, têm sido propostos vários métodos de aprendizagem automática para prever associações entre genes e doenças apoiados na teoria de redes, construindo redes biológicas. Estes métodos, são geralmente limitados a vizualizações agnósticas dos dados, não tendo acesso ao seu contexto e significado, mas é reconhecido que o desempenho dos métodos de aprendizagem automática pode melhorar significativamente quando o contexto e as relações entre os dados são tidos em conta.

Na última década, a explosão na complexidade, no tamanho e heterogeneidade dos dados biológicos motivou um novo panorama de dados semânticos, onde milhões de entidades biológicas descritas semanticamente (isto quer dizer, com significado) estão disponíveis em grafos de conhecimento. Os grafos de conhecimento são estruturas de dados que representam entidades do mundo real e as suas relações por meio de nós e ligações (arestas) entre esses, de uma forma que incorpore o contexto e significado proveniente das ontologias. Uma ontologia é uma especificação formal e explícita sobre um domínio em específico, na qual cada classe (ou conceito) está precisamente definida e as relações entre classes estão parametrizadas ou restringidas.

Apesar dos avanços facilitados pelas ontologias na investigação biológica e biomédica, a maioria dos trabalhos apresenta uma lacuna significativa na forma como as doenças são representadas. Normalmente, as doenças são representadas pelos seus fenótipos, as características ou traços observáveis, sem uma descrição detalhada da doença em si. Esta abordagem ignora a complexidade e o contexto completo das doenças, incluindo conceitos de doenças relacionadas no vocabulário médico. Para além disso, a integração de ontologias em fluxos de trabalhos biológicos e biomédicos é acompanhada pelo desafio de integrar as várias descrições para uma mesma classe quando são combinadas múltiplas ontolodias. A falha na integração destas descrições pode resultar em inconsistências e redundância na análise dos dados, dificultando a capacidade de capturar todo o espetro do conhecimento biológico.

A crescente integração de ontologias biomédicas na forma de grafos de conhecimento tem impulsionado o desenvolvimento de métodos combinados de aprendizagem automática. Um desafio significativo é transformar os dados provenientes dos grafos numa representação que possa ser processada pelos algoritmos populares de aprendizagem automática. Atualmente, os métodos de aprendizagem automática dependem de heurísticas definidas pelo utilizador para extrair recursos que codificam informações estruturais do grafo, como as *degree statistics* e as *kernel functions*. No entanto, estas abordagens podem não

capturar toda a semântica subjacente aos grafos uma vez que se baseiam em contagens. Uma alternativa consiste em transformar as entidades e as relações dos grafos em vetores que capturam a semântica e a informação estrutural do grafo original utilizando *Knowledge Graph Embeddings*. Deste modo, as abordagens mais recentes para prever associações entre genes e doenças baseiam-se neste modelos para gerar representações e em algoritmos populares de aprendizagem automática para prever associações.

O problema da associação gene-doença é tipicamente modelado como uma tarefa de classificação binária de pares. Embora esta seja a lógica subjacente a uma abordagem de aprendizagem automática, a eficácia das abordagens de classificação de ligações é limitada pela necessidade de gerar exemplos negativos, pela ausência de relações entre genes e doenças, e porque não é possível prever diretamente associações entre genes e doenças a partir de alguns *Knowledge Graph Embeddings*. Nesta dissertação, investigamos as diferenças entre abordar um problema como uma tarefa de classificação de ligações e uma tarefa de previsão de ligações. A classificação de ligações identifica e classifica relações inicialmente não representadas entre pares de nós no grafo, enquanto a previsão de ligações concentra-se na deteção de relações em falta ou não observadas entre entidades num grafo.

Foi aplicada uma metodologia de classificação de ligações e desenvolvida uma abordagem de previsão de ligações. A metodologia de classificação de ligações e a estratégia de previsão de ligações possuem as duas primeiras e a última etapa em comum: a criação de vários grafos de conhecimento, a aplicação de *Knowledge Graph Embeddings* e a avaliação do desempenho dos modelos, respetivamente. A formulação de grafos de conhecimento diferentes permitiu analisar a riqueza semântica de várias perspetivas, como ter as entidades descritas com mais ontologias e ter mais ligações entre classes das ontologias. Os grafos de conhecimento desenhados para as experiências integraram: as ontologias *Gene Ontology*, *Human Phenotype Ontology* e *Human Disease Ontology*, bem como os anotações dessas; definições lógicas e mapeamentos entre a *Gene Ontology* e a *Human Phenotype Ontology*; os genes, as doenças e suas associações da DisGeNET (nos grafos de conhecimento da previsão de ligações).

Na metodologia de classificação de ligações, depois de transformadas as entidades e relações dos grafos de conhecimento em vetores utilizando *Knowledge Graph Embeddings*, os vetores dos genes e das doenças são combinados em pares gene-doença. Os vetores foram combinados de cinco maneiras diferentes: somando, calculando a média e o produto Hadamard (nestes casos obtem-se um vetor); e calculando as distâncias *Weighted_L1* and *Weighted_L2* (nestes casos obtem-se um escalar). Os pares gene-doença foram divididos em dez partes iguais, onde os algoritmos populares de aprendizagem automática ficam com nove partes para treino e uma parte para teste. Os algoritmos populares de aprendizagem automática utilizados foram o Naive Bayes, Multi-Layer Perceptron, Random Forest e o Extreme Gradient Boosting. No fim, são calculadas a mediana e a distância interquartil da precisão, *recall* e *Weighted Average of F-measures* dos classificadores.

Na abordagem de previsão de ligações, depois de transformar os grafos de conhecimento em vetores, estes são passados na *scoring function* de cada *Knowledge Graph Embeddings*. Se o objetivo for prever as doenças associadas a um gene, a *scoring function* recebe o vetor do gene, o vetor correspondente à relação ”associação” e o vetor de uma entidade candidata a complementar aquela ligação. O resultado final é um valor que reflete a probabilidade da entidade candidata estar associada ao gene. Valores mais altos indicam que o modelo prevê com maior confiança uma determinada ligação real no grafo. Após

várias entidades, obtem-se uma lista de entidades candidatas a complementar uma determinada ligação real no grafo. Esta lista é depois filtrada para conter só doenças (seguindo o exemplo), são selecionados os primeiros 100 resultados, guardadas as classificações das doenças que estão efetivamente ligadas ao gene, e são calculados o Hits@10, 30 e 100.

Os resultados demonstraram que grafos de conhecimento com definições lógicas ou mapeamentos suportam melhor desempenho dos modelos do que grafos simples apenas com as ontologias na classificação de ligações e na previsão de ligações. As anotações da *Human Phenotype Ontology* para os genes suportam melhor desempenho dos métodos de previsão de ligações. Enriquecer a representação semântica das doenças com uma ontologia que descreve as doenças humanas não suporta melhor desempenho dos métodos de classificação de ligações, e dos métodos de previsão de ligações na previsão dos genes associados a uma doença. No entanto, suporta melhor desempenho dos métodos de previsão de ligações na previsão das doenças associadas a um gene. A distinção nos resultados sugere que os métodos de previsão de ligações são melhores a explorar a riqueza semântica incorporada nos grafos de conhecimento através de várias ontologias e de ligações adicionais entre classes das ontologias.

Abordar um problema como uma tarefa de previsão de ligações em vez de abordar um problema como uma tarefa de classificação de ligações oferece diversas vantagens em vários aspetos de desenho e técnicas. Enquanto na classificação de ligações só os algoritmos populares de aprendizagem automática conhecem as ligações entre genes e doenças, os métodos de previsão de ligações aproveitam essas ligações nos grafos, o que permite explorar outro aspeto da riqueza semântica. Para além disso, na previsão de ligações não é necessário gerar exemplos negativos. Na metodologia de classificação de ligações é necessário integrar vários métodos. Ao contrário, na previsão de ligações os algoritmos permitem gerar as previsões finais. As vantagens de abordar um problema como uma tarefa de previsão de ligações reside na sua capacidade de explorar a riqueza semântica incorporada nos grafos de conhecimento, descobrir ligações ocultas entre entidades e facilitar uma modelagem preditiva mais precisa e abrangente no campo da biologia computacional.

Palavras Chave: Ontologias, Grafos de Conhecimento, Aprendizagem Automática, Classificação de Ligações, Previsão de Ligações

Contents

List of Figures

List of Tables

Acronyms

CV	Cross-Validation
DO	Human Disease Ontology
GO	Gene Ontology
HPO	Human Phenotype Ontology
IQR	Interquartile Range
KG	Knowledge Graph
KGE	Knowledge Graph Embeddings
LD	Logical Definitions
ML	Machine Learning
MR	Mean Rank
MRR	Mean Reciprocal Rank
MLP	Multi-Layer Perceptron
NB	Naive Bayes
OBO	Open Biomedical Ontologies
OWL	Web Ontology Language
RDF	Resource Description Framework
RF	Random Forest
WAF	Weighted Average of F-measures
XGB	Extreme Gradient Boosting

Chapter 1

Introduction

The present chapter introduces the contents of this dissertation. **Context and motivation** for this body of work are presented below in Section 1.1. The scope is introduced in Section 1.2 through **objectives and contributions**. To close off the chapter, the **dissertation outline** is provided in Section 1.3.

1.1 Context and Motivation

Identifying gene-disease associations is an important challenge in biological and biomedical domains, as it presents opportunities in tasks such as understanding disease origin and providing accurate disease prevention, diagnosis and treatment response. Various Machine Learning (ML) approaches have recently been proposed to predict gene-disease associations. These approaches are often grounded in network theory, leveraging biological networks [89, 82, 78, 76].

The explosion in complexity, size and heterogeneity of biological data has motivated the integration of ontologies into ML approaches to enhance the predictive power and interoperability of gene-disease association methods [86, 9, 10, 80, 34]. By incorporating ontological representations, such as provided by the Gene Ontology (GO), researchers have a definition of genes, along with their properties and relationships between them [25].

Despite the advancements facilitated by ontologies in biological and biomedical research, most works have a significant shortcoming in how diseases are represented. Typically, diseases are represented by their phenotypes, the observable characteristics or traits, without a detailed description of the disease itself. This approach misses the complexity and full context of the diseases, including related disease concepts within the medical vocabulary.

Integrating ontologies in biological and biomedical analysis pipelines raises several data-level challenges. A significant challenge is integrating the various descriptions for the same thing when multiple ontologies are combined. Failure to integrate these descriptions can result in inconsistencies and redundancies in data analysis, hindering the ability to capture the full spectrum of biological knowledge [50].

When integrating ontologies with other data types, Knowledge Graphs (KG)s represent a graph-based representation of knowledge. These graphs describe real-world entities and their interrelations through links to ontology concepts [24]. Nevertheless, popular ML algorithms often struggle with the complexity of interconnected biological data represented by KGs due to their inherent graph structure [31].

Unlike tabular data commonly used in traditional ML approaches, KGs encode rich, interconnected relationships between entities, making it challenging for popular ML algorithms to capture and analyze such complex patterns. Additionally, popular ML algorithms cannot effectively incorporate the semantic context embedded within KGs, limiting their ability to extract meaningful insights from the interconnected biological data [30].

Graph embedding approaches have emerged as alternative strategies to handle such complex relational data. Knowledge Graph Embeddings (KGE) have gained particular prominence due to their ability to capture the semantics represented by KGs through creating lighter representations of the data in a continuous space [76]. The resulting embeddings are insightful for graph-based ML problems, as traditional categories do not capture the complex patterns inherent in graph-structured data [30].

Link classification and link prediction are fundamental graph-based ML tasks that differ in conceptualizing and addressing the problem. While *link classification* captures non-modeled relationships between pairs of nodes, *link prediction* focuses on detecting relationships between nodes [31]. These variations in representation present an opportunity to enhance understanding and predictive capabilities within complex systems, such as gene-disease associations.

1.2 Objectives and Contributions

This dissertation aims to investigate the differences between approaching the gene-disease association problem as a *link classification* task and a *link prediction* task within biomedical KGs. This investigation addresses two primary research questions:

- **Research Question 1:** Do link prediction methods explore semantic richness better than link classification methods for gene-disease association prediction?
- **Research Question 2:** Does enriching the semantic representation of diseases improve performance in gene-disease association prediction?

When addressing these questions, semantic richness plays a pivotal role. Semantic richness refers to the depth and complexity of the semantic (i.e. meaning) relationships and representations within the KG. This encompasses various aspects, such as the granularity of concepts, inference capabilities, contextual information, and domain specificity. In this work, semantic richness is approached from multiple perspectives:

- **Link classification versus link prediction:** considering both link classification and link prediction approaches allows to explore how different methods leverage semantic richness. While link classification combines KGE with popular ML algorithms, offering insights into the semantic context

of individual entities, link prediction solely utilizing KGE provides a lens into the broader network structure and relational semantics;

- **Utilization of multiple biomedical ontologies:** investigating KGE over multiple biomedical ontologies enhances semantic richness by incorporating diverse domain-specific knowledge and capturing complex relationships between entities across various biomedical domains — in particular, this work focuses on additional enrichment of disease representation;

- **Complexity of KGs:** comparing simpler KGs to those enriched with additional links between specific ontologies and incorporating annotations, such as HPO annotations for genes, provides insights into how the complexity of the KGs as well as the prediction accuracy.

We explored the ontologies GO, Human Phenotype Ontology (HPO) and Human Disease Ontology (DO). GO and HPO are two of the most popular biomedical ontologies, and DO is the formalist biomedical ontology of human disease. Regarding data from a database, we explored the genes, diseases, and their corresponding associations from DisGeNET [58].

The extensive exploration for the assessment of the approaches provided the following contributions, present in this dissertation:

1. A novel link prediction approach to learning KGE models that obtain the top 100 candidate entities to complete a relationship between one entity and another type of entity of interest;

2. A systematic assessment of the current methodologies employing curated gene-disease associations from DisGeNET and GO, HPO and DO ontologies;

3. Poster presentation with preliminary results presented in the 8th LASIGE Workshop.

The code and some of the data used for the experiments presented in this dissertation are provided at a GitHub Repository.

1.3 Dissertation Outline

This dissertation is organized into seven chapters (including the current chapter).

Chapter 2 presents the fundamental concepts to understand this document.

Chapter 3 covers the related work on the gene-disease association problem using ontologies or KGs. In addition, the difficulties in predicting gene-disease associations and the importance of using a suitable representation strategy are also discussed.

Chapter 4 describes the diverse entities that assign semantic richness and domain coverage to the KGs and the integration methods used to build the KGs.

Chapter 5 encompasses all the work on link classification, analyzing the results obtained for the proposed experiments.

Chapter 6 delves into link prediction, explaining our methodology and assessing the acquired outcomes for the proposed experiments.

Chapter 7 offers concluding remarks of the previous chapters and lays possible paths for the future.

Additionally, this dissertation is supplemented by six appendixes plus a GitHub Repository.

Appendix A details the computational environments in which the experiments were carried out.

Appendix B exhibits the parameters used in the KGE algorithms of the link classification approach.

Appendix C provides additional results regarding the comparison of KGE models in link classification.

Appendix D displays the parameters employed in the KGE algorithms utilized in link prediction.

Appendix E offers supplementary findings regarding the predictive accuracy and precision of link prediction methods.

Lastly, Appendix F presents further outcomes regarding the proportion of the correctly predicted entities ranked in the top 10, 30 and 100 among all entities of the same type in the context of link prediction experiments.

Chapter 2

Background

In this chapter, we introduce fundamental concepts needed to understand this dissertation. Section 2.1 establishes a foundational understanding of how to represent relationships between entities. Then, it introduces the concept of ontology and KG, along with their corresponding components. Section 2.2 introduces ML and the algorithms used in this dissertation. Finally, Section 2.3 describes both link classification and link prediction tasks.

2.1 Foundations of Knowledge Representation: From Graphs to Ontologies

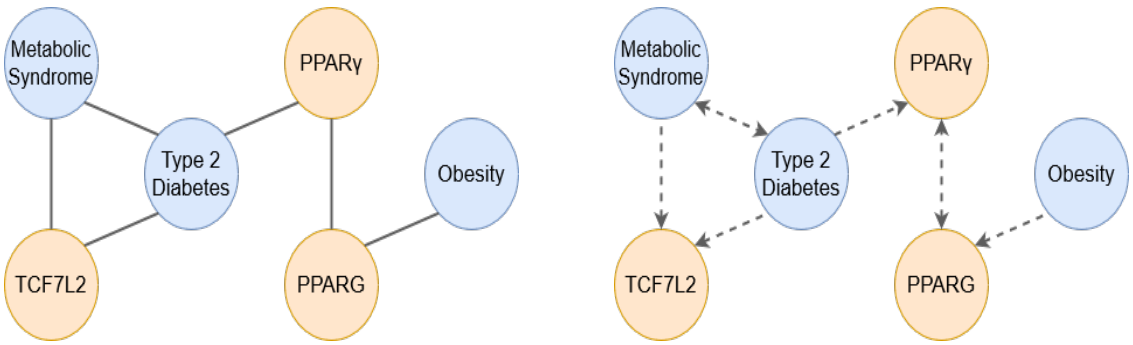
The concept of graphs traces its origins to the pioneering work of mathematicians in the 18th century to model relations between objects using points and lines. Inspired by the network nature of interconnected entities, graphs emerged as a fundamental theory to represent and analyze complex dependencies [38].

2.1.1 Graph Theory

In a simplistic definition, a graph G is expressed as $G = (V, E)$, where V represents a set of vertices (or nodes) and E represents a set of edges (or links). Graphs are used to model relations between objects, with entities represented as vertices and the edges representing the connections between them [16].

The relations between objects are not necessarily symmetric, as shown in Figure 2.1. Undirected graphs represent symmetrical connections (Figure 2.1(a)), whereas directed graphs reveal the direction of interactions (Figure 2.1(b)). Directed graphs provide a deeper understanding of causal relationships and sequential dependencies [5].

Graphs can also be either homogeneous, where all nodes and edges represent entities and relations of the same type, respectively, or heterogeneous, where all nodes and edges represent entities and relations of different types and are labelled accordingly [5]. Moreover, graphs can be weighted by assigning numerical weights to any of their edges, describing connectivity strengths or lengths [24].



(a) Undirected graph (b) Directed graph

Figure 2.1: Examples of undirected and directed graphs. Blue dots represent diseases, and orange dots represent genes related to them. In undirected graphs (a), variables exhibit symmetrical relations, while directed graphs (b) feature asymmetrical relations, occasionally resulting in two distinct edges between two nodes.

2.1.2 Ontologies: Structured Knowledge Representation

An ontology represents a set of conceptual definitions about a domain of interest. It specifies the context and the semantic rules regarding the concepts, allowing for interpreting those concepts through their logical axioms (fundamental assumptions). Ontological representations encompass several conceptual models, e.g., classifications, fully axiomatised theories, and database schemas (Figure 2.2) [25].

The main components that compose an ontology are a set of classes (concepts), a set of domain entities (individuals) [37], and a set of semantic links (relationships) that describe relationships between classes/entities or properties of classes. Thus, domain knowledge is encoded as axioms, natural language labels, synonyms, definitions, and other properties [25].

Ontologies often structure their components as a directed acyclic graph, where the classes are nodes and relations are edges. The combination of expressiveness, community adoption, interoperability, and tool support has propelled Web Ontology Language (OWL) and Open Biomedical Ontologies (OBO) to become prominent ontology languages within the biomedical domain [5].

Utilizing ontologies, the biomedical domain leverages structured representations of medical knowledge, enhancing data integration and analysis [73]. These frameworks facilitate semantic searches and aid in personalized medicine by organizing heterogeneous data for customized treatments. Furthermore, ontologies contribute to healthcare by refining the accuracy of diagnoses and treatment protocols [7].

GO is the most successful biomedical ontology. It describes the universe of concepts associated with gene product functions and how these functions relate to each other. A gene product function in GO is described concerning the biological process, molecular function and cellular component [68]. Other ontologies are HPO, DO and Sequence Ontology.

Classes from one ontology are often linked to classes in another ontology through Logical Definitions

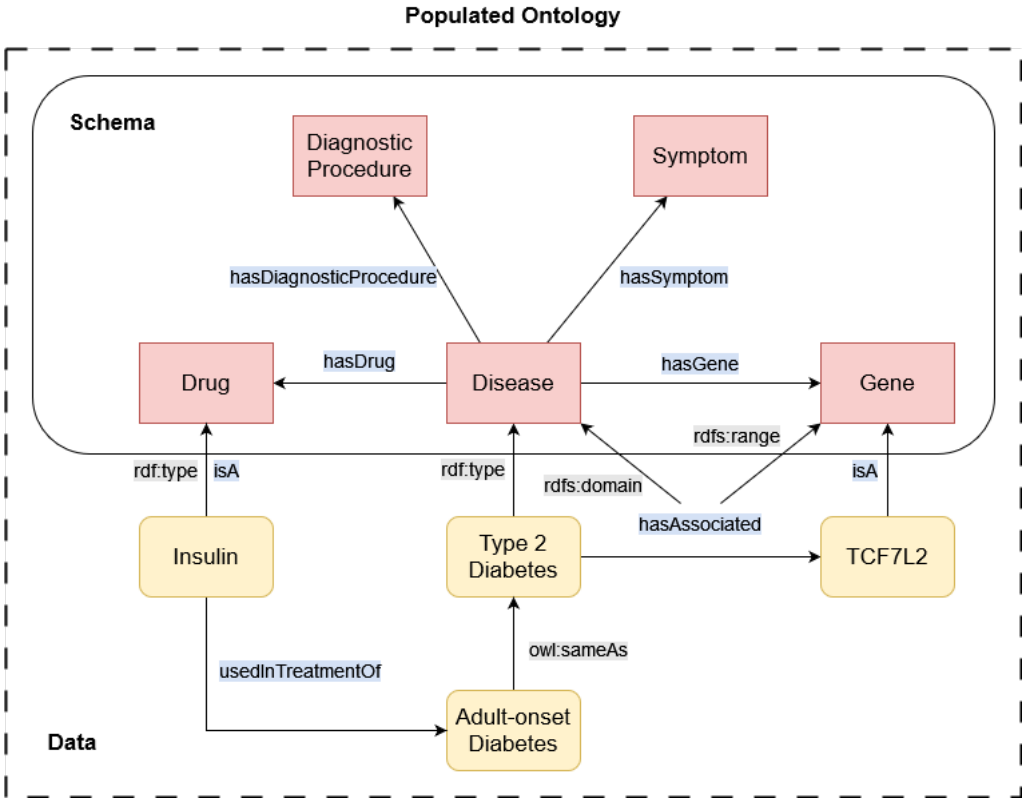


Figure 2.2: Example of an ontology populated with individuals. Red straight-edged rectangles represent classes, yellow rounded rectangles represent individuals, and arrows define relationships between classes/individuals or properties of classes.

(LD) and/or ontology mappings, enabling automated reasoning to be applied directly. While LD specifies the semantics and constraints, mappings facilitate ontological interoperability by aligning similar concepts or entities [20].

An example of a LD between the GO and the HPO is the definition of ”Hearing impairment” (HP:0000365) in HPO. This definition incorporates the concepts of genes associated with deafness and anatomical structures related to auditory function. Examples of mappings are the GO term ”GO:0034220”, which represents the potassium channel activity and the HPO term ”HP:0005952”, which describes ”Hypokalemia”, showcasing the correlation between gene function and phenotype manifestation.

Applying an ontology as a knowledge base facilitates the validation of semantic relationships and the derivation of conclusions from known facts for inference. KGs stand out as a method for structuring knowledge representation, especially when incorporating various ontologies and other data sources.

2.1.3 Knowledge Graphs: Integrating Semantics with Graph Structures

Knowledge Graphs are data structures that represent real-world entities, organizing and interconnecting the data in a way that incorporates the semantic meaning from ontologies. This integration allows for more context-rich data representation from various sources, enabling powerful querying, reasoning, and understanding of relationships between entities [24].

In a KG, the **nodes** represent the entities, **edges** represent connections between two entities, and **labels** indicate the types of relations between two entities. A relationship in a KG is expressed as a fact structured in the form of *(head entity, relation, tail entity)*, indicating that a specific association connects two entities. This architecture is used in the Resource Description Framework (RDF) language, which treats each relation as triple in the form of *{subject-predicate-object}* [77].

Embracing a dynamic approach to data representation, KGs can provide multiple perspectives over an entity, describing it using different properties or multiple portions of the graph [24]. Examples of notable KGs are FreeBase [11], WikiData [75], DBPedia [8] and Yago [70]. Figure 2.3 represents a KG that uses DO as the schema.

KGs are utilized in various domains, including artificial intelligence, data integration, and semantic search, to better understand complex relationships within vast information [5]. In particular, they represent an unparalleled opportunity for ML as it offers a unique source for feature engineering that enriches the input data and potentially leads to improved performance in various tasks [50].

2.2 Machine Learning

Machine Learning is a subfield of computer science focused on developing and applying algorithms that learn automatically from data instead of being explicitly programmed. Among many applications, ML seeks to make predictions (e.g. future events) or discover new patterns using data as feature information [33]. The data can come from nature, collected using devices, handmade by humans or generated by other models.

2.2.1 Supervised Learning Algorithms

Supervised learning represents a pivotal branch within ML where models are trained on input-output pairs (labelled examples), iteratively adjusting its parameters to minimize the difference between the predicted outcomes and the actual labels. This learning method encompasses **regression** problems, where the output is continuous, and **classification** problems, where the output is discrete [33].

We focus on the supervised learning algorithms for classification: Naive Bayes (NB), Multi-Layer Perceptron (MLP), Extreme Gradient Boosting (XGB) and Random Forest (RF). In the following, a brief explanation of their inner-workings is given:

- NB is a probabilistic model based on Bayes' theorem that assumes the independence between

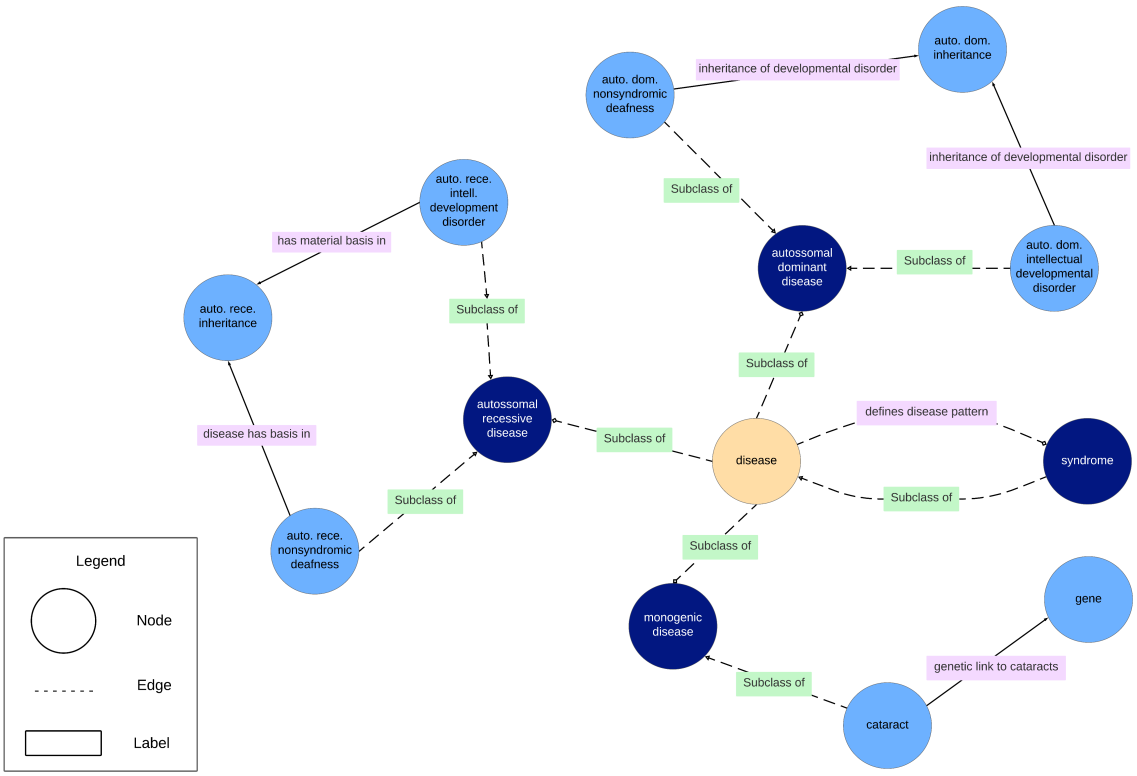


Figure 2.3: KG representation of a concept from the DO [4] with the nodes and edges that define him. The yellow entity is the core entity that represents the general concept of diseases within the ontology. The dark blue entities represent major subclasses of diseases. The light blue entities represent more specific diseases of the first-level nodes, and one represents another type of entity (a gene). Green relations signify a hierarchical organization where diseases are categorized from general to specific levels within the ontology. Pink relations represent more specific associations between diseases and detailed attributes.

features. It calculates the probability of each class given the input features and selects the class with the highest probability [85].

- MLP is a type of artificial neural network that utilizes backpropagation to adjust weights and biases by minimizing error through iterative optimization, effectively handling complex classification problems. Its effectiveness depends on proper tuning, sufficient data, and problem suitability [59].
- XGB is an implementation of gradient-boosted decision trees that combines weak learners sequentially, where each subsequent tree corrects the errors of the previous one. It requires careful tuning of hyperparameters, and the ensemble nature of boosted trees might make it less interpretable [19].
- RF is based on constructing multiple decision trees during training, which collectively determine

the final prediction. This is achieved by selecting the class that represents the mode of the dataset classes in classification tasks or the mean prediction in regression tasks [13].

Several ML applications rely on user-defined heuristics to extract features that encode structural information about a graph (e.g., degree statistics or kernel functions) because popular ML algorithms are not inherently equipped to handle graph data directly. These algorithms traditionally require input in the form of vectors, tensors or other data types.

The central problem of ML on graphs remains to encode the high-dimensional and non-Euclidean information about graph structure into a lower-dimensional space. In recent years, approaches have been increasing that automatically learn to encode graph structure into low-dimensional embeddings, using techniques based on deep learning and nonlinear dimensionality reduction [46].

2.2.2 Knowledge Graph Embeddings

Embeddings serve as lighter representations of the data within a lower dimensional space, offering a way to handle complex information. Under KG theory, KGE plays a crucial role by converting entities and relations into vectors (or other data structures, such as matrices or tensors) that capture the semantics and structural information of the original KG [21].

KGE models encompass core components that facilitate their functionality across various tasks within the realm of KG analysis: embeddings and a loss function [50]. Furthermore, KGE models for link prediction employ a scoring function to assess the plausibility of relations between entities based on their embeddings, enabling the assessment of new candidate triples within the KG [77].

Figure 2.4 represents visually what happens in link prediction KGE models. First, vector representations are generated for the *head entity, relation, tail entity* of the target triple. Vectors encode latent properties of the KG and, for similar entities, tend to be described with similar vectors. Finally, these embeddings serve as input for a scoring function, which returns a value for that association.

Given the large quantity of KGE algorithms in the literature, it is usual to classify them under some criteria. We consider deep learning, matrix factorization, path-based, translational distance and semantic matching models [46, 76]. This dissertation focuses on path-based models, translational distance models, and semantic matching models.

Path-based attempts to preserve local neighborhoods of entities and their properties based on random or non-random walks (paths) in the KG. The KG is transformed into node sequences by performing truncated paths, which preserves the network's structural proximity while capturing structural relationships between entities. Then, natural language methods, such as Word2Vec [47], are applied to the sample paths for graph embedding. DeepWalk [57], Node2Vec [29], RDF2Vec [61], Onto2Vec [64], OPA2Vec [65], and OWL2vec [2] are some of the path-based models. In the following, a brief presentation of the inner-workings of the KGE models used in this dissertation is given:

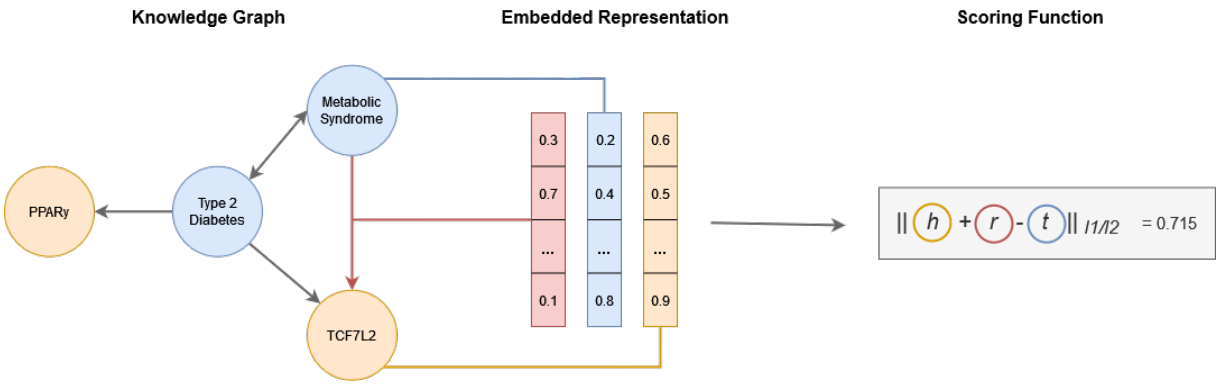


Figure 2.4: Process in link prediction KGE models: From a KG, embedding models generate representations of the KG elements embedded in a lower-dimensional space. In the present example, these representations are vectors. The resulting embeddings are fed to a scoring function that gives a value that reflects the plausibility of a given triple.

- RDF2Vec leverages the structure and semantics of the RDF graph to create vector representations for entities. The idea is first to extract paths connecting entities in the RDF graph. Here, each path is represented as a sequence of entities and predicates. Afterwards, a word embedding method learns vector representations for entities based on these paths. Finally, each entity in the RDF graph is represented as a dense vector in a continuous vector space after training [61];
- OPA2Vec extends RDF2Vec also to incorporate predicates (relations) and attributes (properties of entities). The first step in OPA2Vec is to generate a context for each triple in the KG that captures the surrounding information related to the triple. Next, a word embedding method learns vector representations for relations, entities, and their attributes based on the generated triple contexts. After training, each component in the KG is represented as a dense vector in a continuous vector space [65].

Translational Distance approaches model relations in the KG as translational operations between graph node embeddings. Being f_n a graph embedding, these methods define a translation operation that translates $f_n(h)$ to $f_n(t)$ depending on the relation r. They measure a fact's plausibility as the distance between the two entities after the association translates. The methods considered are TransE [12], TransH [79], TransR [43], TransD [39], TransSparse [40], and KG2E [36]. In the following, a brief presentation of the inner-workings of the KGE models used in this dissertation is given.

Being *h* the *head entity*, *r* the *relation* and *t* the *tail entity*:

- TransE represents entities and relations as vectors in the same space. Given a fact *(h,r,t)*, the relation is interpreted as a translation vector **r** so that the embedded entities **h** and **t** can be connected by **r** with low error [12];

- In TransH, each relation r as a vector $\mathbf{d}_r$ on a hyperplane with $\mathbf{w}_r$ as the normal vector. Given a fact *(h,r,t)*, the entity representations $\mathbf{h}_\perp$ and $\mathbf{t}_\perp$ are first projected onto the hyperplane. Finally, the projections are assumed to be connected by $\mathbf{d}_r$ on the hyperplane with low error [79];

- TransR represents entities as vectors in an entity space, and each relation is associated with a specific space. Then, they are modelled as a translation vector in that space. Given a fact *(h,r,t)*, TransR first projects the entity representations $\mathbf{h}_\perp$ and $\mathbf{t}_\perp$ into the space specific to relation r. Here, $\mathbf{M}_r$ is a projection matrix from the entity space to the relation space of r. The relation is interpreted as a translation vector $\mathbf{r}$ so the entity representations can be connected by $\mathbf{r}$ with low error. Each relation is associated with two matrices in the presented version: project head and tail entities [43];

- TransD simplifies TransR by decomposing the projection matrix into a product of two vectors. Given a fact *(h,r,t)*, TransD introduces additional mapping vectors and entity/relation representations. Finally, the resulting projection matrices are applied to the entity representations to get their projections [39].

Semantic Matching algorithms match the semantics of entities and relations through specific operations like multiplication or neural network convolutions, to measure the plausibility of facts. Standard semantic matching models are RESCAL [51], DistMult [58], HolE [52] and ComplEx [72]. In the following, a brief presentation of the inner-workings of the KGE models used in this dissertation is given:

- For each relation r, DistMult introduces a vector embedding $\mathbf{r}$ and requires a diagonal matrix $\mathbf{M}_r$ from the given vector. The result of the scoring function then captures pairwise interactions between the components of $\mathbf{h}$ and $\mathbf{t}$ along the same dimension [84];

- The term HolE comes from *Holographic Embeddings*. This algorithm represents entities and relations as vectors. Given a fact *(h,r,t)*, the entity representations are first decomposed into $\mathbf{h} \star \mathbf{t}$ by using the circular correlation operation [52];

- ComplEx extends DistMult by introducing complex-valued embeddings to better model asymmetric relations. In ComplEx, entity and relation embeddings **(h,r,t)** no longer lie in a real space where $\bar{\mathbf{t}}$ is the conjugate of $\mathbf{t}$ and Re(.) means taking the real part of a complex value. Then, the scoring function of ComplEx facts from asymmetric relations can receive different scores depending on the order of entities involved [72].

KGE algorithms are a way of representing entities and relationships in a KG as numerical vectors in a continuous space. While some KGE models are more directly usable for graph-related tasks, others involve extracting informative features from paths or structural patterns in the KG and utilizing these features as inputs to other ML algorithms.

2.3 Graph-Related Tasks

ML is a data-driven discipline where the algorithms are often categorized according to the type of task they seek to solve. The usual tasks (such as classification and regression) are not the most informative regarding graphs because of two key differences in the data: properties about relationships between data points provide valuable information to describe the data set, and not all nodes need to have labels to improve the model during training [30].

The most common tasks of graph-based ML are node classification and **link prediction**. Nonetheless, node classification can be redefined as **link classification**, changing the focus from entities to the relationships between them. These tasks address different aspects of graph analysis, feeding to diverse applications across domains like social networks, biological networks, and recommendation systems [31]. We will delve deeper into these tasks in the following sections.

2.3.1 Link Classification

Node classification involves labeling individual nodes within a graph based on their attributes, relations, or embeddings. The label y_u - type or category - is associated with all the nodes $u \in V$ in the *training* set of nodes $V_{train} \subset V$ [31]. This task can be redefined to identify and classify a non-represented relationship between a pair of nodes in the KG, as shown in Figure 2.5. Throughout this dissertation, the task of classifying node pairs is referred to as *link classification*.

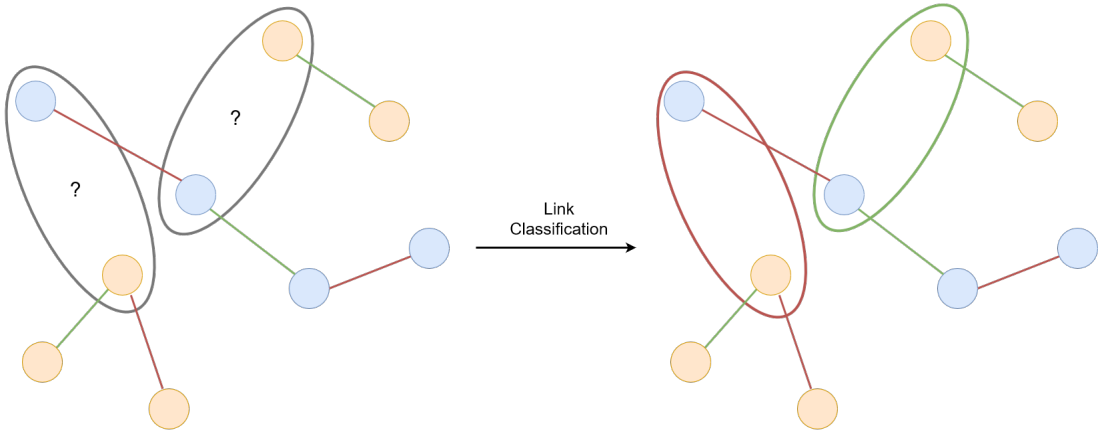


Figure 2.5: Example of a link classification task, where the objective is to determine the relationship between a pair of nodes when the graph does not originally model it. Blue dots represent diseases, and orange dots represent genes. Edges colored green and red represent positive and negative confirmed pairs of nodes, respectively. Figure adapted from Xiao et al. [81].

Several ML algorithms can be employed for link classification, encompassing: traditional classifiers (over features extracted from KGs) [78, 23, 53]; graph-based models, which leverage the inherent struc-

ture of graph data [44, 45]; and deep learning techniques, such as Graph Neural Networks and Deep Neural Networks [35, 87].

Link classification finds applications in various domains, such as citation networks, to label the research topic to which each article belongs in the network [71, 63] and other information tasks. Examples are organizing documents, videos, and web pages or classifying anomalous and potentially dangerous connections [90].

We can also assign several gene ontology types in protein-protein interaction networks for enrichment analysis or identify disease genes by low-dimensional vector representations [69, 83, 78]. The versatility of link classification task allow for their adaptation to various problems across different domains, being an interesting way of looking at a problem in data analysis and knowledge discovery.

2.3.2 Link Prediction

Link prediction, also known as *Knowledge Graph Completion* or *Knowledge Graph Augmentation* [62], predicts missing or unobserved links between entities in a KG by extracting new facts from external sources or by inferring facts from those already in the KG [6]. This work exploits existing facts in the KG to infer missing ones. Figure 2.6 presents an example of a link prediction task.

The link prediction task is also to guess the correct entity that completes <h,r,?> (tail prediction) or <?,r,t> (head prediction) from a list of candidate entities [62]. This assignment requires exploring features, including node attributes, topological features (e.g. Common Neighbor, Jaccard coefficient) and learned relation embeddings.

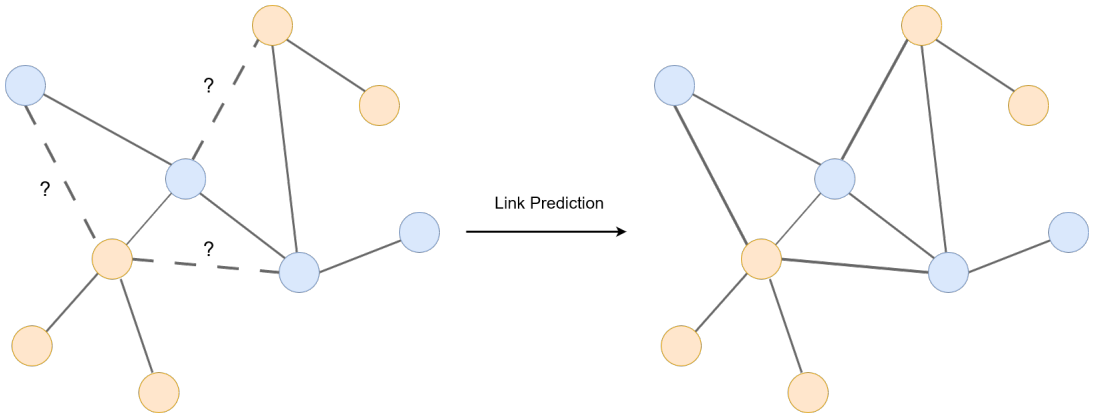


Figure 2.6: Demonstration of a link prediction task, where the objective is to predict whether two nodes are linked. Blue dots represent diseases, and orange dots represent genes. Figure adapted from Xiao et al. [81].

Link prediction employs a range of algorithms that encompass both traditional and modern techniques. Traditional models include similarity-based approaches such as Common Neighbors and Jaccard

coefficient [9, 60]. Probabilistic models like the random walk models are also commonly used [82, 88]. More recent advancements in deep learning have introduced variants of Graph Neural Networks tailored for link prediction, such as Graph Convolutional Networks [89, 80]. These models collectively offer a diverse toolbox for addressing link prediction tasks in various domains, including recommendation systems [22], social networks and community detection [48], and biological networks [10, 73].

Chapter 3

Related Work

This chapter reviews prior research on gene-disease association within biological networks. It starts exploring the challenges in identifying relations between genes and diseases (Section 3.1). Next, it highlights the importance of effective representation strategies for these associations (Section 3.2). It concludes by surveying the existing approaches using ontologies or KGs for predicting gene-disease links and for classifying gene-disease pairs (Section 3.3 and Section 3.4).

3.1 Challenges in Gene-Disease Association

Diseases often result from multiple genetic factors interacting with each other and the environment [17], making it challenging to pinpoint a single gene responsible for a disease. This genetic heterogeneity complicates association studies as multiple genes may contribute to the manifestation of a single disease [15]. Common diseases, such as diabetes, heart disease, and cancer, are polygenic, involving the interplay of numerous genes, each with a small effect on the overall disease activity [74].

Environmental influences, such as diet, lifestyle, exposure to toxins, stress, and even socioeconomic factors, can interact with an individual's genetic makeup, potentially influencing the onset, severity, or progression of diseases. For instance, in some cases, environmental triggers might exacerbate or alleviate the symptoms of a disease [49]. The interplay between inherited genetic predispositions and environmental influences is critical to understanding the complexity of disease development.

Identifying gene-disease associations is a complex and multifaceted process challenged by the genetic heterogeneity of diseases, their polygenic nature, complex inheritance patterns, limited sample sizes for reliable detection of subtle variations, and the need to discover genetic influences from environmental factors [15]. Similarly, the dynamic nature of gene expression, alternative splicing, and post-translational modifications adds further layers of complexity to understanding gene function and its role in disease

pathogenesis.

3.2 Importance of Representation Strategies

A *representation strategy* is a method or approach used to transform raw biological data into structured formats suitable for ML analysis [55]. Link classification and link prediction are common representation strategies used in gene-disease association problems. While link classification characterizes relationships between pairs of genes and diseases, link prediction forecasts new or unseen associations between these entities [31], fulfilling the following objectives:

- **Feature Extraction:** genomic data is high-dimensional and heterogeneous, including sequences, expression levels, and functional annotations. Effective representation strategies can distill these diverse data types into meaningful and informative features, aiding in capturing essential characteristics of genes and diseases [10];
- **Graph Analysis:** biological systems often exhibit network-like structures, such as protein-protein interaction networks or gene co-expression networks. Representation learning methods can encode these relationships into vector representations [30], enabling better analysis and inference of associations between genes and diseases within these networks;
- **Predictive Modelling:** accurate representations are the foundation for predictive models. Effective representations enable ML algorithms to identify patterns and predict novel associations, aiding in the discovery of potential gene-disease associations or therapeutic targets;
- **Biological Interpretability:** well-crafted representations can offer insights into the underlying biology by revealing latent relationships and similarities between genes and diseases. These representations can highlight shared pathways, functions, or molecular mechanisms, providing a deeper understanding of disease etiology and potential therapeutic interventions [31].

Link classification and link prediction exemplify how representation strategies bridge the gap between complex biological data and ML techniques, facilitating the extraction of meaningful associations and insights crucial for understanding genetic contributions to disease susceptibility and progression.

3.3 Overview of Link Prediction Methods

Predicting associations between genes and diseases is an active area of research, which has benefited from the increased available data and the explosion of ML techniques [23]. Table 3.1 provides a concise overview of several link prediction methods that explore ontologies or KGs to predict gene-disease associations.

Utilizing methodologies like Gaussian random projection, similarity networks and network propagation has significantly advanced the prediction of gene-disease associations [34]. Employing Gaussian

Reference	Data Sources	Task	Methods
Yang et al. [86]	DisGeNET, MalaCards, HPO, Menche et al. [38], Orphanet, STRING	Multiple diseases	KGE and similarity networks
Zhu et al. [89]	DisGeNET, HumanNet, Mesh	Multiple diseases	KGE and Graph Convolutional Networks
Xu et al. [82]	HPRD Database, OMIM	Multiple diseases	Multi-path random walk and KGE
Bean et al. [9]	IntAct, GO, DisGeNET, ALSoD, ClinVar	Single disease (ALS)	ML on functional similarity
Biswas et al. [10]	GO, HPO, DO, SwissProt, STRING, SIDER, OMIM, GAD, Comparative Toxicogenomics and Reactome	Single disease (Co-Morbid diseases)	KGE and Markov Clustering
Xiang et al. [80]	IntAct, MINT, BioGRID, HPRD TRANSFAC, KEGG, BIGG, PhospositePlus, CORUM, DisGeNET, HPO, GO	Multiple diseases	Fast network embedding algorithm and dual-layer network reconstruction and propagation
Zhang et al. [88]	GWAS, OMIM, STRING, KEGG, MSigDB	Multiple diseases	Statistical measures, network reconstruction based on local random walk dynamics and random walk with restart
He et al. [34]	HPO, DisGeNET	Single disease (Parkinson disease, Diabetes Mellitus-insulin and Hyperglycemia)	Gaussian random projection, similarity networks and network propagation
Vilela et al. [73]	GO, DisGeNET, Ensembl	Single disease (Autism Spectrum Disorder)	KGE

Table 3.1: Summary of existing works on gene-disease association in the context of link prediction using ontologies or KGs.

random projection, low-dimensional representations of network nodes are generated, facilitating the construction of an enhanced heterogeneous network through the aggregation of adjacency matrices. Subsequently, the associations between all genes/proteins and specific diseases are evaluated by simulating a network propagation process originating from predefined source nodes within the constructed network.

He et al. [34] proposed an approach called DGHNE, which leverages a heterogeneous biomedical network for identifying disease-causing genes. This network is enriched with Cosine Similarity scores derived from phenotype annotation vectors of diseases from resources such as HPO, protein-protein interaction networks and disease-gene associations.

Network reconstruction based on local random walk dynamics offers further insights into biomedical

research [88]. These approaches exploit the dynamics of local random walks within the network to reconstruct and infer meaningful relationships between genes and diseases. By analyzing the connectivity patterns and information flow within the network, these methods provide valuable predictions regarding gene-disease associations, complementing the insights obtained from methodologies like Gaussian random projection and network propagation.

Employing ML algorithms on functional similarity is another strategy for predicting gene-disease associations across various medical conditions. Bean et al. [9] developed a method that inputs a graph containing gene-disease links, along with gene functions sourced from GO and protein-protein interaction data. This approach leverages known disease-associated genes to generate candidate gene lists, assessing each gene's similarity to the disease profile. By applying a similarity score threshold, potential disease-linked genes are identified, with optimization achieved through iterative learning from known disease-gene associations.

Approaches solely relying on link prediction KGE algorithms offer a streamlined method for predicting gene-disease associations [73]. These approaches often involve generating embeddings for genes, diseases and relations within a continuous vector space, applying scoring functions to compute the likelihood of association between gene-disease pairs, and evaluating the predictive performance using metrics like Mean Rank (MR), Mean Reciprocal Rank (MRR), and Hits@k. These algorithms streamline the prediction process by directly inferring associations from the embedded representations within the KG, providing a straightforward approach for identifying potential gene-disease associations.

To establish a comprehensive framework, Vilela et al. [73] constructed a KG incorporating GO and its annotations for genes, gene-disease associations, and Ensembl gene identifiers. Through a 60/20 split for training and testing, they applied TransE, DistMult, and ComplEx on the KG. To evaluate the effectiveness of the KGE models, they employed MR, MRR and Hits@ 1, 3, and 10 evaluation metrics.

Link prediction methods also rely on applying KGE algorithms with complementary methods such as Markov Clustering [10], Graph Convolutional Networks [89] or similarity networks [86], to enhance the prediction of gene-disease associations. These mixed approaches typically apply KGE algorithms first. In the case of Markov Clustering, it can be applied to enrich the gene-disease network by identifying clusters (groups) of correlated diseases. In the case of Graph Convolutional Networks, the embedding representation vectors are fed to the network, and the decoder gives the candidate set of genes. Similarity networks follow the same idea of using Markov Clustering: enrich the gene-disease network.

Yang et al. [86] proposed HerGePred, a methodology that firstly employs embedding algorithms like Node2Vec and LINE to derive embeddings of entities. They then proposed LVR-based similarity prediction (LVRSim) and random walk with restart based on a reconstructed heterogeneous disease-gene network (RW-RDGN). LVRSim utilizes the local and global structural information encoded in low-dimensional vector representations to compute disease-gene similarity using Cosine Similarity. RW-RDGN reconstructs the heterogeneous network incorporating new disease and gene networks, subsequently applying the random walk with restart algorithm to prioritize candidate genes based on their associations with diseases.

While the mentioned works contribute to the field of gene-disease association, several weaknesses exist across these methodologies. Approaches that rely on functional similarity graphs lack direct links between confirmed genes and diseases, limiting their ability to capture disease-specific relationships accurately. Enhancements could involve incorporating ontologies better suited to specific diseases, thereby enriching the understanding of the underlying biology.

3.4 Overview of Link Classification Methods

Table 3.2 offers a comprehensive recap of various link classification methods employed in the gene-disease association problem. These methods leverage ontologies or KGs, providing a valuable reference for understanding the diverse landscape of techniques employed in this critical domain.

Link classification approaches based on neural networks, including methodologies solely based on neural networks [35] and mixed approaches combining neural networks with other techniques [18], have emerged as powerful tools to predict gene-disease associations. By leveraging the graph structure of biological data, Graph Neural Networks can integrate various databases, including gene expression profiles, protein-protein interaction networks, and disease phenotypes. This is achieved through the iterative propagation of information across graph nodes, where each node aggregates information from its neighbors to update its own representation. Graph Neural Networks learn to classify nodes (genes or diseases) based on their features and local and global graph context, enabling them to accurately predict associations between genes and diseases.

Within the methodologies that utilize neural networks and other techniques, some approaches employ specific neural network algorithms, like Deep Belief Networks and Deep Neural Networks, in conjunction with similarity networks or Mashup [44, 45, 87]. Luo et al. [44] proposed dgMDL to predict disease-gene associations with multimodal Deep Belief Networks using two similarity networks built by the K Nearest Neighbours algorithm: protein-protein interaction-based and GO-based. First, two Deep Belief Network sub-models are trained based on the similarity networks. A joint Deep Belief Network combines the two sub-models to learn cross-modality representations. Finally, the Deep Belief Network models the joint model and a sigmoid activation function for decision-making.

A category of approaches employs KGE and pointwise learning-to-rank prediction based on neural networks [18]. KGE first represent genes and diseases as embeddings in a low-dimensional space. The pointwise learning-to-rank model then processes these embeddings and additional features, such as phenotypic characteristics or functional annotations. The model then computes the inner product and applies a sigmoid function to generate a prediction score indicating the likelihood of association between genes and diseases.

Another category of approaches focuses on identifying and classifying a pair of nodes by applying KGEs followed by traditional ML algorithms, such as RF and XGB [78, 53], or more complex models, such as neural networks [23]. These approaches leverage the power of KGE to capture attributes of genes and diseases, utilizing the resulting embeddings as input to traditional or advanced ML techniques to make

Reference	Data Sources	Task	Methods
Luo et al. [44]	OMIM, GO, InWeb_InBioMap	Multiple diseases	Similarity networks, Restricted Boltzmann Machine and multimodal Deep Belief Networks
Luo et al. [45]	OMIM, GO, HPO InWeb_InBioMap	Single disease (Lung Cancer and Bladder Cancer	Similarity networks and euclidean and geodesic distance
Wang et al. [78]	CTD, HumanNet, OMIM, PubMed	Multiple diseases	KGE and RF
Biswas et al. [10]	GO, HPO, DO, SwissProt, STRING, SIDER, OMIM, GAD, Comparative Toxicogenomics and Reactome	Single disease (Co-Morbid diseases)	KGE and Markov Clustering
Chen et al. [18]	GO, HPO, AberOWL, PhenomeNET, UBERON, MP, MGI database, GTEx dataset, STRING and UniProt	Multiple diseases	KGE and pointwise learning-to-rank prediction based on neural networks
Du et al. [23]	GO, HPO, KEGG, DisGeNET and Menche et al. [X]	Single disease (Diabetes Mellitus)	KGE, neural networks and supervised learning models
He et al. [35]	GO, HPO, STRING, DisGeNET and Menche	Single disease (Gait abnormality and Congenital Epicanthus)	FactorHNE (Graph Neural Networks)
Ye et al. [87]	GO, KEGG, STRING and MIPS	Multiple diseases	Mashup and Deep Neural Network algorithms
Nunes et al. [53]	GO, HPO, DisGeNET	Multiple diseases	KGE, RF and XGB

Table 3.2: Outline of existing research on gene-disease association in the context of link classification applying ontologies or KGs.

accurate predictions regarding some labeled examples. This combined methodology demonstrates the collaboration between KGE algorithms and ML models in addressing the complexities of gene-disease association.

Nunes et al. [53] proposes a novel approach for gene-disease association by enriching KGs with semantic links between ontologies. They experimented with KGs with different characteristics, including GO and HPO ontologies and selected gene-disease links. They generated low-dimensional representations of nodes by implementing TransE, DistMult, RDF2Vec, OPA2Vec and OWL2Vec*. After that, they combine the gene and disease vectors through different vector operations. Finally, they analyze the performance of RF and XGB classifiers against a more straightforward approach based on semantic

similarity - Cosine Similarity.

The methodologies discussed for link classification in gene-disease association share certain weaknesses that merit attention for enhancement. One prevalent limitation is the reliance on generic similarity networks and ontologies, which may not fully capture the complexities of disease-specific relationships. To address this, incorporating specialized ontologies tailored to specific diseases, such as DO, could significantly improve prediction accuracy.

Refinement would provide a more nuanced understanding of gene-disease associations and better align with the unique characteristics of each medical condition. Additionally, many approaches employ heterogeneous networks and KGE models, yet there is room for improvement in integrating diverse features and data types. Exploring alternative embedding techniques, considering disease-specific features, and refining feature integration strategies would contribute to a more comprehensive representation of the complex relationships within biological networks.

Chapter 4

Data Integration and Experimental Design

This chapter describes the entities that can appear in the KGs (Section 4.1) and explains how the various datasets are managed to create the final KGs (Section 4.2). Appendix A describes the computational environments in which the experiments were carried out. The KGs used in the experiments can be found at a GitHub Repository.

4.1 Data Characteristics

This dissertation considered genes and diseases extracted from DisGeNET [58], along with their associations, and integrated this data with GO, HPO, and DO ontologies to establish diverse KGs. Therefore, the approaches employed in this dissertation took as input ontology files, gene and disease annotation files, and a list detailing gene-disease pairs. In this section, we begin by explaining the dataset where *gene-disease associations* were extracted. Then, we explore the characteristics of the considered *ontologies*. Finally, we explain the *LD* and *ontology mappings* as additional links between ontology classes.

4.1.1 Gene-Disease Associations

DisGeNET [58] is one of the largest available collections of genes and variants involved in human diseases. It includes gene-disease associations extracted from multiple sources, including Uniprot [3], OMIM [32], or Orphanet [56], which are the same sources used to create some of the ontology annotations. The current version of DisGeNET (v7.0) comprises 1,134,942 pairs, 21,671 genes and 30,170 clinical or abnormal human diseases, disorders, traits, and phenotypes. DisGeNET also contains 369,554 variant-disease associations among 194,515 variants and 14,155 diseases, traits, and phenotypes.

We obtained 16,378 gene-disease associations, with 50% positive pairs and 50% negative pairs from Nunes et al. [53]. The negative pairs were generated by randomly sampling gene-disease pairs since high-quality negative experimental data (unassociated disease-gene pairs) are hardly available. Random sampling is based on the assumption that the expected number of negatives is several orders of magnitude

greater than the number of positives so that negative space is randomly sampled with a greater probability than positive space [54].

4.1.2 Ontologies

Gene Ontology was initially described in Section 2.1. A gene product function corresponds to the protein and non-coding RNA molecules produced by genes. GO resources include the GO itself and the corpus of GO annotations. Every term has a human-readable name and an ID and belongs to one of three sub-ontologies. All terms (other than the root terms representing each aspect above) have an ”is_a” subclass relationship to another term. The current release of GO (2023-01-01) includes 43,248 valid terms, 7,503,460 annotations, and 1,475,947 annotated gene products. GO terms are cross-referenced to corresponding concepts from several external vocabularies [7], including Uniprot [3], KEGG [41], Reactome Pathways [26], OMIM [32] and Orphanet [56].

Human Phenotype Ontology is a comprehensive biological and informatics resource for analyzing phenotypic abnormalities in human diseases. It organizes information into six independent sub-ontologies: phenotype abnormalities, clinical modifier, mode of inheritance, past medical history, blood group, and frequency of phenotypic abnormalities. Terms within HPO form a directed acyclic graph and are linked by ”is_a” (subclass-of) edges, indicating that a term represents a more specific or limited instance of its parent term(s). The latest release of HPO (v2022-12-15) encompasses over 13,000 terms and is developed using medical literature [1], Orphanet [56], DECIPHER [28], and OMIM [32].

Disease Ontology describes human diseases, their phenotypic characteristics, and related disease concepts within the medical vocabulary. It categorizes diseases into various types, including those caused by infectious agents, anatomical entities, cellular proliferation, mental health, metabolism, genetics, physical disorders, and syndromes. The 100th GitHub Release of the Disease Ontology (v2021-08-17) encompasses 10,862 disease terms, with 76% (8,312) defined by textual descriptions. Additionally, it includes 35,984 clinical vocabulary cross-references, identifying terms from MeSH, NCI thesaurus, and SNOMED CT through the bi-annual Unified Medical Language System integration [4], along with Orphanet [56], OMIM [32], and other sources.

4.1.3 Logical Definitions and Ontology Mappings

Logical Definitions within an ontology involve defining the meaning of terms using formal logical language. These definitions are structured to eliminate ambiguity and ensure a consistent understanding of concepts within the ontology. So, LD can be explored to bridge domains and contextualize relations between entities, such as genes and diseases [20]. The HPO includes LD, which defines classes as a com-

position of classes from different ontologies with complex semantic relations, facilitating interoperability and data integration.

Mappings between ontologies involve establishing correspondences or relationships between terms, concepts, or entities across these knowledge structures. These correspondences aim to align and reconcile ontologies' terminology, structure, or semantic differences. To uncover additional links between GO and HPO, Nunes et al. [53] used AML-Compound, an AgreementMakerLight ontology matching system variant, to retrieve relations between ontology classes [27]. They used an empirically determined threshold of 0.8 and found 494 mappings, where 37 were identical to the existing LD.

The original LD imposes restrictions involving four ontologies. However, we aimed to facilitate the relationship between HPO and a specific ontology - GO. We ensure a more direct connection between GO and HPO classes by selecting the simplified version of LD and mappings (example in Figure 4.1). This simplification allows the extraction of a single triple containing classes from each ontology, supporting triple-based approaches and shortest paths linking the ontologies to support random walk-based KGE models.

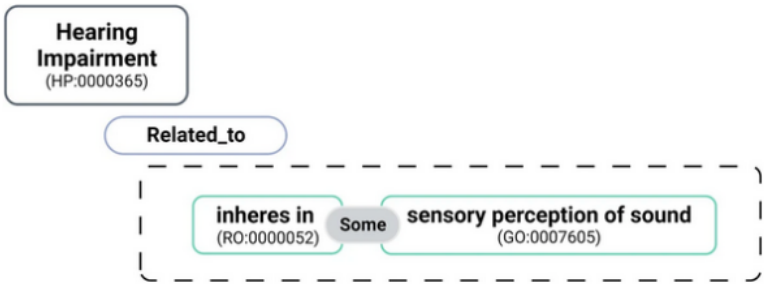


Figure 4.1: Example of simplified LD with a direct relation between two classes. The HPO term for "Hearing Impairment" (HP:0000365) is related to a restriction that involves the GO term "sensory perception of sound" (GO:0007605). Figure extracted from Nunes et al. [53].

4.2 Knowledge Graph Integration

The practical construction of KGs requires a selection of libraries, packages, and methodologies tailored to the intended objectives and domain-specific requirements. In this construction, the methods intertwine with the underlying KGE algorithms and the libraries that implement these algorithms. They directly influence the architecture, semantic richness, and inferential capabilities of the resulting KGs.

We used the *RDFLib*[1] (version 5.0.0) to create the KGs for input from RDF2Vec and KGE algorithms of the link prediction approach. *RDFLib* is a *Python* library designed to work with RDF data. RDF is

[1] https://rdflib.readthedocs.io/en/stable/index.html/

a standard model for data interchange on the web. RDFLib allows developers to manipulate, parse, serialize, store, and query RDF data within *Python* applications.

We mainly used *.parse* and *.add* methods to read and load OWL ontology files, add annotations, and include associations between genes and diseases (in the case of link prediction) to the KGs. Finally, we use *.serialize* method to save the KGs in XML format. RDF2Vec and the link prediction methods take the annotation files in the format 'url_entity *tab* list with annotations'. RDF2Vec also needs a file with all the entities appearing in the graphs, one entity per line with the full URL.

In the case of OPA2Vec, we used *ROBOT*[2] tool (version 1.8.0) to combine multiple OWL files as it only accepts one ontology file and, consequently, one annotation file. ROBOT is a command-line tool and *Java* library developed by the University of Manchester that facilitates the manipulation and management of OWL ontologies. It is part of the more extensive set of tools the OBO Foundry provides for working with OBO. To join the ontology files, we used ROBOT's *merge* functionality by typing in the command line:

```
robot merge --input file1.owl --input file2.owl --output merged.owl
```

OPA2Vec requires the annotations in the format 'entity *tab* <URL>' and all entities that can appear in the KGs in the final annotation file. We combine the TSV annotation files using a unique *Python* script and the *Pandas*[3] library to merge and consolidate the data into a single TSV file. To evaluate the impact that KG semantic richness and domain coverage have on gene-disease association prediction, we created the following KGs:

(i) **GO+HPO:** composed by GO and GO annotations for genes, and HPO and HPO annotations for genes and diseases;

(ii) **GO+HPO+LD:** composed by GO and GO annotations (for genes), HPO and HPO annotations (for genes and diseases), and logical definitions for HPO classes that reference GO;

(iii) **GO+HPO+Mappings:** composed by GO and GO annotations (for genes), HPO and HPO annotations (for genes and diseases), and mappings between GO and HPO classes;

(iv) **GO+HPO+LD+Mappings:** the union of *GO+HPO+LD* and *GO+HPO+Mappings*;

(v) **GO+HPO*+LD+Mappings:** composed by GO and GO annotations (for genes), HPO and HPO annotations (just for diseases), logical definitions and mappings between GO and HPO;

(vi) **GO+DO:** composed by GO and GO annotations (for genes), and DO and DO annotations (for diseases);

(vii) **GO+HPO+DO:** composed by the GO, HPO and DO ontologies and their annotations;

[2] http://robot.obolibrary.org/merge.html/
[3] https://pandas.pydata.org/

(viii) **GO+HPO+LD+DO:** the union of *GO+HPO+LD* and DO and DO annotations (for diseases);

(ix) **GO+HPO+Mappings+DO:** the union of *GO+HPO+Mappings* and DO and DO annotations (for diseases);

(x) **GO+HPO+LD+Mappings+DO:** the union of *GO+HPO+LD+Mappings* and DO and DO annotations (for diseases);

(xi) **GO+HPO*+LD+Mappings+DO:** the union of *GO+HPO*+LD+Mappings* and DO and DO annotations (for diseases).

Figure 4.2 represents the maximum semantic model that KGs can have, where genes and diseases are depicted within circles, ontologies within squares, and some connections among these elements are described.

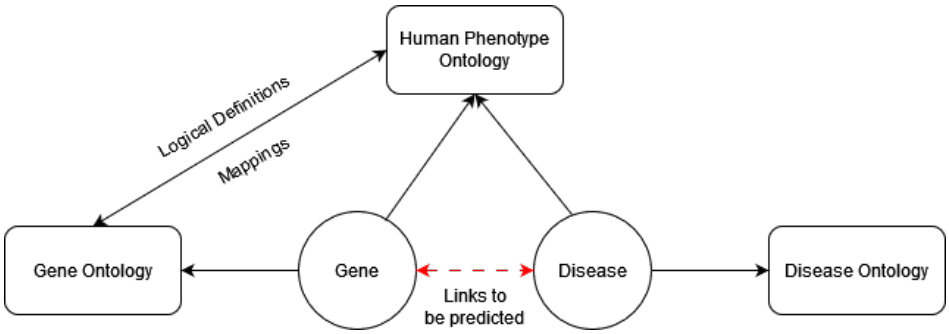


Figure 4.2: Semantic model of KGs: Visual depiction illustrating interconnected relationships among genes, diseases, and ontologies, offering insights into complex biological associations within a structured KG.

Table 4.1 provides the counts of genes, diseases, and corresponding pairs that the **GO + HPO + LD + Mappings + DO** graph encompasses. This KG represents the maximum number of genes and diseases involved in the experiments of this dissertation. Additionally, it is notable that only 50% of gene-disease pairs were included in the link prediction experiments.

Category	**Count**
Genes	8881
Diseases	36028
Gene-Disease Pairs	16378

Table 4.1: Gene-Disease Associations: counts of genes, diseases, and corresponding pairs, providing an overview of the scale of associations within the dataset.

Table 4.2 summarizes relevant statistics regarding the KGs, namely the: Classes (number of classes), A. Genes (number of annotations for genes), A. Diseases (number of annotations for diseases), Logic D. (number of logical definitions) and Mappings (number of mappings) between GO and HPO.

Knowledge Graphs	**Classes**	**A. Genes**	**A. Diseases**	**Logic D.**	**Mappings**
GO + HPO	294766	5901	1848	N/A	N/A
GO + HPO + LD	295118	5901	1848	350	N/A
GO + HPO + Mappings	295261	5901	1848	N/A	494
GO + HPO + LD + Mappings	295580	5901	1848	350	494
GO + HPO* + LD + Mappings	301532	2716	1848	350	494
GO + DO	282517	2716	6851	N/A	N/A
GO + HPO + DO	324407	5901	8699	N/A	N/A
GO + HPO + LD + DO	324759	5901	8699	350	N/A
GO + HPO + Mappings + DO	324902	5901	8699	N/A	494
GO + HPO + LD + Mappings + DO	325221	5901	8699	350	494
GO + HPO* + LD + Mappings + DO	324752	2716	8699	350	494

Table 4.2: Graph-wise ontology statistics: Comparative table detailing class counts, gene and disease annotations, LD and mappings across individual KGs, offering insights into ontology content variations among different graphs.

Chapter 5

Classifying Gene-Disease Pairs

In this section, we address the identification and classification of a pair of nodes. We start with the methodology overview in Section 5.1. Then, we explain the prediction strategy in Section 5.2. The following section (Section 5.3) examines the performance measures used to assess the model's skill and capability. Finally, we discuss the results obtained in the proposed experiments in Section 5.4. Section 2.2 and Section 2.3.1 provided a brief introduction and background of the models and methods that are cited in this chapter. The link classification approach can be found at a GitHub Repository.

5.1 Methodology

Addressing the gene-disease association problem entails constructing and analyzing biological networks, extracting relevant features, and applying ML techniques to predict associations. This workflow aims to decipher and understand the complex relationships between genes and diseases encoded in biological networks and spread the knowledge on these associations. The approach to identify and classify a non-represented relationship between a pair of nodes in the KG has five main components (Figure 5.1):

1. **KG Integration:** integrating the ontologies, annotation data, LD and mappings between GO and HPO to build different KGs;

2. **KG Embeddings:** using path-based KGE algorithms to obtain gene and disease embeddings according to their annotations;

3. **Vector Operations:** combining the obtained embeddings using different vector operators for each gene-disease pair;

4. **Machine Learning:** training supervised learning algorithms over the combined embeddings to predict gene-disease associations;

5. **Predictions Evaluation:** evaluating the performance of the supervised learning models.

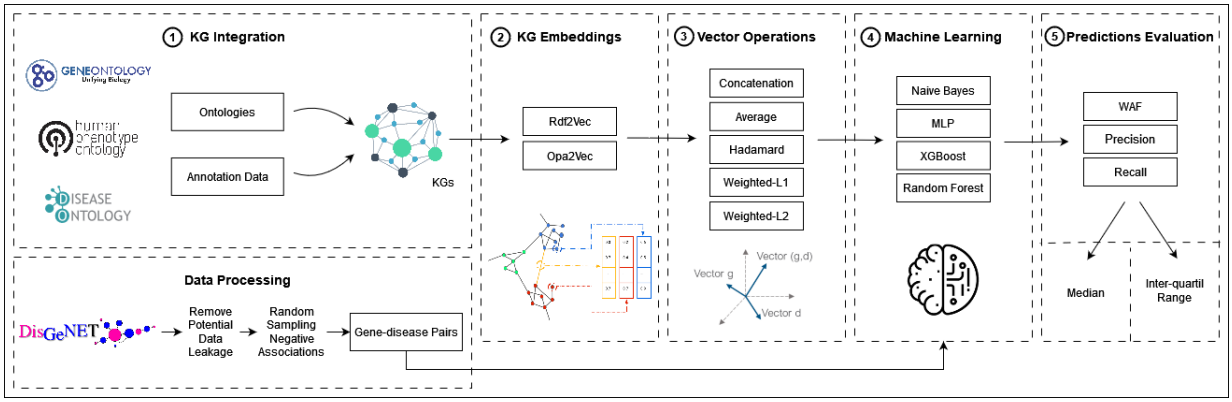


Figure 5.1: Workflow of the link classification approach with five steps: 1) building the KGs with ontologies and annotations; 2) creating embeddings to represent genes and diseases; 3) producing a final vector of the pairs in the dataset; 4) predicting gene-disease associations; and 5) evaluating the model's performance.

Our approach is based on Nunes et al [53]. Regarding KG integration, we considered the two previously considered ontologies (GO and HPO) and investigated the impact of adding a third ontology (DO) to the KGs. The integration of DO to enrich the semantic representation of diseases was motivated by the scarcity of methods leveraging comprehensive disease ontologies, including the study of Nunes et al. [53]. As for the embeddings and vector operations, we applied the two best-performing KGE models (RDF2Vec and OPA2Vec), as well as all vector operations previously considered.

We tested the performance of four classification ML algorithms, complementing the original approach that considered two algorithms (RF and XGB) with two other approaches (NB and MLP). We decided to complement the two originally chosen algorithms because of their popularity as benchmark algorithms, proven effectiveness in classification tasks, and broad support and understanding within the ML community. As per the evaluation metrics, we considered the Weighted Average of F-measures (WAF), precision, and recall to be popular measures for evaluating ML algorithms.

5.2 Gene-Disease Prediction

After constructing the KGs, we used RDF2Vec and OPA2Vec to learn feature vectors for the KGs (described in Section 4.2) and created a representation of two distinct vectors for each gene-disease pair of the dataset. The final embeddings present 200 features and cover the two types of KGE inside path-based methods (Appendix B for default parameters):

- **RDF2Vec (random walk)** with sequences generated using Weisfeiler-Lehman algorithm with walks depth 8 and a limited number of 500 by entity. The corpora of sequences were used to build a Skip-Gram model with the default parameters for Word2Vec [61];

- **OPA2Vec (non-random walk)** with default parameters [65].

We used *PyRDF2Vec* [67] to apply RDF2Vec algorithm. *PyRDF2Vec* is a *Python* library that uses a Skip-Gram model, similar to Word2Vec, to learn vector representations of entities based on their neighborhood information in the graph. This library creates embeddings that capture the relational information in the KG by traversing the graph and considering the context in which entities occur. To employ OPA2Vec, we used a folder containing the algorithm implementation and the default pre-trained PubMed model (consisting of two files) from the original OPA2Vec GitHub Repository.

After the KGE models, each gene-disease pair corresponds to two vectors, $f_i(g)$ and $f_i(d)$, associated with a gene g and a disease d, respectively. To generate the pair representation $r(g, d)$ such that r: $V \times V \rightarrow \mathbb{R}^{d'}$ where d' is the size of the pair (g, d), we applied five different mathematical expressions (or operations) over the corresponding vectors, as summarized in Table 5.1. However, there are several choices for the mathematical expressions from a set of commonly employed operations with KGE [29].

Operator	Definition
Concatenation	$f_i(g)\|\|g_i(d)$
Average	$\frac{f_i(g)+g_i(d)}{2}$
Hadamard	$f_i(g) \times g_i(d)$
Weighted-L1	$\|f_i(g) - g_i(d)\|$
Weighted-L2	$\|f_i(g) - g_i(d)\|^2$

Table 5.1: Mathematical operations.

The fourth step in the link classification approach consists of training supervised learning algorithms with the pair representations to make predictions for unseen pairs (during training). We applied NB, MLP, XGB and RF (described in Section 2.2.1) and used the set of hyperparameters that yield the best model performance (of each of them) according to Grid-Search exploration.

Grid-Search is a hyperparameter optimization technique that systematically searches through a predefined grid of hyperparameters for an ML model. This method exhaustively explores the hyperparameter space, calculating model performance metrics (such as accuracy, F1 score, etc.) for each configuration [14]. We tested the parameters experimented in Nunes et al. [53], as shown in Table 5.2.

5.3 Link Classification Accuracy Measures

Measuring the performance of supervised learning models involves using evaluation metrics adequate to the task or application for which we use these methods. We analyzed the WAF, precision and recall to

Algorithm	Parameter	Values
RF	Maximum Depth:	2, 4, 6, None
	Nr of Estimators:	50, 100, 200
XGB	Maximum Depth:	2, 4, 6
	Nr of Estimators:	50, 100, 200
	Learning Rate:	0.1, 0.01, 0.001
MLP	Hidden Layer Sizes:	(50, 50, 50), (50, 100, 50), (100)
	Activation:	tanh, relu
	Solver:	sgd, adam
	Alpha:	0.0001, 0.05
	Learning Rate:	constant, adaptive

Table 5.2: Grid-Search parameters for the supervised learning algorithms.

assess the performance of supervised learning algorithms. These metrics are often used to understand how well classifiers predict various classes.

WAF accounts for class unbalance by computing the F-measure for each interacting and non-interacting class and then calculating the average of both computed F-measures, weighted by the number of instances of each class:

$$WAF = \frac{\sum_{c \epsilon C} F - measure_c \times Support_c}{\sum_{c \epsilon C} Support_c}, \tag{5.1}$$

where C is the set of classes, $F - measure_c$ is the F-measure computed for class c, and $Support_c$ is the number of instances in class c. The F-measure (for a class c) is the weighted harmonic mean of the precision and recall such that

$$\textit{F-measure} = 2 \times \frac{Precision \times Recall}{Precision + Recall}, \tag{5.2}$$

where

$$\textit{Precision} = \frac{\textit{Number of instances correctly classified as class c}}{\textit{Number of instances (correctly and incorrectly) classified as class c}}, \tag{5.3}$$

and

$$\textit{Recall} = \frac{\textit{Number of instances correctly classified as class c}}{\textit{Number of instances correctly classified}}. \tag{5.4}$$

K-fold Cross-Validation (CV) is a popular method for performance evaluation in classification. The method consists of splitting the data into several subsets to evaluate the performance of the classifier using one subset as test set and the remaining as training set.

Stratified CV is when the ”folds” are stratified so that the class distribution of the triples in each fold is approximately the same as that in the initial data. This ensures balanced representation, aids in reliable model evaluation on varied data subsets, reduces overfitting risks and facilitates effective parameter tuning through multiple iterations [14].

We did a 10 (K)-fold Stratified CV in each experiment, evaluating and reporting the WAF, precision and recall of classifications as median and Interquartile Range (IQR) between all folds [42]. Therefore, we obtain 1638 gene-disease pairs in each fold.

5.4 Results and Discussion

Several factors of the link classification methodology applied for rich semantic representations can impact the performance of gene-disease association prediction: semantic richness and domain coverage of the KGs, the KGE models, and the operators used to combine gene and disease vectors. Given these factors, three critical aspects need to be considered when elucidating the performance impact:

1. How can gene and disease vectors be combined?
2. Which KGE model is more suitable for link classification?
3. What is the impact of enriching the semantic representation of diseases and the impact of including additional links between ontology classes?

In response to these queries, we begin by comparing the vector operations, presenting the results using the best KG with the embedding models. Then, we explore the performance of KGE models by presenting the results using the standout classifier with the different KGs. Finally, we contrast the different KGs by presenting the most informative and representative combined vectors with the ML classifiers.

5.4.1 Comparison of Vector Combination Approaches

One of the challenges in exploring the best semantic representation of genes and diseases when using KGEs is to define a suitable approach to combine the gene and disease vectors. Table 5.3 provides a detailed view of the median and IQR of the WAF scores achieved for each operator using all embedding models and ML algorithms in the same KG.

The Hadamard operator outperformed other operators using OPA2Vec combined with the XGB algorithm. It also achieved promising results by combining the same KGE model with the RF classifier. The second operator that best contributed to predicting associations between genes and diseases was Concatenation when combined with OPA2Vec and XGB. However, OPA2Vec combined with the MLP algorithm is similar when combined with Hadamard and Concatenation.

Varying the embedding model and the supervised learning algorithm in each way of combining vectors, NB was the classifier with the lowest performance in RDF2Vec and OPA2Vec. MLP was generally

Operator	Supervised Algorithm	Embedding Model			
		RDF2Vec		OPA2Vec	
Concatenation	**NB**	0.504	(0.0060)	0.493	(0.0041)
	MLP	**0.753**	(0.0150)	0.761	(0.0129)
	XGB	0.712	(0.0117)	0.762	(0.0122)
	RF	0.699	(0.0066)	0.750	(0.0105)
Average	**NB**	0.622	(0.0160)	0.569	(0.0175)
	MLP	0.732	(0.0098)	0.729	(0.0169)
	XGB	0.715	(0.0147)	0.696	(0.0089)
	RF	0.728	(0.0147)	0.690	(0.0107)
Hadamard	**NB**	0.626	(0.0221)	0.547	(0.0061)
	MLP	0.739	(0.0124)	0.761	(0.0169)
	XGB	0.740	(0.0180)	**0.773**	(0.0186)
	RF	0.743	(0.0109)	0.770	(0.0117)
Weighted-L1	**NB**	0.681	(0.0117)	0.563	(0.0163)
	MLP	0.694	(0.0206)	0.701	(0.0158)
	XGB	0.699	(0.0089)	0.686	(0.0108)
	RF	0.702	(0.0180)	0.682	(0.0139)
Weighted-L2	**NB**	0.664	(0.0150)	0.528	(0.0076)
	MLP	0.705	(0.0143)	0.707	(0.0810)
	XGB	0.699	(0.0086)	0.686	(0.0099)
	RF	0.702	(0.0162)	0.687	(0.0080)

Table 5.3: Assessment of vector combination methods (WAF scores) utilizing the **GO + HPO + Mappings** graph. The values within parentheses represent the IQR. The best possible result in each KGE is highlighted in bold.

best using the non-random walk-based method to create node representations for genes and diseases, achieving the best results for RDF2Vec concatenating these entity vectors. For both XGB and RF, this pattern does not occur, being in most cases better to use the random walk-based method.

While a higher WAF indicates better classifier performance on the median across the ten folds, a lower IQR suggests greater consistency across folds. The Concatenation operator outperformed other operators in terms of uniformity between folds. The best value for the IQR was with the NB algorithm, which demonstrated worse results in predicting the gene-disease associations, reaching an IQR of 0.0041 with OPA2Vec. The second-best result was also with the NB algorithm but for RDF2Vec.

The Weigthed-L2 operator showed promising results concerning the XGB algorithm for RDF2Vec and OPA2Vec. Finally, Average and Weighted-L1 performed similarly against XGB with OPA2Vec and against XGB for RDF2Vec, respectively. However, performance varied greatly within each vector com-

bination method, pointing only to Weighted-L2 as the second method that best contributed to consistency between folds.

Overall, the Hadamard operator outperforms other operators in classifying gene-disease pairs with XGB and RF. The Concatenation operator achieves optimistic results combining OPA2Vec with the XGB algorithm. However, the results are less consistent with Hadamard than with Concatenation. Considering the small losses in performance using Hadamard with XGB, all the remaining results consider the combination of the Hadamard operator with the XGB algorithm.

5.4.2 Comparison of Knowledge Graph Embedding Methods

The choice of an embedding method can significantly influence the performance of gene-disease association prediction. A well-selected embedding technique like OPA2Vec can capture complex relationships and semantic nuances within the biological context. Table 5.4 compares the KGE models, presenting the performance obtained for the XGB algorithm for all the possible combinations of KGE algorithms, operators, and KGs.

Embedding Model	**Operator**	**Knowledge Graph**							
		GO + HPO	**GO + HPO + LD**	**GO + HPO + Map**	**GO + HPO + LD + Map**	**GO + HPO + DO**	**GO + HPO + LD + DO**	**GO + HPO + Map + DO**	**GO + HPO + LD + Map + DO**
RDF2Vec	**Concatenation**	0.732	0.721	0.712	0.718	0.716	0.718	0.715	0.719
	Average	0.707	0.707	0.715	0.710	0.711	0.708	0.703	0.705
	Hadamard	0.742	0.739	0.740	0.741	0.747	0.735	0.739	0.738
	Weighted-L1	0.707	0.697	0.699	0.694	0.699	0.700	0.705	0.696
	Weighted-L2	0.707	0.697	0.699	0.694	0.701	0.700	0.705	0.697
OPA2Vec	**Concatenation**	0.762	0.764	0.762	0.764	0.761	0.757	0.759	0.755
	Average	0.694	0.696	0.696	0.701	0.703	0.701	0.698	0.706
	Hadamard	**0.769**	**0.764**	**0.773**	**0.769**	**0.763**	**0.764**	**0.767**	**0.766**
	Weighted-L1	0.675	0.680	0.686	0.679	0.677	0.680	0.682	0.679
	Weighted-L2	0.676	0.682	0.686	0.679	0.678	0.682	0.682	0.679

Table 5.4: Median of WAF scores for the competing combination of the KGE and vector operators for the different KGs using XGB. The best possible result in each KG is highlighted in bold and the best possible result in each KGE is underlined.

The OPA2Vec embedding model outperformed the RDF2Vec in all KGs. The optimal outcome with RDF2Vec was achieved with **GO + HPO + DO**, losing 2.6% in WAF for the best result. RDF2Vec was

more consistent when analyzing performance across multiple operators and KGs. While RDF2Vec was better with Concatenation, Average and Hadamard, OPA2Vec was better with just Concatenation and Hadamard. In both embedding methods, Weighted-L1 and Weighed-L2 performed similarly.

OPA2Vec generally achieved the best results when combined with the Hadamard operator, with the best score of 0.773 as the median of the 10 folds and the slightest difference between folds (0.0036). Multiple factors can explain the better performance of OPA2Vec: it uses asserted and inferred logical axioms in ontologies using a reasoner; it combines them with vector representations for the lexical component of the ontologies learned over PubMed abstracts using the Word2Vec model.

RDF2Vec was thus positioned as the second-best performer, with 0.747 WAF, showing the potential of random walk-based methods. A clear difference between the two embedding models is the use of rich OWL axioms and word embeddings, which may explain the observed differences. Biomedical ontologies are rich in synonyms, and exploring their similarities in the scientific literature can be very informative.

Table C.1, which is located in Appendix C, analyzes the KGE methods regarding the IQR of WAF scores, showing the performance reached for the XGB algorithm for all possible combinations of KGE approaches, operators and KGs.

5.4.3 Comparison of Knowledge Graphs

The quality, diversity, and representativeness of a KG directly impact its effectiveness in enabling models to discover patterns and make accurate predictions in real-world scenarios. Table 5.5 presents the performance obtained across all KGs, KGE models and supervised learning algorithms.

The GO and HPO ontologies combined with LD and mappings provide the best ontological combinations, achieving a median WAF of 0.773 when combining OPA2Vec with XGB and RF. However, the full KG (without DO) aggregated to RF offered greater consistency between folds. The **GO + HPO + LD** and **GO + HPO + Map + DO** graphs similarly contributed to gene-disease association prediction when incorporated with the MLP algorithm.

Unlike the other supervised learning algorithms, the NB classifier continuously performed better when RDF2Vec is applied over the KGs. The best result for this classifier was reached with all ontologies, and LD between GO and HPO were used. Nonetheless, the **GO + HPO + Map** graph allows for greater homogeneity between folds utilizing OPA2Vec as an embedding method.

When combining only the GO with HPO, the simple version without LD and mappings performed better than the version with both types of links. The same pattern occurs for XGB and RF when DO is one of the ontologies in the KGs, suggesting that this additional information could generate background noise irrelevant to gene-disease association prediction. However, both embedding models drop little to no performance between these KGs.

Adding DO does not significantly improve the results for either embedding technique, especially when using ensemble models (XGB and RF). This also applied to the difference in performance between folds, notably for MLP and RF. DO may not add relevant information to the problem in link classification

Knowledge Graph	Embedding Model	Supervised Algorithm							
		NB		MLP		XGB		RF	
GO + HPO	RDF2Vec	0.622	(0.0143)	0.739	(0.0114)	0.742	(0.0144)	0.758	(0.0112)
	OPA2Vec	0.549	(0.0167)	0.767	(0.0066)	0.769	(0.0097)	0.769	(0.0088)
GO + HPO + LD	RDF2Vec	0.630	(0.0137)	0.738	(0.0111)	0.739	(0.0244)	0.739	(0.0174)
	OPA2Vec	0.547	(0.0216)	**0.768**	(0.0053)	0.764	(0.0099)	0.769	(0.0088)
GO + HPO + Map	RDF2Vec	0.626	(0.0221)	0.739	(0.0124)	0.740	(0.0180)	0.743	(0.0109)
	OPA2Vec	0.547	(0.0061)	0.761	(0.0169)	**0.773**	(0.0186)	0.770	(0.0117)
GO + HPO + LD + Map	RDF2Vec	0.629	(0.0083)	0.737	(0.0078)	0.741	(0.0115)	0.751	(0.0116)
	OPA2Vec	0.550	(0.0149)	0.758	(0.0113)	0.769	(0.0160)	**0.773**	(0.0053)
GO + HPO + DO	RDF2Vec	0.633	(0.0124)	0.727	(0.0128)	0.747	(0.0121)	0.745	(0.0109)
	OPA2Vec	0.548	(0.0144)	0.763	(0.0206)	0.763	(0.0700)	0.768	(0.0107)
GO + HPO + LD + DO	RDF2Vec	**0.638**	(0.0141)	0.734	(0.0059)	0.735	(0.0055)	0.742	(0.0158)
	OPA2Vec	0.547	(0.0210)	0.766	(0.0135)	0.764	(0.0144)	0.767	(0.0102)
GO + HPO + Map + DO	RDF2Vec	0.631	(0.0070)	0.738	(0.0148)	0.739	(0.0129)	0.742	(0.0127)
	OPA2Vec	0.550	(0.0114)	**0.768**	(0.0128)	0.767	(0.0097)	0.768	(0.0129)
GO + HPO + LD + Map + DO	RDF2Vec	0.634	(0.0186)	0.742	(0.0103)	0.738	(0.0089)	0.733	(0.0167)
	OPA2Vec	0.547	(0.0111)	0.766	(0.0135)	0.766	(0.0036)	0.770	(0.0102)

Table 5.5: WAF scores for the combinations of KGE and supervised learning algorithms for the different KGs using the Hadamard operator. The values within parentheses represent the IQR. The best possible result in each supervised learning algorithm is highlighted in bold.

because embedding models only generate representations for genes and diseases without knowing true gene-disease associations.

In brief, the best combinations of KG with an embedding method and a supervised learning algorithm obtained 0.773 WAF by using the XGB and RF classifiers with the Hadamard operator and OPA2Vec model. MLP also showed promising results using the non-random-walk technique in both KGs with and without DO. The results are aligned with the fact that ensemble methods are among the most popular and best-performing classifier algorithms.

The poor results achieved with the NB classifier may be justified by the fact that this algorithm's main limitation is the assumption of independent predictor features. This algorithm implicitly assumes that all the attributes are mutually independent, which is invalid in the context of linked data.

Analyzing the precision and recall of a model provides insights into its performance by measuring the trade-off between correctly identified positive instances (precision) and the ability to capture all positive instances (recall). This evaluation aids in understanding the model's effectiveness in correctly classifying relevant instances and identifying areas for improvement in classification outcomes.

Figure 5.2 and Figure 5.3 show precision and recall values for both embedding methods using XGB and Hadamard. For RDF2Vec, these values reveal that the simplest versions of the KGs (with and without

the DO) are preferred. This is observed for both precision and recall. In the case of OPA2Vec, performance generally increased with more links between GO and HPO. The best KG was the complete version without the DO.

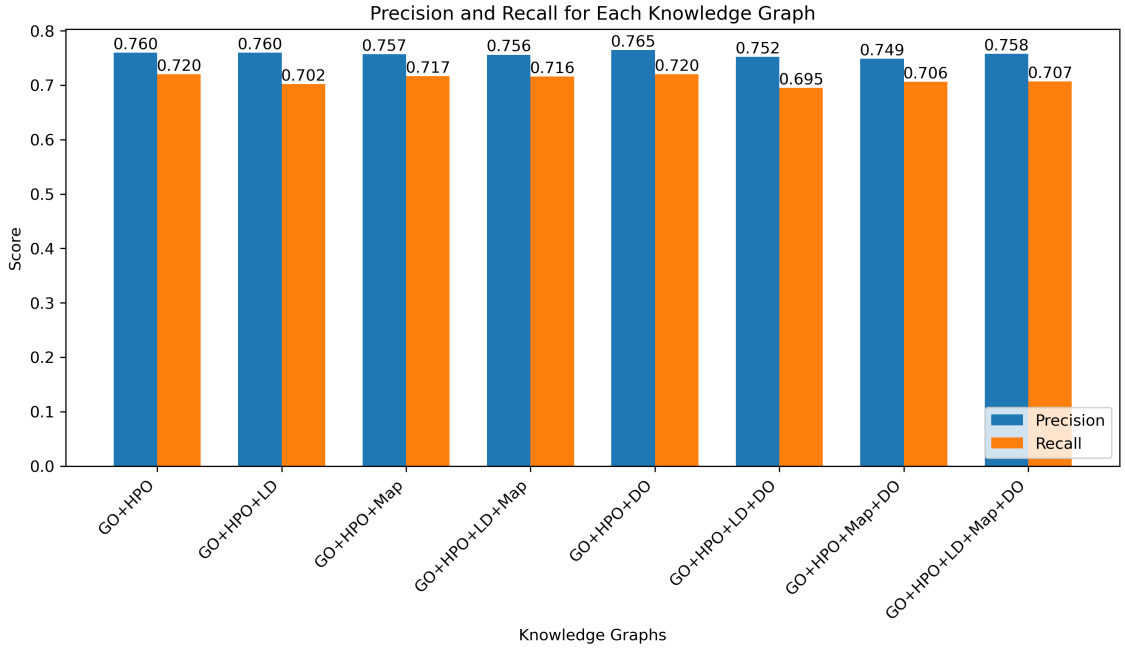


Figure 5.2: Precision and recall for RDF2Vec using XGB and Hadamard. The KGs appear on the x-axis, and the scores on the y-axis.

Overall, the differences between using GO and HPO or combining these two ontologies with LD, mappings or DO were comparatively small regardless of the best graph which uses GO, HPO and mappings between these two ontologies. Comparing the embedding methods, OPA2Vec generally performed better. Additionally, it showed a smaller difference between precision and recall, resulting in a more balanced and effective classifier that accurately identifies positive instances while capturing a higher proportion of all positive instances.

The small contribution to performance observed when adding LD can be partially explained by only 350 (out of 3,293) relations that effectively link the GO and HPO ontologies. The higher recall scores when adding mappings can be interpreted as these links allow for a more comprehensive representation of relevant relationships across diverse knowledge domains. Finally, adding DO does not significantly improve gene-disease association prediction as DO may introduce redundant information or overlap in biological annotations, minimizing the additional discriminative value it provides in the context of the existing ontological experiments.

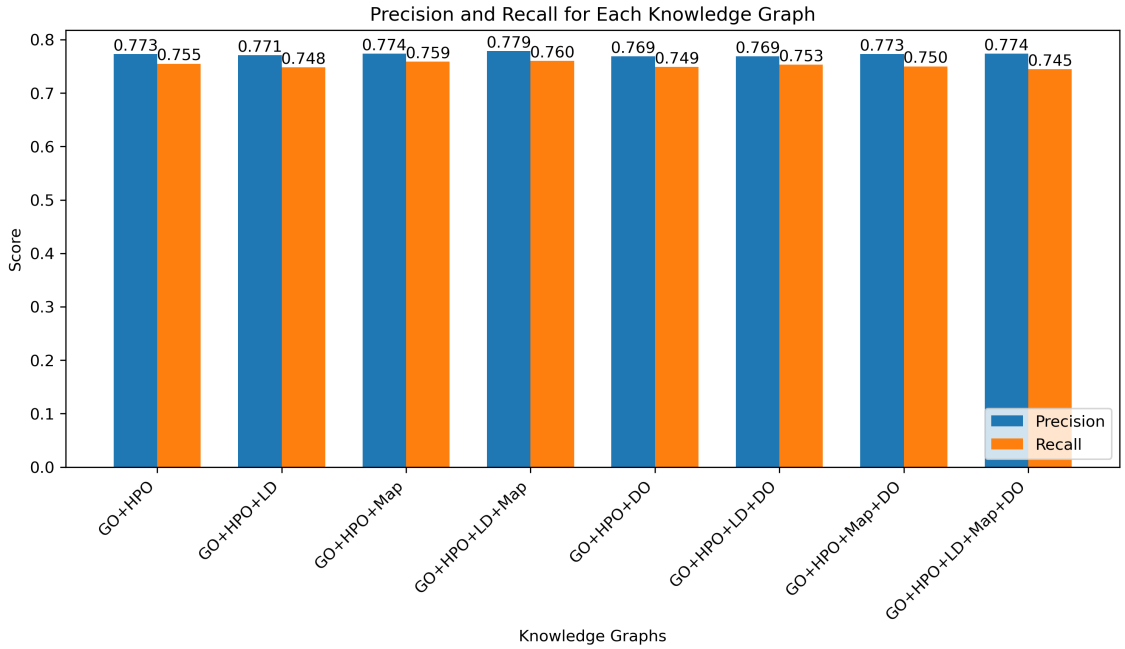


Figure 5.3: Precision and recall for OPA2Vec using XGB and Hadamard. The KGs appear on the x-axis, and the scores on the y-axis.

Chapter 6

Predicting Gene-Disease Links

This section presents and describes the link prediction methodology developed during this work. Section 6.1 outlines the strategy to predict links. Section 6.2 clarifies the prediction strategy. Section 6.3 introduces the performance measures used to assess the model's skill and capability. Finally, Section 6.4 examines the results obtained in the proposed experiments. Section 2.2.2 and Section 2.3.2 provide a brief introduction and background of the models and methods that are cited in this chapter. The link prediction approach can be found at a GitHub Repository.

6.1 Methodology

The link prediction approach proposed in this dissertation is divided into four main steps (Figure 6.1):

1. **KG Integration:** integrating the ontologies, annotation data, LD and mappings between GO and HPO, and (positive) training gene-disease pairs to build different KGs;

2. **KG Embeddings:** using translational distance models and semantic matching models to obtain gene, disease and relation embeddings;

3. **Scoring Function:** passing the resulting embeddings in the scoring function of the embedding models to get the top 100 candidate entities for each unique gene and disease in the test set;

4. **Predictions Evaluation:** applying rank-based evaluation metrics to assess KGE model's performance.

Distinguishing the link prediction task from the link classification task involves including some positive relationships (those that exist) between the entities that are intended to establish new associations. We performed a 70/30 split on the positive gene-disease pairs so that the training set had all types of nodes and relationships, and the test set only had "association" relationships between genes and diseases.

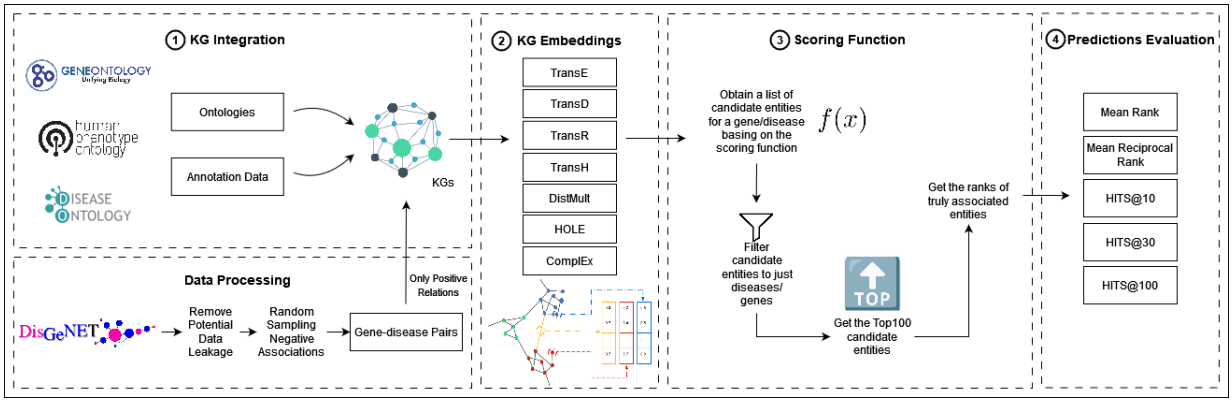


Figure 6.1: Workflow of the link prediction approach with four basic steps: 1) building the KGs with ontologies, annotations and gene-disease associations; 2) creating embeddings; 3) getting the top 100 candidate entities for each gene and disease in the testing set; and 4) evaluating the model's performance.

The decision not to proceed with a 10-fold CV was motivated by the type of task and the algorithms that would learn from the KGs.

The division of pairs in link prediction happens before the embeddings are generated. Using a 10-fold CV could be problematic because a fraction of the data in each fold would be removed, which could significantly change the context of each entity and affect the models' ability to generate representative embeddings. KGE for link prediction requires a lot of processing time. Therefore, frequently restructuring the KG and re-training the model at each fold would drastically increase the computational load and processing time, potentially leading to program interruption.

Regarding the KG integration, our proposed approach considers the GO, HPO, and DO ontologies and the gene-disease associations extracted from DisGeNET. We also explored the full version of KGs (with and without DO) without HPO annotations for genes. The **GO+DO** graph is specifically tailored for link prediction tasks because the gene-disease associations added to the KGs (in this task) establish explicit links between these two ontologies.

As for the embeddings, we applied translational distance models (TransE, TransD, TransR and TransH) and semantic matching models (DistMult, HolE and ComplEx). The use of these link prediction KGE algorithms was motivated by their widespread popularity and proven effectiveness in capturing relationships within KGs. As per the evaluation metrics, we considered a modified version of the Hits@k metric.

6.2 Gene-Disease Prediction

The link classification and link prediction approaches share core principles for predicting associations between genes and diseases. However, their distinct focus dictates the specific techniques and algorithms used to address the unique challenges of each task within this complex problem space. Also, different

KGE are tailored to serve specific purposes and excel in capturing various aspects of information within the graph. Each embedding model is designed with particular strengths, offering varied capabilities to suit different use cases and applications within the realm of KGs.

The link classification strategy combines the KGE algorithms with ML classifiers as path-based models focus on extracting structural information to derive insights or features for various tasks. Examples of insights are connectivity patterns, shortest paths and frequent sequences of relationships. The KGE applied for link prediction, such as translational models, excels at capturing translation-based patterns, which are prevalent in many real-world relationships in KGs. After that, they define scoring functions that measure the plausibility or probability of a triple in the KG [76].

After constructing the KGs, we applied translational distance algorithms (TransE, TransH, TransR and TransD) and semantic matching algorithms (DistMult, HolE and ComplEx), as summarized in Table 6.1. We implemented via the *OpenKE* library (powered by *TensorFlow*) to learn feature vectors for each KG. However, utilizing the *OpenKE* library requires representing KGs through files adhering to a specific format, where a unique identifier denotes each entity and relation of the KG. This was done using the *RDFLib* graph and *Python* dictionaries.

Model	**Scoring Function** $f_t(h,t)$	**Memory Complexity**
TransE	$\|\|h+r-t\|\|_{l_1/l_2}$	$O(N_e d + N_r k)(d=k)$
TransD	$\|\|(r_p h_p^{\parallel} + I)h + r - (r_p t_p^{\parallel} + I)t\|\|_2^2$	$O(N_e d + N_r k)$
TransH	$\|\|(h - w_r^{\parallel} h w_r) + d_r - (t - w_r^{\parallel} t w_r)\|\|_2^2$	$O(N_e d + N_r k)(d=k)$
TransR	$\|\|M_r h + r - M_r t\|\|_2^2$	$O(N_e d + N_r dk)$
DistMult	$h^{\parallel} diag(r) t$	$O(N_e d + N_r k)(d=k)$
HolE	$r^{\parallel}(h \star t)$	$O(N_e d + N_r k)(d=k)$
ComplEx	$Re(h^{\parallel} diag(r)\bar{t})$	$O(N_e d + N_r k)(d=k)$

Table 6.1: KGE models for link prediction, comparing them in terms of scoring functions and memory complexity [21].

The result is a file for the training triples, the test triples, all nodes, all relationships, and all entities with which the ”association” relationship may be related (just genes and diseases). The first two files start with the number of triples, and the following lines are in the format *(e1,e2,rel)*, indicating a relation between the entities *e1* and *e2*. Files with all nodes and all relationships have all entities and corresponding identifiers, one per line.

We implemented the KGE models with the default parameters, as detailed in Appendix D, and conducted training over 100 epochs to optimize their performance. The process culminated in the generation of final embeddings, each characterized by 200 distinct features. This configuration was chosen to balance computational efficiency with the richness of the representation, aiming to capture the complex

relationships within the KGs effectively.

Assuming we are predicting the genes associated with a disease, the third step in the link prediction approach consists of applying the scoring function to embeddings of the known disease and relationship, along with different potential entities. The scoring function calculates scores for each potential entity, indicating how well it fits or aligns with the disease and relationship. We did this for every gene and disease in the test set for every experiment. The result is a list with the top *k* candidate entities to complete the triple. However, we were just interested in genes.

To obtain a list of candidate genes, we filtered the results obtained. Potential gene scores are ranked in order of decreasing likelihood or ”plausibility”. Higher scores for potential genes suggest a stronger likelihood of those entities being the correct match for the disease and relationship. Then, we take the top 100 candidates for the given disease. As a disease is associated with one or more genes (and vice versa), we collected the ranks of all genes truly associated with the disease.

6.3 Link Prediction Assessment Metrics

A list of candidate entities for a gene/disease is similar to a list of recommended movies, such that evaluation metrics typical of recommendation systems are used in link prediction problems. To assess KGE performance, we analyzed a modified version of the Hits@k metric for the top 10, 30 and 100. These metrics are often used to evaluate the performance of a retrieval system based on how well it ranks and recommends relevant items.

Hits@k is a metric and measure of the model used to evaluate the proportion of the correctly predicted entities ranked in the top k among all entities of the same type. The idea is to measure the effectiveness of the algorithm by considering the presence of candidate entities within the top k positions of the candidates' list:

$$\textit{Hits@k} = \frac{1}{n}\sum_{i=1}^{n} hits_i \tag{6.1}$$

where n is the number of possible pairs and $hits_i$ is a binary indicator that is 1 if the candidate entity for the i-th gene/disease is within the top k positions and 0 otherwise [62].

We analyzed Hits@k as the proportion of the correctly predicted entities ranked in the top 10 (30 and 100) among the total number of gene-disease and disease-gene pairs found in the top 100. The goal is to balance computational feasibility and accurately capture true gene-disease associations, reducing the influence of unconfirmed associations.

The distinction between the two Hits@k metrics lies in their application scope and purpose within model evaluation. The first Hits@k metric assesses the precision of the model's predictions in a specific

range, focusing on the model's ability to rank the highest relevant entities. In contrast, the second application of the Hits@k metric evaluates the model's performance in a context that simulates real-world conditions, where the goal is not only to identify relevant associations but to do so in a computationally efficient and practical way for real-world applications.

The rationale for focusing on the top 100 first-ranked entities stems from the necessity to balance computational efficiency with the accuracy of capturing true gene-disease relationships. By concentrating on the top 100 entities, the evaluation method aims to mirror practical scenarios where only the most promising candidates can be feasibly explored further due to time or resource constraints. This approach helps filter out the noise from less likely associations and focuses on those with the highest likelihood of being true positives.

To calculate the modified Hits@k of each model for each experiment, we gathered the ranks of candidate entities in the top 100 truly associated with the given gene or disease and applied the metric described

6.4 Results and Discussion

In this subchapter, we initially focus on the predictive performance of algorithms in predicting diseases associated with a gene and then predicting genes associated with a disease. Our analysis delves into the results from two perspectives regarding gene-disease association within the link prediction task. First, we study which KGE models excel in capturing the relationships within the gene-disease graphs. Then, our analyses extend to the impact of semantic richness and domain coverage of the KGs.

Our analysis focuses on the modified Hits@k metric as it allows for a precision evaluation that considers the relevance of recommendations for a specific entity. Table 6.2 depicts the Hits@10 scores for the different KGE models across all KGs in predicting the diseases associated with a gene.

The TransH algorithm outperformed the other KGE algorithms using KGs with GO, LD and mappings, achieving a Hits@10 of 0.447. This means that 45% of the diseases in the top 10 are truly associated with a given gene. Nonetheless, TransH achieved similar performance when using **GO+HPO+LD+ DO**. TransE was the second-best algorithm, however, employing GO, HPO, DO and mappings between the first two ontologies.

TransD was the third-best algorithm in ranking the diseases associated with a given gene, reaching a Hits@10 of 0.375. HolE, DistMult and ComplEx performed similarly, utilising KGs with GO and HPO ontologies and LD between these ontologies. Nevertheless, HolE performed best when including mappings between GO and HPO and ComplEx when including the DO. The best result with TransR was achieved with the simplest graph applying the GO and HPO ontologies.

Varying the KG across each KGE model, TransH performance did not vary significantly. TransE, TransD and HolE exhibit the same behaviour. However, TransH and TransD performance increased as KGs are more semantically rich, whereas TransE and HolE performance decreased. The only difference between TransE and HolE was that while TransE's performance was better with more links between GO and HPO in the presence of DO, HolE's performance was better in the absence of DO.

Knowledge Graph	Embedding Model						
	TransE	**TransD**	**TransH**	**TransR**	**DistMult**	**HolE**	**ComplEx**
GO + HPO	0.399	0.306	0.374	**0.285**	0.299	0.322	0.268
GO + HPO + LD	0.369	0.321	0.412	0.252	**0.311**	0.314	0.271
GO + HPO + Map	0.403	0.329	0.408	0.127	0.254	0.313	0.271
GO + HPO + LD + Map	0.392	0.333	0.413	0.246	0.263	**0.326**	0.272
GO + HPO + DO	0.400	0.335	0.433	0.205	0.232	0.290	0.289
GO + HPO + LD + DO	0.411	0.324	0.443	0.187	0.239	0.239	**0.304**
GO + HPO + Map + DO	**0.430**	0.349	0.415	0.237	0.258	0.277	0.290
GO + HPO + LD + Map + DO	0.426	0.303	**0.447**	0.199	0.228	0.324	0.256
GO + HPO* + LD + Map	0.394	0.365	**0.447**	0.211	0.180	0.241	0.170
GO + DO	0.413	**0.375**	0.421	0.220	0.273	0.308	0.241
GO + HPO* + LD + Map + DO	0.423	0.338	0.401	0.179	0.148	0.319	0.207

Table 6.2: Assesment of Hits@10 for KGE models over all experiments in predicting the diseases associated with a gene. The best possible score in each link prediction model is highlighted in bold.

The difference between the highest and lowest TransR, DistMult and ComplEx results was greater than 1%. Whereas the performance of TransR decreased with larger graphs, the results of DistMult and CompLEx generally increased. This may be related to the fact that as the KG grows in complexity with additional elements, the translational distances between entities may become less discriminative, leading to decreased performance. Otherwise, semantic matching models leverage the semantic similarities between entities to better capture the complex relationships within the KG.

Overall, the performance of the algorithms increases as more links between GO and HPO are added. It is equally preferable that KGs include LD or mappings when DO is not included. However, the same does not occur in the presence of DO, where the predictive capabilities of KGE methods are higher with mappings between GO and HPO. The presence or absence of DO is also the best scenario for predicting diseases associated with a gene.

Table 6.3 provides a detailed view of the Hits@30 scores for the different ontology combinations across the KGE algorithms in predicting the diseases associated with a gene. Considering the top 100 candidate diseases for a specific gene and relation, we will be evaluating 30% of this dataset. Although TransH had the best result, its performance only differs by 1.6% from TransE. The best result with TransH was achieved with the **GO+HPO*+LD+Map** graph, hitting 70% of the diseases in the top 30.

TransE had the same result with the most complete graph (with DO) and with one of the simplest graphs - the one that uses GO and DO ontology data and annotations. The TransD algorithm also had the best score with this simplest KG. In the case of the other algorithms, their performance can be considered quite similar, below 1% between the best result of each one. The algorithm with the lowest predictive capacity to predict diseases associated with a gene was TransR, with a Hits@30 value of 0.524.

Among the different experiments, the results for TransH and DistMult consistently improved when-

Knowledge Graph	Embedding Model						
	TransE	**TransD**	**TransH**	**TransR**	**DistMult**	**HolE**	**ComplEx**
GO + HPO	0.633	0.633	0.655	0.523	0.581	0.575	0.559
GO + HPO + LD	0.612	0.596	0.665	0.519	**0.585**	0.580	0.556
GO + HPO + Map	0.666	0.605	0.646	0.428	0.548	0.576	0.557
GO + HPO + LD + Map	0.659	0.629	0.684	0.487	0.549	0.583	0.571
GO + HPO + DO	0.652	0.629	0.686	0.478	0.510	0.566	0.583
GO + HPO + LD + DO	0.684	0.624	0.691	0.472	0.535	0.505	**0.587**
GO + HPO + Map + DO	0.668	0.624	0.667	**0.524**	0.547	0.546	0.574
GO + HPO + LD + Map + DO	**0.692**	0.612	0.677	0.458	0.509	**0.610**	0.549
GO + HPO* + LD + Map	0.673	0.646	**0.708**	0.508	0.433	0.528	0.435
GO + DO	**0.692**	**0.648**	0.654	0.478	0.567	0.560	0.530
GO + HPO* + LD + Map + DO	0.677	0.641	0.662	0.448	0.419	0.467	0.491

Table 6.3: Assessment of Hits@30 for KGE models over all experiments in predicting the diseases associated with a gene. The best possible score in each link prediction model is highlighted in bold.

ever LD were added to the KGs. TransE performance especially decreases with the addition of links between GO and HPO in the absence of DO but increases when the description and features of diseases are included. The other algorithms were more inconsistent across the various experiments. In general, the algorithms that showed greater consistency were translational distance-based algorithms.

Including LD or mappings in the KGs doesn't seem to substantially impact the performance of several models. However, DistMult and ComplEx performed better with LD, and TransR performed better with mappings between GO and HPO. The results are generally worse when we omit the HPO annotations for genes, especially for expressive embedding models designed to capture complex relationships in KGs - DistMult, HolE and ComplEx.

Overall, the translational distance algorithms perform best and worst in identifying the most relevant candidate diseases within the top 30. Semantic matching models have similar execution but with very different graphs. In most algorithms, there is little difference between a simple KG and a KG with more links between GO and HPO. Without DO, it is better with LD than with mappings. With DO, it is equally preferable to have LD or mappings. KGs with DO provide accurate embeddings to predict disease-associated genes.

Predicting the genes associated with a disease is expected to be more challenging due to the genetic heterogeneity of diseases, their polygenic nature, and complex inheritance patterns, among other factors described in Section 3.1. These factors introduce complexity in identifying consistent patterns, making the accurate prediction of gene-disease associations a more complex problem. Table 6.4 compares the Hits@10 values through the KGs for the different KGE algorithms in predicting the genes associated with a disease.

The TransD embedding method surpassed the other algorithms with a Hits@10 value of 0.345 uti-

Knowledge Graph	Embedding Model						
	TransE	**TransD**	**TransH**	**TransR**	**DistMult**	**HolE**	**ComplEx**
GO + HPO	0.138	0.281	0.115	0.091	0.233	0.169	0.249
GO + HPO + LD	0.101	**0.345**	0.132	**0.164**	**0.288**	0.140	0.234
GO + HPO + Map	0.090	0.334	0.133	0.111	0.214	0.150	0.250
GO + HPO + LD + Map	0.142	0.318	0.122	0.085	0.231	**0.224**	0.225
GO + HPO + DO	0.105	0.334	0.172	0.147	0.175	0.154	0.276
GO + HPO + LD + DO	0.109	0.332	0.122	0.140	0.244	0.180	0.258
GO + HPO + Map + DO	**0.158**	0.318	0.106	0.104	0.236	0.202	0.238
GO + HPO + LD + Map + DO	0.151	0.328	**0.181**	0.109	0.245	0.179	0.244
GO + HPO* + LD + Map	0.129	0.172	0.122	0.093	0.086	0.141	0.082
GO + DO	0.118	0.338	0.145	0.116	0.242	0.170	**0.277**
GO + HPO* + LD + Map + DO	0.136	0.158	0.122	0.092	0.035	0.104	0.084

Table 6.4: Evaluation of Hits@10 for KGE models over all experiments in predicting the genes associated with a disease. The best possible score in each link prediction model is highlighted in bold.

lizing **GO+HPO+LD**. DistMult and ComplEx were the second and third-best models, respectively, with DistMult leveraging the same KG as TransD and ComplEx leveraging only GO and DO ontology data and annotations. Nonetheless, with similar performance, DistMult hit another 1.1% of the genes associated with a disease.

The TransH and HolE algorithms also had very similar performance, using both semantically richer graphs, with the difference that TransH used mappings in addition to LD. TransR and TransE were the worst algorithms to rank genes associated with a disease in the top 10. They had similar results, however, using very different KGs. While TransR uses the same graph as the two best algorithms in this task, TransE uses all ontologies and mappings between GO and HPO.

For the same KGs, the performance of link prediction methods was generally better in the presence of DO. In 40% of the models, it was better with just the mappings than just with the LD (without the DO), and in 90% of the models, it was better with just the LD (with the DO). When the HPO annotations for genes are omitted, a part of the algorithms achieved better results without DO, and another part of the algorithms achieved similar outcomes. The only exception was TransE, which made accurate predictions with DO.

There were more algorithms that differed by more than 10% between the worst and best results. Within these, the KGs they employ differ in ontologies and knowledge domain. For instance, the best result with TransD and DistMult was with **GO+HPO+LD**, while the worst result for these algorithms was with **GO+HPO*+LD+Map+DO**. For algorithms that do not vary much throughout the KGs, the difference from the worst to the best value includes LD or DO.

In summary, most algorithms perform better without DO. Whereas the best graph represents diseases by their phenotypic characteristics, the second-best graph represents diseases by their description and

attributes. There is a notable dependence on HPO annotations for genes, except for TransH and TransR. The result of TransH is the same without DO and better with DO. For TransR, Hits@10 is highest without DO and lowest with DO when HPO annotations for genes were not included.

Ultimately, we analyze the proportion of genes in the top 30 that are confirmed to be associated with a disease. Table 6.5 displays the Hits@30 results for the different KGE methods across all KGs. TransD was the best algorithm, performing similarly in two KGs - better with the GO, HPO and DO ontologies and annotation data, and also good with **GO+HPO+Map**. This suggests that TransD might be a robust model that is less sensitive to the specific details or additional complexities the KGs introduce.

Knowledge Graph	**Embedding Model**						
	TransE	**TransD**	**TransH**	**TransR**	**DistMult**	**HolE**	**ComplEx**
GO + HPO	0.326	0.502	0.331	0.299	0.510	0.415	0.494
GO + HPO + LD	0.336	0.557	**0.375**	**0.359**	**0.524**	0.380	0.485
GO + HPO + Map	0.277	0.600	0.305	0.289	0.454	0.420	0.487
GO + HPO + LD + Map	**0.390**	0.563	0.333	0.294	0.476	0.438	0.447
GO + HPO + DO	0.323	**0.602**	0.359	0.339	0.419	0.374	**0.542**
GO + HPO + LD + DO	0.289	0.591	0.293	0.345	0.476	0.405	0.510
GO + HPO + Map + DO	0.336	0.560	0.341	0.331	0.503	**0.446**	0.490
GO + HPO + LD + Map + DO	0.325	0.581	0.368	0.318	0.501	0.444	0.480
GO + HPO* + LD + Map	0.294	0.395	0.304	0.209	0.258	0.396	0.356
GO + DO	0.312	0.594	0.339	0.335	0.505	0.384	0.534
GO + HPO* + LD + Map + DO	0.336	0.370	0.367	0.310	0.244	0.381	0.310

Table 6.5: Evaluation of Hits@30 for KGE models over all experiments in predicting the genes associated with a disease. The best possible score in each link prediction model is highlighted in bold.

TransD, DistMult and HolE performance significantly decreased when HPO annotations for genes were not included, being the worst-case scenarios (with and without DO) for these models. ComplEx and DistMult were the second and third-best KGE algorithms, capturing at least 50% of the genes associated with a disease. The other algorithms performed similarly, with translational distance models being greater without DO and HolE better with semantically rich and domain coverage KGs.

Examining the results across different KG compositions, the performance was similar between algorithms but significantly changed within each algorithm. TransD was noticeably better in the majority of the KGs. However, TransH and HolE showed greater consistency between the various KG compositions. There was an inconsistency in the results within each KG, especially greater in **GO+HPO+Map** and **GO+HPO+DO**, and smaller in graphs without HPO annotations for the genes.

Without DO, graphs with LD between GO and HPO were a greater choice for predicting the disease-associated genes in 60% of the link prediction models. The performance of TransD and ComplEx was higher with simpler KGs with all ontologies. Conversely, the performance of DistMult and HolE was better with mappings than LD. Simpler graphs with one type of links between GO and HPO were preferable

to a full version, both with and without DO.

The results show that semantic matching models were more consistent between algorithms of the same category. Adding the DO does not significantly provide accurate embeddings for predicting the genes associated with a disease. It was preferable to simpler graphs with one type of links between GO and HPO than a full version of KGs. The HPO annotations for genes supply relevant information, especially in DistMult.

In the context of evaluating the performance of the KGE algorithms with the modified version of the Hits@k metric, the ratio between the number of items in the top 100 and the number of pairs found in the top 100 consistently equals 1. This occurs because each confirmed entity in the top 100 contributes to one pair when combined with given subjects and predicates, comprising valid triples. Consequently, the count of pairs found in the top 100 matches the count of items in the same list.

KGs with multiple ontologies significantly enhance the prediction of diseases associated with a gene. The presence of DO is key, offering a common language for disease definitions. In predicting the genes linked to a disease, the HPO is sufficient to represent diseases. HPO's detailed descriptions and attributes of disease phenotypes offer essential insights that bridge the gap between clinical manifestations and genetic mechanisms.

At an algorithmic level, translational distance-based models are better choices for gene-disease association prediction. Specifically, TransH outperforms other algorithms in predicting diseases linked to a gene. TransD shows a greater ability to predict disease-associated genes. Despite the varied performance in these specific tasks, TransD consistently ranks among the top three algorithms in predicting gene-disease associations. The performance of semantic matching models is competitive. However, TransR usually had worse results than other translational distance algorithms.

The superior performance of the TransD in predicting the genes associated with a disease may be related to its design incorporating relation-specific translation vectors, separate entity and relation embeddings, and relation-specific projection matrices, allowing it to effectively capture diverse semantic relationships and complex patterns.

DistMult, HolE, and ComplEx's similar performance may be explained by the fact that these algorithms are all expressive embedding models designed to capture complex relationships in KGs. Also, the quantity of embeddings produced by these models might be consistently high, resulting in competitive performance regardless of the KGs composition.

The poor results of TransR may be attributed to limitations in capturing complex relation patterns or handling entity-specific translation operations in the knowledge representation. This is a possible justification for the fact that this algorithm has better performance on simpler graphs.

Results for MR and MRR are presented in Appendix E, and the outcomes for the original Hits@k metric are displayed in Appendix F.

Chapter 7

Conclusion

This chapter concludes this dissertation. Section 7.1 summarises the contents of this work and discusses the results of link classification and link prediction, which answers the research questions. Finally, we point out some possible paths for future work in Section 7.2.

7.1 Discussion

Discovering gene-disease links is an important area of research with applications to understand disease origin and develop new prevention, diagnosis and therapy techniques. Computational approaches based on KG and ML provide a robust framework for addressing the complexity and scale of biological data. In particular, KGs enriched with ontological information offer a structured representation of biological entities and their relationships, facilitating predictive modelling tasks.

While aiming at understanding if link prediction methods explore semantic richness better than link classification methods and whether enriching the semantic representation of diseases improves performance in gene-disease association prediction, this dissertation aimed to explore the differences between approaching a problem as a link prediction task and a link classification task. The link classification approach combines KGE with popular ML algorithms, and the link prediction approach solely relies on KGE.

Regarding the link classification approach, the embeddings generated by OPA2Vec, combined with the Hadamard operator and used by the XGB or RF algorithm, offer the most effective algorithmic combinations for gene-disease association. Concerning the link prediction approach, translational distance-based methods are the top-performing KGE algorithms for accurately predicting gene-disease associations, especially the TransD model.

After studying which methods excel in predicting gene-disease associations, we analyze the impact of the semantic richness of the KGs. KGs with additional links between ontology classes support better performance than simpler KGs with just ontologies when identifying and classifying a pair of nodes.

However, this occurs specifically in graphs containing only GO and HPO ontologies. For graphs with two or three ontologies, using LD or mappings is better than using both types of links. Using OPA2Vec, the more links between GO and HPO, the better the accuracy and completeness of positive predictions.

KGs with LD or mappings better predict gene-disease links than simpler graphs with two or three ontologies. Nonetheless, simpler graphs outperform KGs with one of those type of links in predicting the genes linked to a disease when including DO in the KG. We also found that KGs with one type of link between GO and HPO are better than KGs with both types (of links) in link prediction. The only exception is predicting gene-associated diseases in the absence of the DO.

HPO annotations for genes significantly enhance the performance of link prediction models. This improvement can be related to the phenotypic manifestations associated with specific disease-related genes (in the KGs) the HPO annotations provide. Our results suggest that, in the absence of HPO annotations for genes, we have better results without DO than enriching the semantic representation of diseases.

The two best KGs in identifying and classifying a pair of nodes were found to be **GO+HPO+Map** and **GO+HPO+LD+Map**. To predict the diseases associated with a gene, **GO+HPO+LD+Map+DO** and **GO+HPO*+LD+Map** graphs outperformed the other KGs. To predict the genes associated with a disease, **GO+HPO+LD** and **GO+HPO+DO** graphs provided better performance to the link prediction algorithms.

The results indicate that both link classification and link prediction methods leverage the semantic richness of KGs for gene-disease association prediction. However, the optimal composition of KGs differs based on the specific task. The distinction in results suggests that link prediction methods are particularly better at using the semantic richness encoded in KGs, which are incorporated through various ontologies and additional links between ontology classes.

Link prediction tasks focus on discovering potential links based on the structure and properties of the graph, possibly benefiting more from the multi-dimensional semantic information provided by combining GO, HPO, DO, LD and mappings. This answers **Research Question 1** (in Section 1.2) as the distinction in performance across different tasks and KG compositions suggests that link prediction methods explore semantic richness better than link classification methods.

The performance of link classification methods in gene-disease association prediction does not significantly improve when enriching the semantic representation of diseases with DO. This also occurs when predicting genes associated with a disease, where the description and attributes of disease phenotypes provide essential insights for the task. Nonetheless, predicting diseases associated with a gene improves when enriching the semantic representation of diseases, answering **Research Question 2** (in Section 1.2).

Employing a link prediction task over a link classification task as a representation strategy offers several advantages across various design aspects and techniques. Unlike the link classification approach in this dissertation, where associations between genes and diseases are only seen by supervised learning algorithms, link prediction approaches leverage these relationships within the KGs, able to explore another aspect of the semantic richness. Furthermore, link prediction does not require the synthetic generation of negative examples.

Whereas the link classification approach requires integrating various techniques and algorithms, which can result in greater complexity and a potential loss of information, KGE used in link prediction strategy are an end-to-end approach, allowing them to generate predictions directly from the learned embeddings. Factors such as feature space complexity, computational efficiency, overfitting prevention, and interoperability can explain why link classification methods perform better with simpler graphs.

By directly learning embeddings representing relationships between entities, link prediction models can quickly generate predictions without extensive post-processing or feature engineering. This streamlined process enhances the efficiency of predictive modeling in computational biology, allowing researchers to rapidly identify potential gene-disease associations and prioritize further investigation.

In summary, the advantages of employing a link prediction task over a link classification task as a representation strategy lie in its ability to leverage the rich semantic information encoded within KGs, uncover hidden associations between entities of interest and facilitate more accurate and comprehensive predictive modeling in the field of computational biology.

7.2 Future Work

This work cleared the way for further studies, developments and optimisations. Regarding the KGs, one possible path could be to include a protein-protein interaction network. The hypothesis of adding a protein-protein interaction network relies upon the notion that interactions between proteins provide a functional context for molecular perturbations, including some disease mutations [66]. Indeed, some studies note that the integration of mutations with protein interactions has been used to identify sets of genes important for diseases, as in [10, 86, 88].

Exploring further possibilities regarding the KGs, alternative approaches could involve testing other ontologies related to genes and diseases (e.g. Sequence Ontology), exploring different databases (e.g. ClinVar), examining sub-graphs within KGs, and employing clustering algorithms to identify groups of related genes and diseases (e.g. Markov Clustering). These methods could provide additional insights and perspectives, enriching our understanding of the complex relationships within biological systems and disease pathways.

Going forward, it could be worthwhile to explore other KGE methods for link prediction (e.g. TransSparse and KG2E) or enrich the embeddings of a KGE algorithm with the embeddings of other algorithms (specific to link prediction or not). Another option could be to use techniques that also directly perform link prediction, such as Graph Neural Networks, as in [35, 87, 89]. Graph Neural Networks are a type of ML model designed to work directly with graph data.

Graph Neural Network models process the KG in layers, where each layer aggregates information from the neighbourhoods of nodes. By aggregating neighbourhood information, the GNN updates the embeddings of each node in the graph. Then, a scoring function is applied to predict missing links, enabling the generation of a ranked list of candidate entities to complete a triple [89].

Integrating Graph Neural Networks into our link prediction approach would involve several steps.

First, the KGs are fed into a Graph Neural Network model, which then learns low-dimensional representations of the nodes. These representations are then combined with embeddings obtained from KGE algorithms to enrich the feature space. Alternatively, the representations generated by the KGE algorithms are fed to the model to enrich the feature space [89].

References

[1] The human phenotype ontology in 2021. *Nucleic Acids Research*, 49:D1207–D1217, 1 2021. ISSN 0305-1048. doi: 10.1093/nar/gkaa1043. 26

[2] Owl2vec*: embedding of owl ontologies. *Machine Learning*, 110:1813–1845, 7 2021. ISSN 0885-6125. doi: 10.1007/s10994-021-05997-6. 10

[3] Uniprot: the universal protein knowledgebase in 2021. *Nucleic Acids Research*, 49:D480–D489, 1 2021. ISSN 0305-1048. doi: 10.1093/nar/gkaa1100. 25, 26

[4] The human disease ontology 2022 update. *Nucleic Acids Research*, 50:D1255–D1261, 1 2022. ISSN 0305-1048. doi: 10.1093/nar/gkab1063. XIII, 9, 26

[5] Knowledge graphs. *ACM Computing Surveys*, 54:1–37, 5 2022. ISSN 0360-0300. doi: 10.1145/3447772. 5, 6, 8

[6] R. Abboud and İsmail İlkan Ceylan. Node classification meets link prediction on knowledge graphs, 2021. 14

[7] M. Ashburner, C. A. Ball, J. A. Blake, D. Botstein, H. Butler, J. M. Cherry, A. P. Davis, K. Dolinski, S. S. Dwight, J. T. Eppig, M. A. Harris, D. P. Hill, L. Issel-Tarver, A. Kasarskis, S. Lewis, J. C. Matese, J. E. Richardson, M. Ringwald, G. M. Rubin, and G. Sherlock. Gene ontology: tool for the unification of biology. *Nature Genetics*, 25:25–29, 5 2000. ISSN 1061-4036. doi: 10.1038/75556. 6, 26

[8] S. Auer, C. Bizer, G. Kobilarov, J. Lehmann, R. Cyganiak, and Z. Ives. *DBpedia: A Nucleus for a Web of Open Data*, pages 722–735. 2007. doi: 10.1007/978-3-540-76298-0_52. 8

[9] D. M. Bean, A. Al-Chalabi, R. J. B. Dobson, and A. Iacoangeli. A knowledge-based machine learning approach to gene prioritisation in amyotrophic lateral sclerosis. *Genes*, 11:668, 6 2020. ISSN 2073-4425. doi: 10.3390/genes11060668. 1, 15, 19, 20

[10] S. Biswas, P. Mitra, and K. S. Rao. Relation prediction of co-morbid diseases using knowledge graph completion. *IEEE/ACM Transactions on Computational Biology and Bioinformatics*, 18:708–717, 3 2021. ISSN 1545-5963. doi: 10.1109/TCBB.2019.2927310. 1, 15, 18, 19, 20, 22, 55

[11] K. Bollacker, C. Evans, P. Paritosh, T. Sturge, and J. Taylor. Freebase. pages 1247–1250. ACM, 6 2008. ISBN 9781605581026. doi: 10.1145/1376616.1376746. 8

[12] A. Bordes, N. Usunier, A. García-Durán, J. Weston, and O. Yakhnenko. Translating embeddings for modeling multi-relational data. In *Neural Information Processing Systems*, 2013. URL https://api.semanticscholar.org/CorpusID:14941970. 11

[13] L. Breiman. Random forests. *Machine Learning*, 45:5–32, 2001. ISSN 08856125. doi: 10.1023/A:1010933404324. 10

[14] A. Burkov. *The Hundred-page Machine Learning Book*. Andriy Burkov, 2019. ISBN 9781999579500. URL https://books.google.pt/books?id=ZF3KwQEACAAJ. 33, 35

[15] M. Cerrone, C. A. Remme, R. Tadros, C. R. Bezzina, and M. Delmar. Beyond the one gene–one disease paradigm. *Circulation*, 140(7):595–610, Aug 2019. doi: 10.1161/circulationaha.118.035954. 17

[16] I. Chami, S. Abu-El-Haija, B. Perozzi, C. Ré, and K. Murphy. Machine learning on graphs: A model and comprehensive taxonomy. 2022. 5

[17] P. Chandak, K. Huang, and M. Zitnik. Building a knowledge graph to enable precision medicine. *Scientific Data*, 10(1), Feb 2023. doi: 10.1038/s41597-023-01960-3. 17

[18] J. Chen, A. Althagafi, and R. Hoehndorf. Predicting candidate genes from phenotypes, functions and anatomical site of expression. *Bioinformatics*, 37:853–860, 5 2021. ISSN 1367-4803. doi: 10.1093/bioinformatics/btaa879. 21, 22

[19] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. 3 2016. doi: 10.1145/2939672.2939785. 9

[20] N. Choi, I.-Y. Song, and H. Han. A survey on ontology mapping. *ACM SIGMOD Record*, 35:34–41, 9 2006. ISSN 0163-5808. doi: 10.1145/1168092.1168097. 7, 26

[21] Y. Dai, S. Wang, N. N. Xiong, and W. Guo. A survey on knowledge graph embedding: Approaches, applications and benchmarks. *Electronics*, 9:750, 5 2020. ISSN 2079-9292. doi: 10.3390/electronics9050750. 10, 45

[22] Y. Dong, J. Tang, S. Wu, J. Tian, N. V. Chawla, J. Rao, and H. Cao. Link prediction and recommendation across heterogeneous social networks. In *2012 IEEE 12th International Conference on Data Mining*, pages 181–190, 2012. doi: 10.1109/ICDM.2012.140. 15

[23] J. Du, D. Lin, R. Yuan, X. Chen, X. Liu, and J. Yan. Graph embedding based novel gene discovery associated with diabetes mellitus. *Frontiers in Genetics*, 12, 11 2021. ISSN 1664-8021. doi: 10.3389/fgene.2021.779186. 13, 18, 21, 22

[24] L. Ehrlinger and W. Wöß. Towards a definition of knowledge graphs. 2016. 2, 5, 8

[25] J. Euzenat and P. Shvaiko. *Ontology Matching*. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-38720-3. doi: 10.1007/978-3-642-38721-0. 1, 6

[26] A. Fabregat, S. Jupe, L. Matthews, K. Sidiropoulos, M. Gillespie, P. Garapati, R. Haw, B. Jassal, F. Korninger, B. May, M. Milacic, C. D. Roca, K. Rothfels, C. Sevilla, V. Shamovsky, S. Shorser, T. Varusai, G. Viteri, J. Weiser, G. Wu, L. Stein, H. Hermjakob, and P. D'Eustachio. The reactome pathway knowledgebase. *Nucleic Acids Research*, 46:D649–D655, 1 2018. ISSN 0305-1048. doi: 10.1093/nar/gkx1132. 26

[27] D. Faria, C. Pesquita, E. Santos, I. F. Cruz, and F. M. Couto. Agreementmakerlight results for oaei 2013. 2013. 27

[28] H. V. Firth, S. M. Richards, A. P. Bevan, S. Clayton, M. Corpas, D. Rajan, S. V. Vooren, Y. Moreau, R. M. Pettett, and N. P. Carter. Decipher: Database of chromosomal imbalance and phenotype in humans using ensembl resources. *The American Journal of Human Genetics*, 84:524–533, 4 2009. ISSN 00029297. doi: 10.1016/j.ajhg.2009.03.010. 26

[29] A. Grover and J. Leskovec. node2vec: Scalable feature learning for networks. *CoRR*, abs/1607.00653, 2016. URL http://arxiv.org/abs/1607.00653. 10, 33

[30] W. L. Hamilton. Graph representation learning. *Synthesis Lectures on Artificial Intelligence and Machine Learning*, 14(3):1–159. 2, 13, 18

[31] W. L. Hamilton, R. Ying, and J. Leskovec. Representation learning on graphs: Methods and applications. *IEEE Data Eng. Bull.*, 40:52–74, 2017. URL https://api.semanticscholar.org/CorpusID:3215337. 2, 13, 18

[32] A. Hamosh. Online mendelian inheritance in man (omim), a knowledgebase of human genes and genetic disorders. *Nucleic Acids Research*, 33:D514–D517, 12 2004. ISSN 1362-4962. doi: 10.1093/nar/gki033. 25, 26

[33] J. Han, M. Kamber, and J. Pei. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 3rd edition, 2011. ISBN 0123814790. 8

[34] B. He, K. Wang, J. Xiang, P. Bing, M. Tang, G. Tian, C. Guo, M. Xu, and J. Yang. Dghne: network enhancement-based method in identifying disease-causing genes through a heterogeneous biomedical network. *Briefings in Bioinformatics*, 23, 11 2022. ISSN 1467-5463. doi: 10.1093/bib/bbac405. 1, 18, 19

[35] M. He, C. Huang, B. Liu, Y. Wang, and J. Li. Factor graph-aggregated heterogeneous network embedding for disease-gene association prediction. *BMC Bioinformatics*, 22:165, 12 2021. ISSN 1471-2105. doi: 10.1186/s12859-021-04099-3. 14, 21, 22, 55

[36] S. He, K. Liu, G. Ji, and J. Zhao. Learning to represent knowledge graphs with gaussian embedding. pages 623–632. ACM, 10 2015. ISBN 9781450337946. doi: 10.1145/2806416.2806502. 11

[37] R. Hoehndorf, P. N. Schofield, and G. V. Gkoutos. The role of ontologies in biological and biomedical research: a functional perspective. *Briefings in Bioinformatics*, 16:1069–1080, 11 2015. ISSN 1467-5463. doi: 10.1093/bib/bbv011. 6

[38] R. J. *Introduction to Graph Theory*. Dover Publications, 2nd edition, 1976. ISBN 0486678709, 9780486678702. 5

[39] G. Ji, S. He, L. Xu, K. Liu, and J. Zhao. Knowledge graph embedding via dynamic mapping matrix. pages 687–696. Association for Computational Linguistics, 2015. doi: 10.3115/v1/P15-1067. 11, 12

[40] G. Ji, K. Liu, S. He, and J. Zhao. Knowledge graph completion with adaptive sparse transfer matrix. In *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, AAAI'16, page 985–991. AAAI Press, 2016. 11

[41] M. Kanehisa, M. Furumichi, M. Tanabe, Y. Sato, and K. Morishima. Kegg: new perspectives on genomes, pathways, diseases and drugs. *Nucleic Acids Research*, 45:D353–D361, 1 2017. ISSN 0305-1048. doi: 10.1093/nar/gkw1092. 26

[42] K. Leung. Micro, macro and weighted averages of f1 score, clearly explained, 2022. URL https://towardsdatascience.com/micro-macro-weighted-averages-of-f1-score-clearly-explained-b603420b292f. 35

[43] Y. Lin, Z. Liu, M. Sun, Y. Liu, and X. Zhu. Learning entity and relation embeddings for knowledge graph completion. In *AAAI Conference on Artificial Intelligence*, 2015. URL https://api.semanticscholar.org/CorpusID:2949428. 11, 12

[44] P. Luo, Y. Li, L.-P. Tian, and F.-X. Wu. Enhancing the prediction of disease–gene associations with multimodal deep learning. *Bioinformatics*, 35:3735–3742, 10 2019. ISSN 1367-4803. doi: 10.1093/bioinformatics/btz155. 14, 21, 22

[45] P. Luo, Q. Xiao, P.-J. Wei, B. Liao, and F.-X. Wu. Identifying disease-gene associations with graph-regularized manifold learning. *Frontiers in Genetics*, 10, 4 2019. ISSN 1664-8021. doi: 10.3389/fgene.2019.00270. 14, 21, 22

[46] I. Makarov, D. Kiselev, N. Nikitinsky, and L. Subelj. Survey on graph embeddings and their applications to machine learning problems on graphs. *PeerJ Computer Science*, 7:e357, 2 2021. ISSN 2376-5992. doi: 10.7717/peerj-cs.357. 10

[47] T. Mikolov, K. Chen, G. Corrado, and J. Dean. Efficient estimation of word representations in vector space. 2013. 10

[48] A. Mohan, R. Venkatesan, and K. Pramod. A scalable method for link prediction in large real world networks. *Journal of Parallel and Distributed Computing*, 109:89–101, 2017. ISSN 0743-7315. doi: https://doi.org/10.1016/j.jpdc.2017.05.009. URL https://www.sciencedirect.com/science/article/pii/S0743731517301600. 15

[49] National Academies of Sciences, Engineering, and Medicine; Division on Earth and Life Studies; Thévenon A, Liao J, Bremer A, et al., editor. *Pivotal Interfaces of Environmental Health and Infectious Disease Research to Inform Responses to Outbreaks, Epidemics, and Pandemics: Proceedings of a Workshop—in Brief*. National Academies Press (US), Washington, DC, Sept. 2021. doi: 10.17226/26270. Available from: https://www.ncbi.nlm.nih.gov/books/NBK573776/. 17

[50] M. Nayyeri, G. M. Cil, S. Vahdati, F. Osborne, A. Kravchenko, S. Angioni, A. Salatino, D. R. Recupero, E. Motta, and J. Lehmann. Link prediction of weighted triples for knowledge graph completion within the scholarly domain. *IEEE Access*, 9:116002–116014, 2021. ISSN 2169-3536. doi: 10.1109/ACCESS.2021.3105183. 1, 8, 10

[51] M. Nickel, V. Tresp, and H.-P. Kriegel. A three-way model for collective learning on multi-relational data. ICML'11, page 809–816, Madison, WI, USA, 2011. Omnipress. ISBN 9781450306195. 12

[52] M. Nickel, L. Rosasco, and T. Poggio. Holographic embeddings of knowledge graphs. 2015. 12

[53] S. Nunes, R. T. Sousa, and C. Pesquita. Multi-domain knowledge graph embeddings for gene-disease association prediction. *Journal of Biomedical Semantics*, 14:11, 8 2023. ISSN 2041-1480. doi: 10.1186/s13326-023-00291-x. 13, 21, 22, 25, 27, 32, 33

[54] F. Olken and D. Rotem. Random sampling from databases: a survey. *Statistics and Computing*, 5: 25–42, 3 1995. ISSN 0960-3174. doi: 10.1007/BF00140664. 26

[55] K. Opap and N. Mulder. Recent advances in predicting gene–disease associations. *F1000Research*, 6:578, Apr 2017. doi: 10.12688/f1000research.10788.1. 18

[56] S. Pavan, K. Rommel, M. E. M. Marquina, S. Höhn, V. Lanneau, and A. Rath. Clinical practice guidelines for rare diseases: The orphanet database. *PLOS ONE*, 12:e0170365, 1 2017. ISSN 1932-6203. doi: 10.1371/journal.pone.0170365. 25, 26

[57] B. Perozzi, R. Al-Rfou, and S. Skiena. Deepwalk. *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 701–710, 8 2014. doi: 10.1145/2623330.2623732. 10

[58] J. Piñero, Àlex Bravo, N. Queralt-Rosinach, A. Gutiérrez-Sacristán, J. Deu-Pons, E. Centeno, J. García-García, F. Sanz, and L. I. Furlong. Disgenet: a comprehensive platform integrating information on human disease-associated genes and variants. *Nucleic Acids Research*, 45:D833–D839, 1 2017. ISSN 0305-1048. doi: 10.1093/nar/gkw943. 3, 12, 25

[59] M.-C. Popescu, V. Balas, L. Perescu-Popescu, and N. Mastorakis. Multilayer perceptron and neural networks. *WSEAS Transactions on Circuits and Systems*, 8, 07 2009. 9

[60] E. U. Rehman, A. Saeed, N. Minallah, and A. Hafeez. Knowledge graph embedding for link prediction models. *Preprints*, February 2022. doi: 10.20944/preprints202202.0212.v1. URL https://doi.org/10.20944/preprints202202.0212.v1. 15

[61] P. Ristoski and H. Paulheim. *RDF2Vec: RDF Graph Embeddings for Data Mining*, pages 498–514. 2016. doi: 10.1007/978-3-319-46523-4_30. 10, 11, 32

[62] A. Rossi, D. Barbosa, D. Firmani, A. Matinata, and P. Merialdo. Knowledge graph embedding for link prediction: A comparative analysis. *ACM Transactions on Knowledge Discovery from Data*, 15(2):1–49, Jan. 2021. ISSN 1556-472X. doi: 10.1145/3424672. URL http://dx.doi.org/10.1145/3424672. 14, 46

[63] P. Sen, G. Namata, M. Bilgic, L. Getoor, B. Gallagher, and T. Eliassi□Rad. Collective classification in network data. *AI Mag.*, 29(3):93–106, sep 2008. ISSN 0738-4602. doi: 10.1609/aimag.v29i3.2157. URL https://doi.org/10.1609/aimag.v29i3.2157. 14

[64] F. Z. Smaili, X. Gao, and R. Hoehndorf. Onto2vec: joint vector-based representation of biological entities and their ontology-based annotations. *Bioinformatics*, 34:i52–i60, 7 2018. ISSN 1367-4803. doi: 10.1093/bioinformatics/bty259. 10

[65] F. Z. Smaili, X. Gao, and R. Hoehndorf. Opa2vec: combining formal and informal content of biomedical ontologies to improve similarity-based prediction. *Bioinformatics*, 35:2133–2140, 6 2019. ISSN 1367-4803. doi: 10.1093/bioinformatics/bty933. 10, 11, 33

[66] Z. Stanfield, M. Coşkun, and M. Koyutürk. Drug response prediction as a link prediction problem. *Scientific Reports*, 7:40321, 1 2017. ISSN 2045-2322. doi: 10.1038/srep40321. 55

[67] B. Steenwinckel, G. Vandewiele, T. Agozzino, and F. Ongenae. pyrdf2vec: A python implementation and extension of rdf2vec. In *European Semantic Web Conference*, pages 471–483. Springer Nature Switzerland, 2023. doi: 10.1007/978-3-031-33455-9_28. 33

[68] R. Stevens, C. Wroe, P. Lord, and C. Goble. *Ontologies in Bioinformatics*, pages 635–657. Springer Berlin Heidelberg, 2004. doi: 10.1007/978-3-540-24750-0_32. 6

[69] A. Subramanian, P. Tamayo, V. K. Mootha, S. Mukherjee, B. L. Ebert, M. A. Gillette, A. Paulovich, S. L. Pomeroy, T. R. Golub, E. S. Lander, and J. P. Mesirov. Gene set enrichment analysis: A knowledge-based approach for interpreting genome-wide expression profiles. *Proceedings of the National Academy of Sciences*, 102:15545–15550, 10 2005. ISSN 0027-8424. doi: 10.1073/pnas.0506580102. 14

[70] F. M. Suchanek, G. Kasneci, and G. Weikum. Yago. pages 697–706. ACM, 5 2007. ISBN 9781595936547. doi: 10.1145/1242572.1242667. 8

[71] J. Tang, M. Qu, M. Wang, M. Zhang, J. Yan, and Q. Mei. Line: Large-scale information network embedding. In *Proceedings of the 24th International Conference on World Wide Web*, WWW '15, page 1067–1077, Republic and Canton of Geneva, CHE, 2015. International World Wide Web Conferences Steering Committee. ISBN 9781450334693. doi: 10.1145/2736277.2741093. URL https://doi.org/10.1145/2736277.2741093. 14

[72] T. Trouillon, J. Welbl, S. Riedel, Éric Gaussier, and G. Bouchard. Complex embeddings for simple link prediction. 2016. 12

[73] J. Vilela, M. Asif, A. R. Marques, J. X. Santos, C. Rasga, A. Vicente, and H. Martiniano. Biomedical knowledge graph embeddings for personalized medicine: Predicting disease□gene associations. *Expert Systems*, 40, 6 2023. ISSN 0266-4720. doi: 10.1111/exsy.13181. 6, 15, 19, 20

[74] P. M. Visscher, L. Yengo, N. J. Cox, and N. R. Wray. Discovery and implications of polygenicity of common diseases. *Science*, 373(6562):1468–1473, Sep 2021. doi: 10.1126/science.abi8206. 17

[75] D. Vrandečić and M. Krötzsch. Wikidata. *Communications of the ACM*, 57:78–85, 9 2014. ISSN 0001-0782. doi: 10.1145/2629489. 8

[76] M. Wang, L. Qiu, and X. Wang. A survey on knowledge graph embeddings for link prediction. *Symmetry*, 13:485, 3 2021. ISSN 2073-8994. doi: 10.3390/sym13030485. 1, 2, 10, 45

[77] Q. Wang, Z. Mao, B. Wang, and L. Guo. Knowledge graph embedding: A survey of approaches and applications. *IEEE Transactions on Knowledge and Data Engineering*, 29(12):2724–2743, 2017. doi: 10.1109/TKDE.2017.2754499. 8, 10

[78] X. Wang, Y. Gong, J. Yi, and W. Zhang. Predicting gene-disease associations from the heterogeneous network using graph embedding. pages 504–511. IEEE, 11 2019. ISBN 978-1-7281-1867-3. doi: 10.1109/BIBM47256.2019.8983134. 1, 13, 14, 21, 22

[79] Z. Wang, J. Zhang, J. Feng, and Z. Chen. Knowledge graph embedding by translating on hyperplanes. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence*, AAAI'14, page 1112–1119. AAAI Press, 2014. 11, 12

[80] J. Xiang, N.-R. Zhang, J.-S. Zhang, X.-Y. Lv, and M. Li. Prgefne: Predicting disease-related genes by fast network embedding. *Methods*, 192:3–12, 8 2021. ISSN 10462023. doi: 10.1016/j.ymeth.2020.06.015. 1, 15, 19

[81] S. Xiao, S. Wang, Y. Dai, and W. Guo. Graph neural networks in node classification: survey and evaluation. *Machine Vision and Applications*, 33:4, 1 2022. ISSN 0932-8092. doi: 10.1007/s00138-021-01251-0. 13, 14

[82] B. Xu, Y. Liu, S. Yu, L. Wang, J. Dong, H. Lin, Z. Yang, J. Wang, and F. Xia. A network embedding model for pathogenic genes prediction by multi-path random walking on heterogeneous network. *BMC Medical Genomics*, 12:188, 12 2019. ISSN 1755-8794. doi: 10.1186/s12920-019-0627-z. 1, 15, 19

[83] J. Xu and Y. Li. Discovering disease-genes by topological features in human protein–protein interaction network. *Bioinformatics*, 22:2800–2805, 11 2006. ISSN 1367-4811. doi: 10.1093/bioinformatics/btl467. 14

[84] B. Yang, W. tau Yih, X. He, J. Gao, and L. Deng. Embedding entities and relations for learning and inference in knowledge bases. 2015. 12

[85] F.-J. Yang. An implementation of naive bayes classifier. *2018 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 301–306, 12 2018. doi: 10.1109/CSCI46756.2018.00065. 9

[86] K. Yang, R. Wang, G. Liu, Z. Shu, N. Wang, R. Zhang, J. Yu, J. Chen, X. Li, and X. Zhou. Hergepred: Heterogeneous network embedding representation for disease gene prediction. *IEEE Journal of Biomedical and Health Informatics*, 23:1805–1815, 7 2019. ISSN 2168-2194. doi: 10.1109/JBHI.2018.2870728. 1, 19, 20, 55

[87] J. Ye, S. Wang, X. Yang, and X. Tang. Gene prediction of aging-related diseases based on dnn and mashup. *BMC Bioinformatics*, 22:597, 12 2021. ISSN 1471-2105. doi: 10.1186/s12859-021-04518-5. 14, 21, 22, 55

[88] Y. Zhang, J. Xiang, L. Tang, J. Li, Q. Lu, G. Tian, B.-S. He, and J. Yang. Identifying breast cancer-related genes based on a novel computational framework involving kegg pathways and ppi network modularity. *Frontiers in Genetics*, 12, 8 2021. ISSN 1664-8021. doi: 10.3389/fgene.2021.596794. 15, 19, 20, 55

[89] L. Zhu, Z. Hong, and H. Zheng. Predicting gene-disease associations via graph embedding and graph convolutional networks. pages 382–389. IEEE, 11 2019. ISBN 978-1-7281-1867-3. doi: 10.1109/BIBM47256.2019.8983350. 1, 15, 19, 20, 55, 56

[90] F. Zola, L. Segurola-Gil, J. Bruse, M. Galar, and R. Orduna-Urrutia. Network traffic analysis through node behaviour classification: a graph-based approach with temporal dissection and data-level preprocessing. *Computers and Security*, 115:102632, 2022. ISSN 0167-4048. doi: https://doi.org/10.1016/j.cose.2022.102632. URL https://www.sciencedirect.com/science/article/pii/S0167404822000311. 14

Appendix A

Experimental Setup

We used an AMD Ryzen 9 6900HS machine with eight cores and 32GB RAM to employ the concepts and methodologies described in this work. However, we could carry out the task on a machine with more or less capabilities. We conducted the experiments on a server machine with a 12-core processor, 128GB RAM and two NVIDIA Geforce RTX 2060 Super graphic cards, each boasting 8GB of VRAM.

The existence of this second machine was essential in this work since the operating system it has is more versatile for installing libraries and packages and because it has a graphics card by the requirements of the library that we use to implement the KGE models for link prediction. This second machine also helped us run the experiments at a regular time for the development of the work.

We accessed the second machine through an SSH and telnet client - *Putty*[1] (version 0.80), and controlled all folders and files through an SFTP and FTP client for Microsoft Windows - *WinSCP*[2] (version 6.1.2). Inside our home, we installed *Miniconda*[3] (Conda version 23.11.0) to create two environments: one with the *Python* version 3.6 (for node classification) and another with a more recent *Python* version - 3.10 (for link prediction).

In the environments, we installed: *grpcio*[4] (version 1.48.2); *HDF5*[5] (version 3.1.0); *NumPy*[6] (version 1.19.5); *Pandas*[7] (version 1.1.5); *RDFLib* (version 5.0.0); *Scikit-learn*[8] (version 0.24.2); *Scipy*[9] (version 1.5.4); and *TensorFlow*[10] (version 1.13.1). Installing all these libraries and packages guarantees the normal functioning of the experiments.

[1] https://www.chiark.greenend.org.uk/~sgtatham/putty/latest.html/
[2] https://winscp.net/eng/index.php/
[3] https://docs.conda.io/projects/miniconda/en/latest/
[4] https://pypi.org/project/grpcio/
[5] https://docs.h5py.org/en/stable/index.html/
[6] https://numpy.org/
[7] https://pandas.pydata.org/
[8] https://scikit-learn.org/stable/index.html/
[9] https://pypi.org/project/scipy/
[10] https://www.tensorflow.org/?hl=pt/

We also used Linux's `screen` command, which is a terminal multiplexer, to create a new window with a shell in it, run a command (e.g. a process), and then push the window to the background (called "detaching"). This tool is excellent for long processes we do not want to accidentally terminate by closing the terminal window.

Appendix B

Default Parameters of KGE Models in Link Classification Approach

Algorithm	Parameters
RDF2Vec	Word2Vec default parameters: sentences=None, corpus file=None, alpha=0.025, window=5, min count=5, max vocab size=None, sample=0.001, seed=1, workers=3, min alpha=0.0001, sg=0, hs=0, negative=5, ns exponent=0.75, hashfxn=0, epochs=5, null word=0, trim rule=None, sorted vocab=1, batch words=10000, compute loss=False, callbacks=0, comment=None, max final vocab=None, shrink windows=True.
OPA2Vec	Annotations [metadata annotations]: All annotation properties. Pretrained [pre-trained model]: Default pre-trained model from http://bio2vec.net/data/pubmed_model/ Reasoner [reasoner]: Elk Debug [debug]: Set to no, in which case no intermediate files are kept once the program exits.

Table B.1: Default parameters for the KGE models in link classification experiments.

Appendix C

Results of IQR for KGE Models in Link Classification Approach

Embedding Model	Operator	Knowledge Graph							
		GO + HPO	GO + HPO + LD	GO + HPO + Map	GO + HPO + LD + Map	GO + HPO + DO	GO + HPO + LD + DO	GO + HPO + Map + DO	GO + HPO + LD + Map + DO
RDF2Vec	Concatenation	0.0160	**0.0081**	0.0117	0.0118	0.0151	0.0176	0.0151	0.0131
	Average	0.0163	0.0135	0.0147	**0.0101**	0.0124	0.0112	0.0118	0.0108
	Hadamard	0.0144	0.0244	0.0180	0.0115	0.0121	**0.0055**	0.0129	0.0089
	Weighted-L1	0.0155	0.0323	0.0089	0.0197	0.0190	0.0202	0.0154	0.0134
	Weighted-L2	0.0145	0.0324	**0.0086**	0.0180	0.0187	0.0211	0.0139	0.0129
OPA2Vec	Concatenation	0.0096	0.0141	0.0122	0.0143	0.0168	0.0139	0.0118	0.0119
	Average	0.0113	0.0131	0.0089	0.0149	**0.0093**	0.0091	0.0104	0.0075
	Hadamard	0.0097	0.0099	0.0186	0.0160	0.0700	0.0144	**0.0097**	**0.0036**
	Weighted-L1	0.0144	0.0154	0.0108	0.0116	0.0162	0.0188	0.0134	0.0076
	Weighted-L2	0.0149	0.0120	0.0099	0.0115	0.0167	0.0188	0.0134	0.0075

Table C.1: IQR of WAF scores for the competing combination of the KGEs and vector operators for the different KGs using XGB. The best possible result in each KG is highlighted in bold and the best possible result in each KGE is underlined.

Appendix D

Default Parameters of KGE Models in Link Prediction Approach

Algorithm	Parameters
ComplEx	work threads=8, nr batches=100, alpha=0.5, lambda=0.05, bern=1, entity negative rate=1, relation negative rate=0, optimization method=Adagrad
DistMult	work threads=8, nr batches=100, alpha=0.5, lambda=0.05, bern=1, entity negative rate=1, relation negative rate=0, optimization method=Adagrad
HolE	work threads=8, nr batches=100, alpha=0.1, bern=0, margin=0.2, entity negative rate=1, relation negative rate=0, optimization method=Adagrad
TransD	work threads=8, nr batches=100, alpha=1.0, bern=1, margin=4.0, entity negative rate=1, relation negative rate=0, optimization method=SGD
TransE	work threads=8, nr batches=100, alpha=0.001, bern=0, margin=1.0, entity negative rate=1, relation negative rate=0, optimization method=SGD
TransH	work threads=8, nr batches=100, alpha=0.001, bern=0, margin=1.0, entity negative rate=1, relation negative rate=0, optimization method=SGD
TransR	work threads=8, nr batches=100, alpha=1.0, lambda=4.0, margin=1.0, entity negative rate=1, relation negative rate=0, optimization method=SGD

Table D.1: Default parameters for the KGE models in link prediction experiments.

Appendix E

Results of MR and MRR in Link Prediction Approach

MR is a measure that represents the average position of the candidate entities in a list of truly associated entities for a gene/disease:

$$MR = \frac{\sum_{i=1}^{n} rank_i}{n} \tag{E.1}$$

where n is the number of truly candidate entities for a gene/disease and $rank_i$ indicates the rank position of the candidate entity for the i-th gene/disease.

MRR is a probability metric used to evaluate the performance on an algorithm based on a sample of candidate entities arranged by the probability of accuracy:

$$MRR = \frac{1}{n} \sum_{i=1}^{n} \frac{1}{rank_i} \tag{E.2}$$

where n and $rank_i$ indicate the same as in the MR metric.

Unranked entities are entities outside the top 100, which we represent with a value of 1000. These entities are truly associated with the gene/disease (confirmed gene-disease pair). However, they are not among the first 100 entities. Assigning 1000 to these items helps ensure that the calculated metrics accurately reflect the model's performance and are not unduly influenced by unranked items. It promotes more robust and interpretable evaluations of the model's predictive capabilities.

To calculate the MR and MRR of each model for each experiment, we gathered all the ranks of all unique genes and diseases into a single list and applied the metrics described.

Knowledge Graph	Embedding Model						
	TransE	**TransD**	**TransH**	**TransR**	**DistMult**	**HolE**	**ComplEx**
GO + HPO	788.449	596.985	**784.762**	820.149	630.264	728.754	615.767
GO + HPO + LD	787.943	595.987	789.856	799.318	617.606	726.967	627.32
GO + HPO + Map	783.817	591.703	785.895	827.023	645.656	730.639	**600.069**
GO + HPO + LD + Map	**779.633**	589.33	787.179	796.605	647.298	715.695	618.592
GO + HPO + DO	798.969	589.271	799.49	803.596	693.802	740.923	609.994
GO + HPO + LD + DO	797.86	589.665	798.192	808.669	681.905	793.493	605.589
GO + HPO + Map + DO	792.704	586.302	804.182	793.997	681.559	735.683	613.232
GO + HPO + LD + Map + DO	807.522	592.828	785.289	803.911	645.332	**681.259**	606.187
GO + HPO* + LD + Map	806.041	712.331	814.157	814.059	802.85	825.243	804.838
GO + DO	798.523	**570.457**	798.115	**786.964**	**613.51**	705.706	631.642
GO + HPO* + LD + Map + DO	813.601	704.808	811.459	821.711	828.411	776.102	781.151

Table E.1: MR evaluation of link prediction embedding methods throughout the experiments. The best possible result in each KGE is highlighted in bold.

Knowledge Graph	Embedding Model						
	TransE	**TransD**	**TransH**	**TransR**	**DistMult**	**HolE**	**ComplEx**
GO + HPO	0.001	0.001	0.001	0.001	**0.013**	0.001	0.001
GO + HPO + LD	0.001	**0.01**	0.001	0.001	**0.167**	**0.05**	**0.143**
GO + HPO + Map	0.001	**1.0**	0.001	0.001	0.001	**0.019**	0.001
GO + HPO + LD + Map	0.001	0.001	0.001	0.001	0.001	0.001	0.001
GO + HPO + DO	0.001	0.001	0.001	0.001	0.001	0.001	0.001
GO + HPO + LD + DO	0.001	0.001	0.001	0.001	0.001	0.001	**0.04**
GO + HPO + Map + DO	0.001	**1.0**	0.001	0.001	**0.01**	**0.5**	**0.1**
GO + HPO + LD + Map + DO	0.001	0.001	0.001	0.001	0.001	0.001	0.001
GO + HPO* + LD + Map	0.001	0.001	0.001	0.001	0.001	0.001	0.001
GO + DO	0.001	**0.125**	0.001	0.001	**0.038**	0.001	0.001
GO + HPO* + LD + Map + DO	0.001	0.001	0.001	0.001	0.001	0.001	0.001

Table E.2: MRR assessment of KGE methods across the experiments. The best values for all KGE are highlighted in bold.

Appendix F

Results of Hits@10, 30 and 100 in Link Prediction Approach

To calculate the Hits@10, 30 and 100 of each model for each experiment, we gathered all the ranks of all unique genes and diseases into a single list and applied the equation described in Section 6.3.

Knowledge Graph	Embedding Model						
	TransE	**TransD**	**TransH**	**TransR**	**DistMult**	**HolE**	**ComplEx**
GO + HPO	0.08	0.124	0.076	0.047	0.106	0.08	0.104
GO + HPO + LD	0.074	0.137	0.081	**0.049**	**0.12**	0.075	0.1
GO + HPO + Map	0.08	0.139	0.083	0.022	0.089	0.075	0.109
GO + HPO + LD + Map	0.082	0.139	0.082	0.045	0.093	0.087	0.102
GO + HPO + DO	0.075	0.142	0.083	0.04	0.069	0.068	0.115
GO + HPO + LD + DO	0.078	0.138	0.082	0.035	0.079	0.048	**0.118**
GO + HPO + Map + DO	**0.084**	0.145	0.076	0.046	0.083	0.07	0.11
GO + HPO + LD + Map + DO	0.077	0.13	**0.09**	0.037	0.086	**0.093**	0.103
GO + HPO* + LD + Map	0.07	0.098	0.076	0.038	0.034	0.04	0.031
GO + DO	0.077	**0.161**	0.081	0.044	0.105	0.083	0.096
GO + HPO* + LD + Map + DO	0.073	0.093	0.07	0.03	0.023	0.064	0.042

Table F.1: Hits@10 performance of link prediction embedding models across the experiments. The best possible result in each KGE is highlighted in bold.

Knowledge Graph	Embedding Model						
	TransE	**TransD**	**TransH**	**TransR**	**DistMult**	**HOLE**	**ComplEx**
GO + HPO	0.13	0.247	0.136	0.09	0.214	0.15	0.214
GO + HPO + LD	0.127	0.244	0.136	0.102	**0.222**	0.148	0.207
GO + HPO + Map	0.136	0.254	0.134	0.072	0.191	0.149	**0.222**
GO + HPO + LD + Map	**0.142**	0.257	0.139	0.095	0.193	0.159	0.212
GO + HPO + DO	0.126	0.263	0.133	0.093	0.154	0.139	0.23
GO + HPO + LD + DO	0.132	0.26	0.132	0.089	0.171	0.103	0.23
GO + HPO + Map + DO	0.133	0.258	0.126	**0.106**	0.176	0.142	0.22
GO + HPO + LD + Map + DO	0.128	0.253	**0.14**	0.088	0.186	**0.186**	0.216
GO + HPO* + LD + Map	0.122	0.179	0.123	0.09	0.083	0.091	0.085
GO + DO	0.132	**0.279**	0.128	0.1	0.219	0.157	0.203
GO + HPO* + LD + Map + DO	0.12	0.18	0.12	0.078	0.069	0.123	0.104

Table F.2: Hits@30 performance of link prediction embedding models across the experiments. The best possible result in each KGE is highlighted in bold.

Knowledge Graph	Embedding Model						
	TransE	**TransD**	**TransH**	**TransR**	**DistMult**	**HOLE**	**ComplEx**
GO + HPO	0.218	0.416	**0.222**	0.187	0.383	0.281	0.398
GO + HPO + LD	0.219	0.417	0.217	0.209	0.395	0.283	0.386
GO + HPO + Map	0.223	0.421	0.221	0.181	0.368	0.279	**0.414**
GO + HPO + LD + Map	**0.227**	0.423	0.219	0.212	0.366	0.294	0.395
GO + HPO + DO	0.207	0.423	0.206	0.205	0.318	0.269	0.403
GO + HPO + LD + DO	0.208	0.423	0.208	0.199	0.33	0.215	0.408
GO + HPO + Map + DO	0.213	0.427	0.202	0.214	0.33	0.274	0.4
GO + HPO + LD + Map + DO	0.198	0.42	0.221	0.205	0.368	**0.329**	0.408
GO + HPO* + LD + Map	0.2	0.297	0.191	0.193	0.206	0.182	0.204
GO + DO	0.208	**0.442**	0.208	**0.222**	**0.4**	0.305	0.203
GO + HPO* + LD + Map + DO	0.192	0.305	0.194	0.186	0.18	0.232	0.228

Table F.3: Hits@100 performance of link prediction embedding models across the experiments. The best possible result in each KGE is highlighted in bold.