Biomedical Knowledge Graph Embeddings with Negative Statements

Rita T. Sousa[1][0000-0002-7241-8970], Sara Silva[1][0000-0001-8223-4799], Heiko Paulheim[2][0000-0003-4386-8195], and Catia Pesquita[1][0000-0002-1847-9393]

[1] LASIGE, Faculdade de Ciências da Universidade de Lisboa, Portugal
{risousa,sgsilva,clpesquita}@ciencias.ulisboa.pt
[2] Data and Web Science Group, Universität Mannheim, Germany
heiko.paulheim@uni-mannheim.de

Abstract. A knowledge graph is a powerful representation of real-world entities and their relations. The vast majority of these relations are defined as positive statements, but the importance of negative statements is increasingly recognized, especially under an Open World Assumption. Explicitly considering negative statements has been shown to improve performance on tasks such as entity summarization and question answering or domain-specific tasks such as protein function prediction. However, no attention has been given to the exploration of negative statements by knowledge graph embedding approaches despite the potential of negative statements to produce more accurate representations of entities in a knowledge graph.

We propose a novel approach, TrueWalks, to incorporate negative statements into the knowledge graph representation learning process. In particular, we present a novel walk-generation method that is able to not only differentiate between positive and negative statements but also take into account the semantic implications of negation in ontology-rich knowledge graphs. This is of particular importance for applications in the biomedical domain, where the inadequacy of embedding approaches regarding negative statements at the ontology level has been identified as a crucial limitation.

We evaluate TrueWalks in ontology-rich biomedical knowledge graphs in two different predictive tasks based on KG embeddings: protein-protein interaction prediction and gene-disease association prediction. We conduct an extensive analysis over established benchmarks and demonstrate that our method is able to improve the performance of knowledge graph embeddings on all tasks.

Keywords: Knowledge Graph · Knowledge Graph Embedding · Negative Statements · Biomedical Applications.

1 Introduction

Knowledge Graphs (KGs) represent facts about real-world entities and their relations and have been extensively used to support a range of applications from question-answering and recommendation systems to machine learning and analytics [17]. KGs have taken to the forefront of biomedical data through their ability to describe and interlink information about biomedical entities such as genes, proteins, diseases and patients, structured

according to biomedical ontologies. This supports the analysis and interpretation of biological data, for instance, through the use of semantic similarity measures [32]. More recently, a spate of KG embedding methods [42] have emerged in this space and have been successfully employed in a number of biomedical applications [28]. The impact of KG embeddings in biomedical analytics is expected to increase in tandem with the growing volume and complexity of biomedical data. However, this success relies on the expectation that KG embeddings are semantically meaningful representations of the underlying biomedical entities.

Regardless of their domain, the vast majority of KG facts are represented as positive statements, e.g. $(\textit{hemoglobin}, \textit{hasFunction}, \textit{oxygen transport})$. Under a Closed World Assumption, negative statements are not required, since any missing fact can be assumed as a negative. However, real-world KGs reside under the Open World Assumption where non-stated negative facts are formally indistinguishable from missing or unknown facts, which can have important implications across a variety of tasks.

The importance of negative statements is increasingly recognized [2,10]. For example, in the biomedical domain, the knowledge that a patient does not exhibit a given symptom or a protein does not perform a specific function is crucial for both clinical decision-making and biomedical insight. While ontologies are able to express negation and the enrichment of KGs with interesting negative statements is gaining traction, existing KG embedding methods are not able to adequately utilize them [21], which ultimately results in less accurate representations of entities.

We propose True Walks, to the best of our knowledge, the first-ever approach that is able to incorporate negative statements into the KG embedding learning process. This is fundamentally different from other KG embedding methods, which produce negative statements by negative random sampling strategies to train representations that bring the representations of nodes that are linked closer, while distancing them from the negative examples. TrueWalks uses explicit negative statements to produce entity representations that take into account both existing attributes and lacking attributes. For example, for the negative statement $(\textit{Bruce Willis}, \textit{NOT birthPlace}, \textit{U.S.})$, our representation would be able to capture the similarity between Bruce Willis and Ryan Gosling, since neither was born in the U.S (see Figure 1). The explicit declaration of negative statements such as these is an important aspect of more accurate representations, especially when they capture unexpected negative statements (i.e., most people would expect that both actors are U.S. born). Using TrueWalks, Bruce Willis and Ryan Gosling would be similar not just because they are both actors but also because neither was born in the U.S.

True Walks generates walks that can distinguish between positive and negative statements and consider the semantic implications of negation in KGs that are rich in ontological information, particularly in regard to inheritance. This is of particular importance for applications in the biomedical domain, where the inadequacy of embedding approaches regarding negative statements has been identified as a crucial limitation [21]. We demonstrate that the resulting embeddings can be employed to determine semantic similarity or as features for relation prediction. We evaluate the effectiveness of our approach in two different tasks, protein-protein interaction prediction and gene-disease association prediction, and show that our method improves performance over state-of-the-art embedding methods and popular semantic similarity measures.

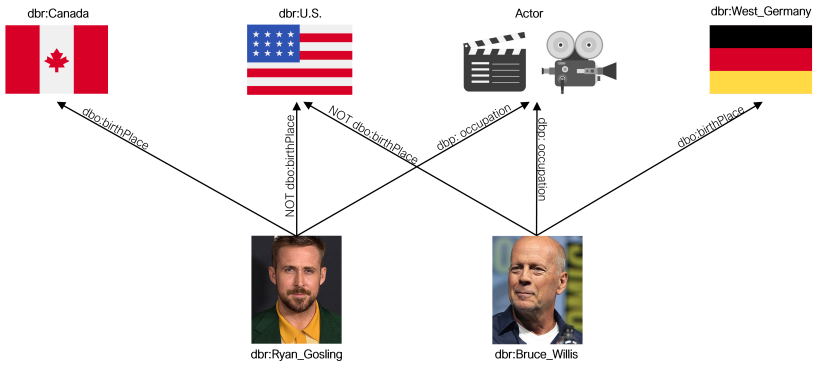

Fig. 1. A DBPedia example motivating the negative statements problem. The author of Bruce Willis' picture is Gage Skidmore.

Our contributions are as follows:

- We propose TrueWalks, a novel method to generate random walks on KGs that are aware of negative statements and results in the first KG embedding approach that considers negative statements.
- We develop extensions of popular path-based KG embedding methods implementing the TrueWalks approach.
- We enrich existing KGs with negative statements and propose benchmark datasets for two popular biomedical KG applications: protein-protein interaction (PPI) prediction and gene-disease association (GDA) prediction.
- We report experimental results that demonstrate the superior performance of TrueWalks when compared to state-of-the-art KG embedding methods.

2 Related Work

2.1 Exploring Negative Statements

Approaches to enrich existing KGs with interesting negative statements have been proposed both for general-purpose KGs such as Wikidata [3] and for domain-specific ones such as the Gene Ontology (GO) [11,44]. Exploring negative statements has been demonstrated to improve the performance of various applications. [2] developed a method to enrich Wikidata with interesting negative statements and its usage improved the performance on entity summarization and decision-making tasks. [44] have designed a method to enrich the GO [14] with relevant negative statements indicating that a protein does not perform a given function and demonstrated that a balance between positive and negative annotations supports a more reasonable evaluation of protein function prediction methods. Similarly, [11] enriched the GO with negative statements and demonstrated an associated increase in protein function prediction performance. The relevance of negative annotations has also been recognized in the prediction of gene-phenotype

associations in the context of the Human-Phenotype Ontology (HP) [22], but the topic remains unexplored [25]. It should be highlighted that KG embedding methods have not been employed in any of these approaches to explore negative statements.

2.2 Knowledge Graph Embeddings

KG embedding methods map entities and their relations expressed in a KG into a lower-dimensional space while preserving the underlying structure of the KG and other semantic information [42]. These entity and relation embedding vectors can then be applied to various KG applications such as link prediction, entity typing, or triple classification. In the biomedical domain, KG embeddings have been used in machine learning-based applications in which they are used as input in classification tasks or to predict relations between biomedical entities. [21] provides an overview of KG embedding-based approaches for biomedical applications.

Translational models, which rely on distance-based scoring functions, are some of the most widely employed KG embedding methods. A popular method, TransE [6], assumes that if a relation holds between two entities, the vector of the head entity plus the relation vector should be close to the vector of the tail entity in the vector space. TransE has the disadvantage of not handling one-to-many and many-to-many relationships well. To address this issue, TransH [43] introduces a relation-specific hyperplane for each relation and projects the head and tail entities into the hyperplane. TransR [23] builds entity and relation embeddings in separate entity space and relation spaces.

Semantic matching approaches are also well-known and use similarity-based scoring functions to capture the latent semantics of entities and relations in their vector space representations. For instance, DistMult [48] employs tensor factorization to embed entities as vectors and relations as diagonal matrices.

2.3 Walk-Based Embeddings

More recently, random walk-based KG embedding approaches have emerged. These approaches are built upon two main steps: (i) producing entity sequences from walks in the graph to produce a corpus of sequences that is akin to a corpus of word sequences or sentences; (2) using those sequences as input to a neural language model [27] that learns a latent low-dimensional representation of each entity within the corpus of sequences.

DeepWalk [31] first samples a set of paths from the input graph using uniform random walks. Then it uses those paths to train a skip-gram model, originally proposed by the word2vec approach for word embeddings [27]. Node2vec [16] introduces a different biased strategy for generating random walks and exploring diverse neighborhoods. The biased random walk strategy is controlled by two parameters: the likelihood of visiting immediate neighbors (breadth-first search behavior), and the likelihood of visiting entities that are at increasing distances (depth-first search behavior). Neither DeepWalk nor node2vec take into account the direction or type of the edges. Metapath2vec [8] proposes random walks driven by metapaths that define the node type order by which the random walker explores the graph. RDF2Vec [35] is inspired by the node2vec strategy but it considers both edge direction and type making it particularly suited to KGs.

OWL2Vec* [7] was designed to learn ontology embeddings and it also employs direct walks on the graph to learn graph structure.

2.4 Tailoring Knowledge Graph Embeddings

Recent KG embedding approaches aim to tailor representations by considering different semantic, structural or lexical aspects of a KG and its underlying ontology. Approaches such as EL [20] and BoxEL [45] embeddings are geometric approaches that account for the logical structure of the ontology (e.g., intersection, conjunction, existential quantifiers). OWL2Vec* [7] and OPA2Vec [37] take into consideration the lexical portion of the KG (i.e., labels of entities) when generating graph walks or triples. OPA2Vec also offers the option of using a pre-trained language model to bootstrap the KG embedding. Closer to our approach, OLW2Vec* contemplates the declaration of inverse axioms to enable reverse path traversal, however, this option was found lacking for the biomedical ontology GO. Finally, different approaches have been proposed to train embeddings that are aware of the order of entities in a path, such as [51] and [34], which extend TransE and RDF2Vec, respectively.

3 Methods

3.1 Problem Formulation

In this work, we address the task of learning a relation between two KG entities (which can belong to the same or different KGs) when the relation itself is not encoded in the KG. We employ two distinct approaches: (1) using the KG embeddings of each entity

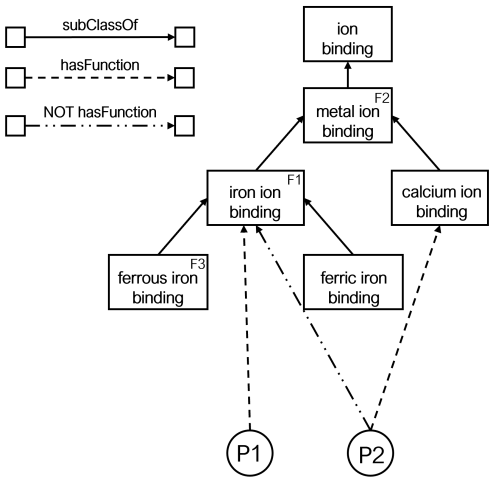

Fig. 2. A GO KG subgraph motivating the *reverse inheritance* problem.

as features for a machine learning algorithm and (2) comparing the KG embeddings directly through a similarity metric.

We target ontology-rich KGs that use an ontology to provide rich descriptions of real-world entities instead of focusing on describing relations between entities themselves. These KGs are common in the biomedical domain. As a result, the KG's richness lies in the TBox, with a comparatively less complex ABox, since entities have no links between them. We focus on Web Ontology Language (OWL) [15] ontologies since biomedical ontologies are typically developed in OWL or have an OWL version.

Biomedical entities in a KG are typically described through positive statements that link them to an ontology. For instance, to state that a protein P performs a function F described under the GO, a KG can declare the axiom $P \sqsubseteq \exists hasFunction.F$. However, the knowledge that a given protein does not perform a function can also be relevant, especially to declare that a given protein does not have an activity typical of its homologs [12]. Likewise, the knowledge that a given disease does not exhibit a particular phenotype is also decisive in understanding the relations between diseases and genes [25]. We consider the definition of grounded negative statements proposed by [2] as $\neg(s, p, o)$ which is satisfied if $(s, p, o) \notin KG$ and expressed as a *NegativeObjectPropertyAssertion*[3]. Similar to what was done in [2], we do not have a negative object property assertion for every missing triple. Negative statements are only included if there is clear evidence that a triple does not exist in the domain being captured. Taking the protein example, negative object property assertions only exist when it has been demonstrated that a protein does not perform a particular function.

An essential difference between a positive and a negative statement of this kind is related to the implied inheritance of properties exhibited by the superclasses or subclasses of the assigned class. Let us consider that $(P_1, \textit{hasFunction}, F_1)$ and $(F_1, \textit{subClassOf}, F_2)$. This implies that $(P_1, \textit{hasFunction}, F_2)$, since an individual with a class assignment also belongs to all superclasses of the given class, e.g., a protein that performs *iron ion binding* also performs *metal ion binding* (see Figure 2). This implication is easily captured by directed walk generation methods that explore the declared subclass axioms in an OWL ontology. However, when we have a negative statement, such as $\neg(P_2, \textit{hasFunction}, F_1)$, it does not imply that $\neg(P_2, \textit{hasFunction}, F_2)$. There are no guarantees that a protein that does not perform *iron ion binding* also does not perform *metal ion binding*, since it can very well, for instance, perform *calcium ion binding*. However, for $(F_3, \textit{subClassOf}, F_1)$ the negative statement $\neg(P_2, \textit{hasFunction}, F_1)$ implies that $\neg(P_2, \textit{hasFunction}, F_3)$, as a protein that does not perform *iron ion binding* also does not perform *ferric iron binding* nor *ferrous iron binding*. Therefore, we need to be able to declare that protein P_1 performs both functions F_1 and F_3, but that P_2 performs F_1 but not F_3. Since OWL ontologies typically declare subclass axioms, there is no opportunity for typical KG embedding methods to explore the reverse paths that would more accurately represent a negative statement.

The problem we tackle is then two-fold: how can the *reverse inheritance* implied by negative statements be adequately explored by walk-based KG embedding methods, and how can these methods distinguish between negative and positive statements.

[3] https://www.w3.org/TR/owl2-syntax/#Negative_Object_Property_Assertions

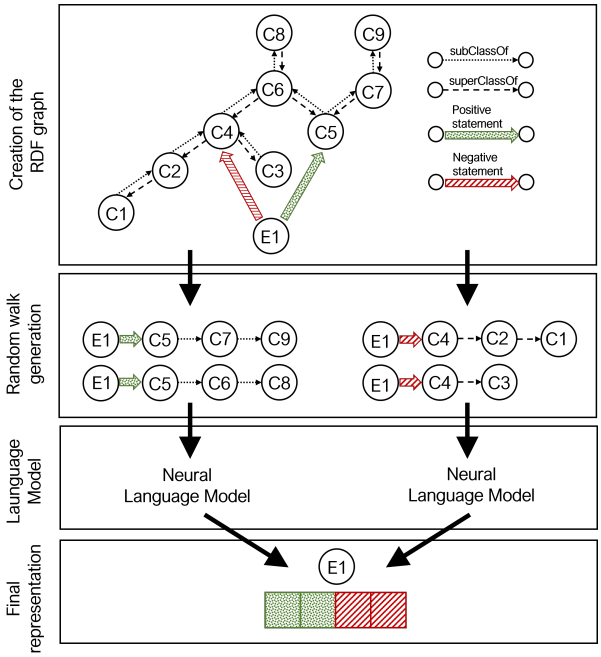

Fig. 3. Overview of the TrueWalks method with the four main steps: (i) creation of the RDF graph, (ii) random walk generation with negative statements; (iii) neural language models, and (iv) final representation.

3.2 Overview

An overview of TrueWalks, the method we propose, is shown in Figure 3. The first step is the transformation of the KG into an RDF Graph. Next, our novel random walk generation strategy that is aware of positive and negative statements is applied to the graph to produce a set of entity sequences. The positive and negative entity walks are fed to neural language models to learn a dual latent representation of the entities. TrueWalks has two variants: one that employs the classical skip-gram model to learn the embeddings (TrueWalks), and one that employs a variation of skip-gram that is aware of the order of entities in the walk (TrueWalksOA, i.e. order-aware).

3.3 Creation of the RDF Graph

The first step is the conversion of an ontology-rich KG into an RDF graph. This is a directed, labeled graph, where the edges represent the named relations between two resources or entities, represented by the graph nodes[4]. We perform the transformation according to the *OWL to RDF Graph Mapping* guidelines defined by the W3C[5]. Simple

[4]https://www.w3.org/RDF/

[5]https://www.w3.org/TR/owl2-mapping-to-rdf/

axioms can be directly transformed into RDF triples, such as subsumption axioms for atomic entities or data and annotation properties associated with an entity. Axioms involving complex class expressions are transformed into multiple triples which typically require blank nodes.

Let us consider the following existential restriction of the class *obo:GO_0034708 (methyltransferase complex)* that encodes the fact that a methyltransferase complex is part of at least one intracellular anatomical structure:

ObjectSomeValuesFrom(obo:BFO_0000050 (part of),
obo:GO_0005622 (intracellular anatomical structure))

Its conversion to RDF results in three triples:

(obo:GO_0034708, rdfs:subClassOf, _:x)
(_:x, owl:someValuesFrom,obo:GO_0005622)
(_:x, owl:onProperty,obo:BFO_0000050)

where _:x denotes a blank node.

3.4 Random Walk Generation with Negative Statements

The next step is to generate the graph walks that will make up the corpus (see Algorithm 1). For a given graph $G = (V, E)$ where E is the set of edges and V is the set of vertices, for each vertex $v_r \in V_r$, where V_r is the subset of individuals for which we want to learn representations, we generate up to w graph walks of maximum depth d rooted in vertex v_r. We employ a depth-first search algorithm, extending on the basic approach in [35]. At the first iteration, we can find either a positive or negative statement. From then on, walks are biased: a positive statement implies that whenever a subclass edge is found it is traversed from subclass to superclass, whereas a negative statement results in a traversal of subclass edges in the opposite direction (see also Figure 3). This generates paths that follow the pattern $v_r \rightarrow e_{1i} \rightarrow v_{1i} \rightarrow e_{2i}$. The set of walks is split in two, negative statement walks and positive statement walks. This will allow the learning of separate latent representations, one that captures the positive aspect and one that captures the negative aspect.

An important aspect of our approach is that, since OWL is converted into an RDF graph for walk-based KG embedding methods, a negative statement declared using a simple object property assertion (e.g. *notHasFunction*) could result in the less accurate path: *Protein P* $\rightarrow$ *notHasFunction* $\rightarrow$ *iron ion binding* $\rightarrow$ *subClassOf* $\rightarrow$ *ion binding*. Moreover, random walks directly over the *NegativeObjectPropertyAssertion*, since it is decomposed into multiple triples using blank nodes, would also result in inaccurate paths. However, our algorithm produces more accurate paths, e.g.: *Protein P* $\rightarrow$ *notHasFunction* $\rightarrow$ *iron ion binding* $\rightarrow$ *superClassOf* $\rightarrow$ *ferric iron binding* by adequately processing the *NegativeObjectPropertyAssertion*.

3.5 Neural Language Models

We employ two alternative approaches to learn a latent representation of the individuals in the KG. For the first approach, we use the skip-gram model [27], which predicts the context (neighbor entities) based on a target word, or in our case a target entity.

Algorithm 1 Walk generation for one entity using TrueWalks. The function GET NON VISITED NEIGHBOURS(status) is used to generate the random walks using a depth-first search. It gets the neighbors of a given node that have not yet been visited in previous iterations. If the status is negative (which means that the first step in the walk was made with a negative statement), the neighbors will include all the non-visited neighbors except those connected through subclass statements, and if the status is positive, it will include all the neighbors except those connected through superclass statements.

$d \leftarrow max\_depth\_walks$
$w \leftarrow max\_number\_of\_walks$
$ent \leftarrow root\_entity$
function GET TRUEWALKS(ent)
  $pos\_walks \leftarrow$ GET RANDOM WALKS($ent, positive$)
  $neg\_walks \leftarrow$ GET RANDOM WALKS($ent, negative$)
  return pos_walks, neg_walks
function GET RANDOM WALKS($ent, status$)
  while $len(walks) < w$ **do**
    $walk \leftarrow ent$
    $depth \leftarrow 1$
    while $depth < d$ **do**
      $last \leftarrow len(walk) == d$
      $e, v \leftarrow$ GET NEIGHBOR($walk, status, last$)
      if $e, v == None$ **then**
        break
      $walk$.append(e, v)
      $depth++$
    $walks$.append($walk$)
  return $walks$
function GET NEIGHBOR($walk, status, last$)
  $n \leftarrow$ GET NON VISITED NEIGHBORS($status$)
  if $len(n) == 0$ & $len(walk) > 2$ **then**
    $e, v \leftarrow walk[-2], walk[-1]$
    ADD VISITED NEIGHBORS($e, v, len(walk) - 2, status$)
    return $None$
  $e, v \leftarrow n[rand()]$
  if $last$ **then**
    ADD VISITED NEIGHBORS($e, v, len(walk), status$)
  return e, v

Let $f : E \rightarrow \mathbb{R}^d$ be the mapping function from entities to the latent representations we will be learning, where d is the number of dimensions of the representation (f is then a matrix $|E| \times d$). Given a context window c, and a sequence of entities $e_1, e_2, e_3, ..., e_L$, the objective of the skip-gram model is to maximize the average log probability p:

$$\frac{1}{L} \sum_{l=1}^{L} \log p(e_{l+c}|e_l) \tag{1}$$

where $p(e_{l+c}|e_l)$ is calculated using the softmax function:

$$p(e_{l+c}|e_l) = \frac{\exp(f(e_{l+c}) \cdot f(e_l))}{\sum_{e=1}^{E} \exp(f(e) \cdot f(e_l))} \quad (2)$$

where $f(e)$ is the vector of the entity e.

To improve computation time, we employ a negative sampling approach based in [27] that minimizes the number of comparisons required to distinguish the target entity, by taking samples from a noise distribution using logistic regression, where there are k negative samples for each entity.

The second approach is the structured skip-gram model [24], a variation of skip-gram that is sensitive to the order of words, or in our case, entities in the graph walks. The critical distinction of this approach is that, instead of using a single matrix f, it creates $c \times 2$ matrices, $f_{-c}, ..., f_{-2}, f_{-1}, f_1, ..., f_c$, each dedicated to predicting a specific relative position to the entity. To make a prediction $p(e_{l+c}|e_l)$, the method selects the appropriate matrix f_l.

The neural language models are applied separately to the positive and negative walks, producing two representations for each entity.

3.6 Final Representations

The two representations of each entity need to be combined to produce a final representation. Different vector operations can, in principle, be employed, such as the Hadamard product or the L1-norm. However, especially since we will employ these vectors as inputs for machine learning methods, we would like to create a feature space that allows the distinction between the negative and positive representations, motivating us to use a simple concatenation of vectors.

4 Experiments

We evaluate our novel approach on two biomedical tasks: protein-protein interaction (PPI) prediction and gene-disease association (GDA) prediction[39]. These two challenges have significant implications for understanding the underlying mechanisms of biological processes and disease states.

Both tasks are modeled as relation prediction tasks. For PPI prediction, we employ TrueWalks embeddings both as features for a supervised learning algorithm and directly for similarity-based prediction. For GDA prediction, since embeddings for genes and diseases are learned over two different KGs, we focus only on supervised learning. We employ a Random Forest algorithm across all classification experiments with the same parameters (see the supplementary file for details).

4.1 Data

Our method takes as input an ontology file, instance annotation file and a list of instance pairs. We construct the knowledge graph (KG) using the RDFlib package [5], which

Table 1. Statistics for each KG regarding classes, instances, nodes, edges, positive and negative statements.

	GO_{PPI}	GO_{GDA}	HP_{GDA}
Classes	50918	50918	17060
Literals and blank nodes	532373	532373	442246
Edges	1425102	1425102	1082859
Instances	440	755	162
Positive statements	7364	10631	4197
Negative statements	8579	8966	225

parses the ontology file in OWL format and processes the annotation file to add edges to the RDFlib graph. The annotation file contains both positive and negative statements which are used to create the edges in the graph.

Protein-Protein Interaction Prediction Predicting protein-protein interactions is a fundamental task in molecular biology that can explore both sequence and functional information [18]. Given the high cost of experimentally determining PPI, computational methods have been proposed as a solution to the problem of finding protein pairs that are likely to interact and thus provide a selection of good candidates for experimental analysis. In recent years, a number of approaches for PPI prediction based on functional information as described by the GO have been proposed [50,20,37,38,21]. The GO contains over 50000 classes that describe proteins or genes according to the molecular functions they perform, the biological processes they are involved in, and the cellular components where they act.

The GO KG is built by integrating three sources: the GO itself [14], the Gene Ontology Annotation (GOA) data [13], and negative GO annotations [44] (details on the KG building method and data sources are available in the supplementary file). A GO annotation associates a Uniprot protein identifier with a GO class that describes it. We downloaded the GO annotations corresponding to positive statements from the GOA database for human species. For each protein P in the PPI dataset and each of its association statements to a function F in GOA, we add the assertion $(P, \mathit{hasFunction}, F)$. We employ the negative GO associations produced in [44], which were derived from expert-curated annotations of protein families on phylogenetic trees. For each protein P in the PPI dataset and each of its association statements to a function F in the negative GO associations dataset, we add a negative object property assertion. To do so, we use metamodeling (more specifically, punning [6]) and represent each ontology class as both a class and an individual. This situation translates into using the same IRI. Then, we use a negative object property assertion to state that the individual representing a biomedical entity is not connected by the object property expression to the individual representing an ontology class. Table 1 presents the GO KG statistics.

The target relations to predict are extracted from the STRING database [40]. We considered the following criteria to select protein pairs: (i) protein interactions must be

[6] https://www.w3.org/TR/owl2-new-features/#F12:_Punning

extracted from curated databases or experimentally determined (as opposed to computationally determined); (ii) interactions must have a confidence score above 0.950 to retain only high confidence interaction; (iii) each protein must have at least one positive GO association and one negative GO association. The PPI dataset contains 440 proteins, 1024 interacting protein pairs, and another 1024 pairs generated by random negative sampling over the same set of proteins.

Gene-Disease Association Prediction Predicting the relation between genes and diseases is essential to understand disease mechanisms and identify potential biomarkers or therapeutic targets [9]. However, validating these associations in the wet lab is expensive and time-consuming, which fostered the development of computational approaches to identify the most promising associations to be further validated. Many of these explore biomedical ontologies and KGs [41,49,36,4,26] and some recent approaches even apply KG embedding methods such as DeepWalk [1] or OPA2Vec [37,30].

For GDA prediction, we have used the GO KG, the Human Phenotype Ontology (HP) KG (created from the HP file and HP annotations files), and a GDA dataset. Two different ontologies are used to describe each type of entity. Diseases are described under the HP and genes under the GO. We built GO KG in the same fashion as in the PPI experiment, but instead of having proteins linked to GO classes, we have genes associated with GO classes. Regarding HP KG, HP [22] describes phenotypic abnormalities found in human hereditary diseases. The HP annotations link a disease to a specific class in the HP through both positive and negative statements.

The target relations to predict are extracted from DisGeNET [33], adapting the approach described in [30] to consider the following criterion: each gene (or disease) must have at least one positive GO (or HP) association and one negative GO (or HP) association. This resulted in 755 genes, 162 diseases, and 107 gene-disease relations. To create a balanced dataset, we sampled random negative examples over the same genes and diseases. Table 1 describes the created KGs.

4.2 Results and Discussion

We compare TrueWalks against ten state-of-the-art KG embedding methods: TransE, TransH, TransR, ComplEx, distMult, DeepWalk, node2vec, metapath2vec, OWL2Vec* and RDF2Vec. TransE, TransH and TransR are representative methods of translational models. ComplEx and distMult are semantic matching methods. They represent a bottom-line baseline with well-known KG embedding methods. DeepWalk and node2vec are undirected random walk-based methods, and OWL2Vec* and RDF2Vec are directed walk-based methods. These methods represent a closer approach to ours, providing a potentially stronger baseline. Each method is run with two different KGs, one with only positive statements and one with both positive and negative statements. In this second KG, we declare the negative statements as an object property, so positive and negative statements appear as two distinct relation types. The size of all the embeddings is 200 dimensions across all experiments (details on parameters can be found in the supplementary file), with TrueWalks generating two 100-dimensional vectors, one for the positive statement-based representation and one for the negative, which are concatenated to produce the final 200-dimensional representation.

Relation Prediction using Machine Learning To predict the relation between a pair of entities e_1 and e_2 using machine learning, we take their vector representations and combine them using the binary Hadamard operator to represent the pair: $r(e_1, e_2) = v_{e_1} \times v_{e_2}$. The pair representations are then fed into a Random Forest algorithm for training using Monte Carlo cross-validation (MCCV) [46]. MCCV is a variation of traditional k-fold cross-validation in which the process of dividing the data into training and testing sets (with β being the proportion of the dataset to include in the test split) is repeated M times. Our experiments use MCCV with $M = 30$ and $\beta = 0.3$. For each run, the predictive performance is evaluated based on recall, precision and weighted average F-measure. Statistically significant differences between TrueWalks and the other methods are determined using the non-parametric Wilcoxon test at $p < 0.05$.

Table 2 reports the median scores for both PPI and GDA prediction. The top half contains the results of the first experiment where we compare state-of-the-art methods using only the positive statements to TrueWalks (at the bottom) which uses both types. The results reveal that the performance of TrueWalks is significantly better than the

Table 2. Median precision, recall, and F-measure (weighted average F-measure) for PPI and GDA prediction. TrueWalks performance values are italicized/underlined when improvements are statistically significant with p-value < 0.05 for the Wilcoxon test against the positive (Pos)/positive and negative (Pos+Neg) variants of other methods. The best results are in bold.

	Method	PPI Prediction			GDA Prediction		
		Precision	**Recall**	**F-measure**	**Precision**	**Recall**	**F-measure**
Pos	TransE	0.553	0.546	0.554	0.533	0.538	0.531
	TransH	0.566	0.562	0.566	0.556	0.563	0.548
	TransR	0.620	0.607	0.616	0.594	0.600	0.592
	ComplEx	0.680	0.659	0.679	0.597	0.625	0.598
	distMult	0.765	0.737	0.754	0.585	0.600	0.575
	DeepWalk	0.813	0.836	0.822	0.618	0.646	0.629
	node2vec	0.826	0.741	0.794	0.643	0.616	0.644
	metapath2vec	0.562	0.563	0.561	0.554	0.531	0.549
	OWL2Vec*	0.833	0.806	0.823	0.652	0.656	0.646
	RDF2Vec	0.831	0.826	0.828	0.623	0.625	0.615
Pos+Neg	TransE	0.584	0.582	0.585	0.597	0.585	0.586
	TransH	0.573	0.572	0.570	0.563	0.554	0.554
	TransR	0.722	0.678	0.704	0.633	0.625	0.630
	ComplEx	0.750	0.720	0.740	0.549	0.545	0.545
	distMult	0.813	0.740	0.784	0.530	0.523	0.534
	DeepWalk	0.843	0.834	0.841	0.615	0.646	0.630
	node2vec	0.847	0.734	0.798	0.614	0.594	0.621
	metapath2vec	0.557	0.569	0.558	0.527	0.531	0.522
	OWL2Vec*	0.860	0.812	0.840	0.654	0.600	0.645
	RDF2Vec	0.847	**0.844**	0.845	0.625	**0.661**	0.630
	TrueWalks	***0.870***	0.817	*0.846*	***0.667***	0.625	***0.661***
	TrueWalksOA	*0.868*	0.836	***0.858***	*0.661*	0.616	*0.654*

other methods, improving both precision and F-measure. An improvement in precision, which is not always accompanied by an increase in recall, confirms the hypothesis that embeddings that consider negative statements produce more accurate representations of entities, which allows a better distinction of true positives from false positives.

A second experiment employs a KG with both negative and positive statements for all methods. Our method can accurately distinguish between positive statements and negative statements, as discussed in subsection 3.4. For the remaining embedding methods, we declare the negative statements as an object property so that these methods distinguish positive and negative statements as two distinct types of relation. This experiment allows us to test whether TrueWalks, which takes into account the positive or negative status of a statement, can improve the performance of methods that handle all statements equally regardless of status.

The bottom half of Table 2 shows that both variants of TrueWalks improve on precision and F-measure for both tasks when compared with the state-of-the-art methods using both positive and negative statements. This experiment further shows that the added information given by negative statements generally improves the performance of most KG embedding methods. However, no method surpasses TrueWalks, likely due to its ability to consider the semantic implications of inheritance and walk direction, especially when combined with the order-aware model.

Comparing the two variants of TrueWalks demonstrates that order awareness does not improve performance in most cases. However, TrueWalksOA improves on precision and F-measure for all other state-of-the-art methods. These results are not unexpected since the same effect was observed in other order-aware embedding methods [34].

Regarding the statistical tests, TrueWalks performance values are italicized/underlined in Table 2 when improvements over all other methods are statistically significant, except when comparing TrueWalks with OWL2Vec* for GDA, since in this particular case the improvement is not statistically significant.

Relation Prediction using Semantic Similarity We also evaluate all methods in PPI prediction using KG embedding-based semantic similarity, computed as the cosine similarity between the vectors of each protein in a pair. Adopting the methodology employed by [20] and [45], for each positive pair e_1 and e_2 in the dataset, we compute the similarity between e_1 and all other entities and identify the rank of e_2. The performance was measured using recall at rank 10[7], recall at rank 100, mean rank, and the area under the ROC curve (Table 3). Results show that TrueWalksOA achieves the top performance across all metrics, but TrueWalks is bested by RDF2Vec on all metrics except Hits@10, by OWL2Vec* on Hits@100 and by node2vec on Hits@10.

To better understand these results, we plotted the distribution of similarity values for positive and negative pairs in Figure 4. There is a smaller overlap between negative and positive pairs similarities for TrueWalksOA, which indicates that considering both the status of the function assignments and the order of entities in the random walks

[7]Since we compute the similarity score for all possible pairs to simulate a more realistic scenario where a user is presented with a ranked list of candidate interactions, the task is several degrees more difficult to performant and all KG embedding methods have a recall score of 0 at rank 1. As a result, we have excluded the results for this metric from our analysis.

Table 3. Hits@10, Hits@100, mean rank, and ROC-AUC for PPI prediction using cosine similarity obtained with different methods. In bold, the best value for each metric.

	Method	Hits@10	Hits@100	MeanRank	AUC
Pos	TransE	0.013	0.125	103.934	0.538
	TransH	0.013	0.134	102.703	0.543
	TransR	0.037	0.196	81.916	0.636
	ComplEx	0.080	0.261	64.558	0.689
	distMult	0.112	0.340	46.512	0.803
	DeepWalk	0.125	0.380	35.406	0.847
	node2vec	0.163	0.375	37.275	0.827
	metapath2vec	0.017	0.151	98.445	0.558
	OWL2Vec*	0.152	0.386	33.192	0.860
	RDF2Vec	0.133	0.391	32.419	0.870
Pos + Neg	TransE	0.022	0.161	94.809	0.576
	TransR	0.100	0.274	60.120	0.732
	TransH	0.025	0.174	91.553	0.594
	ComplEx	0.132	0.334	45.268	0.805
	distMult	0.149	0.378	35.351	0.853
	DeepWalk	0.148	0.383	35.365	0.849
	node2vec	**0.166**	0.389	34.305	0.840
	metapath2vec	0.020	0.165	93.374	0.578
	OWL2Vec*	0.160	0.397	32.234	0.869
	RDF2Vec	0.155	0.401	30.281	0.879
TrueWalks		0.161	0.392	32.089	0.869
TrueWalksOA		**0.166**	**0.407**	**28.128**	**0.889**

results in embeddings that are more meaningful semantic representations of proteins. Furthermore, the cosine similarity for negative pairs is consistently lower when using both variants of TrueWalks, which supports that the contribution of negative statement-based embeddings is working towards filtering out false positives.

5 Conclusion

Knowledge graph embeddings are increasingly used in biomedical applications such as the prediction of protein–protein interactions, gene-disease associations, drug-target relations and drug-drug interactions [28]. Our novel approach, TrueWalks, was motivated by the fact that existing knowledge graph embedding methods are ill-equipped to handle negative statements, despite their recognized importance in biomedical machine learning tasks [21]. TrueWalks incorporates a novel walk-generation method that distinguishes between positive and negative statements and considers the semantic implications of negation in ontology-rich knowledge graphs. It generates two separate embeddings, one for each type of statement, enabling a dual representation of entities that can be explored by downstream ML, focusing both on features entities have and those they lack. TrueWalks outperforms representative and state-of-the-art knowledge

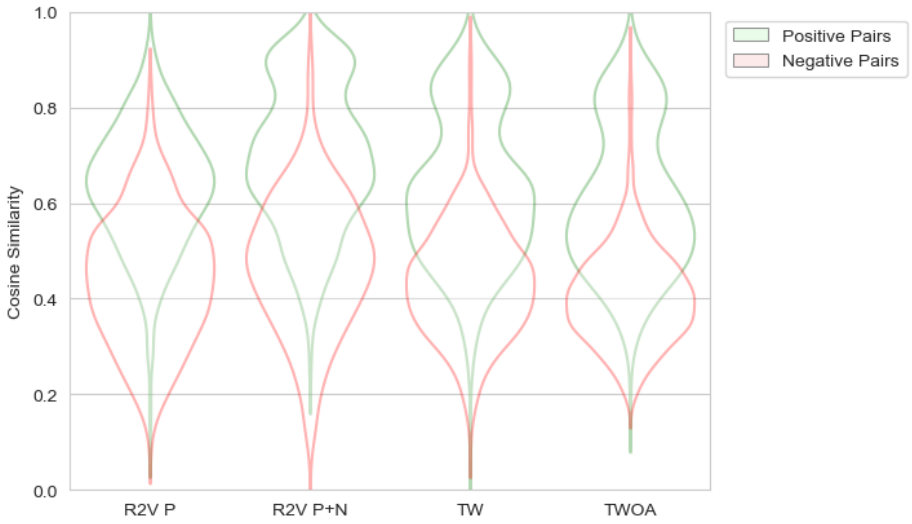

Fig. 4. Violin plot with embedding similarity obtained with RDF2Vec with positive statements (R2V P), RDF2Vec with both positive and negative statements (R2V P+N), TrueWalks (TW), and TrueWalksOA (TWOA).

graph embedding approaches in the prediction of protein-protein interactions and gene-disease associations.

We expect TrueWalks to be generalizable to other biomedical applications where negative statements play a decisive role, such as predicting disease-related phenotypes [47] or performing differential diagnosis [19]. In future work, we would also like to explore counter-fitting approaches, such as those proposed for language embeddings [29], to consider how opposite statements can impact the dissimilarity of entities.

Supplemental Material Statement: The source code for True Walks is available on GitHub (`https://github.com/liseda-lab/TrueWalks`). All datasets are available on Zenodo (`https://doi.org/10.5281/zenodo.7709195`). A supplementary file contains the links to the data sources, the parameters for the KG embedding methods and ML models, and the results of the statistical tests.

Acknowledgements C.P., S.S., and R.T.S. are funded by FCT, Portugal, through LASIGE Research Unit (ref. UIDB/00408/2020 and ref. UIDP/00408/2020). R.T.S. acknowledges the FCT PhD grant (ref. SFRH/BD/145377/2019) and DAAD Contact Fellowship grant. This work was also partially supported by the KATY project, which has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 101017453, and in part by projeto 41, HfPT: Health from Portugal, funded by the Portuguese Plano de Recuperação e Resiliência. The authors are grateful to Lina Aveiro and Carlota Cardoso for the fruitful discussions that inspired this work.

References

1. Alshahrani, M., Khan, M.A., Maddouri, O., Kinjo, A.R., Queralt-Rosinach, N., Hoehndorf, R.: Neuro-symbolic representation learning on biological knowledge graphs. Bioinformatics **33**(17), 2723–2730 (2017)
2. Arnaout, H., Razniewski, S., Weikum, G., Pan, J.Z.: Negative statements considered useful. Journal of Web Semantics **71**, 100661 (2021), publisher: Elsevier
3. Arnaout, H., Razniewski, S., Weikum, G., Pan, J.Z.: Wikinegata: a knowledge base with interesting negative statements. Proceedings of the VLDB Endowment **14**(12), 2807–2810 (2021), publisher: VLDB Endowment Inc.
4. Asif, M., Martiniano, H., Couto, F.: Identifying disease genes using machine learning and gene functional similarities, assessed through Gene Ontology. PLOS ONE **13**, e0208626 (12 2018)
5. Boettiger, C.: rdflib: A high level wrapper around the redland package for common rdf applications (2018)
6. Bordes, A., Usunier, N., Garcia-Durán, A., Weston, J., Yakhnenko, O.: Translating embeddings for modeling multi-relational data. In: Proceedings of NIPS 2013. p. 2787–2795. Curran Associates Inc., Red Hook, NY, USA (2013)
7. Chen, J., Hu, P., Jimenez-Ruiz, E., Holter, O.M., Antonyrajah, D., Horrocks, I.: OWL2Vec*: Embedding of OWL ontologies. Machine Learning pp. 1–33 (2021)
8. Dong, Y., Chawla, N.V., Swami, A.: metapath2vec: Scalable representation learning for heterogeneous networks. In: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 135–144 (2017)
9. Eilbeck, K., Quinlan, A., Yandell, M.: Settling the score: variant prioritization and mendelian disease. Nature Reviews Genetics **18**(10), 599–612 (2017)
10. Flouris, G., Huang, Z., Pan, J.Z., Plexousakis, D., Wache, H.: Inconsistencies, negations and changes in ontologies. In: Proceedings of the 21st National Conference on Artificial Intelligence-Volume 2. pp. 1295–1300 (2006)
11. Fu, G., Wang, J., Yang, B., Yu, G.: NegGOA: negative GO annotations selection using ontology structure. Bioinformatics **32**(19), 2996–3004 (06 2016)
12. Gaudet, P., Dessimoz, C.: Gene Ontology: pitfalls, biases, and remedies. In: The Gene Ontology Handbook, pp. 189–205. Humana Press, New York, NY (2017)
13. GO Consortium: The Gene Ontology resource: enriching a GOld mine. Nucleic Acids Research **49**(D1), D325–D334 (2021)
14. GO Consortium: The Gene Ontology Resource: 20 years and still GOing strong. Nucleic Acids Research **47**(D1), D330–D338 (11 2018)
15. Grau, B.C., Horrocks, I., Motik, B., Parsia, B., Patel-Schneider, P., Sattler, U.: OWL 2: The next step for OWL. Journal of Web Semantics **6**(4), 309–322 (2008)
16. Grover, A., Leskovec, J.: node2vec: Scalable feature learning for networks. In: Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 855–864 (2016)
17. Hogan, A., Blomqvist, E., Cochez, M., d'Amato, C., Melo, G.d., Gutierrez, C., Kirrane, S., Gayo, J.E.L., Navigli, R., Neumaier, S., et al.: Knowledge graphs. ACM Computing Surveys (CSUR) **54**(4), 1–37 (2021)
18. Hu, L., Wang, X., Huang, Y.A., Hu, P., You, Z.H.: A survey on computational models for predicting protein–protein interactions. Briefings in Bioinformatics **22**(5), bbab036 (2021)
19. Köhler, S., Øien, N.C., Buske, O.J., Groza, T., Jacobsen, J.O., McNamara, C., Vasilevsky, N., Carmody, L.C., Gourdine, J., Gargano, M., et al.: Encoding clinical data with the Human Phenotype Ontology for computational differential diagnostics. Current Protocols in Human Genetics **103**(1), e92 (2019)

20. Kulmanov, M., Liu-Wei, W., Yan, Y., Hoehndorf, R.: EL embeddings: geometric construction of models for the description logic EL++. Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence (2019)
21. Kulmanov, M., Smaili, F.Z., Gao, X., Hoehndorf, R.: Semantic similarity and machine learning with ontologies. Briefings in Bioinformatics **22**(4), bbaa199 (2021)
22. Köhler, S., Gargano, M., Matentzoglu, N., Carmody, L.C., Lewis-Smith, D., Vasilevsky, N.A., Danis, D.e.a.: The Human Phenotype Ontology in 2021. Nucleic Acids Research **49**(D1), D1207–D1217 (12 2020)
23. Lin, Y., Liu, Z., Sun, M., Liu, Y., Zhu, X.: Learning entity and relation embeddings for knowledge graph completion. Proceedings of the AAAI conference on artificial intelligence **29**(1) (2015)
24. Ling, W., Dyer, C., Black, A.W., Trancoso, I.: Two/too simple adaptations of word2vec for syntax problems. In: Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies. pp. 1299–1304 (2015)
25. Liu, L., Zhu, S.: Computational methods for prediction of human protein-phenotype associations: A review. Phenomics **1**(4), 171–185 (2021)
26. Luo, P., Xiao, Q., Wei, P.J., Liao, B., Wu, F.X.: Identifying disease-gene associations with graph-regularized manifold learning. Frontiers in Genetics **10** (2019)
27. Mikolov, T., Chen, K., Corrado, G., Dean, J.: Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781 (2013)
28. Mohamed, S.K., Nounu, A., Nováček, V.: Biological applications of knowledge graph embedding models. Briefings in Bioinformatics **22**(2), 1679–1693 (2021)
29. Mrksic, N., Séaghdha, D.Ó., Thomson, B., Gasic, M., Rojas-Barahona, L.M., Su, P.H., Vandyke, D., Wen, T.H., Young, S.J.: Counter-fitting word vectors to linguistic constraints. In: HLT-NAACL (2016)
30. Nunes, S., Sousa, R.T., Pesquita, C.: Predicting gene-disease associations with knowledge graph embeddings over multiple ontologies. In: ISMB Annual Meeting - Bio-Ontologies (2021)
31. Perozzi, B., Al-Rfou, R., Skiena, S.: Deepwalk: Online learning of social representations. In: Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. pp. 701–710 (2014)
32. Pesquita, C., Faria, D., Falcao, A.O., Lord, P., Couto, F.M.: Semantic similarity in biomedical ontologies. PLoS Computational Biology **5**(7), e1000443 (2009)
33. Piñero, J., Ramírez-Anguita, J.M., Saüch-Pitarch, J., Ronzano, F., Centeno, E., Sanz, F., Furlong, L.I.: The DisGeNET knowledge platform for disease genomics: 2019 update. Nucleic Acids Research **48**(D1), D845–D855 (11 2019)
34. Portisch, J., Paulheim, H.: Putting RDF2Vec in order. In: CEUR Workshop Proceedings. vol. 2980, pp. 1–5. RWTH (2021)
35. Ristoski, P., Paulheim, H.: RDF2Vec: RDF graph embeddings for data mining. In: International Semantic Web Conference. pp. 498–514. Springer (2016)
36. Robinson, P., Köhler, S., Oellrich, A., Genetics, S., Wang, K., Mungall, C., Lewis, S., Washington, N., Bauer, S., Seelow, D., Krawitz, P., Gilissen, C., Haendel, M., Smedley, D.: Improved exome prioritization of disease genes through cross-species phenotype comparison. PCR Methods and Applications **24**(2), 340–348 (Feb 2014)
37. Smaili, F.Z., Gao, X., Hoehndorf, R.: OPA2Vec: combining formal and informal content of biomedical ontologies to improve similarity-based prediction. Bioinformatics **35**(12), 2133–2140 (2019)
38. Sousa, R.T., Silva, S., Pesquita, C.: Evolving knowledge graph similarity for supervised learning in complex biomedical domains. BMC Bioinformatics **21**(1), 1–19 (2020)

39. Sousa, R.T., Silva, S., Pesquita, C.: Benchmark datasets for biomedical knowledge graphs with negative statements (2023)
40. Szklarczyk, D., Gable, A.L., Nastou, K.C., Lyon, D., Kirsch, R., Pyysalo, S., Doncheva, N.T., Legeay, M., Fang, T., Bork, P., Jensen, L.J., von Mering, C.: The STRING database in 2021: customizable protein–protein networks, and functional characterization of user-uploaded gene/measurement sets. Nucleic Acids Research **49**(D1), D605–D612 (11 2020)
41. Vanunu, O., Magger, O., Ruppin, E., Shlomi, T., Sharan, R.: Associating genes and protein complexes with disease via network propagation. PLoS Computational Biology **6** (2010)
42. Wang, Q., Mao, Z., Wang, B., Guo, L.: Knowledge graph embedding: A survey of approaches and applications. IEEE Transactions on Knowledge and Data Engineering **29**(12), 2724–2743 (2017)
43. Wang, Z., Zhang, J., Feng, J., Chen, Z.: Knowledge Graph Embedding by Translating on Hyperplanes. In: Proceedings of the 28th AAAI Conference on Artificial Intelligence. pp. 1112–1119. AAAI Press (2014)
44. Warwick Vesztrocy, A., Dessimoz, C.: Benchmarking Gene Ontology function predictions using negative annotations. Bioinformatics **36**(Supplement_1), i210–i218 (07 2020)
45. Xiong, B., Potyka, N., Tran, T.K., Nayyeri, M., Staab, S.: Faithful Embeddings for EL++ Knowledge Bases. In: International Semantic Web Conference. pp. 22–38. Springer (2022)
46. Xu, Q.S., Liang, Y.Z.: Monte Carlo cross validation. Chemometrics and Intelligent Laboratory Systems **56**(1), 1–11 (2001)
47. Xue, H., Peng, J., Shang, X.: Predicting disease-related phenotypes using an integrated phenotype similarity measurement based on HPO. BMC Systems Biology **13**(2), 1–12 (2019)
48. Yang, B., tau Yih, W., He, X., Gao, J., Deng, L.: Embedding entities and relations for learning and inference in knowledge bases (2015)
49. Zakeri, P., Simm, J., Arany, A., ElShal, S., Moreau, Y.: Gene prioritization using bayesian matrix factorization with genomic and phenotypic side information. Bioinformatics **34**, i447 – i456 (2018)
50. Zhang, S.B., Tang, Q.R.: Protein–protein interaction inference based on semantic similarity of Gene Ontology terms. Journal of Theoretical Biology **401**, 30–37 (2016)
51. Zhu, Y., Liu, H., Wu, Z., Song, Y., Zhang, T.: Representation learning with ordered relation paths for knowledge graph completion. In: Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP). pp. 2662–2671 (2019)