Link Analysis meets Ontologies: Are Embeddings the Answer?

Sebastian Mežnar[a] (Researcher), Matej Bevec[a] (Researcher), Nada Lavrač[a,b,c] (Researcher) and Blaž Škrlj[a,b] (Researcher)

[a]*Jožef Stefan Institute, Jamova 39, Ljubljana, Slovenia*
[b]*Jožef Stefan International Postgraduate School, Jamova 39, Ljubljana, Slovenia*
[c]*University of Nova Gorica, Glavni trg 8, Vipava, Slovenia*

ARTICLE INFO

ABSTRACT

The increasing amounts of semantic resources offer valuable storage of human knowledge; however, the probability of wrong entries increases with the increased size. The development of approaches that identify potentially spurious parts of a given knowledge base is thus becoming an increasingly important area of interest. In this work, we present a systematic evaluation of whether structure-only link analysis methods can already offer a scalable means to detecting possible anomalies, as well as potentially interesting novel relation candidates. Evaluating thirteen methods on eight different semantic resources, including Gene Ontology, Food Ontology, Marine Ontology and similar, we demonstrated that structure-only link analysis could offer scalable anomaly detection for a subset of the data sets. Further, we demonstrated that by considering symbolic node embedding, explanations of the predictions (links) could be obtained, making this branch of methods potentially more valuable than the black-box only ones. To our knowledge, this is currently one of the most extensive systematic studies of the applicability of different types of link analysis methods across semantic resources from different domains.

1. Introduction

Researchers and companies often encounter tasks that, for their solution, need domain knowledge. This knowledge often contains complex relations between entities and human-defined terms that can be modelled using ontologies [10, 52]. Ontologies can range from small ones that describe people, their activities, and relations to other people (FOAF ontology [26]) up to large ones such as the Gene ontology that describes e.g., protein functions and cellular processes [3].

Ontologies have long been used to represent and reason over domain knowledge but have recently shown further potential in conjunction with machine learning methods. They have been used for relation prediction tasks [65, 12], much like graphs, or to expand features with background knowledge in other machine learning tasks. Commonly applied methods range from more traditional semantic similarity approaches [49, 65] to recently successful entity embedding algorithms, whether it be graph-based [27, 9, 40, 14], syntactic [56, 57], or hybrid [13]. Alternatively, ontologies have been utilized to constrain the output of machine learning and optimization models to conform to certain rules [55]. Machine learning has also been used to help experts with ontology construction tasks. *Ontology matching* refers to finding a semantic mapping between two inter operable ontologies [18], while *ontology completion* refers to finding missing, non-present links that are not logically deducible from the existing ontology but are plausible [13, 37].

While small ontologies can easily be annotated due to a limited number of possible relations, even highly skilled domain experts can make mistakes when considering larger ontologies, by missing some links or adding non-existent ones. These mistakes can have an impact on our understanding of the domain and can produce models and solutions that do not perform well. In our work, we aim to adopt machine learning methods that perform well on the link prediction task to find the missing and the redundant links in ontologies solely based on their structure. We do this by representing ontologies as graphs, and applying link prediction methods. The contributions of this work can be summarized as follows:

- A proposed methodology for finding missing and redundant links in real-life ontologies.

- Demonstrated utility of the considered link analysis methods on multiple ontologies with different properties.

- Simple to use software for prediction of missing and redundant links and evaluation of the proposed methodology on a given ontology.[1]

- Proposed temporal approach for evaluating the quality of the methods, investigating how the predicted links are accounted for in the future versions of the same ontology.

- An investigation of how link predictions can be *explained* via symbolic node embedding.

2. Related work

In this section, we first introduce ontologies and their use in machine learning, followed by the relevant work from the field of anomaly detection, machine learning on graphs and link prediction.

ORCID(s):

[1]Found at https://github.com/smeznar/link-analysis-meets-ontologies

2.1. Ontologies

Ontologies refer to machine-readable representations of knowledge in a given application domain, usually defined in a declarative knowledge modelling language, such as OWL (Web Ontology Language) [29] that is based on description logic (DL). Ontologies operate with individuals, classes (sets of individuals), and properties (relations between individuals), for which they define semantics through a set of logical statements – axioms.

These statements fall into two categories. The terminological box (also called the T-box, vocabulary, or schema) contains statements defining classes, their characteristics and hierarchy. In contrast, the assertional box (A-box) consists of assertions about individuals (concrete facts), which use the vocabulary of the T-box. Given a complete ontology, reasoners can infer additional implicit facts from the explicitly defined set based on rules defined in the T-box. For example, the A-box fact *"Mary is a mother"* implies *"Mary is a parent"* since the T-box defines that *"mother is a subclass of parent"*.

Some of the T-box statements in OWL are class subsumption axioms (*"mother SubclassOf: woman"*, meaning that every mother is a woman), property restrictions (*"parent EquivalentTo: HasChild Some person"*, meaning that everyone who has at least one person as a child is a parent) and set operators (*"parent EquivalentTo: mother Or father"*, meaning that a parent is either a mother or a father). A-box statements include membership axioms (*"Mary Types: mother"* meaning Mary is a mother) and property assertions (*John Facts: HasWife Mary*, meaning that Mary is John's wife). These examples are taken from W3C's OWL primer [29] and are presented in the Manchester syntax, a notation that will also be used in further examples.

Ontologies are used in various fields and vary significantly in their purpose, content, and implementation. However, they correspond to one of the two archetypes outlined below.

- The first refers to small, ungrounded ontologies that lack an A-box and serve as a semantic schema of high-level terms (classes) in a particular domain. They are often used in the scope of the semantic web and are intended to be referenced by various web sources and thereby populated with "external" facts. Examples include FOAF (12 classes) and Marine ontology (104 classes). In the former, the classes represent agents, documents, and other entities on the web, while the latter describes high-level terms relating to marine biology (see Section 4.1).

- Conversely, there are larger, more grounded ontologies attempting to comprehensively capture knowledge in a domain as a complex hierarchy with many concrete facts. An illustrative example is the Gene ontology [3] with (at the time of writing this paper) 62,201 classes representing gene products and their functions.

Notably, such ontologies, or at least their graph representations, could be considered knowledge graphs by some definitions that define the latter as a schema (T-box) accompanied by a large number of (A-box) facts [47, 8]. However, knowledge graphs do not have a well-established definition yet, with other work giving alternate proposals, such as a collection of facts without the schema [31], or a system encompassing an ontology and a reasoning engine [22].

In practice, different grounded ontologies take different approaches to capturing A-box (ground level) facts in OWL. Some, such as HeLiS [21], use individuals and OWL's object property assertion axioms to define that two "ground level entities" are related by a property. Others, such as the Gene ontology and Food ontology [20], do not to use individuals at all, representing even very "individual-like" entities as classes and thus blurring the line between T-box and A-box or between an ontology and a knowledge graph (KG). Instead, such ontologies define the equivalent of a property assertion with a construct like the following (from FoodOn):

pecan pie SubclassOf: HasIngredient Some sugar

Directly, this expresses that *"pecan pie is a subclass of the class of all dishes, which have some sort of a sugar as an ingredient"*, but is intended to be read as *"pecan pie has sugar as an ingredient"*.

2.2. Mining ontologies with network-based approaches

As described, ontologies have recently shown potential when used with machine learning methods, such as to provide additional background information or to constrain the learning process, but they can also be seen as standalone resources, much like knowledge graphs. In our work, we attempt to systematically approach mining ontologies with graph-based methods to identify anomalies and potentially novel relations.

The related work comprises of several papers that present ontology-specific embedding algorithms. Onto2Vec [56] and OPA2Vec [57] constructs sentences from OWL axioms and trains a language model, On2Vec [14] is based on translational graph embeddings and OWL2Vec* [13] combines the language model approach with random walks on the ontology graph. These methods are typically evaluated against knowledge graph embeddings on a limited number of large ontologies in a relation prediction task.

Other examples are some approaches to semantic data mining [34] and domain-specific applications, most often in the biomedical domain, where ontologies are mined for tasks such as protein-protein interaction (PPI) prediction and gene function prediction, gene-disease prediction. Here several ontologies are usually combined into a single data set. As for methods, semantic similarity approaches [65, 66], often heavily tailored to the task, are the traditional choice, but many recent works adopt ontology-specific embeddings [45, 12, 2].

None of the above could be considered a systematic study. Perhaps the closest work to ours and a valuable resource about the topic is a recent survey on the state of machine

learning with ontologies [36] that covers both traditional semantic similarity methods and recent embedding-based methods. It looks at simple graph embeddings, knowledge graph embeddings and ontology-specific embeddings, categorizing them into graph-based, syntactic and semantic approaches. The study also includes an experimental evaluation of a subset of these methods in a protein-protein interaction task. However, it only considers two subsets of GO as data sets, since its focus is on theoretic categorization of the field and on the biomedical domain in particular.

Comparatively, some key characteristics of our study are:

- We evaluate a large number of graph-based methods and compare simple graph embeddings to KG-specific embeddings. However, we limit outselves to structure-only methods due to the nature of the ontology-to-graph conversion.

- We test our methodology on substantially different ontologies both in terms of the domain they cover and their size, ranging from small schemas to large knowledge bases of ground level facts.

- We propose a temporal approach for evaluating the quality of the methods.

A more extensive overview of related work is summarized in Table 1.

2.3. Graph-based machine learning

Graph-based machine learning has seen a rise in popularity in recent years due to its potential to work with complex data structures such as relational databases and structures commonly found in biology and chemistry [15]. This branch of machine learning mainly focuses on node and graph classification [7], node clustering, and link prediction tasks [39].

Machine learning tasks on graphs are usually solved in one of three ways. Traditionally, tasks on graphs were solved using label propagation [63], PageRank [46], and proximity-based measures such as Adamic/Adar [1], Jaccard coefficient [54], and preferential attachment. Another group of approaches embed graphs into tabular data which is used together with traditional machine learning methods such as logistic regression to generate predictions. These approaches include well-established and tested methods such as node2vec [27] and Deepwalk [48], as well as some new ones such as SNoRe [41]. Recently, with new research in deep learning approaches, neural network models such as graph convolutional networks (GCN) [33], and graph attention networks (GAT) [62] have emerged as end-to-end learners.

2.4. Anomaly detection

Anomaly detection studies patterns in data and searches for instances that do not conform to them. These instances usually represent anomalies, cases that need further examination to stop malicious attempts or give additional information about the observed environment. Anomaly detection can be found in many different domains such as intrusion detection [17], fraud detection [24], medical anomaly detection [30], and image protection [50].

Detection methods usually depend on the domain of focus and its data. The concept of an anomaly is not well defined, so what might be considered an anomaly in the medical field might not be considered an anomaly in fraud detection due to different nature of the data. Broadly, anomalies can be split into point anomalies, contextual anomalies, and collective anomalies. Point anomalies are those where each instance can be considered an anomaly regarding other data. Contextual anomalies are the ones considered anomalous in some context but might not be otherwise. An example of a contextual anomaly is the high temperature during winter. Lastly, collective anomalies are a collection of instances that are anomalous in regards to other data, i.e. one second of readings during a electrocardiogram scan [11].

Anomalies can be found using different approaches. One approach is to use classifiers that output whether an instance is anomalous or not. The selection of the classifier does not matter much, but neural networks, Bayesian networks, and support vector machines have been mostly used in recent years. Other approaches include statistical and information theoretic-based detection, nearest-neighbour based detection, where we assume anomalies lie far from their neighbours, clustering-based detection, where we assume anomalies lie outside clusters, while non-anomalous data lie inside. The last approach is spectral-based detection and embeds instances into latent space where anomalies differ significantly from non-anomalous data.

The proposed approach is closely related to contextual anomaly detection and link prediction tasks. The distribution is, in our case, the link structure of the ontology; an anomaly is a connection that exists (or is missing) but should not, based on the structure of the ontology. The contextual part of the approach comes from the fact that some connections might exist in one context (one domain) but not in a different one. Most algorithms we use first create embeddings of a network's nodes, subsequently used in classification. These algorithms closely resemble those that are used in spectral-based anomaly detection. This methodology is explained in more detail in Section 3.3.

2.5. Link prediction

Link prediction is one of the most widely addressed tasks concerning network-based data. Predicting whether there exists an edge between two nodes without any additional information is almost impossible, but with some additional information about the network, the nodes, and with some assumptions, various approaches predict such existence well [38, 39].

The most common assumption used in link prediction is that two nodes are connected if they are similar. This similarity might be due to them sharing similar node features or having common neighbours (or, more commonly, neighbourhoods). When applied to real-life networks, this assumption is very reasonable since, for example, two individuals who went to the same school, lived in the same city,

Table 1

Overview of some of the related approaches, their aim and a short description. The horizontal lines separate different types of research articles.

Article Title	Article type	Considered data	Method type	Description
Onto2Vec: joint vector-based representation of biological entities and their ontology-based annotations [56]	method	Gene ontology	syntactic ontology embedding	Represents ontology axioms as sentences and trains a word2vec model to generate embeddings. Evaluated on a PPI task.
OPA2Vec: Combining formal and informal content of biomedical ontologies to improve similarity-based prediction [57]	method	PhenomeNET and Gene ontology	syntactic ontology embedding	Extends Onto2Vec by including informal information such as class descriptions in the axiom sentences.
On2Vec : Embedding-based Relation Prediction for Ontology Population [14]	method	Yago, ConceptNet and DBPedia ontology	graph-based ontology embedding	Adapts translational KG embeddings for ontologies by accounting for hierarchical relations. Evaluated on a relation prediction task.
OWL2Vec*: Embedding of OWL Ontologies [13]	method	HeLiS, FoodOn, and Gene ontology	hybrid ontology embedding	Combines concepts from OPA2Vec with biased random walks on the ontology graph.
Predicting Gene-Disease Associations with Knowledge Graph Embeddings over Multiple Ontologies [45]	application	Several biomedical ontologies	various KG and ontology embeddings	Combines several biomedical ontologies along with annotations and applies several embeddings for the task of gene-disease prediction.
Predicting Candidate Genes From Phenotypes, Functions, And Anatomical Site Of Expression [12]	application and a method	Several biomedical ontologies	various ontology embeddings	Combines data from several biomedical ontologies. Presents a novel domain-specific embedding model and evaluates it against existing ontology embeddings on a gene-disease prediction task.
Ontology-based prediction of cancer driver genes [2]	application	Several biomedical ontologies	syntactic ontology embeddings	Combines data from several biomedical ontologies, generates features with OPA2Vec and trains a model to predict cancer genes.
Gene function prediction based on Gene Ontology Hierarchy Preserving Hashing [66]	application and a method	Gene ontology	novel semantic similarity method	Gene ontology terms are represented by a hierarchy-preserving hash function before computing semantic similarity for gene-function prediction.
Protein–protein interaction inference based on semantic similarity of Gene Ontology terms [65]	application	Gene ontology	various semantic similarity methods	Integrates multiple semantic similarity measures to improve PPI prediction on the Gene ontology.
Semantic similarity and machine learning with ontologies [36]	survey	Gene ontology	various SSM, simple graph embeddings, KG embeddings and ontology embeddings	Survey on ML with ontologies. Covers both traditional SSM and recent ontology embedding methods.

and had similar hobbies are more likely to know each other and thus be friends on social media networks. Another assumption that is commonly used is that nodes are likely to be connected to nodes with a high number of neighbours. This also mirrors real-life networks as, for example, a paper with many citations is more visible and thus more likely to be cited by a new paper than a paper only cited once.

Link prediction is traditionally solved using proximity-based methods that model the networks using the assumptions mentioned above. These methods most commonly predict the links based on the first and second neighbours of their nodes, e.g., the number of common neighbours. These include the Jaccard index [54], Adamic/Adar index [1], and others.

With the rise of graph-based machine learning, new methods were developed that usually perform very well on all networks, even the ones where the discussed (structure) assumptions do not hold. These methods mostly embed nodes into a dense low dimensional, or sparse representation that maintains the network's structure. The most common such embedding method is node2vec [27], which uses random walks as sentences for training the skip-gram model [42].

Another approach, SNoRe [41], creates a sparse embedding where a node is represented as a vector of similarities between the hashed neighbourhood of the node and the neighbourhoods of nodes selected as features.

Similar to embedding methods, graph neural networks have recently been used for different machine learning tasks on real-life networks These approaches jointly exploit the adjacency matrix of a network alongside node features. The most popular graph neural network approaches include graph convolutional networks (GCN) [33], graph attention networks (GAT) [62] and graph isomorphism networks (GIN) [64].

For *knowledge graphs*, i.e., graphs, where nodes and edges usually contain some additional information (mostly types), specialized approaches can be used. One such approach is metapath2vec [19] that works similarly to node2vec, but the sampled random walks have a predetermined structure (meta paths). Other approaches such as TransE [9] and RotatE [60] embed nodes and relations in such a way that a combination of their embeddings creates a vector that has a norm close to zero if the triplet (subject, predicate, object) is inside the graph and close to one if it is not.

3. Methodology

In this research, we aim to represent ontologies as graphs and employ link prediction methods to find missing and redundant connections. The proposed methodology consists of the following steps: ontology to graph transformation, link prediction, and finding of missing and redundant connections described below.

3.1. Ontology to graph transformation

As mentioned in Section 1, machine learning has approached the use of ontologies in various ways, such as treating them as documents [57, 56]. One common approach is to represent ontologies as graphs where nodes represent classes or individuals, and links encode semantic relationships defined by the ontology. This addresses some limitations of existing non-graph methods [13, 36], but more importantly, it enables the use of many powerful graph-based machine learning methods that are being developed for other problem domains and are rapidly evolving.

Since an ontology can be understood as a set of logical expressions and is usually modelled as such, there exist multiple possible conversions of a given ontology into a graph. Certain expressions, like property assertions between individuals (facts) and class taxonomy (*"subclass of"* relations) directly map to nodes and edges. Others, like property restrictions, domain-range axioms, and set operators, do not have an obvious representation. Given our aim to learn about ontologies using graph-based methods, the conversion needs to be such that semantics expressed with OWL axioms are sufficiently reflected in the resulting graph's topology.

A number of different approaches to converting OWL ontologies to graphs have been developed for various tasks. However, there is not yet an established standard or agreement on what the most appropriate representation is. W3C

provides a specification for translating the OWL syntax directly to a heterogeneous graph represented by RDF triples [16]. Conversely, most other methods attempt to approximate semantic relations between entities by introducing new links that are somehow justified by the ontology's axioms. This branch of methods includes Onto2Graph [53], OBO Graphs [58] and so-called projection rules [59, 13].

We consider two of these approaches: the OWL to RDF-triplet mapping following the W3C specification and conversion by projection rules. We choose the former because it is a W3C standard and the latter because it has been used in a similar machine learning context. However, we note that these conversion algorithms could be substituted with others without significant changes to the rest of our methodology.

- The first approach [16] exactly reproduces the OWL file in a graph (RDF triple) form, including all axioms without any reductions. Simple relations, such as (*"pecan pie SubclassOf: food product"*), are transformed directly. Complex expressions, like property restrictions, are translated by introducing so-called *blank nodes*. For example, the previously mentioned FoodOn construct *"pecan pie SubclassOf: HasIngredient some sugar"* is transformed into 4 triples: $\langle pecan\ pie,\ SubclassOf,\ x\rangle$, $\langle x,\ SomeValuesFrom,\ sugar\rangle$, $\langle x,\ type,\ Restriction\rangle$, $\langle x,\ OnProperty,\ HasIngredient\rangle$. In theory, this means all the expressiveness is kept, and a reasoner can traverse such a graph, inferring new facts. However, in a machine learning setting, where such rigour is not necessary, all the intermediate syntactic nodes and edges might only act as noise when learning associations among the entities of interest.

- The second strategy is based on so-called projection rules proposed in [59] and has been used for machine learning with ontologies [13], producing good results. Here, not all exact logical relationships are kept. Simple relations like class subsumption and property assertions between individuals are again transformed directly. However, complex logical expressions, like property restrictions, are approximated with simple triples. The same example *"pecan pie SubclassOf: HasIngredient some sugar"* becomes the intuitive triple $\langle pecan\ pie,\ HasIngredient,\ sugar\rangle$, meaning *pecan pie* and *sugar* are directly connected by an edge, without any intermediate nodes. The result is a graph that presumably at least approximately captures all relationships but doesn't contain noisy syntactic structures.

Since our work focuses on methods that only consider graph structure, data properties (like *"Mary has age 35"*) and other lexical information like labels and descriptions of entities (annotation properties) are discarded in the conversion.

As described, both conversion methods produce a directed heterogeneous multigraph, i.e. a set of triples (edges) of the form $\langle s, p, o\rangle \in T$, where s and o are nodes (classes, individuals, or blank nodes) and p is a label, representing the relation between them. This is the input to our chosen

knowledge graph embedding methods that operate on heterogeneous graphs. For other methods, we further convert this graph into an undirected simple graph $G(N, E)$ so that $\{o, s\} \in E \Leftrightarrow \exists p : \langle o, p, s \rangle \in T$, meaning two nodes are at most connected by a single undirected anonymous edge.

Previous work has taken different approaches, like including all available lexical information along with graph topology and training language models to produce embeddings [13].

In our experiments, we found conversion by projection rules to outperform the OWL to RDF mapping across the board. We, therefore, choose to only present results using graphs obtained with conversion by projection rules (see Section 5.1).

3.2. Link prediction benchmark

Link prediction and anomaly detection tasks are closely related since the anomaly detection model must be able to reconstruct the graph to be able to predict if an edge (connection) between two nodes is an anomaly or not. Because of this, it is crucial to determine how well our model works on the link prediction task. High accuracy on the link prediction task means that the model will be able to reconstruct the graph well and thus accurately predict which edges (connections) are missing or redundant. In our work, we use the following methodology to test how well our methods perform on the link prediction task. First, we transform an ontology into a graph as described in Section 3.1.

After this, we create positive (existent) and negative (non-existent) examples. We shuffle and split them into five folds. We then use the edges from four out of five folds to create the adjacency matrix used in training.

For each fold, we then train the baseline models using the adjacency matrix generated from the other four folds and the corresponding positive and negative edges, if they are needed as input. We use these models to predict the existence of the positive and negative edges in this fold. We evaluate the performance using the ROC-AUC and average precision. An overview of the link prediction process can be seen in Figure 1.

3.3. Finding missing and redundant edges

Annotations of data are not perfect, and often an annotator might miss some relations in the ontology, or sufficient experiments might not have been conducted to determine if some relation exists. Because of this, methods for finding and recommending missing or redundant edges help improve the ontology. We create recommendations for such edges in the following way.

We first transform the ontology as presented in Section 3.1. Then we embed the generated network into matrix R using SNoRe with the default parameters, i.e. 1024 walks of maximum length 5 for each node, inclusion threshold 0.005, cosine similarity as the distance metric, and less than $|N| \cdot 256$ non-zero values. Using R we create the link prediction matrix $L = R \cdot R^T$ that is used to find candidates for the missing and redundant connections. We split the link prediction matrix into two matrices, one that represents the score of existing edges and one that represents the score of non-existing edges. The matrix for the scores of existing edges can be obtained by using the adjacency matrix as the mask, while the matrix with scores for the non-existing edges can be obtained by subtracting the matrix with scores of existing edges from the link prediction matrix L. Recommendations for non-existing edges are obtained by selecting the elements in the matrix with scores of the non-existing edges with the highest scores. Similarly, recommendations for existing edges are obtained by selecting elements with the lowest score in the matrix with scores of existing edges. An overview of this methodology is shown in Figure 2. In the figure, the graph in the top left represents the K_n graph weighted by the values in matrix L. The edge scores are then split into those belonging to existent edges and non-existent edges. We chose the non-existent edges with the highest score (green) as candidates for missing edges and existent edges with the lowest score (red) as candidates for redundant edges.

Because the space complexity of the link prediction matrix is quadratic, such an approach might not be feasible for ontologies with many entities. One way to avoid this is to create predictions only for a small number of nodes $P \subset V$. This can be done by creating a prediction matrix $L[P] = R[P] \cdot R^T$, where $A[B]$ represents rows of nodes from B in matrix A. To obtain recommendations, we use the same technique as before, the only difference being that we use only the mask for the selected nodes. For the approaches that do not generate an embedding, this is done by only generating scores for the selected pairs of nodes (a subset).

We benchmark our approach for the missing and redundant edges using a temporal approach. We take different versions of one ontology, order them by their release date, and evaluate our methodology on them using the following approach shown in Figure 3. We take two subsequent ontologies, one published at time t and the other at time $t + 1$, and transform them. Then we create candidates for missing and redundant edges on the ontology published at time t and test whether these candidates occur in the ontology published at time $t + 1$. If a candidate for the missing edge occurs in the ontology published at time $t + 1$ we classify the candidate as correct, otherwise as incorrect. Similarly, we classify the candidate for the redundant edge as correct if the edge does occur in ontology published at time t but not in the one published at time $t + 1$. Otherwise, the candidate is classified as incorrect. We test different numbers of top k candidates and get the score (accuracy) for two subsequent years as the number of correctly classified candidates is divided by the total number of candidates.

3.4. Explaining recommendations

In Section 3.3 we create recommendations for missing and redundant links using SNoRe. Besides good performance, SNoRe creates an embedding with symbolic features, which make the embedding and thus the recommendations easier to interpret. The following sections present our methodology to create both local interpretations (for a single recom-
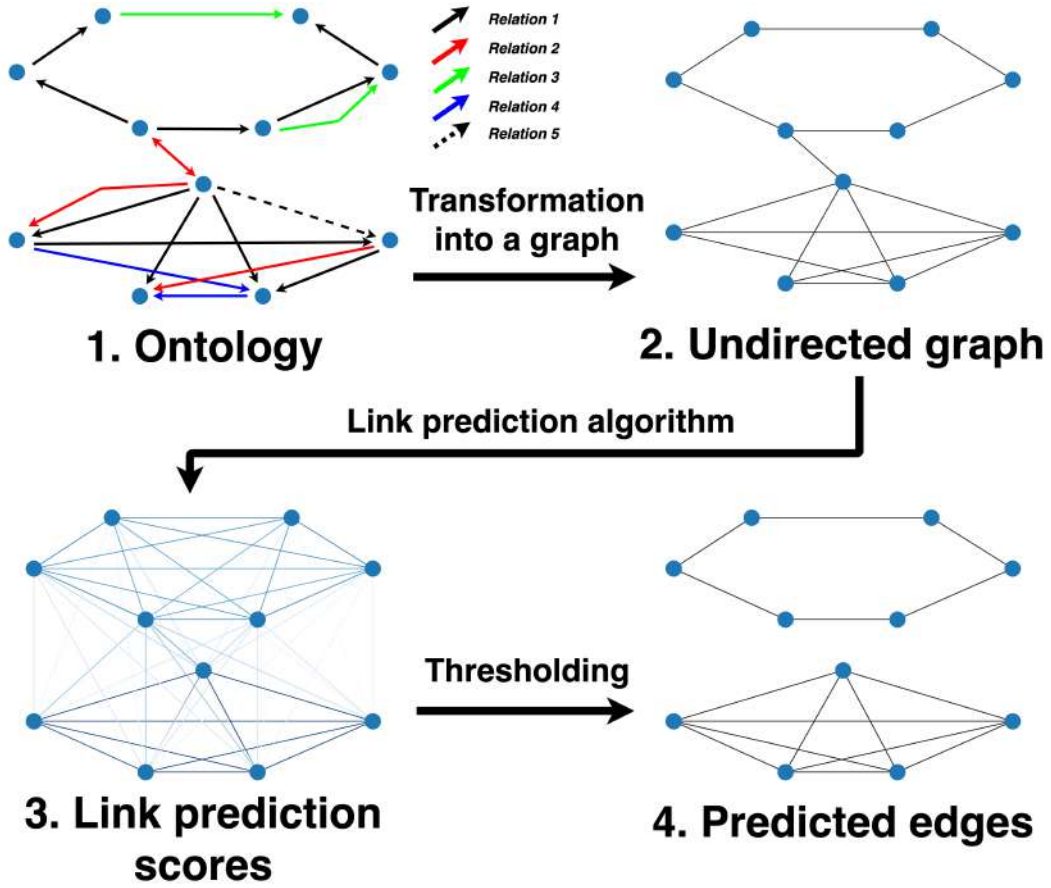
Figure 1: Overview of the link prediction methodology. We start with an ontology in stage 1 and transform it into an undirected graph to get to stage 2. Using link prediction algorithms, we get to stage 3, the link prediction scores. Lastly, we threshold these scores to get the predicted edges (stage 4).

mendation) as well as global ones (which features contribute most to the connectivity of an ontology).

3.4.1. Global interpretation

Using global interpretations, we can further explore the domain and how much the selected (symbolic) features influence the existence of a link between two nodes. The importance of a global interpretation can be given with the toy example where nodes represent pages (people and interests) on a social network and links represent the connection between two pages (friendship or a follow). Let us say that the similarity to the neighbourhood of a node representing a famous person and a local high school are selected as the relevant features. Since two people are more likely to know each other if they went to the same high school, the global interpretation will tell us that the feature representing the similarity to the neighbourhood of the local high school node is more important when determining the friendship between two people.

We create a global interpretation using logistic regression in the following way. First we create the node embedding using SNoRe and select edges that will be used as positive and negative examples during the training (similarly as

in the link prediction benchmark 3.2). After this we create training data by multiplying the embeddings of nodes incident to the selected edges element-wise. We use this data to train the logistic regression model. The importance of features can be estimated by the absolute value of its t-statistic [43]. The t-statistic is calculated as the weight of the feature divided by its standard error. For j-th feature with the weight β_j, the t-statistic is calculated using the formula

$$\text{SE}(\beta_j) = \sqrt{(\boldsymbol{X}^T \boldsymbol{W} \boldsymbol{X})_{jj}^{-1}},$$

where $\boldsymbol{X}$ represents the matrix with training data used as input to the logistic regression, $\boldsymbol{W}$ is a diagonal matrix where

$$\boldsymbol{W}_{jj} = \frac{e^{\sum_{i=1}^p \beta_i \boldsymbol{X}_{ji}}}{(1 + e^{\sum_{i=1}^p \beta_i \boldsymbol{X}_{ji}})^2},$$

and p the number of features in the embedding.

3.4.2. Local interpretation

Usually, we are more interested in the local interpretations of the recommendations. These give us an insight into what contributed to the prediction of an edge between two

Link prediction scores
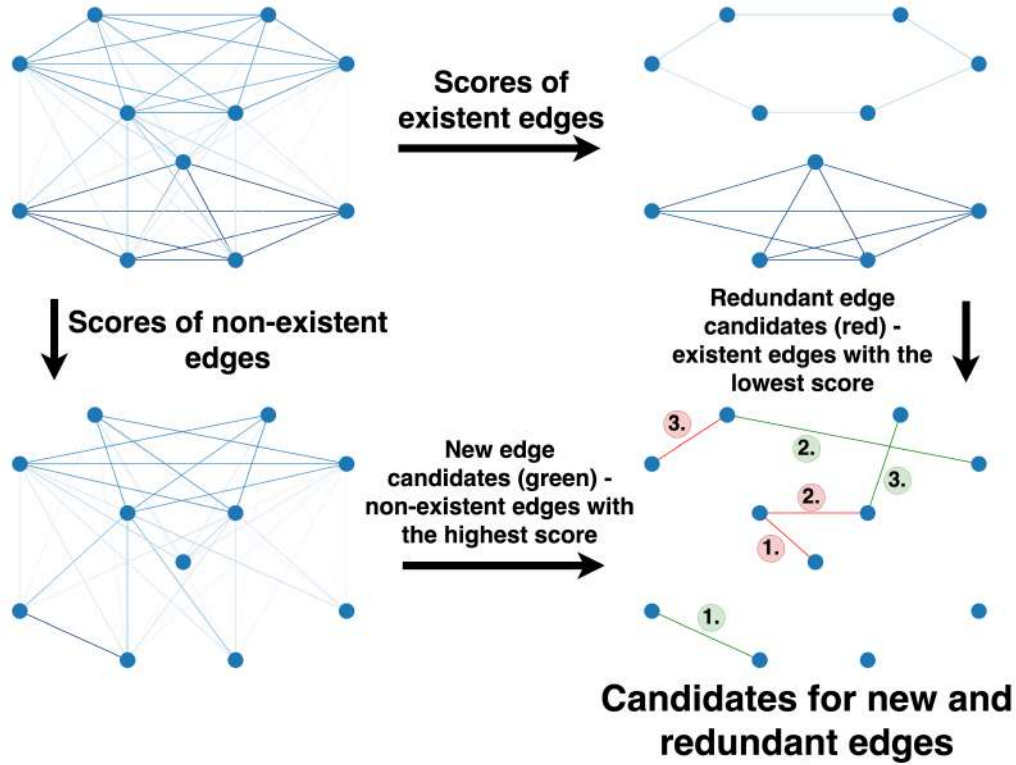
Figure 2: Overview of our methodology for finding missing and redundant edges. We start by splitting the link prediction scores into scores of the existent edges and those of non-existent edges. We take the non-existent edges with the highest score (candidates for missing edges) and existent edges with the lowest score (candidates for redundant edges) as our candidates.

nodes. Let us show an example of such interpretation with the same toy scenario as for the global interpretation. In this scenario, one user might be recommended to another one due to the high value of the local high school feature, but another one due to the high value of the feature representing the similarity to the neighbourhood of the famous person node.

We create a local interpretation by multiplying the embeddings of two nodes incident to the selected edge element-wise. We can interpret the recommendation by sorting features based on their value and looking at which features contributed the most to the confidence score for that edge. Further, if we use a classifier such as logistic regression, features can be weighted using the parameters of that model.

4. Experimental setting

In this section, we first present the considered data sets (ontologies), followed by description of the baselines and the evaluation procedures.

4.1. Data sets

In our work, we tested ontologies of different sizes and with different properties. These ontologies are the following:

Marine TLO [61] A small top-level ontology of concepts related to biodiversity data in the marine domain. It is intended to help integrate new information about marine species (linked data) by providing a hierarchy of generic classes like *legislative zone* or *ecosystem*.

Anatomy ontology (AEO) [5] A high-level vocabulary of anatomical structures common across species. It aims to enable interoperability between different anatomy ontologies (such as EHDAA2) and describes anatomical entities such as *artery*, *bone* or *mucous membrane*.

SCTO [23] Captures upper-level terms from the Systematized Nomenclature of Medicine (SNOMED CT), a comprehensive medical terminology used to manage electronic health data, as an ontology. It is just a taxonomy (only *subclass of* relations), with diverse classes such as *symptom*, *laboratory test* or *anatomical structure*.

Emotion ontology (MFOEM) [28] It aims to describe affective phenomena (emotions and moods), their different building blocks, and their effects on human behaviour (expressions). Similar to the Anatomy ontology, this ontology includes more numerous and more
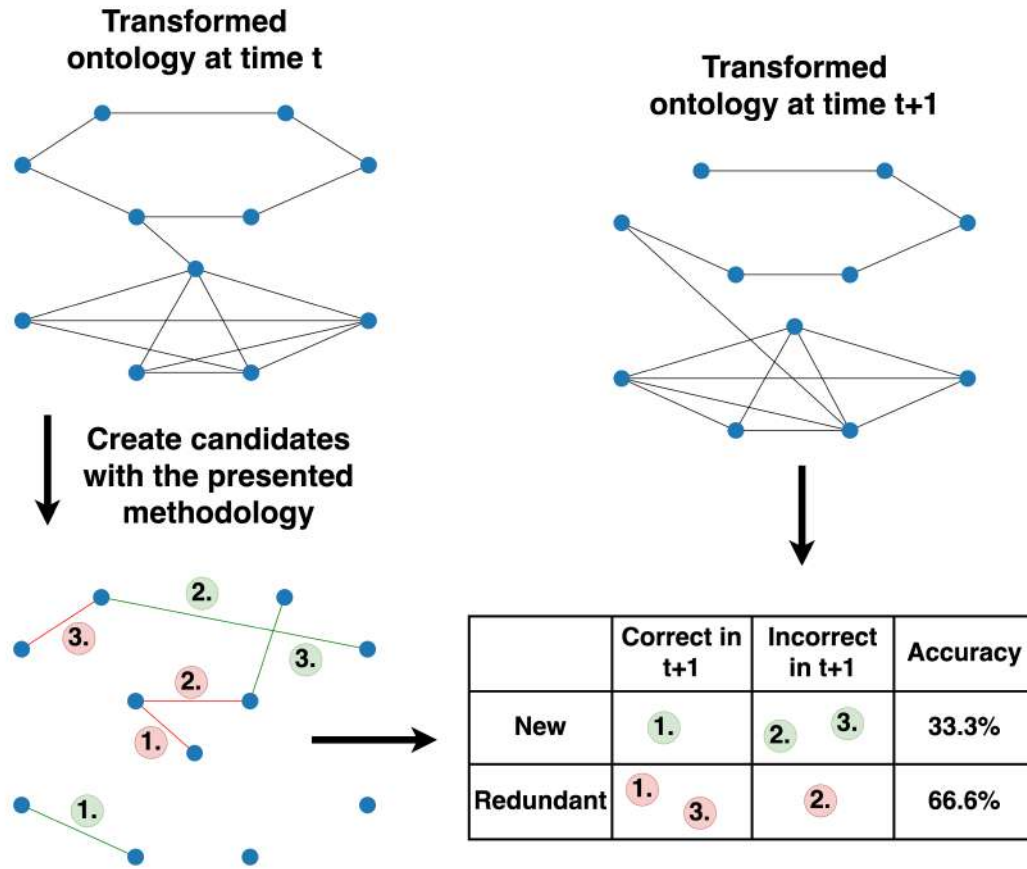
Figure 3: Overview of our temporal-based benchmark. We start with two transformed ontologies published at time t and $t+1$. We create candidates for missing and redundant edges on the ontology published at time t and test them on the ontology published at time $t+1$. The final score is obtained by dividing the number of correctly classified candidates by the total number of candidates.

specific terms than FOAF or SCTO, but is not as grounded as, for example, the Gene Ontology. It models classes such as *anxiety*, *negative valence* or *blushing* and properties such as *has occurrent part*.

Human Developmental Anatomy v2 (EHDAA2) [6] An ontology that is primarily structured around the parts of organ systems and their development in the first 49 days (Carnegie stages (CS)1–20). It includes more than 2000 anatomical entities (AEs) and aims to include as much information about human developmental anatomy as is practical and as is available in the literature.

Food ontology (FOODON) [20] It aims to name all parts of animals, plants, and fungi that can bear a food role for humans, as well as derived food products and the processes used to make them. It is a large, fairly grounded ontology with upper-level entities like *part of organism* and leaf classes as specific as *Pinot noir wine* or *chickpea*. Some properties include *has ingredient*, *derives from* and *has quality*.

LKN [51] A biological *knowledge graph* constructed from multiple different sources of information, including temporal expression data, small RNA-based interactions and protein-protein interactions. This source was obtained in the process of semi-automatic curation.

Gene ontology (GO) [3] A comprehensive source of information on cellular processes. It describes three types of entities - molecular functions, cellular components and biological processes - and their relations in a complex class hierarchy linked mostly by *is a* (subsumption), *part of* (meronymy) and *regulates* properties. Among the used ontologies, GO is the largest and most grounded with entities (classes) ranging from *molecular function* to, for example, *DNA alpha-glucosyl transferase activity*.

Some basic statistics of these ontologies are shown in Tables 2 and 3. We have four smaller ontologies, one medium sized, and three bigger ones. Of the three bigger ontologies, the Food ontology has a very tree-like structure, while LKN and gene ontologies are more connected. We also ran more extensive and automated experiments on different versions

Ontology	|N|	|E|	Components
Marine [61]	108	156	2
Anatomy [5]	249	366	1
SCTO [23]	321	370	1
Emotions [28]	631	773	1
EHDAA2 [6]	2743	12894	15
Food [20]	28740	35897	107
LKN [51]	20011	68503	2427
GO [3]	44167	101504	1

Table 2
Basic statistics of the tested ontologies' graph forms (by projection rules), where |N| denotes the number of nodes and |E| the number of edges.

of the Gene ontologies between the years 2015 and 2021.

4.2. Baselines

We test our approach with the following baselines:

Adamic/Adar [1] An edge between nodes u and v is scored with the formula $\sum_{x \in N(u) \cap N(v)} \frac{1}{\log |N(x)|}$ where $N(x)$ is the neighborhood of node x. These scores are normalized and thresholded to obtain link prediction.

Jaccard coefficient [54] An edge between nodes u and v is scored with the formula $\frac{|N(u) \cap N(v)|}{|N(u) \cup N(v)|}$, where $N(u)$ is the neighborhood of node u. These scores are normalized and thresholded to obtain link prediction.

Preferential attachment [4] An edge between nodes u and v is scored with the formula $|N(u)| \cdot |N(v)|$, where $N(u)$ is the neighborhood of node u. These scores are normalized and thresholded to obtain link prediction.

GAE [32] Generates a node representation with a variational graph auto-encoder that uses latent variables to learn an interpretable model.

GAT [62] Includes the attention mechanism that helps learn the importance of neighboring nodes. In our tests, we adapt the implementation from PyTorch Geometric [25].

GCN [33] A method that introduced convolution to graph neural networks and revolutionized the field. This approach aggregates feature information from the node's neighborhood. In our tests, we adapt the implementation from PyTorch Geometric [25].

GIN [64] Learns a representation that can provably achieve the maximum discriminative power. In our tests, we adapt the implementation from PyTorch Geometric [25].

SNoRe [41] A node embedding algorithm that produces an interpretable embedding by calculating the similarity between vectors generated by hashing random walks.

node2vec [27] A node embedding algorithm that learns a low dimensional representation of nodes that maximizes the likelihood of neighborhood preservation using random walks.

metapath2vec [19] A node embedding algorithm that learns a low dimensional representation of nodes. The algorithm works similarly to node2vec, but samples random walks based on predetermined metapaths.

TransE [9] creates a knowledge graph embedding in such way that the distance between the embedding of the second node and the embedding of the first node translated by the embedding of the relation is small.

RotatE [60] creates a knowledge graph embedding. This approach is similar to TransE, but instead of translating the embedding of the first node by the embedding of the relation, it rotates the embedding in complex vector space.

Spectral clustering [44] Generates a node embedding by using a non-linear dimensionality reduction method based on spectral decomposition of the graph Laplacian matrix.

4.3. Evaluation

We evaluate the link prediction capabilities on transformed ontologies by using five-fold cross-validation. We create these folds as follows. We start with a directed (multi)graph with multiple edges between each pair of nodes. We transform this graph into a simple undirected graph and remove elements on the diagonal of the adjacency matrix (self-loops). Afterwards, we take the upper triangle of the adjacency matrix, put the elements into an array, and shuffle them to create positive examples. This is crucial for fair evaluation since each edge is chosen exactly once and thus contained inside exactly one fold. For negative examples, we randomly sample pairs of nodes, test if the edge between them exists, discard them in this case, and make sure they do not repeat. We use the same amount of positive and negative examples. We split positive and negative examples into five equally sized parts (last being a few examples shorter if the number of edges is not divisible by five).

We get the score of a baseline on the selected data set by taking the mean value of scores for each fold. A fold is scored by training the model with data from other folds and using either ROC-AUC or average precision to obtain prediction scores for edges in this fold.

To predict the edge score for the TransE and RotatE baseline, we also need to input the relation we want to predict. To bypass this, we generate predictions for each relation and output the most probable one. We do this because if there is no edge between two nodes, all predictions should have a low score, and otherwise, at least one should have a high score (the one we select).

We run our tests on a machine with 64GB of RAM and 12 threads. To make our experiments reproducible, we initialize the random number generators of the data splitting algorithm and all baselines with a predetermined seed. This way, data splits are the same for each baseline.

Ontology	Classes	Individuals	Object properties	Subsumption axioms	Restrictions	Set axioms
Marine	104	3	92	105	0	0
Anatomy	250	0	11	366	101	0
SCTO	394	18	8	341	251	111
Emotions	688	36	29	774	94	40
EHDAA2	2734	0	9	13366	10283	0
Food	45942	381	68	39155	8860	2543
GO	62201	0	9	90583	30704	23493

Table 3
Basic statistics of the tested OWL ontologies. Set axioms encompass class equivalence, disjointness, union and intersection. Imported ontologies are considered part of the importing ontology — this applies to Emotion, Food and Gene ontology.

5. Results

In this section we present the results of link prediction, results of temporal approach presented in section 3.3, and show examples of explanations for the generated recommendations.

5.1. Link prediction results

The results of link prediction using the ROC-AUC metric using the methodology described in Section 3.2 are presented in Table 4. From the results we can observe two aspects that generally hold for all baselines: the variance of results falls with the number of edges, and that baselines perform significantly worse on ontologies where the ratio between nodes and edges is close to one. We can also observe that on smaller ontologies embedding methods that do not rely on random walks such as spectral embedding, TransE, and RotatE work best, while on bigger ones SNoRe and TransE generally outperform the others. By grouping baselines of similar kinds together we see that proximity-based approaches usually give mediocre performance, graph neural networks work well on most data sets but usually fall just below the best performing approaches, node embedding algorithms based on random walks generally perform great on all data sets but are near the top on larger ones, approaches designed for knowledge graphs perform similarly to other embedding methods, and spectral embedding generally performs better on smaller ontologies, the exceptions being Marine and LKN. Overall, the best performing baselines are SNoRe and TransE.

Similar results can be observed when the average precision metric is used. These results are presented in Table 5. We can see that the baselines perform better on bigger graphs where the ratio between edges and nodes is not close to one. The biggest difference in performance using this metric can be observed on the GAE baseline that achieves the best score on the SCTO and EHDAA2 ontologies and is close the the best one on a few others. Another notable result is the performance of the baseline RotatE on Anatomy, when the baseline performs the worse with the ROC-AUC metric, but the best when average precision is used.

Lastly, Figure 4 shows the average time needed for link prediction on a given ontology. We can see that the running time of smaller ontologies is less than a second for (almost) any given baseline. On the three bigger ontologies, proximity-based methods are the fastest even though they sometimes achieve great results, see for example the Preferential attachment on LKN. Overall, the slowest method is RotatE that needs almost twice as much time on the Gene ontology than TransE that is the second slowest. We can see that graph neural network methods perform similarly or a bit slower than SNoRe, while achieving lower results overall.

These results show that link prediction performance and thus also the presented anomaly detection approach works best on ontologies that have enough nodes and more importantly edges.

5.2. Detection of missing and redundant edges

We tested our methodology for finding missing and redundant edges using the temporal approach described in Section 3.3 on the Gene ontology. We did this by selecting seven ontologies published between 2015 and 2021. The number of edges that were added and removed in each year is shown in Table 5.2. We can see that on average there are around 8400 edges added and 7500 edges removed each year.

We generated a different number of candidates (10, 100, 500) for both missing and redundant edges and tested how many of them were added to or removed from the ontology of the following year.

The accuracy of top-k missing candidates added in the following year is shown in Table 7. We can see that the candidates with a high score are more likely to be added in the following year. This is best shown on predictions generated on the ontology from 2019, where 3 of the top 10 generated candidates were added, but then only 5 more in the next 90 and 18 in the predictions in the range $100 - 500$.

The accuracy of top-k redundant candidates added in the following year is shown in Table 8. These results are a bit different, mainly for $k = 10$ where only one predicted edge gets removed in the following year. Results for $k = 100$ and $k = 500$ are very similar to those of the missing edges.

5.3. Interpretation examples

To further make our predictions useful, we interpret them using the methodology presented in section 3.4.

Figure 5 shows the global interpretation of recommendations for the 2019 gene ontology. In the figure, we show the ten features whose parameters have the highest values. We

| Dataset Baseline	Marine	Anatomy	SCTO	Emotions	EHDAA2	FoodOn	LKN	GO
Adamic	61.01 (± 3.79)	51.05 (± 0.90)	56.22 (± 2.59)	50.47 (± 0.68)	71.88 (± 0.46)	50.91 (± 0.06)	62.83 (± 0.50)	65.39 (± 0.13)
Jaccard	60.72 (± 3.41)	51.02 (± 0.92)	56.11 (± 2.47)	50.47 (± 0.68)	62.30 (± 0.81)	50.91 (± 0.06)	62.77 (± 0.50)	64.95 (± 0.11)
Prefferential	70.76 (± 4.47)	52.31 (± 4.30)	55.70 (± 2.99)	52.34 (± 3.32)	83.38 (± 0.35)	47.54 (± 0.46)	88.75 (± 0.34)	69.53 (± 0.16)
GAE	63.73 (± 5.18)	57.44 (± 4.67)	58.76 (± 4.28)	58.59 (± 4.49)	80.38 (± 0.67)	53.02 (± 0.43)	83.80 (± 2.99)	68.16 (± 0.40)
GAT	45.35 (± 2.33)	54.50 (± 3.63)	50.51 (± 3.43)	51.98 (± 1.36)	72.40 (± 0.72)	54.26 (± 1.39)	63.51 (± 8.88)	77.75 (± 0.21)
GCN	61.75 (± 4.92)	59.03 (± 5.27)	54.73 (± 2.33)	56.51 (± 5.46)	69.69 (± 0.66)	57.09 (± 0.57)	75.69 (± 1.27)	75.66 (± 0.88)
GIN	59.81 (± 6.16)	59.47 (± 4.62)	54.90 (± 3.03)	54.69 (± 3.81)	70.64 (± 0.87)	58.11 (± 0.18)	73.23 (± 0.57)	77.19 (± 0.19)
SNoRe	70.79 (± 2.05)	57.59 (± 2.45)	**59.06 (± 3.23)**	60.47 (± 2.83)	69.06 (± 0.90)	**64.82 (± 0.16)**	86.91 (± 0.29)	**79.82 (± 0.19)**
node2vec	71.01 (± 5.07)	53.20 (± 3.54)	52.25 (± 2.10)	47.71 (± 2.04)	74.51 (± 0.83)	51.25 (± 0.44)	86.47 (± 0.36)	76.37 (± 0.11)
metapath2vec	**76.09 (± 3.55)**	57.36 (± 5.61)	41.42 (± 3.16)	49.20 (± 4.92)	78.93 (± 0.39)	57.68 (± 0.53)	76.91 (± 0.47)	53.76 (± 0.22)
TransE	74.82 (± 6.68)	56.23 (± 3.16)	55.99 (± 2.34)	54.16 (± 1.24)	**84.63 (± 0.23)**	64.56 (± 1.41)	**89.62 (± 0.29)**	75.20 (± 0.54)
RotatE	75.98 (± 5.20)	50.36 (± 4.47)	55.69 (± 2.99)	49.74 (± 2.53)	71.75 (± 0.90)	47.82 (± 0.19)	88.68 (± 0.38)	77.61 (± 0.25)
Spectral	43.48 (± 10.15)	**59.62 (± 5.26)**	55.84 (± 1.49)	**61.50 (± 2.56)**	68.32 (± 2.23)	49.27 (± 3.57)	83.93 (± 0.94)	61.07 (± 1.92)

Table 4
Link prediction results based on the ROC-AUC metric (multiplied by 100 to improve readability).

Dataset Baseline	Marine	Anatomy	SCTO	Emotions	EHDAA2	FoodOn	LKN	GO
Adamic	60.96 (± 3.72)	51.52 (± 0.72)	56.27 (± 3.20)	50.68 (± 0.60)	75.73 (± 0.51)	50.91 (± 0.06)	62.88 (± 0.50)	65.39 (± 0.13)
Jaccard	59.54 (± 2.71)	51.02 (± 0.86)	55.33 (± 2.22)	50.46 (± 0.61)	55.48 (± 0.73)	50.91 (± 0.06)	62.12 (± 0.48)	64.95 (± 0.11)
Prefferential	72.44 (± 5.63)	57.09 (± 2.97)	**66.10 (± 2.72)**	64.29 (± 3.53)	**86.89 (± 0.53)**	47.54 (± 0.46)	89.68 (± 0.33)	69.53 (± 0.16)
GAE	70.74 (± 4.92)	59.61 (± 4.65)	66.06 (± 4.22)	**65.24 (± 3.62)**	84.76 (± 0.76)	53.02 (± 0.43)	83.80 (± 2.99)	68.16 (± 0.40)
GAT	45.75 (± 1.69)	55.29 (± 1.07)	50.35 (± 3.36)	52.24 (± 1.80)	70.96 (± 0.47)	54.26 (± 1.39)	66.73 (± 11.95)	77.75 (± 0.21)
GCN	60.15 (± 4.62)	58.83 (± 3.70)	56.11 (± 2.83)	58.47 (± 3.75)	68.29 (± 1.05)	57.09 (± 0.57)	75.69 (± 1.27)	75.66 (± 0.88)
GIN	63.01 (± 3.65)	58.44 (± 3.39)	55.38 (± 3.09)	57.30 (± 3.25)	69.52 (± 1.22)	58.11 (± 0.18)	73.23 (± 0.57)	77.19 (± 0.19)
SNoRe	70.80 (± 0.68)	58.28 (± 1.31)	61.67 (± 3.35)	61.06 (± 3.07)	65.95 (± 1.15)	**64.82 (± 0.16)**	86.06 (± 0.54)	**79.82 (± 0.19)**
node2vec	73.97 (± 3.99)	57.79 (± 3.27)	53.05 (± 1.38)	50.31 (± 2.49)	72.80 (± 1.08)	51.25 (± 0.44)	87.19 (± 0.39)	76.37 (± 0.11)
metapath2vec	**78.76 (± 4.79)**	61.41 (± 6.01)	46.57 (± 2.42)	57.53 (± 5.23)	79.47 (± 0.73)	57.68 (± 0.53)	73.24 (± 0.49)	53.76 (± 0.22)
TransE	74.94 (± 4.54)	58.89 (± 3.79)	64.10 (± 3.40)	60.66 (± 3.23)	74.55 (± 0.85)	64.56 (± 1.41)	**91.80 (± 0.20)**	75.20 (± 0.54)
RotatE	75.83 (± 4.11)	**62.13 (± 3.38)**	63.27 (± 3.59)	62.18 (± 1.54)	85.37 (± 0.42)	47.82 (± 0.19)	91.19 (± 0.26)	77.61 (± 0.25)
Spectral	55.70 (± 9.71)	61.05 (± 3.79)	58.01 (± 2.87)	59.25 (± 2.07)	64.87 (± 2.49)	49.27 (± 3.57)	81.28 (± 2.39)	61.07 (± 1.92)

Table 5
Link prediction results based on the average precision metric (multiplied by 100 to improve readability).
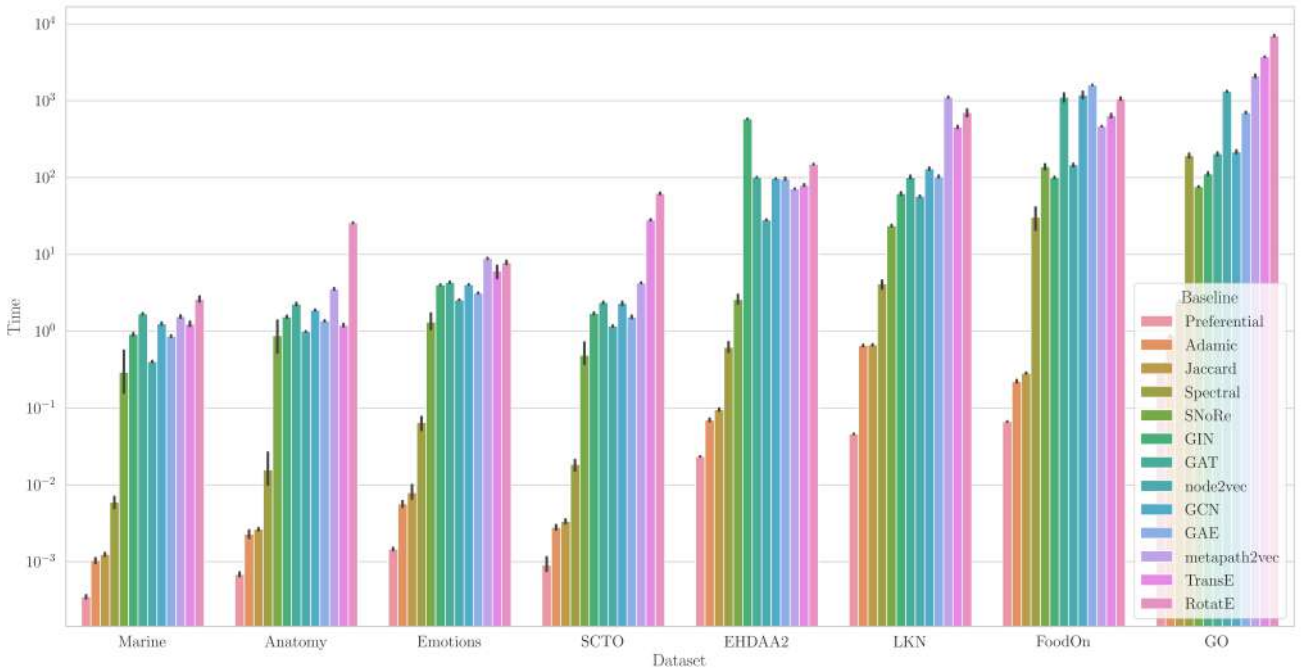
Figure 4: Time needed to train the baselines, shown on the logarithmic scale. Baselines are ordered by the total training time

Year	Added	Removed
2016	7492	3302
2017	7497	2960
2018	15905	13704
2019	3781	3746
2020	2969	4381
2021	12653	17114

Table 6
Number of edges added to and removed from the previous year's Gene Ontology.

k\year (t)	2015	2016	2017	2018	2019	2020
10	0.200	0.200	0.000	0.100	0.300	0.000
100	0.060	0.070	0.040	0.040	0.080	0.000
500	0.022	0.022	0.016	0.044	0.052	0.006

Table 7
Accuracy of top k predicted missing edges at year t that appear in the following year's $(t + 1)$ Gene Ontology.

can see that the parameter of the term *apoptotic process* has the highest value, meaning that when the neighborhood of two nodes is similar to the neighborhood of this node (high value in the embedding for this feature), it is more likely that there is an edge between them. Other features have lower parameter value but their value is still quite high.

The local interpretation of top four recommendations for the biological process node of the same year are shown in Figure 6. For each of the four recommendations, we show four features that contribute the most, the amount they contribute, and the mean value of features in this vector. Overall, we can see that in our examples some features stand out, but even the top feature is in some cases only ten times higher than the mean value. The feature that stands out most in these examples is *GO_0051186 (cofactor metabolic process)* between nodes that represent terms *GO_0008150 (biological process)* and *GO_0006739 (NADP metabolic process)*. This feature has value 0.11 while the second highest *GO_1901360 (organic cyclic compound metabolic process)* has 0.06.

Further, we can look at the distribution of feature values in the local interpretation and determine whether some of these values stand out. An example of this can be seen in figure 7, where distribution of feature values of top four recommendations are shown. We can see that most features of each recommendation are zero, while only a few really stand out.

6. Discussion

This section summarises our work and discusses the main advantages and disadvantages of the proposed approach for finding missing and redundant edges in ontologies.

The main goal of our approach is to generate reasonable predictions for missing and redundant links in ontologies solely based on their structure. This is done by scoring edges using link prediction algorithms. In Section 5.1 we empirically show the results of link prediction on graphs we get by transforming ontologies. We see that link prediction works well on graphs with a high number of average edges per node but badly on graphs whose structure resembles a tree. This means that our methodology will be much more reliable on graphs with many edges and unreliable on ontologies whose structure resembles a tree.

In Section 5.2 we tested our approach by using different versions of the same ontology and testing whether the created candidates occur in other versions. We do this by creating candidates for missing edges on the version t of the ontology and checking their occurrence on version $t+1$. The results for the predicted missing edges in Table 7 show that the candidates with the higher score have a better chance to occur in the following version, while the redundant edges occur with roughly the same frequency regardless of their score.

Since the results show that a significant percentage of candidates for new edges occurs in the following version of
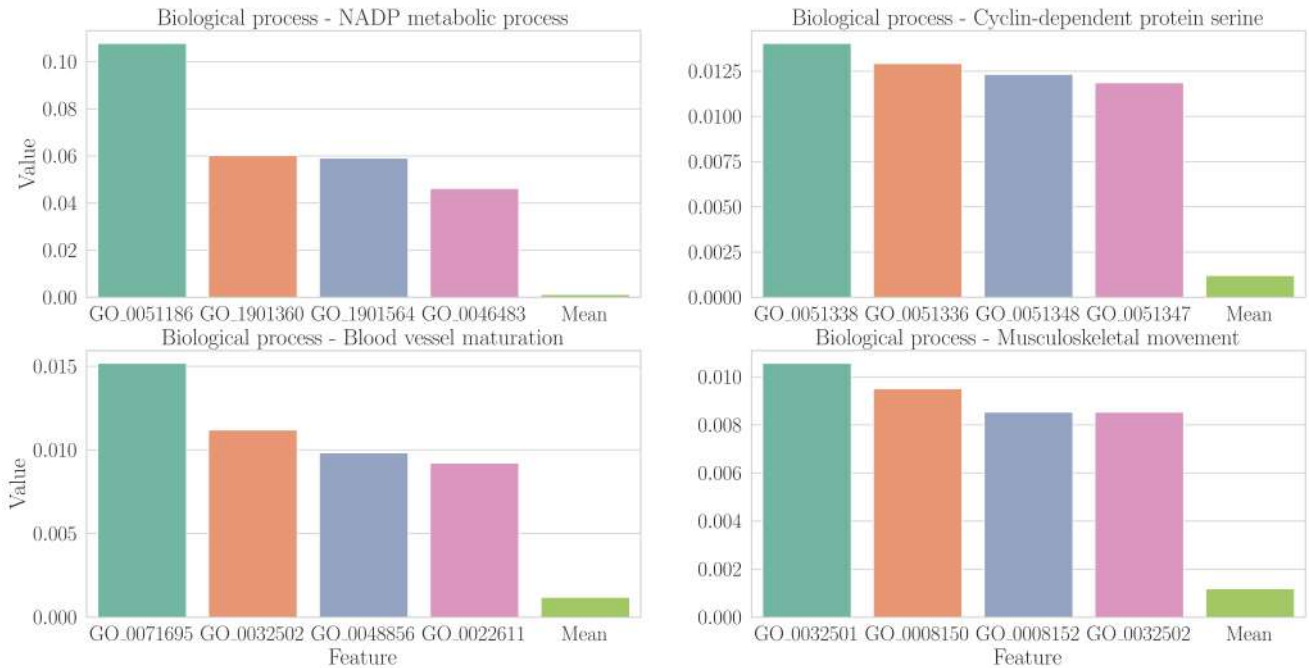
k\year (t)	2015	2016	2017	2018	2019	2020
10	0.000	0.000	0.000	0.100	0.000	0.000
100	0.040	0.020	0.020	0.070	0.130	0.010
500	0.024	0.010	0.028	0.044	0.048	0.028

Table 8
Accuracy of top k predicted redundant edges at year t that were removed in the following year's $(t + 1)$ Gene Ontology.
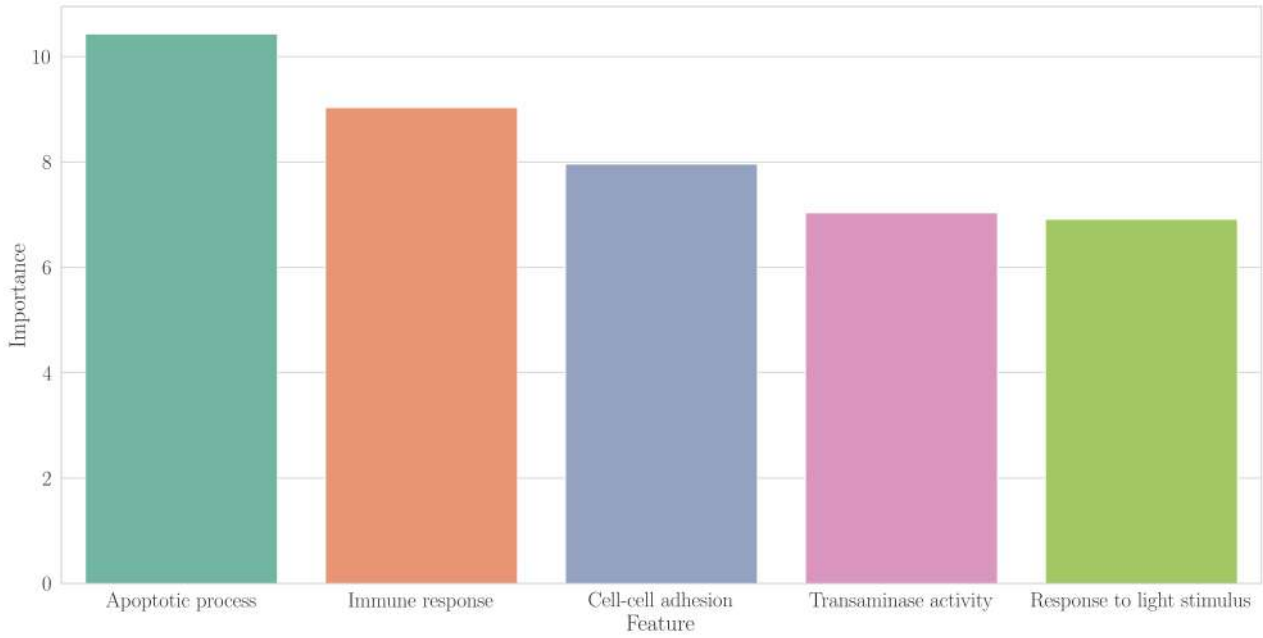
**Figure 5:** An example of a global explanation for the Gene ontology.

Figure 6: Examples of local explanations for four missing edges with the highest score. The sum of all feature values for an edge is equal to the edge's score.

the ontology, this methodology could help annotate larger ontologies where connections can easily be missed. In practice, the project coordinator could set up a web application where the annotator would select nodes he/she is interested in and get the missing edges with the highest, and existing edges with the lowest score. To further increase the possibility of correct annotation, explanations for the relevant edges could also be shown. The main bottleneck for our approach is the calculation of the embedding, which can be done offline. This means that creating recommendations for annotators would be fast and would not need a lot of resources – the developed approach could serve as an on-line ontology annotation assistant.

There are a few disadvantages to the proposed methodol-
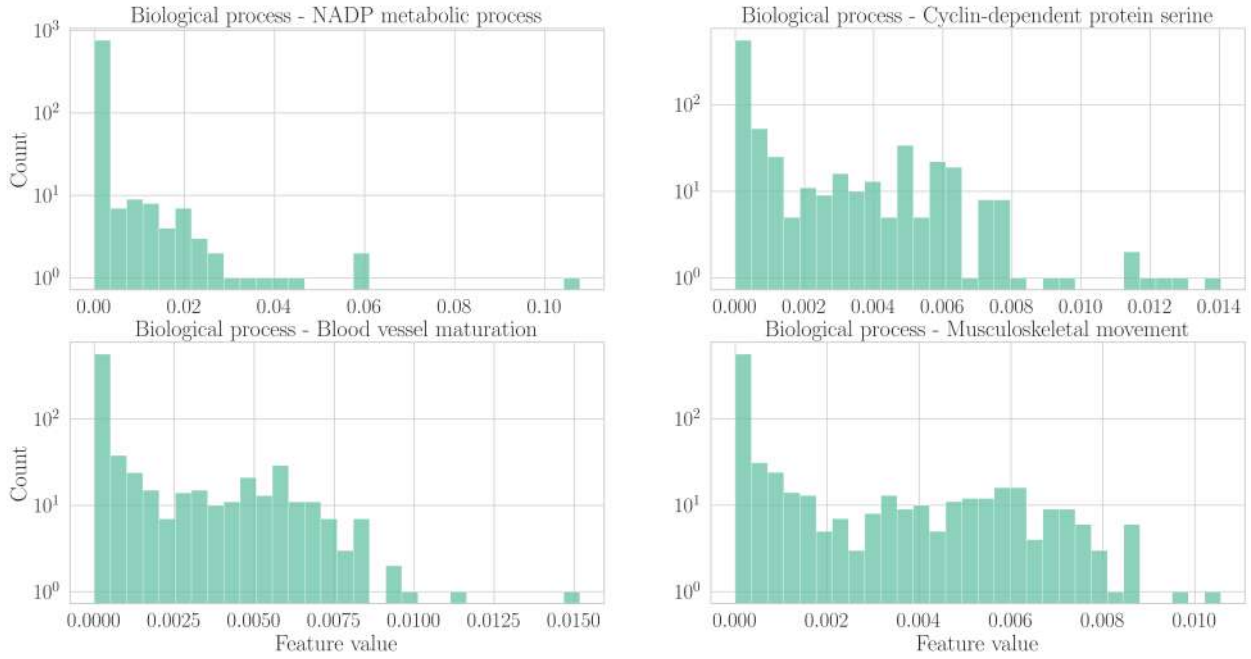
Figure 7: Distribution of feature values for local explanations. The x axis shows the value of the feature, while the y axis (logarithmic scale) represents the number of features with the value.

ogy. The biggest is that the methodology works well on ontologies where each node has many connections but poorly on the ones whose structure resembles a tree. Such ontologies probably do not contain enough information to recommend new connections only based on their structure. A possible solution could be to include additional information such as the description of classes and recommend edges based on this semantic information.

Another disadvantage is that the approach needs quadratic space to store scores for each connection. This could prove problematic for large ontologies where such an approach is needed even more due to the number of possible connections that can easily be missed. In practice, this is not necessarily problematic since embeddings are usually small enough to fit inside the memory and can be used to calculate scores for only a subset of k nodes. This lowers the space complexity to $|N| \cdot k$, which can easily fit inside the memory and gives the same results.

We note that this work primarily focuses on outlining a scalable end-to-end methodology for graph-based link prediction on ontologies. As such, there exist possible improvements to many of the approaches adopted as part of the pipeline.

For example, we demonstrate how KG specific methods compare to methods that operate on simple graphs. However, there exist even more specialized methods, such as ontology-specific embeddings [56, 57, 13, 14, 35]. These approaches are tailored to capture the higher expressivity that ontologies offer compared to knowledge graphs and usually utilize lexical information (meta-data) about nodes and relations, which we ignore. Leveraging this additional information would likely produce better results.

Other choices could also be made when it comes to ontology pre-processing and the ontology-to-graph conversion step. These include extending given ontologies with related ones, ontology pruning, entailment reasoning before conversion and the choice of ontology-to-graph conversion protocol itself.

7. Conclusions

In this work, graph-based machine learning approaches were used in the proposed methodology for finding missing and redundant edges in ontologies. We showed that this approach yields good results on larger ontologies when nodes have a high average degree.

The proposed approach could prove useful for annotators of large ontologies or domain experts (e.g., biologists) to find the connections that are the most likely to belong in the ontology.

In further work, we plan to collaborate with domain experts to further analyze the performance of our methodology in a real-life setting. We also intend to study different approaches for pre-processing ontologies and representing them as graphs, since this is one potential area where incorporating more of the available semantic information into the model could help improve results. Finally, we wish to move our focus to ontology-specific embeddings and other methods that utilize meta-data about entities. We suspect that taking full advantage of these additional features can significantly improve the results and further explore the limits of the presented methodology.

Acknowledgments

References

[1] Adamic, L.A., Adar, E., 2003. Friends and neighbors on the web. Social Networks 25, 211–230. URL: https://www.sciencedirect.com/science/article/pii/S0378873303000091, doi:https://doi.org/10.1016/S0378-8733(03)00009-1.

[2] Althubaiti, S., Karwath, A., Dallol, A., Noor, A., Alkhayyat, S., Alwassia, R., Mineta, K., Gojobori, T., Beggs, A., Schofield, P., Gkoutos, G., Hoehndorf, R., 2019. Ontology-based prediction of cancer driver genes. Scientific Reports 9. doi:10.1038/s41598-019-53454-1.

[3] Ashburner, M., Ball, C.A., Blake, J.A., Botstein, D., Butler, H., Cherry, J.M., Davis, A.P., Dolinski, K., Dwight, S.S., Eppig, J.T., Harris, M.A., Hill, D.P., Issel-Tarver, L., Kasarskis, A., Lewis, S., Matese, J.C., Richardson, J.E., Ringwald, M., Rubin, G.M., Sherlock, G., 2000. Gene ontology: tool for the unification of biology. the gene ontology consortium. Nature genetics 25, 25–29. URL: https://pubmed.ncbi.nlm.nih.gov/10802651, doi:10.1038/75556. 10802651[pmid].

[4] Barabási, A.L., Albert, R., 1999. Emergence of scaling in random networks. Science 286, 509–512. URL: https://science.sciencemag.org/content/286/5439/509, doi:10.1126/science.286.5439.509, arXiv:https://science.sciencemag.org/content/286/5439/509.full.pdf.

[5] Bard, J., 2012a. The aeo, an ontology of anatomical entities for classifying animal tissues and organs. Frontiers in Genetics 3, 18. URL: https://www.frontiersin.org/article/10.3389/fgene.2012.00018, doi:10.3389/fgene.2012.00018.

[6] Bard, J., 2012b. A new ontology (structured hierarchy) of human developmental anatomy for the first 7 weeks (carnegie stages 1-20) 221, 406–416. URL: https://doi.org/10.1111/j.1469-7580.2012.01566.x, doi:10.1111/j.1469-7580.2012.01566.x.

[7] Bhagat, S., Cormode, G., Muthukrishnan, S., 2011. Node Classification in Social Networks. Springer US, Boston, MA. pp. 115–148. URL: https://doi.org/10.1007/978-1-4419-8462-3_5, doi:10.1007/978-1-4419-8462-3_5.

[8] Bonatti, P., Decker, S., Polleres, A., Presutti, V., 2018. Knowledge graphs: New directions for knowledge representation on the semantic web (dagstuhl seminar 18371). Dagstuhl Reports 8, 29–111.

[9] Bordes, A., Usunier, N., García-Durán, A., Weston, J., Yakhnenko, O., 2013. Translating embeddings for modeling multi-relational data, in: Burges, C.J.C., Bottou, L., Ghahramani, Z., Weinberger, K.Q. (Eds.), Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States, pp. 2787–2795. URL: https://proceedings.neurips.cc/paper/2013/hash/1cecc7a77928ca8133fa24680a88d2f9-Abstract.html.

[10] Brank, J., Grobelnik, M., Mladenic, D., 2005. A survey of ontology evaluation techniques, in: Proceedings of the conference on data mining and data warehouses (SiKDD 2005), Citeseer Ljubljana, Slovenia. pp. 166–170.

[11] Chandola, V., Banerjee, A., Kumar, V., 2009. Anomaly detection: A survey. ACM Comput. Surv. 41. doi:10.1145/1541880.1541882.

[12] Chen, J., Althagafi, A., Hoehndorf, R., 2020a. Predicting candidate genes from phenotypes, functions, and anatomical site of expression. Bioinformatics 37. doi:10.1093/bioinformatics/btaa879.

[13] Chen, J., Hu, P., Jiménez-Ruiz, E., Holter, O., Antonyrajah, D., Horrocks, I., 2020b. Owl2vec*: Embedding of owl ontologies .

[14] Chen, M., Tian, Y., Chen, X., Xue, Z., Zaniolo, C., 2018. On2vec: Embedding-based relation prediction for ontology population, in: Ester, M., Pedreschi, D. (Eds.), Proceedings of the 2018 SIAM International Conference on Data Mining, SDM 2018, May 3-5, 2018, San Diego Marriott Mission Valley, San Diego, CA, USA, SIAM. pp. 315–323. URL: https://doi.org/10.1137/1.9781611975321.36, doi:10.1137/1.9781611975321.36.

[15] Costa, L.d.F., Oliveira Jr, O.N., Travieso, G., Rodrigues, F.A., Villas Boas, P.R., Antiqueira, L., Viana, M.P., Correa Rocha, L.E., 2011. Analyzing and modeling real-world phenomena with complex networks: a survey of applications. Advances in Physics 60, 329–412.

[16] Cuenca-Grau, B., Horrocks, I., Parsia, B., Ruttenberg, A., Schneider, M., 2012. OWL 2 Web Ontology Language Mapping to RDF Graphs (Second Edition).

[17] Denning, D., 1987. An intrusion-detection model. IEEE Transactions on Software Engineering SE-13, 222–232. doi:10.1109/TSE.1987.232894.

[18] Doan, A., Madhavan, J., Domingos, P., Halevy, A., 2004. Ontology Matching: A Machine Learning Approach. Springer Berlin Heidelberg, Berlin, Heidelberg. pp. 385–403. doi:10.1007/978-3-540-24750-0_19.

[19] Dong, Y., Chawla, N.V., Swami, A., 2017. metapath2vec: Scalable representation learning for heterogeneous networks, in: Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, August 13 - 17, 2017, ACM. pp. 135–144. URL: https://doi.org/10.1145/3097983.3098036, doi:10.1145/3097983.3098036.

[20] Dooley, D.M., Griffiths, E.J., Gosal, G.S., Buttigieg, P.L., Hoehndorf, R., Lange, M.C., Schriml, L.M., Brinkman, F.S.L., Hsiao, W.W.L., 2018. Foodon: a harmonized food ontology to increase global food traceability, quality control and data integration. npj Science of Food 2, 23. URL: https://doi.org/10.1038/s41538-018-0032-6, doi:10.1038/s41538-018-0032-6.

[21] Dragoni, M., Bailoni, T., Maimone, R., Eccher, C., 2018. Helis: An ontology for supporting healthy lifestyles, in: Vrandečić, D., Bontcheva, K., Suárez-Figueroa, M.C., Presutti, V., Celino, I., Sabou, M., Kaffee, L.A., Simperl, E. (Eds.), The Semantic Web – ISWC 2018, Springer International Publishing. pp. 53–69.

[22] Ehrlinger, L., Wöß, W., 2016. Towards a definition of knowledge graphs.

[23] El-Sappagh, S., Franda, F., Ali, F., Kwak, K.S., 2018. Snomed ct standard ontology based on the ontology for general medical science. BMC Medical Informatics and Decision Making 18, 76. URL: https://doi.org/10.1186/s12911-018-0651-5, doi:10.1186/s12911-018-0651-5.

[24] Fawcett, T., Provost, F., 1999. Activity monitoring: Noticing interesting changes in behavior, in: In Proceedings of the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 53–62.

[25] Fey, M., Lenssen, J.E., 2019. Fast Graph Representation Learning with PyTorch Geometric, in: ICLR Workshop on Representation Learning on Graphs and Manifolds.

[26] Graves, M., Constabaris, A., Brickley, D., 2007. Foaf: Connecting people on the semantic web. Cataloging & classification quarterly 43, 191–202.

[27] Grover, A., Leskovec, J., 2016. node2vec: Scalable feature learning for networks, in: Krishnapuram, B., Shah, M., Smola, A.J., Aggarwal, C.C., Shen, D., Rastogi, R. (Eds.), Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, August 13-17, 2016, ACM. pp. 855–864. URL: https://doi.org/10.1145/2939672.2939754, doi:10.1145/2939672.2939754.

[28] Hastings, J., Ceusters, W., Smith, B., Mulligan, K., 2011. Dispositions and processes in the emotion ontology. CEUR Workshop Proceedings 833.

[29] Hitzler, P., Kroetzsch, M., Parsia, B., Patel-Schneider, P., Rudolph,

S., 2012. OWL Web Ontology Language Primer (Second Edition).

[30] Horn, P., Feng, L., Li, Y., Pesce, A., 2002. Effect of outliers and non-healthy individuals on reference interval estimation. Clinical chemistry 47, 2137–45. doi:10.1093/clinchem/47.12.2137.

[31] Kejriwal, M., 2019. What Is a Knowledge Graph?. Springer International Publishing. pp. 1–7. doi:10.1007/978-3-030-12375-8.

[32] Kipf, T.N., Welling, M., 2016. Variational Graph Auto-Encoders. NIPS Workshop on Bayesian Deep Learning .

[33] Kipf, T.N., Welling, M., 2017. Semi-supervised classification with graph convolutional networks, in: 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings, OpenReview.net. URL: https://openreview.net/forum?id=SJU4ayYgl.

[34] Kralj, J., Robnik-Sikonja, M., Lavrac, N., 2019. Netsdm: Semantic data mining with network analysis. Journal of Machine Learning Research 20, 1–50. URL: http://jmlr.org/papers/v20/17-066.html.

[35] Kulmanov, M., Liu-Wei, W., Yan, Y., Hoehndorf, R., 2019. EL embeddings: Geometric construction of models for the description logic EL++, in: Kraus, S. (Ed.), Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019, ijcai.org. pp. 6103–6109. URL: https://doi.org/10.24963/ijcai.2019/845, doi:10.24963/ijcai.2019/845.

[36] Kulmanov, M., Smaili, F.Z., Gao, X., Hoehndorf, R., 2020. Semantic similarity and machine learning with ontologies. Briefings in Bioinformatics 22. doi:10.1093/bib/bbaa199.

[37] Li, N., Bouraoui, Z., Schockaert, S., 2019. Ontology completion using graph convolutional networks, in: SEMWEB.

[38] Liben-Nowell, D., Kleinberg, J., 2007. The link-prediction problem for social networks. Journal of the American society for information science and technology 58, 1019–1031.

[39] Lü, L., Zhou, T., 2011. Link prediction in complex networks: A survey. Physica A: statistical mechanics and its applications 390, 1150–1170.

[40] Lv, X., Hou, L., Li, J., Liu, Z., 2018. Differentiating concepts and instances for knowledge graph embedding, in: Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Brussels, Belgium. pp. 1971–1979. URL: https://aclanthology.org/D18-1222, doi:10.18653/v1/D18-1222.

[41] Mežnar, S., Lavrač, N., Škrlj, B., 2020. Snore: Scalable unsupervised learning of symbolic node representations. IEEE Access 8, 212568–212588. doi:10.1109/ACCESS.2020.3039541.

[42] Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013. Efficient estimation of word representations in vector space. Proceedings of Workshop at ICLR 2013.

[43] Molnar, C., 2019. Interpretable Machine Learning. https://christophm.github.io/interpretable-ml-book/.

[44] Ng, A.Y., Jordan, M.I., Weiss, Y., 2001. On spectral clustering: Analysis and an algorithm, in: Dietterich, T.G., Becker, S., Ghahramani, Z. (Eds.), Advances in Neural Information Processing Systems 14 [Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, December 3-8, 2001, Vancouver, British Columbia, Canada], MIT Press. pp. 849–856. URL: https://proceedings.neurips.cc/paper/2001/hash/801272ee79cfde7fa5960571fee36b9b-Abstract.html.

[45] Nunes, S., Sousa, R., Pesquita, C., 2021. Predicting gene-disease associations with knowledge graph embeddings over multiple ontologies .

[46] Page, L., Brin, S., Motwani, R., Winograd, T., 1999. The PageRank Citation Ranking: Bringing Order to the Web., in: WWW 1999.

[47] Paulheim, H., 2016. Knowledge graph refinement: A survey of approaches and evaluation methods. Semantic Web 8, 489–508. doi:10.3233/SW-160218.

[48] Perozzi, B., Al-Rfou, R., Skiena, S., 2014. Deepwalk: online learning of social representations, in: Macskassy, S.A., Perlich, C., Leskovec, J., Wang, W., Ghani, R. (Eds.), The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '14, New York, NY, USA - August 24 - 27, 2014, ACM. pp. 701–710. URL: https://doi.org/10.1145/2623330.2623732, doi:10.1145/2623330.2623732.

[49] Pesquita, C., Faria, D., Falcão, A., Lord, P., Couto, F., 2009. Semantic similarity in biomedical ontologies. PLoS Computational Biology 5.

[50] Pokrajac, D., Lazarevic, A., Latecki, L.J., 2007. Incremental local outlier detection for data streams, in: 2007 IEEE Symposium on Computational Intelligence and Data Mining, pp. 504–515. doi:10.1109/CIDM.2007.368917.

[51] Ramšak, Ž., Coll, A., Stare, T., Tzfadia, O., Baebler, Š., Van de Peer, Y., Gruden, K., 2018. Network modeling unravels mechanisms of crosstalk between ethylene and salicylate signaling in potato. Plant physiology 178, 488–499. URL: https://pubmed.ncbi.nlm.nih.gov/29934298, doi:10.1104/pp.18.00450. 29934298[pmid].

[52] Roche, C., 2003. Ontology: a survey. IFAC Proceedings Volumes 36, 187–192.

[53] Rodríguez-García, M., Hoehndorf, R., 2018. Inferring ontology graph structures using owl reasoning. BMC Bioinformatics 19. doi:10.1186/s12859-017-1999-8.

[54] Salton, G., McGill, M., 1983. Introduction to Modern Information Retrieval. International student edition, McGraw-Hill. URL: https://books.google.si/books?id=7f5TAAAAMAAJ.

[55] Silla, C., Freitas, A., 2011. A survey of hierarchical classification across different application domains. Data Mining and Knowledge Discovery 22, 31–72. doi:10.1007/s10618-010-0175-9.

[56] Smaili, F.Z., Gao, X., Hoehndorf, R., 2018a. Onto2vec: joint vector-based representation of biological entities and their ontology-based annotations. Bioinformatics (Oxford, England) 34. doi:10.1093/bioinformatics/bty259.

[57] Smaili, F.Z., Gao, X., Hoehndorf, R., 2018b. Opa2vec: combining formal and informal content of biomedical ontologies to improve similarity-based prediction. Bioinformatics (Oxford, England) 35. doi:10.1093/bioinformatics/bty933.

[58] Smith, B., Ceusters, W., Klagges, B., Köhler, J., Kumar, A., Lomax, J., Mungall, C., Neuhaus, F., Rector, A., Rosse, C., 2005. Relations in biomedical ontologies. Genome biology 6, R46. doi:10.1186/gb-2005-6-5-r46.

[59] Soylu, A., Kharlamov, E., Zheleznyakov, D., Jiménez-Ruiz, E., Giese, M., Skjæveland, M., Hovland, D., Schlatte, R., Brandt, S., Lie, H., Horrocks, I., 2017. Optiquevqs: a visual query system over ontologies for industry. Semantic Web 9. doi:10.3233/SW-180293.

[60] Sun, Z., Deng, Z., Nie, J., Tang, J., 2019. Rotate: Knowledge graph embedding by relational rotation in complex space, in: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019, OpenReview.net. URL: https://openreview.net/forum?id=HkgEQnRqYQ.

[61] Tzitzikas, Y., Alloca, C., Bekiari, C., Marketakis, Y., Fafalios, P., Doerr, M., Minadakis, N., Patkos, T., Candela, L., 2013. Integrating heterogeneous and distributed information about marine species through a top level ontology, in: Proceedings of the 7th Metadata and Semantic Research Conference (MTSR'13), Thessaloniki, Greece.

[62] Velickovic, P., Cucurull, G., Casanova, A., Romero, A., Liò, P., Bengio, Y., 2018. Graph attention networks, in: 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings, OpenReview.net. URL: https://openreview.net/forum?id=rJXMpikCZ.

[63] Xiaojin, Z., Zoubin, G., 2002. Learning from labeled and unlabeled data with label propagation. Tech. Rep., Technical Report CMU-CALD-02–107, Carnegie Mellon University .

[64] Xu, K., Hu, W., Leskovec, J., Jegelka, S., 2019. How powerful are graph neural networks?, in: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019, OpenReview.net. URL: https://openreview.net/forum?id=ryGs6iA5Km.

[65] Zhang, S.B., Tang, Q.R., 2016. Protein–protein interaction inference based on semantic similarity of gene ontology terms. Journal of theoretical biology 401. doi:10.1016/j.jtbi.2016.04.020.

[66] Zhao, Y., Fu, G., Wang, J., Guo, M., Yu, G.X., 2018. Gene function prediction based on gene ontology hierarchy preserving hashing. Genomics 111. doi:10.1016/j.ygeno.2018.02.008.