machine learning & knowledge extraction

MDPI

Article

Ontology Completion with Graph-Based Machine Learning: A Comprehensive Evaluation

Sebastian Mežnar [1,2]*, Matej Bevec [1], Nada Lavrač [1,2,3] and Blaž Škrlj [1]

[1] Jožef Stefan Institute, Jamova 39, 1000 Ljubljana, Slovenia
[2] Jožef Stefan International Postgraduate School, Jamova 39, 1000 Ljubljana, Slovenia
[3] School of Engineering and Management, University of Nova Gorica, Glavni trg 8, 5271 Vipava, Slovenia
* Correspondence: sebastian.meznar@ijs.si

Abstract: Increasing quantities of semantic resources offer a wealth of human knowledge, but their growth also increases the probability of wrong knowledge base entries. The development of approaches that identify potentially spurious parts of a given knowledge base is therefore highly relevant. We propose an approach for ontology completion that transforms an ontology into a graph and recommends missing edges using structure-only link analysis methods. By systematically evaluating thirteen methods (some for knowledge graphs) on eight different semantic resources, including Gene Ontology, Food Ontology, Marine Ontology, and similar ontologies, we demonstrate that a structure-only link analysis can offer a scalable and computationally efficient ontology completion approach for a subset of analyzed data sets. To the best of our knowledge, this is currently the most extensive systematic study of the applicability of different types of link analysis methods across semantic resources from different domains. It demonstrates that by considering symbolic node embeddings, explanations of the predictions (links) can be obtained, making this branch of methods potentially more valuable than black-box methods.

Keywords: machine learning; embedding; ontology completion; link prediction; explainability

Citation: Mežnar, S.; Bevec, M.; Lavrač, N.; Škrlj, B. Ontology Completion with Graph-Based Machine Learning: A Comprehensive Evaluation. *Mach. Learn. Knowl. Extr.* **2022**, *4*, 1107–1123. https://doi.org/10.3390/make4040056

Academic Editors: Andreas Holzinger and Federico Cabitza

Received: 20 October 2022
Accepted: 28 November 2022
Published: 1 December 2022

1. Introduction

Researchers and companies often address tasks that require domain knowledge for their solution. Domain knowledge often contains complex relations between entities and human-defined terms that can be modeled using ontologies [1,2]. Ontologies can range from small ones that describe people, their activities, and relations to other people, such as the FOAF ontology [3], up to large ones such as the Gene Ontology that describes, e.g., protein functions and cellular processes [4].

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 [5,6], much like graphs, or to improve features with background knowledge in other machine learning tasks [7]. Commonly applied methods range from more traditional semantic similarity approaches [5,8] to recently successful entity-embedding algorithms, whether graph-based [9–11], syntactic [12,13], hybrid [14], or rule-based [15]. Alternatively, ontologies have been utilized to constrain the output of machine learning and optimization models to conform to certain rules [16].

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 nonexistent ones. These mistakes can have an impact on our understanding of the domain and can produce models and solutions that do not perform well. This issue is addressed by *ontology completion*, which refers to the task of finding missing relations that are plausible but not logically deducible from the given ontology [17].

This work adopts machine learning methods that perform well on the link prediction task for ontology completion exclusively based on the ontology structure. This is achieved by representing ontologies as graphs and applying link prediction methods, providing experts with a scalable ontology completion tool to help improve the error-prone human annotation process. The contributions of this work include:

- A methodology for ontology completion using link analysis methods;
- A methodology for recommending missing edges using scores obtained through link prediction;
- Demonstrated utility of considered link analysis methods on multiple ontologies with different properties;
- Simple-to-use software for ontology completion and evaluation of the proposed methodology on a given ontology (Available online: https://github.com/smeznar/ontology-completion-with-graph-learners, accessed on 1 November 2022).

The related work is presented in Section 2. The proposed methodology is outlined in Section 3. It consists of an ontology-to-graph transformation (Section 3.1), ontology completion using link prediction (Section 3.2), a recommendation of missing edges (Section 3.3), and an approach for explaining recommendations (Section 3.4). The experimental setting is outlined in Section 4. The results of the evaluation are presented in Section 5. The paper concludes with a discussion in Section 6.

2. Related Work

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

2.1. Ontologies

Ontologies refer to machine-readable representations of knowledge in a given application domain, usually defined in a declarative knowledge modeling language, such as OWL (Web Ontology Language) [18], which 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"*. T-box statements in OWL are class subsumption axioms (e.g., *"mother SubclassOf: woman"*), property restrictions (e.g., *"parent EquivalentTo: HasChild Some person"*, meaning that everyone who has at least one child is a parent) and set operators (e.g., *"parent EquivalentTo: mother Or father"*). A-box statements include membership axioms (e.g., *"Mary Types: mother"*) and property assertions (e.g., *John Facts: HasWife Mary*).

Ontologies are used in various fields and vary significantly in their purpose, content, and implementation. At one end of the spectrum, there are 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 and the Marine Ontology. Conversely, there are larger, more grounded ontologies attempting to comprehensively capture knowledge in a domain as a complex hierarchy with many concrete facts. The Gene Ontology is an illustrative example.

Notably, grounded ontologies, or at least their graph representations, could be considered knowledge graphs (KGs) by some definitions that define the latter as a schema (T-box) accompanied by a large number of (A-box) facts [19]. However, KGs do not have

a well-established definition yet, with other work giving alternate proposals, such as a collection of facts without the schema [20], or a system encompassing an ontology and a reasoning engine [21].

In practice, different grounded ontologies also take different approaches to capturing A-box (ground level) facts in OWL. Some, such as HeLiS [22], 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 [23], do not use individuals at all, representing even very "individual-like" entities as classes and thus blurring the line between T-box and A-box. Because of this, it can be difficult to create a general approach for embedding ontologies.

2.2. Ontology Embeddings

Ontologies can be seen as stand-alone resources, much like knowledge graphs, as means of encoding domain knowledge. On the other hand, ontologies have recently shown potential in combination with machine learning methods as means to provide additional background information or to constrain the learning process. To exploit ontology information using approaches such as linear regression or neural networks, one usually has to embed them into the vector space first. In our work, we propose to embed ontologies with graph-based methods to identify potentially novel relations for ontology completion.

The related work comprises several works that present ontology-specific embedding algorithms. Onto2Vec [12] constructs sentences from OWL axioms and trains a language model; On2Vec [11] is based on translational graph embeddings and OWL2Vec* [14] combines the language model approach with random walks on the ontology graph. These methods have been typically evaluated against knowledge graph embeddings on a limited number of large ontologies for the relation prediction task (predicting the predicate between two entities).

Other examples are domain-specific applications, most often in the biomedical domain, where ontologies are mined for tasks such as protein–protein interaction (PPI) prediction, or gene–disease prediction. Here, several ontologies are usually combined into a single data set and used with semantic similarity approaches [5,24], often heavily tailored to the task. Recently, similar works have adopted ontology-specific embeddings [6,25] that are more general.

None of the above can be considered a systematic study. To our knowledge, the most valuable resource about the topic that is closest to our work is a survey on the state of machine learning with ontologies [26] that covers both traditional semantic similarity methods and recent embedding-based methods. It looked at simple graph embeddings, knowledge graph embeddings, and ontology-specific embeddings, categorizing them into graph-based, syntactic and semantic approaches. The survey included an experimental evaluation of a subset of methods on a protein–protein interaction task. It only considered two subsets of GO as data sets, since its focus was on a theoretic categorization of the field and on the biomedical domain in particular. Compared to the survey [26] that focused on the overview of the field, our study focuses on the evaluation of a large number of graph-based methods, comparing graph embeddings to KG-specific embeddings. However, we limit ourselves to structure-only methods due to the nature of the ontology-to-graph conversion. We also 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.

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

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 [12]	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 [13]	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 [11]	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 [14]	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 [25]	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 [6]	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 [27]	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 [24]	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 [5]	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 [26]	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.

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 [28]. This branch of machine learning mainly focuses on node and graph classification [29], node clustering, and link prediction tasks [30].

Machine learning tasks on graphs are usually solved in three different ways. Traditionally, tasks on graphs are solved using label propagation [31], PageRank [32], and proximity-based measures such as Adamic/Adar [33] or the Jaccard coefficient [34]. Another group of approaches embed graphs into a vector space, used together with traditional machine learning methods such as logistic regression to generate predictions. These approaches include well-established methods such as node2vec [9] and Deepwalk [35], as well as new ones such as SNoRe [36]. More recently, with new research in deep learning approaches, neural network models such as graph convolutional networks (GCN) [37], and graph attention networks (GAT) [38] have emerged as end-to-end learners.

2.4. Link Prediction

Link prediction is one of the most widely addressed tasks concerning graph-based data. Predicting whether there exists an edge between two nodes without any additional information is hard, but with some additional information about the graphs, nodes, and with some assumptions, various approaches can predict the edge existence well [30,39]. 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 similar neighborhoods. Another assumption that is commonly used is that nodes are likely to be connected to nodes with a high number of neighbors. These assumptions are both reasonable and often occur in real-life networks as well as ontologies.

Link prediction is traditionally solved using proximity-based methods that model networks using the assumptions mentioned above. These methods commonly predict the existence of links based on the first and second neighbors of the nodes, e.g., the number of common neighbors. These include Adamic/Adar index [33], preferential attachment, and others. Later, embedding methods such as node2vec [9] and SNoRe [36] were developed. These methods use a random walk to explore and approximate a node's neighborhood. The embedding these methods produce is either low-dimensional or sparse and usually performs well even on networks where structure assumptions do not necessarily hold. Similar to embedding methods, graph neural networks such as GCN [37] and GAT [38] have recently been used for different machine learning tasks on networks. These approaches jointly exploit the adjacency matrix of a network alongside node features. For knowledge graphs, i.e., graphs where nodes and edges usually contain some additional information, specialized approaches such as metapath2vec [40] are used. Other approaches on knowledge graphs such as TransE [10] and RotatE [41] 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, and object) is inside the graph and close to one if it is not.

In our work, we mainly focus on link prediction using embedding and proximity-based methods as they do not require a specific representation or additional knowledge about the graph. By exploiting the semantic information of the knowledge graph, one can find missing links using rule-based approaches [15].

3. Methodology

The following section presents the ontology-to-graph transformation, link prediction, recommendation of missing links, and explanation of the recommendations.

3.1. Ontology-to-Graph Transformation

As outlined above, machine learning has approached the use of ontologies in various ways. One of the common approaches is to represent ontologies as graphs where nodes represent classes or individuals, and links encode semantic relationships defined by the

ontology. This approach 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 modeled as such, there exist multiple possible conversions of a given ontology into a graph. Certain expressions, such as property assertions, directly map to nodes and edges. Others, such as property restrictions, domain-range axioms, and set operators, do not have an obvious representation. Because of 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 for converting OWL ontologies to graphs have been developed; however, there is not yet an established standard or agreement on what the most appropriate representation is. In our work we transform an ontology into a (knowledge) graph using *projection rules* [14,42], as it has been previously used in a similar machine learning context, outperforming methods such as OWL2RDF [14]. However, we note that the conversion algorithm could be substituted by another without significant changes to the rest of our methodology. Projection rules transform class subsumptions and property assertions between individuals directly into predefined triplets, without loss of information, while more complex logical expressions, such as property restrictions, are approximated with simple triplets that do not keep the exact logical relationships. The result is a graph that presumably approximately captures all relationships but does not contain noisy syntactic structures. This approach produces a directed heterogeneous multigraph, i.e., a set of triplets (edges) of the form $\langle s, p, o \rangle \in T$, where s and o are nodes (classes or individuals) and p is a label representing the relation between them. We use this graph directly as the input to our baselines, which operate on knowledge graphs. For other methods, we further convert this graph into an undirected homogeneous 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. In both cases we discard any additional textual information, such as labels and descriptions of entities. We use undirected graphs instead of directed ones because benchmarking link prediction on directed graphs can be problematic; it may result in increasing scores artificially, since a connection between two nodes can occur in both the training and the test set.

3.2. Ontology Completion Using Link Prediction

Link prediction and ontology completion tasks are closely related since one predicts which edges are in the graph, and the other which relations are missing from the ontology. Therefore, it is crucial to determine how well our model works on the link prediction task. A 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.

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 (nonexistent) 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. Proximity-based methods and GNN predict scores directly, while other methods use logistic regression. We evaluate the performance using the ROC-AUC. An overview of the link prediction process can be seen in Figure 1. In knowledge graph baselines, where the existence of a specific relation is predicted, we score all relations and take the highest score.

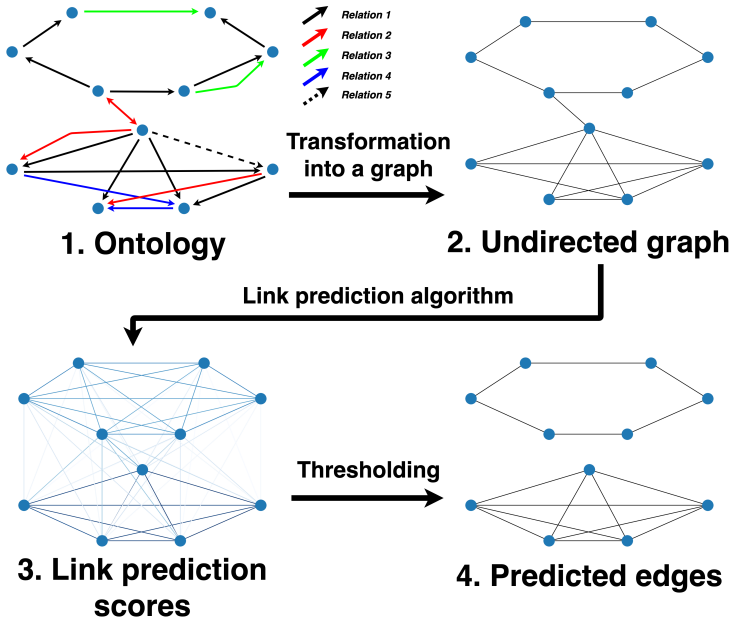


Figure 1. Overview of the link prediction methodology.

3.3. Recommending 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 recommending missing links can help improve the ontology. This section presents an approach for creating such recommendations.

First, we transform the ontology as presented in Section 3.1. Then, we embed the nodes of the generated network into matrix $\boldsymbol{R}$ using a non proximity-based approach (e.g., approaches in Section 4). A row of $\boldsymbol{R}$ represents a node, while a column represents either a symbolic feature (in case of SNoRe) or a latent feature. Using $\boldsymbol{R}$, we then create the link prediction matrix $\boldsymbol{L} = \boldsymbol{R} \cdot \boldsymbol{R}^T$ that is used to find candidates for the missing and redundant connections. For proximity-based approaches, matrix $\boldsymbol{L}$ can be obtained by individually generating scores for each pair of nodes. We split the link prediction matrix into two matrices, one that represents the score of existing edges and one that represents the score of nonexisting edges. The matrix with scores of the existing edges $\boldsymbol{P}$ can be obtained by using the adjacency matrix $\boldsymbol{A}$ as the mask $\boldsymbol{P} = \boldsymbol{L}[\boldsymbol{A}]$ (if matrices $\boldsymbol{A}$ and $\boldsymbol{B}$ are of the same size, we define $\boldsymbol{A}[\boldsymbol{B}] = \boldsymbol{C}$ as $\boldsymbol{C}_{i,j} = \boldsymbol{A}_{i,j}$ if $\boldsymbol{B}_{i,j} = 1$, otherwise $\boldsymbol{C}_{i,j} = 0$). The matrix with scores for the nonexisting edges $\boldsymbol{M}$ can be obtained by subtracting the matrix with the scores of existing edges from the link prediction matrix $\boldsymbol{M} = \boldsymbol{L} - \boldsymbol{P}$. Recommendations for missing edges are obtained by selecting the elements in the matrix of the nonexisting edges with the highest scores. Additionally, recommendations for redundant edges can be obtained by selecting the elements with the lowest (nonzero) scores in the matrix of existing edges. An overview of this methodology is shown in Figure 2.

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

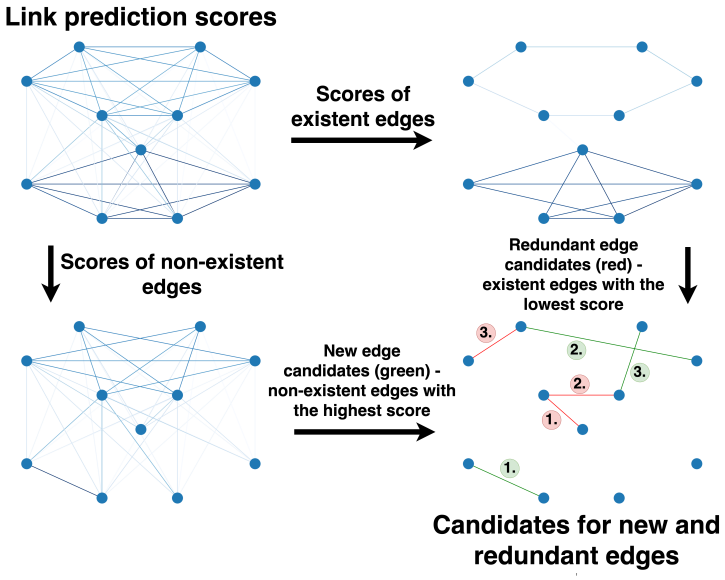


Figure 2. Overview of our methodology for finding missing and redundant edges.

3.4. Explaining Recommendations

An ontology completion tool can have much higher utility to annotators if it can provide some level of explanation for its recommendations. By using a method that produces a symbolic embedding, such as SNoRe, we can explain why the model recommended some specific edges. The following sections presents how to create global explanations, which give the most important features (in our case nodes). Similarly, one can create explanations for each prediction using interpretability methods such as SHAP [43].

Global explanations are useful for finding nodes that contribute the most to the presence of an edge. Since features represent the similarity to some node's neighborhood, this information can be exploited to prioritize nodes with the higher chance of being connected. Specifically, annotators can start with nodes in the neighborhood of the most important feature (a node). To create such explanations, we first train a logistic regression model as in the link prediction benchmark. The importance of features can be estimated by the absolute value of its t-statistic [44]. The t-statistic is calculated as the weight of the feature divided by its standard error. For the jth feature with the weight β_j, the t-statistic is calculated using the formula $\mathrm{SE}(\beta_j) = \sqrt{(\boldsymbol{X}^T \boldsymbol{W} \boldsymbol{X})^{-1}_{jj}}$, where $\boldsymbol{X}$ represents the matrix with training data used as input to the logistic regression, $\boldsymbol{W}$ is a diagonal matrix where $\boldsymbol{W}_{jj} = \frac{e^{\sum_{i=1}^{p} \beta_i x_{ji}}}{(1+e^{\sum_{i=1}^{p} \beta_i x_{ji}})^2}$, and p is the number of features in the embedding.

4. Experimental Setting

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

4.1. Data Sets

In our work, we used the following ontologies:

Marine TLO [45]: 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 such as *legislative zone* or *ecosystem*.

Anatomy Entity Ontology (AEO) [46]: 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 [47]: 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 a *subclass of* relations), with diverse classes such as *symptom*, *laboratory test*, or *anatomical structure*.

Emotion Ontology (MFOEM) [48]: It aims to describe affective phenomena (emotions and moods), their different building blocks, and their effects on human behavior (expressions). Similar to the Anatomy Entity Ontology, this ontology includes more numerous and more 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) [49]: 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) [23]: It aims to name all parts of animals, plants, and fungi that can bear a food role for humans, as well as their derived food products and the processes used to make them. It is a large, fairly grounded ontology with upper-level entities such as *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 [50]: 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 semiautomatic curation.

Gene Ontology (GO) [4] 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*.

We selected these ontologies because they possessed different properties, came from different domains, ranged from small to big ones, and contain both grounded and ungrounded ontologies. Because of this, we believe that the conclusions from our benchmark generalize well to most existing ontologies.

Some basic statistics of these ontologies are shown in Table 2. Of the three bigger ontologies, the Food Ontology has a very treelike structure, while LKN and the Gene Ontology are more connected.

Table 2. Basic statistics of the tested ontologies. Columns $|N|$, $|E|$, and Components describe the converted graphs. The rest describes the raw OWL ontologies.

Ontology	$\|N\|$	$\|E\|$	Components	Classes	Individuals	Object Properties	Subsumption Axioms	Restrictions	Set Axioms
Marine	108	156	2	104	3	92	105	0	0
Anatomy	249	366	1	250	0	11	366	101	0
SCTO	321	370	1	394	18	8	341	251	111
Emotions	631	773	1	688	36	29	774	94	40
EHDAA2	2743	12,894	15	2734	0	9	13,366	10,283	0
Food	28,740	35,897	107	45,942	381	68	39,155	8860	2543
LKN	20,011	68,503	2427	20,011	0	0	68,503	68,503	0
GO	44,167	101,504	1	62,201	0	9	90,583	30,704	23,493

4.2. Baselines

We selected baselines that belonged to different groups. Proximity-based methods such as Adamic/Adar [33], the Jaccard coefficient [34], and preferential attachment [51]

are fast, space-efficient, and require no additional training, but rely on strong assumptions which usually do not hold. Graph neural networks such as GAE [52], GAT [38], GCN [37], and GIN [53] usually work well when we have a lot of data, computational resources, and handcrafted features. Embedding methods SNoRe [36], node2vec [9], and metapath2vec [40] first embed graph nodes into a matrix and then solve a given task with a learner. We classified the existence of a link for this method using logistic regression as it has good performance and interpretability. TransE [10] and RotatE [41] are knowledge graph embedding methods that embed nodes as well as the relationships between them. Lastly, we also used a spectral clustering [54] baseline. A more detailed description of these baselines is the following:

Adamic/Adar [33]: An edge between nodes u and v is scored with $\sum_{x \in N(u) \cap N(v)} \frac{1}{\log |N(x)|}$, where $N(x)$ is the neighborhood of node x. These scores are normalized and thresholded to obtain a link prediction.

Jaccard coefficient [34]: An edge between nodes u and v is scored with $\frac{|N(u) \cap N(v)|}{|N(u) \cup N(v)|}$, where $N(u)$ is the neighborhood of node u. These scores are normalized and thresholded to obtain the link prediction.

Preferential attachment [51]: An edge between nodes u and v is scored with $|N(u)| \cdot |N(v)|$, where $N(u)$ is the neighborhood of node u. These scores are normalized and thresholded to obtain the link prediction.

GAE [52]: Generates a node representation with a variational graph autoencoder that uses latent variables to learn an interpretable model.

GAT [38]: Includes the attention mechanism that helps learn the importance of neighboring nodes. In our tests, we adapted the implementation from PyTorch Geometric [55].

GCN [37]: 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 adapted the implementation from PyTorch Geometric [55].

GIN [53]: Learns a representation that can provably achieve the maximum discriminative power. In our tests, we adapted the implementation from PyTorch Geometric [55].

SNoRe [36]: A node embedding algorithm that produces an interpretable embedding by calculating the similarity between vectors generated by hashing random walks.

node2vec [9]: A node-embedding algorithm that learns a low-dimensional representation of nodes that maximizes the likelihood of neighborhood preservation using random walks.

metapath2vec [40]: 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 [10]: Creates a knowledge graph embedding in such a 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 [41]: 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 a complex vector space.

Spectral clustering [54]: Generates a node embedding by using a nonlinear dimensionality reduction method based on a spectral decomposition of the graph's Laplacian matrix.

4.3. Evaluation

We evaluated the link prediction capabilities on transformed ontologies by using a five-fold cross-validation. We created these folds as follows. We started with a directed (multi)graph with multiple edges between each pair of nodes. We transformed this graph

into a simple undirected graph and removed elements on the diagonal of the adjacency matrix. Afterwards, we took the upper triangle of the adjacency matrix, put the elements into an array, and shuffled them to create positive examples. Selecting only elements from the upper triangle ensured a fair evaluation since each edge was chosen exactly once and thus contained exactly one fold. For negative examples, we randomly sampled pairs of nodes, tested if the edge between them existed, and made sure they did not repeat. We used the same amount of positive and negative examples. We split positive and negative examples into five equally sized parts.

We obtained the score of a baseline on the selected data set by taking the mean value of the scores for each fold. A fold was scored by training the model with data from other folds and using the ROC-AUC to obtain prediction scores for edges in that fold. While the AUC assessed all decision thresholds and thus gave a score that might be too general [56], we believe it sufficiently showed the performance of different baselines in our benchmark. To make our experiments reproducible, we initialized the random number generators of the data splitting algorithm and all baselines with the same, predetermined seed. This way, data splits were the same for each baseline.

TransE and RotatE baselines needed two nodes, as well as the type of the edge between them to predict the edge score. To bypass this, we generated predictions for each relation and output the most probable one. We did this because if there was 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 selected). During the training of these baselines, we maintained the information about the edge type.

5. Results

In this section we present the results of the link prediction and show an example of the global explanations.

5.1. Link Prediction Results

The results of ontology completion using the methodology described in Section 3.2 are presented in Table 3. From the results, we can observe two aspects that generally held for all baselines: the variance of results fell with the number of edges, and baselines performed significantly worse on ontologies where the ratio between nodes and edges was close to one. We can also observe that on smaller ontologies, the embedding methods that did not rely on random walks such as spectral embedding, TransE, and RotatE worked best, while on bigger ones, SNoRe and TransE generally outperformed the others. By grouping baselines of similar kinds together, we see that proximity-based approaches usually gave mediocre performance, and graph neural networks worked well on most data sets but usually fell just below the best performing approaches. Node-embedding algorithms based on random walks generally performed great on all data sets and approaches designed for knowledge graphs performed similarly to other embedding methods. Spectral embedding generally performed better on smaller ontologies, the exceptions being the Marine Ontology. Overall, the best performing baselines (based on the average rank) were SNoRe and TransE.

When comparing training and inference speed, our experiments showed that proximity-based methods performed the fastest, while the knowledge graph embedding methods usually the slowest. We found that the running time of smaller ontologies was less than a second for (almost) any given baseline. On the three bigger ontologies, some proximity-based methods achieved great results even though they were a lot faster; e.g., preferential attachment on LKN. Overall, the slowest method was RotatE, which needed almost twice as much time on the Gene Ontology as TransE, the second slowest. We found that graph neural network methods performed similarly or a bit slower than SNoRe, while achieving lower results overall.

Table 3. Link prediction results based on the ROC-AUC metric (multiplied by 100 to improve readability) and average rank.

Dataset Baseline	Marine	Anatomy	SCTO	Emotions	EHDAA2	FoodOn	LKN	GO	Average Rank
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)	8.88
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)	10.25
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)	6.88
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)	5.38
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)	8.13
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)	6.88
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)	6.63
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)**	**3.63**
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)	7.50
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)	7.63
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)	4.00
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)	7.50
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)	7.75

5.2. Interpretation Examples

To further make our predictions useful, we interpreted them using the methodology presented in Section 3.4. Figure 3 shows an example of the global explanation made for the 2019 Gene Ontology. The term *apoptotic process* had the highest value, meaning that when the neighborhood of two nodes were similar to the neighborhood of this node (a high value in the embedding for this feature), it was more likely that there was an edge between them. In a real-world application, the annotator should start with nodes that are in the neighborhood of the *apoptotic process* node.

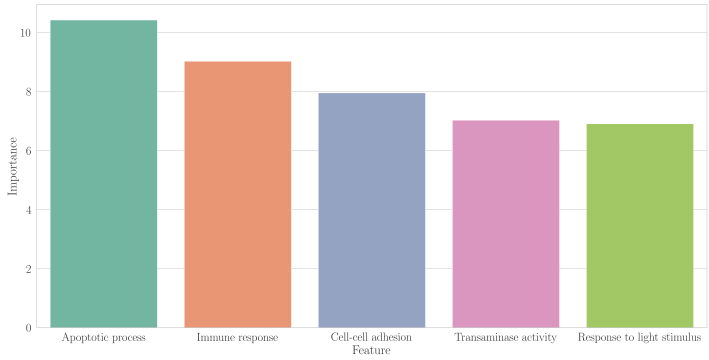
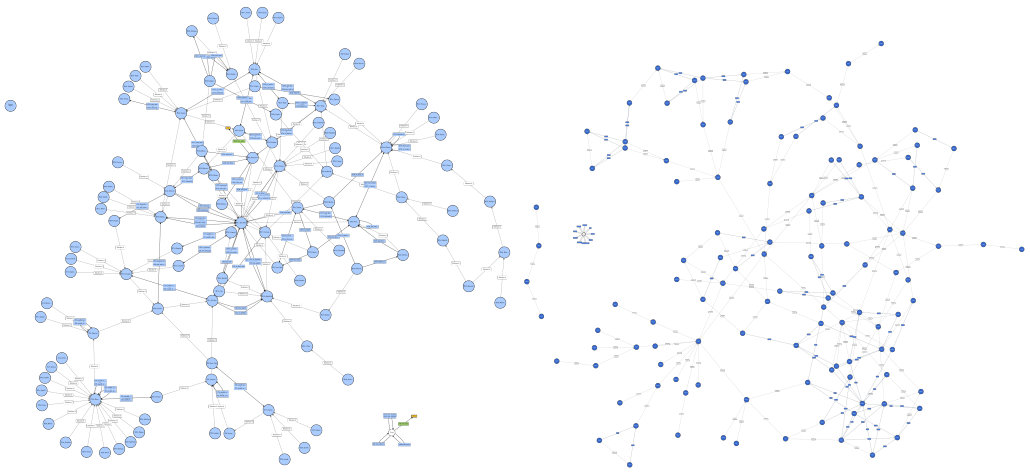


Figure 3. Example of a global explanation for the Gene ontology.

6. Discussion

The main goal of our approach was to simplify and accelerate the process of ontology completion for annotators. We did this by transforming the ontology into a graph and using link prediction approaches to score the edges. We also presented a methodology for recommending the most probable missing (and possibly redundant) links. We empirically evaluated the results of the link prediction on graphs obtained by transforming ontologies. From Tables 2 and 3, we can see that the link prediction worked well on graphs with a high number of average edges per node but badly on graphs whose structure resembled a tree (i.e., average edges per node is close to one). Figure 4a,b show a visualization of the marine and anatomy ontologies. We can see that the marine ontology was well connected through most of the graph but had a lot of leaves, while the anatomy ontology was largely treelike, especially at the bottom left of the figure.



(**a**) Marine ontology. (**b**) Anatomy ontology.

Figure 4. Visualization of the marine ontology (**a**) and anatomy ontology (**b**) using WebVOWL [57].

There are a few disadvantages to the proposed methodology. While the methodology works well on ontologies where nodes are well connected, it performs poorly on the

ones whose structure resembles a tree. Such ontologies probably do not contain enough information to recommend new connections solely based on their structure. A possible solution could be to transform the ontology into a knowledge graph, instead of a simple undirected graph. Another disadvantage is that the approach needs a 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. Since we work with simple graphs, our methodology can only capture simple links and works under the assumption that two nodes are connected if they are similar. This assumption is not necessarily true in ontologies and knowledge graphs, where link can have different semantic meanings (as an example a link might represent that two nodes are opposite). Lastly, link prediction assumes that the distribution of links in the test set corresponds to the distribution of *real* missing links. For ontologies, this assumption might not hold and consequently, results in the paper might not be indicative of the true performance.

Note that this work primarily focused 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 demonstrated how KG specific methods compared to methods that operated on simple graphs. However, there exist even more specialized methods, such as ontology-specific embeddings [12–14]. These approaches are tailored to capture the higher expressivity that ontologies offer compared to knowledge graphs and usually utilize lexical information (metadata) about nodes and relations, which we ignored. Leveraging this additional information would likely produce better results. Other choices could also be made when it comes to ontology preprocessing 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 the ontology-to-graph conversion protocol itself.

Lastly, the presented methodology could help annotate larger ontologies where connections can easily be missed. In practice, a web application can be set up, where the annotator selects some nodes and gets recommended the most probable missing edges.

7. Conclusions

In this work, we proposed a graph-based machine learning approach for ontology completion that is complementary to other domain-knowledge-based approaches. With our approach, an annotator can quickly generate recommendations for the most probable missing relations without the need to fine-tune their approach to the specific ontology. We showed that the methodology yielded good results on larger ontologies when nodes had 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 preprocessing 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 metadata about entities. We suspect that taking full advantage of these additional features could significantly improve the results and further explore the limits of the presented methodology.

Author Contributions: Conceptualization, S.M. and B.Š.; Methodology, S.M., M.B. and B.Š.; Software, S.M.; Resources, N.L.; Data curation, M.B.; Writing—original draft, S.M., M.B. and B.Š.; Writing—review & editing, S.M., N.L. and B.Š.; Visualization, S.M.; Supervision, B.Š.; Project administration, N.L. and B.Š.; Funding acquisition, N.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by European Union's Horizon 2020 grant number 863059.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: All data and code can be found at https://github.com/smeznar/ontology-completion-with-graph-learners (accessed on 1 November 2022).

Acknowledgments: This work was supported by the Knowledge Technologies national research grant, and the European Union's Horizon 2020 research and innovation programme under grant agreement 863059 (FNS-Cloud, Food Nutrition Security).

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Brank, J.; Grobelnik, M.; Mladenić, D. A survey of ontology evaluation techniques. In Proceedings of the Conference on Data Mining and Data Warehouses (SiKDD 2005), Citeseer Ljubljana, Slovenia, 17 October 2005; pp. 166–170.
2. Roche, C. Ontology: A survey. *IFAC Proc. Vol.* **2003**, *36*, 187–192. [CrossRef]
3. Graves, M.; Constabaris, A.; Brickley, D. Foaf: Connecting people on the semantic web. *Cat. Classif. Q.* **2007**, *43*, 191–202. [CrossRef]
4. 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.; et al. Gene ontology: Tool for the unification of biology. The Gene Ontology Consortium. *Nat. Genet.* **2000**, *25*, 25–29. [CrossRef] [PubMed]
5. Zhang, S.B.; Tang, Q.R. Protein–protein interaction inference based on semantic similarity of Gene Ontology terms. *J. Theor. Biol.* **2016**, *401*, 30–37. [CrossRef]
6. Chen, J.; Althagafi, A.; Hoehndorf, R. Predicting Candidate Genes From Phenotypes, Functions, and Anatomical Site of Expression. *Bioinformatics* **2020**, *37*, 853–860. [CrossRef]
7. Jain, N.; Tran, T.K.; Gad-Elrab, M.H.; Stepanova, D. Improving Knowledge Graph Embeddings with Ontological Reasoning. In Proceedings of the Semantic Web (ISWC 2021), Virtual, 24–28 October 2021; Springer International Publishing: Cham, Switzerland, 2021; pp. 410–426.
8. Pesquita, C.; Faria, D.; Falcão, A.; Lord, P.; Couto, F. Semantic Similarity in Biomedical Ontologies. *PLoS Comput. Biol.* **2009**, *5*, e1000443. [CrossRef]
9. 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, San Francisco, CA, USA, 13–17 August 2016; ACM: New York, NY, USA, 2016; pp. 855–864. [CrossRef]
10. Bordes, A.; Usunier, N.; García-Durán, A.; Weston, J.; Yakhnenko, O. Translating Embeddings for Modeling Multi-relational Data. In Proceedings of the Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013, Lake Tahoe, NV, USA, 5–10 December 2013; pp. 2787–2795.
11. Chen, M.; Tian, Y.; Chen, X.; Xue, Z.; Zaniolo, C. On2Vec: Embedding-based Relation Prediction for Ontology Population. In Proceedings of the 2018 SIAM International Conference on Data Mining, SDM, San Diego Marriott Mission Valley, San Diego, CA, USA, 3–5 May 2018; SIAM: Philadelphia, PA, USA, 2018; pp. 315–323. [CrossRef]
12. Smaili, F.Z.; Gao, X.; Hoehndorf, R. Onto2Vec: Joint vector-based representation of biological entities and their ontology-based annotations. *Bioinformatics* **2018**, *34*, i52–i60. [CrossRef]
13. Smaili, F.Z.; Gao, X.; Hoehndorf, R. OPA2Vec: Combining formal and informal content of biomedical ontologies to improve similarity-based prediction. *Bioinformatics* **2018**, *35*, 2133–2140. [CrossRef]
14. Chen, J.; Hu, P.; Jiménez-Ruiz, E.; Holter, O.; Antonyrajah, D.; Horrocks, I. OWL2Vec*: Embedding of OWL Ontologies. *Mach. Learn.* **2021**, *110*, 1813–1845. [CrossRef]
15. Tran, H.D.; Stepanova, D.; Gad-Elrab, M. H.; Lisi, F.A.; Weikum G. Towards Nonmonotonic Relational Learning from Knowledge Graphs. In *International Conference on Inductive Logic Programming*; Springer: Cham, Switzerland, 2017; pp. 94–107.
16. Silla, C.; Freitas, A. A survey of hierarchical classification across different application domains. *Data Min. Knowl. Discov.* **2011**, *22*, 31–72. [CrossRef]
17. Li, N.; Schockaert, S. Ontology Completion Using Graph Convolutional Networks. In Proceedings of the SEMWEB, Auckland, New Zealand, 26–30 October 2019.
18. Hitzler, P.; Kroetzsch, M.; Parsia, B.; Patel-Schneider, P.; Rudolph, S. OWL Web Ontol. Lang. Primer (Second Edition). *W3C Recomm.* **2009**, *27*, 123.

19. Bonatti, P.; Decker, S.; Polleres, A.; Presutti, V. Knowledge Graphs: New Directions for Knowledge Representation on the Semantic Web (Dagstuhl Seminar 18371). *Dagstuhl Rep.* **2018**, *8*, 29–111.
20. Kejriwal, M. What Is a Knowledge Graph? In *Domain-Specific Knowledge Graph Construction*; Springer: Cham, Switzerland, 2019; pp. 1–7. [CrossRef]
21. Ehrlinger, L.; Wöß, W. Towards a Definition of Knowledge Graphs. *SEMANTiCS (Posters Demos SuCCESS)* **2016**, *48*, 2.
22. Dragoni, M.; Bailoni, T.; Maimone, R.; Eccher, C. HeLiS: An Ontology for Supporting Healthy Lifestyles. In *International Semantic Web Conference*; Vrandečić, D., Bontcheva, K., Suárez-Figueroa, M.C., Presutti, V., Celino, I., Sabou, M., Kaffee, L.A., Simperl, E., Eds.; Springer: Cham, Switzerland, 2018; pp. 53–69.
23. 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. FoodOn: A harmonized food ontology to increase global food traceability, quality control and data integration. *NPJ Sci. Food* **2018**, *2*, 23. [CrossRef]
24. Zhao, Y.; Fu, G.; Wang, J.; Guo, M.; Yu, G.-X. Gene Function Prediction based on Gene Ontology Hierarchy Preserving Hashing. *Genomics* **2018**, *111*, 334–342. [CrossRef]
25. Nunes, S.; Sousa, R.; Pesquita, C. Predicting Gene-Disease Associations with Knowledge Graph Embeddings over Multiple Ontologies. *arXiv* **2021**, arXiv:2105.04944.
26. Kulmanov, M.; Smaili, F.Z.; Gao, X.; Hoehndorf, R. Semantic similarity and machine learning with ontologies. *Briefings Bioinform.* **2020**, *22*, bbaa199. [CrossRef]
27. Althubaiti, S.; Karwath, A.; Dallol, A.; Noor, A.; Alkhayyat, S.; Alwassia, R.; Mineta, K.; Gojobori, T.; Beggs, A.; Schofield, P.; et al. Ontology-based prediction of cancer driver genes. *Sci. Rep.* **2019**, *9*, 17405. [CrossRef]
28. Costa, L.F.; Oliveira, O.N., Jr.; Travieso, G.; Rodrigues, F.A.; Villas Boas, P.R.; Antiqueira, L.; Viana, M.P.; Correa Rocha, L.E. Analyzing and modeling real-world phenomena with complex networks: A survey of applications. *Adv. Phys.* **2011**, *60*, 329–412. [CrossRef]
29. Bhagat, S.; Cormode, G.; Muthukrishnan, S. Node Classification in Social Networks. In *Social Network Data Analytics*; Springer: Boston, MA, USA, 2011; pp. 115–148.
30. Lü, L.; Zhou, T. Link prediction in complex networks: A survey. *Phys. A Stat. Mech. Its Appl.* **2011**, *390*, 1150–1170. [CrossRef]
31. Xiaojin, Z.; Zoubin, G. *Learning from Labeled and Unlabeled Data with Label Propagation*; Technical Report CMU-CALD-02–107; Carnegie Mellon University: Pittsburgh, PA, USA, 2002.
32. Page, L.; Brin, S.; Motwani, R.; Winograd, T. The PageRank Citation Ranking: Bringing Order to the Web. Stanford Info Lab Technical Report. November 1999. pp. 1966–1999. Available online: http://ilpubs.stanford.edu:8090/422/ (accessed on 1 November 2022).
33. Adamic, L.A.; Adar, E. Friends and neighbors on the Web. *Soc. Netw.* **2003**, *25*, 211–230. . [CrossRef]
34. Salton, G.; McGill, M.J. *Introduction to Modern Information Retrieval*; International Student Edition; McGraw-Hill: New York, NY, USA, 1983.
35. Perozzi, B.; Al-Rfou, R.; Skiena, S. DeepWalk: online learning of social representations. In Proceedings of the The 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD'14), New York, NY, USA, 24–27 August 2014; ACM: New York, NY, USA, 2014; pp. 701–710. [CrossRef]
36. Mežnar, S.; Lavrač, N.; Škrlj, B. SNoRe: Scalable Unsupervised Learning of Symbolic Node Representations. *IEEE Access* **2020**, *8*, 212568–212588. [CrossRef]
37. Kipf, T.N.; Welling, M. Semi-Supervised Classification with Graph Convolutional Networks. In Proceedings of the 5th International Conference on Learning Representations (ICLR 2017), Toulon, France, 24–26 April 2017.
38. Velickovic, P.; Cucurull, G.; Casanova, A.; Romero, A.; Liò, P.; Bengio, Y. Graph Attention Networks. In Proceedings of the 6th International Conference on Learning Representations (ICLR), Vancouver, BC, Canada, 30 April–3 May 2018.
39. Liben-Nowell, D.; Kleinberg, J. The link-prediction problem for social networks. *J. Am. Soc. Inf. Sci. Technol.* **2007**, *58*, 1019–1031. [CrossRef]
40. Dong, Y.; Chawla, N.V.; Swami, A. metapath2vec: Scalable Representation Learning for Heterogeneous Networks. In Proceedings of the Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Halifax, NS, Canada, 13–17 August 2017; ACM: New York, NY, USA, 2017; pp. 135–144. [CrossRef]
41. Sun, Z.; Deng, Z.-H.; Nie, J.-Y.; Tang, J. RotatE: Knowledge Graph Embedding by Relational Rotation in Complex Space. In Proceedings of the 7th International Conference on Learning Representations (ICLR 2019), New Orleans, LA, USA, 6–9 May 2019.
42. Soylu, A.; Kharlamov, E.; Zheleznyakov, D.; Jiménez-Ruiz, E.; Giese, M.; Skjæveland, M.; Hovland, D.; Schlatte, R.; Brandt, S.; Lie, H.; et al. OptiqueVQS: A Visual Query System over Ontologies for Industry. *Semant. Web* **2017**, *9*, 627–660. [CrossRef]
43. Lundberg, S.M.; Lee, S.-I. A Unified Approach to Interpreting Model Predictions. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017; Curran Associates Inc.: Red Hook, NY, USA, 2017; pp. 4768–4777.
44. Molnar, C. Interpretable Machine Learning. 2019. Available online: https://christophm.github.io/interpretable-ml-book/ (accessed on 1 November 2022).

45. Tzitzikas, Y.; Alloca, C.; Bekiari, C.; Marketakis, Y.; Fafalios, P.; Doerr, M.; Minadakis, N.; Patkos, T.; Candela, L. 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, 19–22 November 2013.
46. Bard, J. The AEO, an Ontology of Anatomical Entities for Classifying Animal Tissues and Organs. *Front. Genet.* **2012**, *3*, 18. [CrossRef]
47. El-Sappagh, S.; Franda, F.; Ali, F.; Kwak, K.-S. SNOMED CT standard ontology based on the ontology for general medical science. *BMC Med. Inform. Decis. Mak.* **2018**, *18*, 76. [CrossRef]
48. Hastings, J.; Ceusters, W.; Smith, B.; Mulligan, K. Dispositions and Processes in the Emotion Ontology. *CEUR Workshop Proc.* **2011**, *833*, 71–78.
49. Bard, J. A new ontology (structured hierarchy) of human developmental anatomy for the first 7 weeks (Carnegie stages 1-20). *J. Anat.* **2012**, *221*, 406–416. [CrossRef]
50. Ramšak, Ž.; Coll, A.; Stare, T.; Tzfadia, O.; Baebler, Š.; Van de Peer, Y.; Gruden, K. Network Modeling Unravels Mechanisms of Crosstalk between Ethylene and Salicylate Signaling in Potato. *Plant Physiol.* **2018**, *178*, 488–499. [CrossRef]
51. Barabási, A.L.; Albert, R. Emergence of Scaling in Random Networks. *Science* **1999**, *286*, 509–512. [CrossRef]
52. Kipf, T.N.; Welling, M. Variational Graph Auto-Encoders. *arXiv* **2016**, arXiv:1611.07308.
53. Xu, K.; Hu, W.; Leskovec, J.; Jegelka, S. How Powerful are Graph Neural Networks? In Proceedings of the 7th International Conference on Learning Representations (ICLR), New Orleans, LA, USA, 6–9 May 2019.
54. Ng, A.Y.; Jordan, M.I.; Weiss, Y. On Spectral Clustering: Analysis and an algorithm. In Proceedings of the Advances in Neural Information Processing Systems 14, Neural Information Processing Systems: Natural and Synthetic, NIPS 2001, Vancouver, BC, Canada, 3–8 December 2001; Dietterich, T.G., Becker, S., Ghahramani, Z., Eds.; MIT Press: Cambridge, MA, USA, 2001; pp. 849–856.
55. Fey, M.; Lenssen, J.E. Fast Graph Representation Learning with PyTorch Geometric. *arXiv* **2019**, arXiv:1903.02428.
56. Carrington, A.M.; Manuel, D.G.; Fieguth, P.; Ramsay, T.O.; Osmani, V.; Wernly, B.; Bennett, C.; Hawken, S.; Magwood, O.; Sheikh, Y.; et al. Deep ROC Analysis and AUC as Balanced Average Accuracy, for Improved Classifier Selection, Audit and Explanation. *IEEE Trans. Pattern Anal. Mach. Intell.* **2022**. [CrossRef] [PubMed]
57. Lohmann, S.; Negru, S.; Haag, F.; Ertl, T. Visualizing Ontologies with VOWL. *Semant. Web* **2016**, *7*, 399–419. [CrossRef]