Predicting missing annotations in Gene Ontology with Knowledge Graph Embeddings and True Path Rule

Özge Erten[1,*], Shervin Mehryar[1], Remzi Çelebi[1] and Christopher Brewster[1,2]

[1] *Institute of Data Science, Maastricht University, Paul-Henri Spaaklaan 1, 6229 GT, Maastricht, Netherlands*
[2] *Data Science Group, TNO, Kampweg, Soesterberg, Netherlands*

Abstract

Gene Ontology (GO) and its Annotations (GOA) provide a controlled and evolving vocabulary for gene products and gene functions widely used in molecular biology. GO & GOA are updated and maintained both automatically from biological publications and manually by curators. These knowledge bases however are often incomplete for two reasons: 1) Research in biological domain itself is still ongoing; 2) The amount of experimental evidence might not be yet sufficient to validate annotations. In this paper, we address the gap in evidence between gene products and their annotations by making link predictions using Knowledge Graph Embedding (KGE) methods. Through the application of the True Path Rule (TPR) in the training stage of KGE, we were able to improve the performance of traditional KGE methods. We report two experimental scenarios with GO and GO Chicken Annotation datasets to show the contribution of embedding TPR to prediction accuracy.

Keywords

Link prediction, True path rule, Knowledge graph embeddings, Predicting Gene Ontology Annotations

1. Introduction

One successful application of KGs and ontologies in bio-medicine is the Gene Ontology (GO). In particular, the Gene Ontology Consortium has used ontologies, namely the Gene Ontology and its Annotations (GOA), to provide a common vocabulary for gene functions since 1998. GO&GOA support biological studies and experiments such as gene enrichment analysis by providing a hierarchical gene function semantic model. Moreover, as new biomedical information emerges, these ontologies are updated on a monthly basis. Despite regular updates, GO remains incomplete because of the inherent complexity of biological domain, and the large size of data which can not be validated easily or rapidly. As a result, several approaches have been developed to address the incompleteness issue in KGs. One of several approaches is to use Knowledge Graph Embeddings (KGEs) to predict the missing links, and this will be our focus in this paper [1, 2, 3].

This paper aims to improve the prediction performance of KGEs for the purpose of GO annotations by using the True Path Rule (TPR) used by the Gene Ontology. Namely, TPR is a natural outcome of the hierarchical structure of GO, and it declares:*"If a gene function can annotate a gene product, the ancestral classes of that gene function can also annotate that gene product. However, if a gene function can not annotate a gene product, then it can not be annotated by the descendant classes of that gene function."*. We propose a methodology for training KGEs that incorporates TPR to make the learned embeddings more suitable for inference rules [4].

Previous work by Valentini [5] has formulated the gene function prediction issue as a hierarchical classification problem based on the True Path Rule. This study introduces an ensemble algorithm based on TPR with three main steps. The first step is that a group of classifiers makes a local decision

SWAT4HCLS 2023: The 14th International Conference on Semantic Web Applications and Tools for Health Care and Life Sciences, February 13–16, 2023, Basel, Switzerland

[*] Corresponding author.

o.erten@maastrichtuniversity.nl (Ö. Erten); shervin.mehryar@maastrichtuniversity.nl (S. Mehryar); remzi.celebi@maastrichtuniversity.nl (R. Çelebi); christopher.brewster@maastrichtuniversity.nl (C. Brewster)

CEUR Workshop Proceedings (CEUR-WS.org)

for each GO class in the graph. The second step is that if the classifier assigns a positive label for a class, the parent classes also have that label, but negative labels do not propagate from the bottom up. The third step is that if it is labeled with a negative label for a class, it also assigns all of its child classes to negative labels. Positive labels do not affect the lower classes in the GO hierarchy. In the experiments, TPR based ensemble performs better than other ensemble algorithms.

Kulmanov et al.[6] describe how ontologies can be used to provide background knowledge in machine learning-based semantic similarity tasks. The distance similarity or the similarity of belonging to a particular subject between the elements in representative learning plays an important role in model training. To observe this similarity contribution, they evaluate various ontology embedding techniques. One of the experiments was done by adding GO semantics with TPR to two neural network-based methods, and the both experimental results show an increase in prediction scores.

In this work, we defined two experimental designs with a dataset that consists of GOA versions from 2018 to current (2022). The GOA versions are considered and treated as pairs. For the training set, both experiments use the earlier version in the pair as well as the subsumption classes in GO. The newer version is used for differentiating comparison with the prior version and detecting newly added annotations. The testing and validation datasets in the first scenario take into account only those captured, newly added annotations. The second scenario adds implicit annotations that are captured in the GO hierarchy by TPR to the test and validation sets.

2. Methodology

KGEs are the vector embeddings learned from a set of triples describing facts in a KG. KGEs can subsequently be used to perform reasoning tasks such as link prediction and entity classification. Typically, KG embedding methods embeds entities and relations onto a vector space directly where each triple (head entity, relation, tail entity) in the KG is assigned a score based on its validity. The sum of scores (i.e. loss) for positive and negative triple set is optimized during training. In this paper, we applied the KGE methods to GO and its Annotations to predict missing or future annotations. To further capture and embed the TPR, we generate and incorporate samples using the TPR in the training data [7].

In detail, we formulate the task as follows: Given a KG $\mathcal{G}$ represented with a relation r between entities e_1 and e_2. First, the optimal vector embeddings are learned for all entities and relations. The corresponding vector space can also be denoted by the following relation: $\vec{e_2} = \vec{e_1} + \vec{r}$, and it represents single triples in the ontology in the form $(e_1, r, e_2) \in \mathcal{G}$. Then, the following criterion is used to learn the embeddings for link prediction with given a set $\mathcal{S}$ of triples representing facts in the KG:

$$L_{tri} = \sum_{(s,r,o)\in\mathcal{S}} ||\vec{s} + \vec{r} - \vec{o}||_2, \tag{1}$$

where as before, $\vec{s}$, $\vec{r}$, and $\vec{o}$ are vector representations in $\mathbb{R}^d$ corresponding to head entity, relation, and tail entity in the ontology. These representations are learned using the triples in the data and embedded as a d-dimensional vector similar to the process in TransE [8]. Our contribution is adding samples following the TPR as shown in Figure 1 to the set $\mathcal{S}$. Essentially, we distinguish between direct gene product-and-function relations and higher level gene product-and-function relations. In the first scenario, embeddings are learned using the TransE model on existing triplets in the dataset. In the second scenario, we enrich the training data with additional samples from gene products inheriting their first-level ancestry functions as well as second-level ancestry functions. These additional samples serve to improve embedding qualities and we refer to this method as TransE+TPR. The detail on dataset creation and the different scenarios are given in the next section, and the code is accessible on our Github: https://github.com/ozyygen/predict-KGE-TPR.

3. Dataset

We generate four datasets by using GOA versions from 2018 to current (2022). Each dataset contains a version pair that is selected with one year window length. We use the prior version to generate the training set, and the latter for generating the testing and validation sets. Namely, each set contains triples consist of a gene product as head, a gene function as tail, and the type in which the gene function annotates that gene product as relation. Additionally, we add "is a" and "part of" semantic of gene functions and TPR-inferred annotations from GO into the training set. In Table 1, triple counts for each dataset are given for training, testing and validation sets.

Table 1
Datasets triple counts

	GOA18-19	GOA19-20	GOA20-21	GOA21-22
Train set	48.924	35.304	29.332	24.948
Valid set scenario-1	3.457	593	871	831
Valid set scenario-2.1	14.310	7.051	4.797	2.869
Valid set scenario-2.2	29.241	16.106	10.304	5.829
Test set scenario-1	3.458	593	872	830
Test set scenario-2.1	14.310	7.052	4.797	2.868
Test set scenario-2.2	29.241	16.106	10.304	5.829

4. Experimental Design

In this work, two experimental designs were studied for KGE methods across four versions and three scenarios. Figure 1 shows training, testing and validation split on a toy data. Node 8, 9,10, 11, representing gene functions, annotate X1, X2, X3 and X4 gene products respectively in the figure. Solid red line denotes the first version relations between gene products and gene functions, and dashed red lines represent relations for the second version of the dataset [2].

Scenario 1: Two consecutive versions of GOA were used to generate the dataset for training the embedding model. Specifically, the prior version was used to generate the training set. We also added the related GO subsumption classes and TPR-inferred annotations in the training set in order to enrich semantic information in the KG. The test and validation sets were created with the latter version of the pair by randomly splitting the triples (annotations) into a test set and a validation set by a ratio of 0.5. We excluded the triples that contain a new gene product or a new gene function which were not present in the prior version. This scenario is denoted by sc-1.

Scenario 2: In Scenario 2, we used the same training set with Scenario 1, but we extended Scenario 1 test set with the implicit relations obtained from the TPR. We added relations that can be infered by TPR to the test set. We infer these relations by applying the following rule; if a gene product is annotated by a gene function in the training set, then the gene product need to be annotated by the ancestral classes of that gene function. The objective of this addition is to observe whether our method can predict the implicit links inferred by the TPR.

To observe the effect of the TPR at the different level depth, we designed Scenario 2.1 and Scenario 2.2 with two different super classes depth:

- `Scenario 2.1`: In this scenario, we generated implicit annotations with TPR using the first level ancestors of gene functions. This scenario is denoted by sc-2.1.
- `Scenario 2.2`: For this scenario, we considered second level ancestors of gene functions in addition to the first level ancestors. This scenario is denoted by sc-2.2.

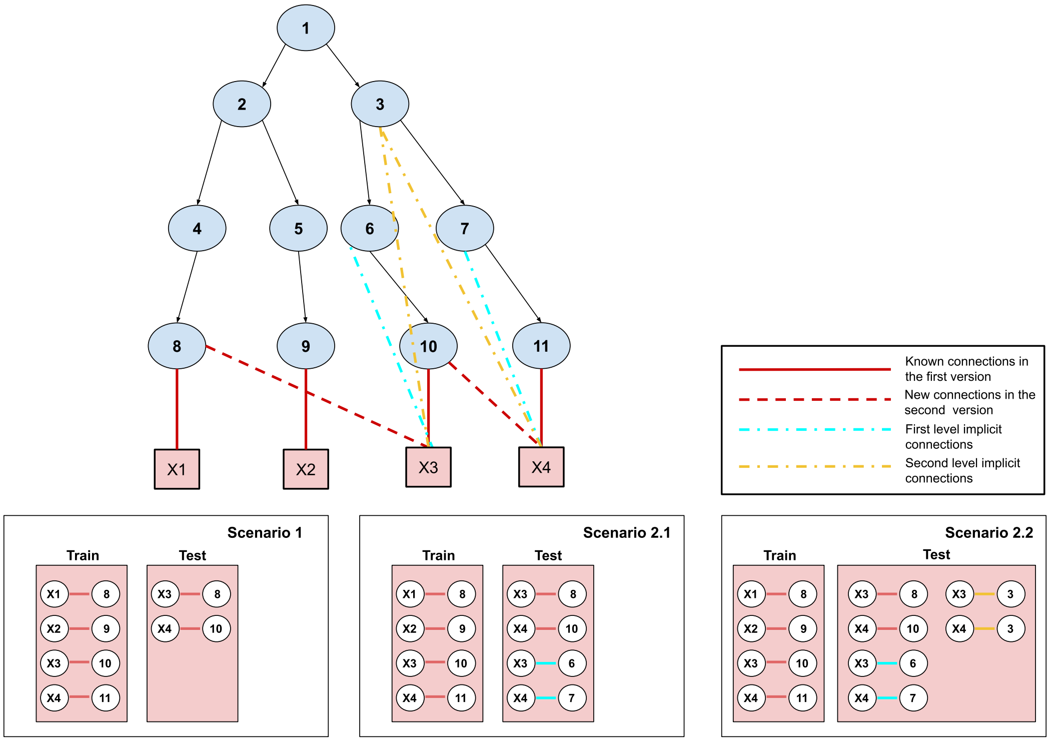
1
2
3
4
5
6
7
8
9
10
11
X1
X2
X3
X4
Known connections in the first version
New connections in the second version
First level implicit connections
Second level implicit connections
Scenario 1
Train
Test
Scenario 2.1
Train
Test
Scenario 2.2
Train
Test

Figure 1: Sample train, test and validation sets. Numbered blue nodes and their relations denote a sample GO hierarchy, and pink boxes denote gene products (X1, X2, X3, X4). The three scenarios generate different test sets. Scenario 1 will generate a test set which contains the newly added annotations in the second version (X3-8 and X4-10, denoted by dashed red line). In addition to these annotations, Scenario 2.1 test set will also include the implicit links (X3-6 and X4-7, denoted by dashed blue line) between gene products of the test data and the super classes of gene functions at one upper level. Scenario 2.2 will further add annotations from the second upper classes of gene functions (X3-3 and X4-3, denoted by dashed yellow line). The test set is then split into a test set and a validation set by a ratio of 0.5.

5. Result and Conclusion

We conducted experiments with TransE and TransE+TPR to find out the efficacy of TPR into the link prediction accuracy. The results are shown in Table 2.

We repeated the experiments with different time windows and scenarios. Four datasets help to observe whether the train, test and validation set triple count has an effect on predictions. Scenario 1 does not have any inferred annotations. For Scenario 2, we added TPR-inferred GO semantic to compare the scores with scenario-1. The table has two methods for four dataset with three scenarios. Accordingly, TransE Hit@10 scores show hierarchical semantic addition in dataset does not have a significant impact on improving prediction accuracy. On the contrary in TransE+TPR method, the rule contribute increasing the accuracy.

Specifically, we implement TransE and TPR and evaluate on different GOA datasets. The dataset is enriched with TPR-inferred annotations and GO subsumption classes. The results show significant increase in the accuracy when rules are applied during training process. Particularly, in the best case scenario the proposed method performance in terms of Hits@10 is at average 0.6275 ± 0.0814 in comparison to average Hits@10 of 0.1037 ± 0.0094 from TransE. This approximately 0.52 gain in performance is attributed to the importance of hierarchical information captured by the model through TPR samples, as explained in the previous section.

Table 2
Link prediction scores for TransE and TransE+TPR

		TransE		TransE+TPR	
	Scenario	MRR	Hits@10	MRR	Hits@10
GOA18-19	sc-1	0.0321	0.0933	0.2552	0.5857
	sc-2.1	0.0397	0.0981	0.1347	0.4186
	sc-2.2	0.0505	0.1098	0.1692	**0.6100**
GOA19-20	sc-1	0.0478	0.1433	0.1971	0.4849
	sc-2.1	0.0338	0.0927	0.2021	**0.6800**
	sc-2.2	0.0438	0.1080	0.1967	0.6609
GOA20-21	sc-1	0.0518	0.1439	0.1807	0.4738
	sc-2.1	0.0441	0.1132	0.2340	**0.7066**
	sc-2.2	0.0501	0.1376	0.1589	0.5674
GOA21-22	sc-1	0.0396	0.1126	0.1387	0.3762
	sc-2.1	0.0426	0.0990	0.1669	**0.5207**
	sc-2.2	0.0466	0.1149	0.1711	0.5700

Even though TransE+TPR method achieved the highest accuracy scores for Scenario 2 almost each dataset, further study requires to determine the optimal depth for hierarchical class addition of GO semantic to receive the best prediction accuracy. Also, distinguishing annotations based on evidence, such as types of experiments or automatically generated, then treating them accordingly might have an impact on prediction accuracy. Furthermore, we think that training a KGE with several versions of the data will enhance the effectiveness of the KGE in link prediction. Lastly, the training-test split, which takes into account gene traits such orthology, can be used to test link prediction stability. We leave these topics open to be covered in future work.

References

[1] J. A. Blake, Ten quick tips for using the gene ontology, PLoS computational biology 9 (2013) e1003343.

[2] G. O. Consortium, The gene ontology resource: 20 years and still going strong, Nucleic acids research 47 (2019) D330–D338.

[3] M. Wang, L. Qiu, X. Wang, A survey on knowledge graph embeddings for link prediction, Symmetry 13 (2021) 485.

[4] Y. Zhao, J. Wang, J. Chen, X. Zhang, M. Guo, G. Yu, A literature review of gene function prediction by modeling gene ontology, Frontiers in Genetics 11 (2020) 400.

[5] G. Valentini, True path rule hierarchical ensembles, in: International Workshop on Multiple Classifier Systems, Springer, 2009, pp. 232–241.

[6] M. Kulmanov, F. Z. Smaili, X. Gao, R. Hoehndorf, Semantic similarity and machine learning with ontologies, Briefings in bioinformatics 22 (2021) bbaa199.

[7] Y. Dai, S. Wang, N. N. Xiong, W. Guo, A survey on knowledge graph embedding: Approaches, applications and benchmarks, Electronics 9 (2020) 750.

[8] A. Bordes, N. Usunier, A. Garcia-Duran, J. Weston, O. Yakhnenko, Translating embeddings for modeling multi-relational data, Advances in neural information processing systems 26 (2013).