1 Introduction

Link prediction is a field in graph-based machine learning that aims to predict novel relationships between entities. When applied to biomedical knowledge graphs, link prediction can be a versatile and powerful method of hypothesis generation, e.g., for drug discovery (Abbas et al., 2021). A vast array of machine learning link prediction algorithms emerged over the last decade. Most of these are black boxes such as embedding- or deep learning based algorithms, where the rationale for a prediction is difficult or impossible to explain. Explainability, however, is important for facilitating scientific insight, judging the plausibility of predictions and improving acceptance by end-users (Adadi and Berrada, 2018).

Previous work on explainable link prediction (Meilicke et al., 2019) has limitations that decrease its utility for real-world applications involving large biomedical knowledge graphs, such as lacking scalability, limited predictive performance, or lack of end-user interfaces for viewing and understanding predictions and their explanations.

To address these issues, we release the LinkExplorer software suite for predicting, exploring and explaining links in large knowledge graphs. It integrates SAFRAN (Ott et al., 2021), a novel, state-of-the-art rule-based link prediction algorithm that our group developed. SAFRAN is highly scalable, outperforms other explainable link prediction methods on standard link prediction benchmarks, narrows the performance gap between explainable and black-box algorithms, and produces highly condensed and transparent explanations (Ott et al., 2021). The LinkExplorer suite offers a web-based interface for navigating existing links between entities and relations together with predicted links and their explanations generated by SAFRAN, thereby enabling unprecedented insight into the predictions of a state-of-the-art link prediction algorithm. We also report highly competitive evaluation results of our explainable link prediction algorithm on several large-scale biomedical knowledge graphs.

2 Software architecture and performance

In the following we provide a brief overview of the SAFRAN and AnyBURL algorithms. For a detailed description of the algorithms we refer to (Ott et al., 2021).

AnyBURL (Meilicke et al., 2019) is a walk-based (bottom-up) method based on the idea that sampled paths (random walks) in a knowledge graph are examples of very specific rules and thus can be transformed into more general logical first-order Horn clauses. In each iteration of the rule mining
algorithm, AnyBURL samples paths and generalizes them into rules of three predefined types. Examples of rules generalized from a sampled path can be seen in Table 1. Furthermore AnyBURL calculates a confidence for each rule, the probability that an entity predicted by the rule is correct. This confidence is the relative proportion of correctly predicted entities in all predicted entities by a rule when applied to the training set. In order to correct for unseen entities a constant $p_c > 0$ is added to the denominator. As can be seen in Figure 1, LinkExplorer displays the confidence, number of correctly predicted and number of all predicted entities of a rule. In the example of Figure 1 $p_c$ was set to 5.

![cytochrome P450 family 2 subfamily C member 18 can be overexpressed in liver](image)

**Fig. 1.** Structured view of rules that triggered a prediction.

The rule-application framework SAFRAN takes rules generated by AnyBURL as input, and improves upon AnyBURL’s rule application method in two important ways. First, SAFRAN was engineered to scale to large knowledge bases, which is essential for many biomedical applications. Second, SAFRAN introduces an algorithm that identifies and clusters redundant rules, i.e. rules that are structurally different but have highly correlated predictions. The predictions of rule clusters can then be aggregated with a noisy-or operation, which would not work well when redundancies are not accounted for. The noisy-or aggregation makes it possible to combine predictions of different rules / rule clusters in a meaningful way and improves predictive accuracy. Furthermore, rule clustering makes generated explanations easier to understand. Each prediction made by SAFRAN can be explained in terms of the rules / rule clusters and instantiations that triggered it. A screenshot of LinkExplorer providing explanations for the prediction cytochrome P450 family 2 subfamily C member 18 can be overexpressed in liver is shown in Figure 1.

LinkExplorer uses RDF* (Hartig and Thompson, 2019) graphs to store nodes and edges of knowledge graphs which are served via a SPARQL* endpoint. Furthermore these graphs include metadata of entities such as labels, descriptions and types. In addition to loading knowledge graphs hosted on the server, users can load custom knowledge graphs by specifying external SPARQL* endpoints.

![cytochrome P450 family 2 subfamily C member 18 can be overexpressed in liver](image)

**Table 1. Examples of rules that can be generalized from the sampled path**

<table>
<thead>
<tr>
<th>Rule</th>
<th>Confidence</th>
<th>Correctly predicted</th>
<th>Predicted</th>
</tr>
</thead>
<tbody>
<tr>
<td>speaks(Y, X) ← lives(X, A), lang(A, Y)</td>
<td>0.58571</td>
<td>41</td>
<td>65</td>
</tr>
<tr>
<td>speaks(english, X) ← lives(X, A), lang(A, english)</td>
<td>0.44218</td>
<td>65</td>
<td>142</td>
</tr>
<tr>
<td>speaks(Y, max) ← lives(X, A), lang(A, Y)</td>
<td>0.42056</td>
<td>90</td>
<td>209</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Model</th>
<th>MRR h@1</th>
<th>h@10</th>
<th>MRR h@1</th>
<th>h@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>RESCAL</td>
<td>0.320</td>
<td>0.212</td>
<td>0.544</td>
<td>0.330</td>
</tr>
<tr>
<td>TransE</td>
<td>0.280</td>
<td>0.175</td>
<td>0.500</td>
<td>0.330</td>
</tr>
<tr>
<td>DistMult</td>
<td>0.300</td>
<td>0.193</td>
<td>0.521</td>
<td>0.351</td>
</tr>
<tr>
<td>ComplEx</td>
<td>0.319</td>
<td>0.211</td>
<td>0.547</td>
<td>0.362</td>
</tr>
<tr>
<td>ConvE</td>
<td>0.288</td>
<td>0.186</td>
<td>0.510</td>
<td>0.290</td>
</tr>
<tr>
<td>RotatE</td>
<td>0.286</td>
<td>0.180</td>
<td>0.511</td>
<td>0.272</td>
</tr>
</tbody>
</table>

**Table 2. Comparison of the fully explainable algorithms SAFRAN and AnyBURL with several established latent models on the datasets**

OpenBioLink, PheKnowLator, Hetionet and ogbl-biokg. SAFRAN (denoted with *) is our approach. The best overall results are marked in bold and the best explainable results are underlined.

The user interface of LinkExplorer was realized using a client-server architecture and is implemented in Javascript and HTML5. The client was created using the ReactJS framework and the server-sided Javascript components are served using NodeJS. The server is used to provide predictions and explanations which are retrieved from SQLite databases generated by SAFRAN. LinkExplorer provides explanation files for all hosted knowledge graphs and allows users to load local prediction files.

We make a public LinkExplorer instance available that hosts three large biomedical knowledge graphs: OpenBioLink (Breit et al., 2020), Hetionet (Himmelstein et al., 2017) and PheKnowLator (Callahan et al., 2020). Additionally, the instance contains two widely-used general-domain evaluation datasets: WN18RR (Dettmers et al., 2018) and YAGO3-10 (Dettmers et al., 2018). As there is no publicly available dataset split of Hetionet and PheKnowLator, they were randomly split using a 90-5-5 split ratio per relation. Prediction files were generated by SAFRAN using rules that were learned by AnyBURL for 1000 seconds using 22 threads of a machine with 24 physical (48 logical) Intel(R) Xeon(R) CPU E5-2650 v4 processor.
LinkExplorer

3 Discussion and future work

LinkExplorer demonstrates that high predictive performance can be achieved while retaining a high level of model explainability and transparency, attributes that are very desirable for biomedical hypothesis generation. Future work should focus on conducting empirical user studies of explanation methods to quantify and understand their utility in real-world biomedical research settings.

Funding

This work received funding from netidee grant 5171 (OpenBiolink).

References


