Proceedings of Machine Learning Research 176:314–318, 2022 NeurIPS 2021 Competition and Demonstration Track

Automated Evaluation of GNN Explanations with Neuro Symbolic Reasoning

Vanya Bannihatti Kumar AE17B045@SMAIL.IITM.AC.IN
IIT Madras, India

Balaji Ganesan BGANESA1@IN.IBM.COM
Arvind Agarwal ARVAGARW@IN.IBM.COM
IBM Research, India

Muhammed Ameen MUHAMMED.ABDUL.MAJEED.AMEEN@IBM.COM
Devbrat Sharma DEVBRAT.SHARMA@IBM.COM
IBM Data and AI, India

Editors: Douwe Kiela, Marco Ciccone, Barbara Caputo

Abstract

Explaining Graph Neural Networks predictions to end users of AI applications in easily understandable terms remains an unsolved problem. In particular, we do not have well developed methods for automatically evaluating explanations, in ways that are closer to how users consume those explanations. Based on recent application trends and our own experiences in real world problems, we propose an automatic evaluation approach for GNN Explanations using Neuro Symbolic Reasoning.

Keywords: Graph Neural Networks, Explainability, Neuro Symbolic Reasoning

1. Introduction

Explaining neural model predictions to the users of an application adds tremendous value, enhancing trust in the model predictions and increases adoption of AI even in sensitive real world applications. While explainability techniques in other areas of machine learning like LIME Ribeiro et al. (2016), Influence functions Koh and Liang (2017), SHAP Lundberg and Lee (2017), and Anchors Ribeiro et al. (2018) have been widely adopted in real world applications, adoption of graph neural networks (GNN) explainability techniques has faced some challenges.

Recent works in GNN explanations include Ying et al. (2019), Yuan et al. (2020), Vu and Thai (2020), Schlichtkrull et al. (2020), Wang et al. (2021), Lin et al. (2021). Many of them tend to produce a subgraph of important features, nodes and edges, as explanation for a GNN model prediction. These explanations can be referred to as *subgraph explanations*.

Our experience from real world applications is that these subgraph explanations are usually hard for end users to understand. Even for model developers it is difficult to determine what is an acceptable explanation for a prediction. In industrial applications, this problem is sought to be solved with user-studies Ganesan et al. (2020b) and active learning based approaches.

One of the methods that is often proposed to evaluate explanations is the following. Given groups of people A and B, we show a number of input samples and model predictions to both the groups. To group A, we additionally show explanations for the predictions.

Then during inference time, we show only the input samples and ask both groups A and B, to guess what the model will predict. It is generally expected that group A will be able to better guess the model predictions because of having seen the explanations during training.

We have been using this method to evaluate GNN explanations with mixed results. In Ganesan et al. (2020a), we presented different kinds of explanations and asked the users to *guess an explanation* for a predicted link rather the predict the model behaviour itself. This is because, GNN explanations tend to be not that easy to understand and seldom make it easier to understand model behaviour just looking at few samples.

However, designing such user-studies could be cost prohibitive, take a lot of time and effort, and still may not lead to desired outcomes. Explainability is subjective and users may not be able to provide feedback unless different explanations are presented to them. Further, different human annotators who annotate explanations or verify the correctness of annotations may have significant knowledge of the domain or the corpus, which makes them overlook missing parts of an explanation. So what may be a reasonable explanation to one user might be indecipherable to another.

Hence we need a framework to automatically evaluate explanations before they are presented to human evaluators. We have since been exploring solutions, which go beyond measuring the fidelity of the post-hoc models, but fall short of requiring human evaluators to spend lot of time annotating the explanations.

2. Automated Evaluation of Explanations

We believe the trick is to do what Ontology designers have always done. If we can capture the major concepts and relations of a graph in the form of an ontology, or as a set of axioms, we can evaluate the explanations by running the axioms through a reasoner.

$$\frac{\text{T-BOX} : \text{Man} \sqsubseteq \text{Mortal} \qquad \text{A-BOX} : \text{Man(Socrates)}}{\text{Socrates} \in \text{Mortal}}$$

Figure 1: Example of conclusion by reasoner

In BK et al. (2021), we proposed using reasoners (typically used in semantic web and ontologies), to automatically evaluate the explanations from GNN explainability techniques, provided we are able to convert them into axioms. Typically a reasoner (which is complete i.e, can prove all statements that are true given the knowledge base) is fed the axioms (universal truth) along with the instance level axioms, and the goal. Then using several rules of inference such as *modus ponens*, *modus tollens*, rules of simplification, addition etc, the reasoner can generate the proof for a given goal.

As shown in Figure 1, an example rule of modus ponens can be, if "All men are mortal" is the universal truth and "Socrates is a man" is instance level assertion and the neural graph prediction is "Socrates is mortal", then this prediction can be easily concluded using the modus ponens rule of inference which states that if "P implies Q" and P occurs in the knowledge base then Q can be concluded. The complete proof generated would be "All men are mortal. Socrates is a man. Socrates is a mortal", which can be interpreted easily by the end user.

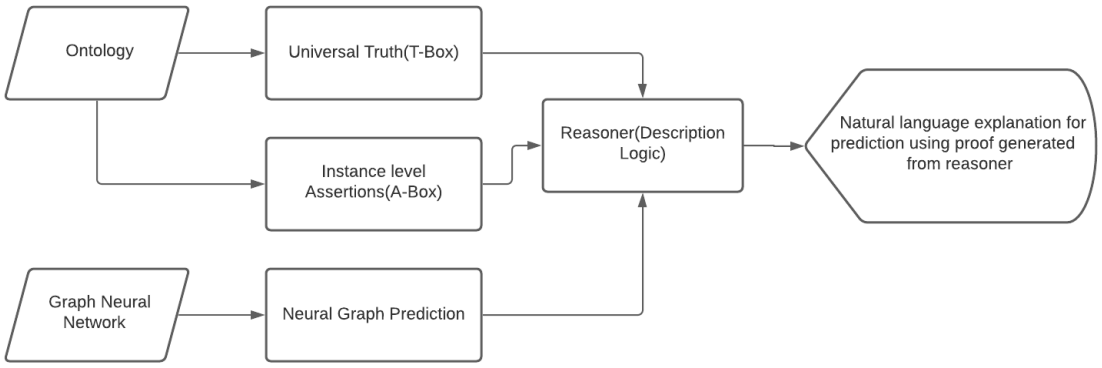
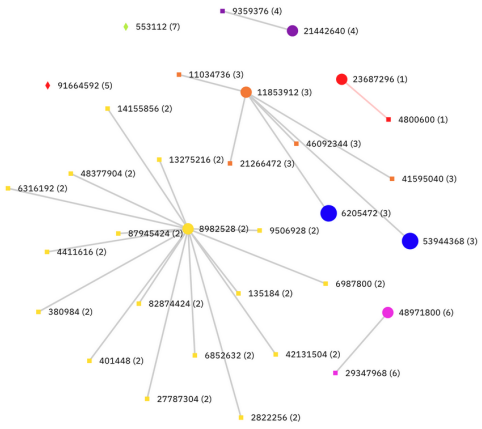


Figure 2: Reasoning mechanism

A neuro-symbolic reasoning based approach to this problem as shown in Figure 2, will involve using description logic to represent the universal truth (axioms) in terms of concepts and relations between them. The universal truths themselves comes from an ontology. The GNN model predictions are the instance level axioms. Then the reasoner will reason over the universal truths and instance level axioms to generate proofs. Optionally these proofs can also be converted as natural language explanations.

3. Demo



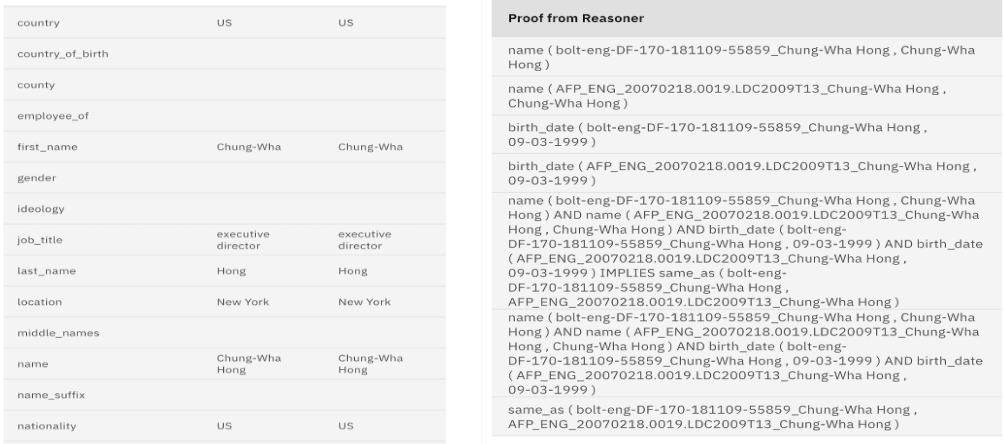
(a) Entity matching

Attribute	Value
city	New York (0.616)
country	US (0.699)
first_name	Chung-Wha (0.641)
job_title	executive director (0.646)
last_name	Hong (0.669)
location	New York (0.636)
name	Chung-Wha Hong (0.658)
nationality	US (0.671)
organization	White House (0.661)
state	New York (0.604)

(b) GNN Explanation

Figure 3: In our demo, we show the entity matches predicted by a GNN model, and important features for the predictions identified by a GNN explainer.

We describe our solution using the entity matching task in the industry. Entity matching is the task of predicting if two nodes in a graph (or records in a table) belong to the same



country	US	US
country_of_birth		
county		
employee_of		
first_name	Chung-Wha	Chung-Wha
gender		
ideology		
job_title	executive director	executive director
last_name	Hong	Hong
location	New York	New York
middle_names		
name	Chung-Wha Hong	Chung-Wha Hong
name_suffix		
nationality	US	US

(*a*) Comparison of nodes

Proof from Reasoner

name (bolt-eng-DF-170-181109-55859_Chung-Wha Hong , Chung-Wha Hong)

name (AFP_ENG_20070218.0019.LDC2009T13_Chung-Wha Hong , Chung-Wha Hong)

birth_date (bolt-eng-DF-170-181109-55859_Chung-Wha Hong , 09-03-1999)

birth_date (AFP_ENG_20070218.0019.LDC2009T13_Chung-Wha Hong , 09-03-1999)

name (bolt-eng-DF-170-181109-55859_Chung-Wha Hong , Chung-Wha Hong) AND name (AFP_ENG_20070218.0019.LDC2009T13_Chung-Wha Hong , Chung-Wha Hong) AND birth_date (bolt-eng-DF-170-181109-55859_Chung-Wha Hong , 09-03-1999) AND birth_date (AFP_ENG_20070218.0019.LDC2009T13_Chung-Wha Hong , 09-03-1999) IMPLIES same_as (bolt-eng-DF-170-181109-55859_Chung-Wha Hong , AFP_ENG_20070218.0019.LDC2009T13_Chung-Wha Hong)

name (bolt-eng-DF-170-181109-55859_Chung-Wha Hong , Chung-Wha Hong) AND name (AFP_ENG_20070218.0019.LDC2009T13_Chung-Wha Hong , Chung-Wha Hong) AND birth_date (bolt-eng-DF-170-181109-55859_Chung-Wha Hong , 09-03-1999) AND birth_date (AFP_ENG_20070218.0019.LDC2009T13_Chung-Wha Hong , 09-03-1999)

same_as (bolt-eng-DF-170-181109-55859_Chung-Wha Hong , AFP_ENG_20070218.0019.LDC2009T13_Chung-Wha Hong)

(*b*) Proof from the reasoner

Figure 4: A typical explanation presented to the human evaluators

real world entity. This task is critical for managing *master data* in enterprises, governments and many commercial applications. Master data refers to the critical customer data that organizations maintain. Master Data Management (MDM) Oberhofer et al. (2014) refers to a group of products that help organizations manage this master data.

An entity here is a person or organization, and the task is to determine if two nodes represent the same real world entity and hence can be merged into a single node. As shown in 3(*a*), these different nodes can be linked to each other using *same_as* links. Graph Neural Networks models can be used to predict these links. We then generate GNN Explanations (typically subgraphs where important nodes and features are highlighted) for the links using the state of the art methods like Ying et al. (2019). We get universal truth from an ontology or as rules provided by experts. Then the neural model predictions are converted to axioms. Together these are fed to a reasoner to check for correctness.

The proof from the reasoner can be presented as shown in Figure 4(*b*), along with the comparison of nodes like in Figure 4(*a*). This solution can be used to complement an active learning or case-study to evaluate the explanations. We can also automate the evaluation of explanations by reporting the number of times GNN model explanations are found correct by the reasoner. This automated evaluation can be used to correct any obvious errors in the GNN model prediction, before these explanations are presented to human annotators.

Conclusion

We described an interactive demo where we present predictions from a GNN model, subgraph explanations from GNN explainability techniques, and proofs generated from a reasoner. The proofs are then used to evaluate GNN explanations. Our method can be used to automate the evaluation of GNN model explanations in real-world applications.

References

Vanya BK, Balaji Ganesan, Aniket Saxena, Devbrat Sharma, and Arvind Agarwal. Towards automated evaluation of explanations in graph neural networks. *XAI Workshop at ICML*, 2021.

Balaji Ganesan, Matheen Ahmed Pasha, Srinivas Parkala, Neeraj R Singh, Gayatri Mishra, Jim O'Neill, Sumit Bhatia, Hima Patel, Sameep Mehta, and Somashekar Naganna. xlp: Explainable link prediction. NeurIPS Demo, 2020a.

Balaji Ganesan, Hima Patel, and Sameep Mehta. Explainable link prediction for privacy-preserving contact tracing. SpicyFL 2020 Workshop at NeurIPS 2020, 2020b.

Pang Wei Koh and Percy Liang. Understanding black-box predictions via influence functions. In *International Conference on Machine Learning*, pages 1885–1894. PMLR, 2017.

Wanyu Lin, Hao Lan, and Baochun Li. Generative causal explanations for graph neural networks. In *International Conference on Machine Learning*, pages 6666–6679. PMLR, 2021.

Scott M Lundberg and Su-In Lee. A unified approach to interpreting model predictions. In *Advances in neural information processing systems*, pages 4765–4774, 2017.

Martin Oberhofer, Eberhard Hechler, Ivan Milman, Scott Schumacher, and Dan Wolfson. *Beyond big data: Using social MDM to drive deep customer insight*. IBM Press, 2014.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. " why should i trust you?" explaining the predictions of any classifier. In *Proceedings of the 22nd ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1135–1144, 2016.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. Anchors: High-precision model-agnostic explanations. In *Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

Michael Sejr Schlichtkrull, Nicola De Cao, and Ivan Titov. Interpreting graph neural networks for nlp with differentiable edge masking. *arXiv preprint arXiv:2010.00577*, 2020.

Minh N. Vu and My T. Thai. Pgm-explainer: Probabilistic graphical model explanations for graph neural networks, 2020.

Xiang Wang, Yingxin Wu, An Zhang, Xiangnan He, and Tat-Seng Chua. Towards multi-grained explainability for graph neural networks. *Advances in Neural Information Processing Systems*, 34, 2021.

Zhitao Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik, and Jure Leskovec. Gnnexplainer: Generating explanations for graph neural networks. In *Advances in neural information processing systems*, pages 9244–9255, 2019.

Hao Yuan, Jiliang Tang, Xia Hu, and Shuiwang Ji. Xgnn: Towards model-level explanations of graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 430–438, 2020.