Effects Of Node Features On Logical Explainers In The EDGE Framework

Eugene Agbor Egbe[1] eugene@mail.uni-paderborn.de

Paderborn University, Warburger Straße 100, 33098, Germany

Abstract. The use and dependence of machine learning and deep learning has shown a significant rise in both simple and complex tasks made applicable to real life situations. Bundled with the innovations made by these techniques, there's also an important need to understand how the results produced are arrived at even as these systems are considered as complex "black boxes" and also how they might behave when several changes are made to the data and/or systems they feed. Hence the need for explainability also is inevitable to continue to enforce trust in these complex models and also there is also increase in variants to address this need. To attempt this, the EDGE framework is proposed to evaluate diverse knowledge graph explanations by assessing logical rule-based explanations of various explainers in terms of prediction accuracy and fidelity to the particular Graph Neural Network (GNN) model applied. Our approach seeks to experiment on the effects of capturing more heterogeneous node features to the evaluations made by the EDGE framework. The evaluations of this experiment shows that logical methods are a better approximation to explanations for complex datasets with complicated semantics and their explanations are human interpretable.

Keywords: Explainable AI · Knowledge Graphs · Node Classification · Node Features

1 Introduction

The increasing adoption of black-box machine learning models, such as graph neural networks (GNNs), in knowledge graph analysis has brought significant improvements in predictive performance [9]. Today, these models are being used significantly for node classification and also graph classification tasks [14]. However, these models often lack transparency, making it difficult for users to understand the rationale behind their predictions. This lack of interpretability poses challenges in sensitive domains where trust, accountability, and validation are essential. Recently, interpretability and explainability have gained more and more attention in the field of machine learning, as they are crucial when it comes to high-stakes decisions and troubleshooting [8].

In order to provide explainability to GNNs, several methods have emerged, each with explainers, for instance PGExplainer [10], a parameterized explainer which has been proposed in the recent literature. In recent literature, it is arguably being shown that it is even more important to explore the possibility of explainability by expressing learned concepts as logical expressions which provide more expressivity [6].

With these many approaches, there is increasing interest in evaluating and comparing the performance of all these approaches to be able to tell which method better approximates the explanation, hence the work done in the EDGE (Evaluation of Diverse Knowledge Graph Explanations) framework [11]. EDGE uses two main criteria to evaluate the performance of the explainers, these are:

- Perdiction perfomance: How well an explainer's predictions align with ground truth labels,
- Explanation performance: Consistency between the explainer's predictions and those generated by the graph neural network.

This study seeks to add more flavor to the EDGE framework by observing how the framework behaves when global and local heterogeneous node features are added to the input features used by the explainer on the Relational Graph Attention Network (RGAT) [2] leveraged by the framework. As node local features, we use Node degree [1], One-hot encoding [12] and clustering coefficient [13]. The global features attempted were PageRank [3] and Betweeness centrality [4] of the different node types. We then report on the results and explain the learnt class expressions as positive or negative instances.

2 Data Analysis

As part of the data analysis for this project, two functionalities were implemented: one to compute heterogeneous node features and another to provide a summary of the data set information. Two types of features are tried; the local feature context, which systematically extracts and normalizes structural features for each node type in the heterogeneous graph. Specifically, it computes the **node degree** (as the sum of incoming edges for each node type), applies **one-hot encoding** to represent node types, and calculates the local **clustering coefficient** for each node using the formula:

$$C_i = \frac{2T_i}{k_i(k_i-1)};$$

where T_i is the number of triangles through node i and k_i is the degree of node i. These features are then normalized to ensure comparability across node types. The dataset description function provides a comprehensive summary of the dataset, including statistics on node and edge types, feature availability, label distribution, and graph density, where density is computed as:

$$D = \frac{2E}{N(N-1)};$$

with E being the number of edges and N the number of nodes. Together, these steps offer both quantitative and qualitative insight into the structure and properties of the graph, laying a solid foundation for subsequent modeling and explainability tasks.

As global features, we also looked at **PageRank**, a global importance score for each node, normalized per node type; and **Betweenness centrality**,

$$g(v) = \sum_{s \neq v \neq t} \frac{\sigma_{st}(v)}{\sigma_{st}};$$

where σ_{st} are the shortest routes from node s to node t, and $\sigma_{st}(v)$ is the number of those paths that go through node v. All features are concatenated to form a comprehensive feature vector for each node, supporting downstream learning and explainability.

3 Model Training and Evaluation

3.1 Dataset And Training

The dataset used in this experiment is the AIFB dataset which satisfies the requirements of selecting a small-yet generalizable graph dataset. All datasets come along with training and testing sets. For training the RGCN and RGAT models, we use 80% of the training set for training and 20% for early stopping.

The Relational Graph Attention Network (RGAT) was used as the default model for node classification on heterogeneous knowledge graphs. RGAT extends the standard Graph Attention Network (GAT) by incorporating relation-specific attention mechanisms, allowing it to effectively model multi-relational data commonly found in knowledge graphs. The model is trained using node features that include structural and semantic information, as described in the data analysis section. Logical approaches are trained with the positive and negative examples obtained from the GNN predictions on the whole training set.

During training, the RGAT model receives as input the heterogeneous graph along with the computed node features. The model is optimized using the cross-entropy loss function, with early stopping based on validation accuracy to prevent overfitting. Hyperparameters such as learning rate, hidden dimension size, number of attention heads, and dropout rate are set according to best practices and validated on the dataset.

After training, we evaluate the model's predictive performance on the test set using standard metrics such as accuracy, precision, recall, and F1-score. To provide interpretable explanations for the model's predictions, we integrate the EvoLearner symbolic explainer. EvoLearner operates by searching for concise OWL class expressions that best describe the set of nodes predicted to belong to a particular class. These expressions are generated using evolutionary algorithms and are evaluated for their coverage and specificity with respect to the model's outputs.

3.2 Model Evaluation

The model evaluation is based on the same matrics as the EDGE framework which is prediction performance and explanation performance. The prediction performance assesses the ability of the explainer to predict the original ground truth labels of data; the explanation performance assesses the ability of the explainer to make the same prediction as the GNN. Those comparisons are done in terms of accuracy, precision, recall, and F1-score with the main goal of exploring how the explainer's perfomance reflects the ground truth while maintaining fidelity to the GNN.

For the logical approaches, the RDF dataset is converted into OWL Knowledge Graphs using the ROBOT tool [7]. Both transformations were performed on

Table 1. Performance of explainers on AIFB.

Approach	**Pred. Perf.**				**Expl. Perf.**			
	A	P	R	F1	A	P	R	F1
EvoLearner	0.650	0.545	**1.000**	**0.705**	0.661	0.559	**1.000**	**0.717**

the same original RDF datasets, ensuring conversion accuracy through a series of test cases within the framework. To evaluate our approach, we selected the Relational Graph Attention (RGAT) model as the base GNN model throughout the experiments. The model was trained on the AIFB dataset for a node classification task. The model was optimized with the same recommendations of the EDGE framework using a learning rate of 0.005 and a weight decay of 0.0005 to prevent over-fitting. The implemented RGAT model validation accuracy is 0.93 for AIFB dataset.

Table 1 shows the prediction performance and explanation performance of the EvoLeaner explainerin terms of accuracy, precision, recall and F1-Measure on the AIFB dataset over 5 independent CELOE was tried as well, but EvoLearner outperforms it based on the experiments (1.0 for its prediction and explanation recall).

Comparing the results of the RGAT with those of the original RGCN model used by EDGE with the same feature combinations (added on github), there was a slight increase in the explainers performance accuracy. As highlighted in EDGE, better results could be achieved with bigger datasets whcih demonstrate structural characteristics like the MUTAG dataset, which has a higher frequency when dealing with logical approaches [5].

4 Model Explanation

In this study, we applied the `EDGE-Hetero` framework to extract interpretable semantic patterns from heterogeneous project knowledge graphs using Description Logic (DL) class expressions. These expressions encode human-readable concepts, such as:

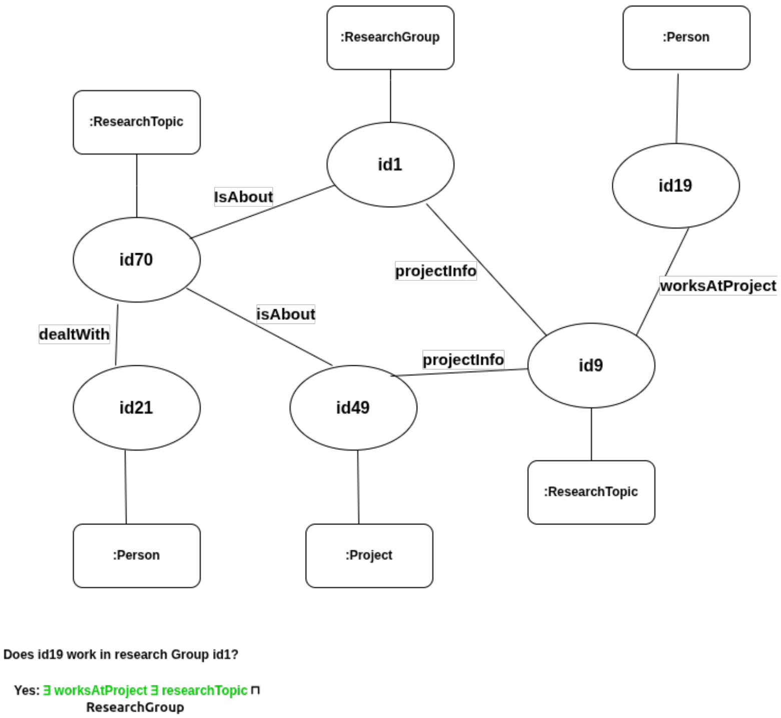
$$\forall \text{worksAtProject}.(\forall \text{projectInfo}.(\leq \exists \text{ isAbout}.(\leq 6 \text{ dealtWithIn}\top)))$$

This expression describes a set of individuals (e.g., employees, researchers) who satisfy a very specific condition related to the projects they are involved in. In plain language, it identifies any individual for whom the following statement holds true:

> “For **every single project** this individual works at, it must be the case that for **every piece of project information** associated with that project, that information is ‘isAbout‘ **at most three** distinct topics or items. Furthermore, each of those topics or items must, in turn, be ‘dealtWithIn‘ by **at most six** other entities.”

This creates a highly constrained class, ensuring that the individuals only work on projects whose information structure is strictly limited in its relationships and their cardinality. The structural reasoner from owlpy was then used to evaluate each expression across the dataset, identifying between 2495 and 2511 matching individuals per expression, thereby validating their semantic coherence.

To complement the explanations, DL queries were run on protege specifying the given class expressions against the ontology to get the terminological, assertional and inferred axioms which which will serve as a more in-depth explanation for the individuals which satisfy the class expression. We then visualize those in an image as shown below: Based on the image above, we can ask questions which are entailed in the original ontology meaning that it validates the the patterns which the explainers discovered and given that it keeps fidelity to the GNN, then this explanations are considered as valid.



Does id19 work in research Group id1?

Yes: ∃ worksAtProject ∃ researchTopic ⊓ ResearchGroup

Fig. 1. Class expression with axioms in a graph.

5 Conclusion

In this experiment, we evaluated the effects of capturing node features on a local and global point of view on the EDGE framework. It was noticed that the logic-based methods provide a high recall and also, no matter how complex a dataset is, and complex in semantics, the explainers are sill able to study these patterns (class expressions) even though there are several known issues working with multi-class classification [11].

It might be interesting in the future to explore ontologies with more complicated semantics such as MUTAG and BGS to evaluate the effect of these local and global features and also to add a module which generates the explanations which are interpretable directly on the framework. This is currently work in progress in the EDGE-Hetero experiment.

6 Contributions

This study was done by the author [Egbe Eugene Agbor] with the assistance of the Professor [Dr. Stefan Heindor] with the following contributions. All the code and documentation on the project and not limited to:

- Adding a function to capture and compute the local and global features of nodes.
- Adding a describe function display statistics of the currently train dataset.
- Writing custom parser for the learnt class expression since recent owlpy parsers won't work with current version of EDGE framework.
- Running the ontology and reasoner on protege and interpreting the reasoner results.

Acknowledgments. A huge thanks to the professor of the XAI course Dr. Stefan Heindorf for the ideas given in the course of this experiment and to all the authors of the EDGE framework.

References

1. Chapter 4 - node degree and strength. In Alex Fornito, Andrew Zalesky, and Edward T. Bullmore, editors, *Fundamentals of Brain Network Analysis*, pages 115–136. Academic Press, San Diego, 2016.
2. Dan Busbridge, Dane Sherburn, Pietro Cavallo, and Nils Y. Hammerla. Relational graph attention networks, 2019.
3. Ying Ding, Erjia Yan, Arthur Frazho, and James Caverlee. Pagerank for ranking authors in co-citation networks, 2010.
4. Changjun Fan, Li Zeng, Yuhui Ding, Muhao Chen, Yizhou Sun, and Zhong Liu. Learning to identify high betweenness centrality nodes from scratch: A novel graph neural network approach, 2019.
5. Hui Han, Tianyu Zhao, Cheng Yang, Hongyi Zhang, Yaoqi Liu, Xiao Wang, and Chuan Shi. Openhgnn: An open source toolkit for heterogeneous graph neural network. In *Proceedings of the 31st ACM International Conference on Information & Knowledge Management*, CIKM '22, page 3993–3997, New York, NY, USA, 2022. Association for Computing Machinery.
6. Stefan Heindorf, Lukas Blübaum, Nick Düsterhus, Till Werner, Varun Nandkumar Golani, Caglar Demir, and Axel-Cyrille Ngonga Ngomo. Evolearner: Learning description logics with evolutionary algorithms. In *Proceedings of the ACM Web Conference 2022*, WWW '22, page 818–828, New York, NY, USA, 2022. Association for Computing Machinery.
7. Rebecca Jackson, James Balhoff, Eric Douglass, Nomi Harris, Christopher Mungall, and James Overton. Robot: A tool for automating ontology workflows. *BMC Bioinformatics*, 20, 07 2019.
8. Benjamin Leblanc and Pascal Germain. On the relationship between interpretability and explainability in machine learning, 2024.
9. Meng Liu, Hongyang Gao, and Shuiwang Ji. Towards deeper graph neural networks. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, page 338–348, New York, NY, USA, 2020. Association for Computing Machinery.
10. Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, and Xiang Zhang. Parameterized explainer for graph neural network. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 19620–19631. Curran Associates, Inc., 2020.
11. Rupesh Sapkota, Dominik Köhler, and Stefan Heindorf. Edge: Evaluation framework for logical vs. subgraph explanations for node classifiers on knowledge graphs. In *CIKM*. ACM, 2024.
12. Cencheng Shen, Qizhe Wang, and Carey E. Priebe. One-hot graph encoder embedding. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(6):7933–7938, June 2023.
13. Thibaud Trolliet, Nathann Cohen, Frédéric Giroire, Luc Hogie, and Stéphane Pérennes. Interest clustering coefficient: a new metric for directed networks like twitter, 2020.
14. Muhan Zhang, Zhicheng Cui, Marion Neumann, and Yixin Chen. An end-to-end deep learning architecture for graph classification. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence and Thirtieth Innovative Applications of Artificial Intelligence Conference and Eighth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'18/IAAI'18/EAAI'18. AAAI Press, 2018.