Effects of Graph Neural Network Aggregation Functions on Generalizability for Solving Abstract Argumentation Semantics

Dennis Craandijk[1,2,*], Floris Bex[2,3]

[1]National Police Lab AI, Netherlands Police
[2]Utrecht University, Netherlands
[3]Tilburg University, Netherlands

Abstract

This paper investigates the effects of different graph neural network aggregation functions on the generalizability of these models for solving abstract argumentation semantics. We systematically compare the performance of Sum, Mean, Max, Attention, and Convolution aggregation functions on predicting sceptical argument acceptance under the preferred semantics. Our experiments utilize a variety of benchmark datasets to evaluate scalability and out-of-domain generalization. Results show that while most aggregators perform well on scalability, Max and Attention aggregators significantly outperform others on generalization to data not seen during training. This study provides valuable insights into the design of accurate and robust GNN-based approximate solvers for abstract argumentation frameworks, emphasizing the importance of the aggregation function.

Keywords

abstract argumentation, graph neural network, approximate solver

1. Introduction

Abstract argumentation has gained significant attention as a formal framework for representing and reasoning about complex decision-making processes [1]. Advancements in various domains, such as legal reasoning, multi-agent systems and human-computer interaction, highlight the importance of developing efficient solvers for various computational reasoning problems within this approach. Computational problems, such as enumerating extensions or deciding whether an argument is accepted in one or all such extensions, are commonly solved with exact solvers [2, 3]. While exact solvers are effective for small-scale problems, these programs can struggle to handle the computational demands of large and complex argumentation frameworks. Several recent approaches have been proposed for defining approximate algorithms, which aim to provide solutions that are close to the exact one but can be computed more efficiently [4, 5, 6, 7].

Graph neural networks (GNNs) are a promising architecture for approximate solvers due

SAFA'24: Fifth International Workshop on Systems and Algorithms for Formal Argumentation, September 17, 2024, Hagen, Germany
*Corresponding author.
d.f.w.craandijk@uu.nl (D. Craandijk); f.bex@uu.nl (F. Bex)
www.florisbex.com (F. Bex)
0000-0001-6815-7053 (D. Craandijk); 0000-0002-5699-9656 (F. Bex)

to their ability to learn from graph-structured data. Since an argumentation framework (AF) can naturally be represented as graph, GNNs can effectively capture the interactions between arguments and counterarguments, allowing them to approximate exact reasoning about argumentation semantics. The allure of this approach is that GNNs can solve problems with linear time complexity relative to the input size, while being learned from data and thus alleviating the need for manual feature engineering and expert knowledge.

Several works have shown that GNNs are able to predict argument acceptance under various argumentation semantics with high accuracy [4, 5, 6, 7]. These studies employ various GNN architectures, mainly characterized by distinct aggregation and update functions, and different training and evaluation regimes. The lack of a systematic and uniform evaluation protocol has hindered direct comparisons between these approaches, thereby missing insights into the most effective design choices. To address this gap, this work aims at uniformly comparing different GNN architectures based on their performance on a number of benchmark datasets. We specifically focus on the most effective aggregation function, contributing to the development of accurate and robust models.

2. Preliminaries

2.1. Abstract argumentation

We recall Dung's abstract argumentation frameworks [8].

Definition 1. *An abstract* argumentation framework *(AF) is a pair $F = (A, R)$ where A is a (finite) set of arguments and $R \subseteq A \times A$ is the attack relation. The pair $(a, b) \in R$ means that a* attacks *b. A set $S \subseteq A$ attacks b if there is an $a \in S$, such that $(a, b) \in R$. An argument $a \in A$ is* defended *by $S \subseteq A$ iff, for each $b \in A$ such that $(b, a) \in R$, S attacks b.*

Dung-style semantics define the sets of arguments that can jointly be accepted (*extensions*). A σ-extension refers to an extension under semantics σ. We consider admissible sets and preferred and grounded semantics with the following functions respectively adm, prf, grd.

Definition 2. *Let $F = (A, R)$ be an AF. A set $S \subseteq A$ is* conflict-free *(in F), if there are no $a, b \in S$, such that $(a, b) \in R$. The collection of sets which are conflict-free is denoted by $cf(F)$. For $S \in cf(F)$, it holds that: $S \in adm(F)$, if each $a \in S$ is defended by S; $S \in prf(F)$, if $S \in adm(F)$ and for each $T \in adm(F)$, $S \not\subset T$ and for each $a \in A$ defended by S it holds that $a \in S$; $S \in grd(F)$.*

Furthermore, for some argument a that is part of F, we can determine if it is *credulously accepted* under semantics $\sigma - a$ is contained in at least one σ-extension – or *sceptically accepted* under $\sigma - a$ is contained in all σ-extensions.

2.2. Graph neural networks

We recall graph neural networks as used for abstract argumentation [4]. Let $G = (V, E)$ be a graph representation of an AF F, where at message passing step $t = 0$ each node i is assigned a real-valued vector $v_i^t \in V$ and each edge between node i and j is assigned a vector $e_{ij} \in E$

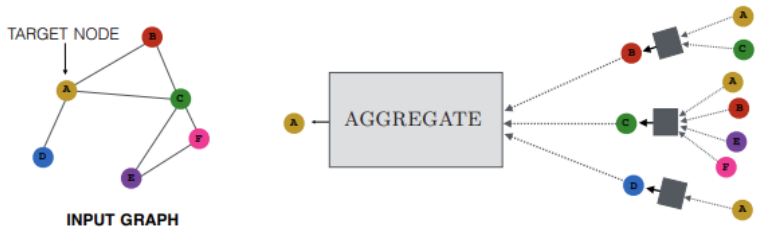

Figure 1: The aggregation functions in a graph neural network combines feature information from neighboring nodes, expressing the local graph structure when updating node representations. To generate the embedding for node A, the messages from A's graph neighbors are aggregated. In turn, the messages coming from these neighbors are based on the aggregated messages from their respective neighbors, and so on. (source: Hamilton et al. [9])

indicating whether it represents a directed, reciprocal or self-loop edge. At subsequent message passing steps t, each node i aggregates messages m_{ij}^t from its neighbours j and updates its vector representation, such that

$$m_{ij}^{t+1} = \text{MSG}(v_i^t, v_j^t, e_{ij}) \tag{1}$$

$$v_i^{t+1} = \text{UPDT}(v_i^t, \underset{j \in N(i)}{\text{AGGR}}(m_{ij}^{t+1})) \tag{2}$$

where $N(i)$ denotes all neighbours of node i. The message function MSG computes a messages based on the vectors of two connected nodes along with the edge representations. The update function UPDT updates the node representation based on the previous node representation and the messages aggregated by AGGR. The message and update functions are parameterized neural networks which, in conjunction with the aggregation function, yield a neural message passing algorithm whose parameters can be optimised (i.e. learned) based on data. After each step t, node representations can be read out with the readout function READ that maps a node representation to a likelihood $o_{v_i}^t = \text{READ}(v_i^t)$ of the respective argument being accepted.

3. Aggregation function

A central component of GNNs is the aggregation function, which summarizes the information from neighboring nodes (see Figure 1). Different aggregation functions can impact the model's performance and the type of information it captures from the graph. We discuss five common aggregation functions: Sum, Max, Mean, Attention, and Convolution.

The Sum aggregation function adds up the feature vectors of neighboring nodes, allowing each node to accumulate information from its neighbors equally. This approach is straightforward and shown to be one of the most expressive aggregation functions [10]. However, it may lead to unstable node embeddings when scaling up to AFs larger than those seen during training [11].

Max aggregation selects the maximum value for each feature dimension across the neighbors, making it useful for highlighting the most significant features. This method might lose some nuanced information since it does not consider the entire neighborhood's contribution. Mean aggregation computes the average of the feature vectors of neighboring nodes, providing a balanced representation. This approach can help mitigate the influence irrelevant neighbors but may fail to capture the unique importance of individual nodes. Attention mechanisms compute a weighted Mean of the neighbors' features based on their relevance to the central node. By assigning higher weights to more important neighbors, Attention aggregation enables the model to focus on the most relevant information and improve its discriminative power [12]. Finally, convolutional aggregation functions, inspired by Convolutional Neural Networks (CNNs), apply learnable filters to the neighborhood features [13]. This approach allows the model to capture spatial patterns and local structures within the graph, making it particularly suitable for tasks that require understanding of geometric properties.

In the field of computational argumentation, different works have used different aggregation functions in their GNN architectures. Where Kuhlmann and Thimm [5], Malmqvist et al. [6] use the Convolution aggregation function, Craandijk and Bex [4] use the Sum aggregator and Cibier and Mailly [7] use the Attention aggregator. Whereas Craandijk and Bex [14] use a combination of aggregators, to the best of our knowledge no work has yet evaluated the performance of Max and Mean aggregators on predicting argument acceptability. Additionally, all mentioned works employ the various aggregators in combination with different update and message functions and training and evaluation regimes, making them difficult to compare. In this work we compare all mentioned aggregators uniformly in terms of architectural design and evaluation setup.

4. Data

Since GNNs learn to solve problems based on data, the characteristics of the data used to train and evaluate the architecture are consequential to their performance. The different works on using GNNs to learn argumentation semantics use different datasets, hindering direct comparison between methods. Additionally most works train and validate on datasets with the same characteristics. Kuhlmann and Thimm [5] show that, while different GNN architectures generally yield high quality results when tested on in-distribution AFs, performance severely degrades when generalizing to AF types not seen during training. This indicates that GNNs tend to learn superficial features of the data, rather than a general applicable rule (a fundamental problem found in various fields of deep learning).

In this work we aim at comparing GNN design choices with the goal of developing accurate and robust approximate solvers. Therefore we adopt the evaluation datasets of Kuhlmann and Thimm [5] (i.e. PBBG, KWT and ICCMA) with the aim of testing scalability and generalization of different aggregation functions. Generalization and scalability ensure that a learned GNN solver for abstract argumentation can be reliably deployed in various settings. A scalable solver ensures that the system can handle increasingly larger argumentation frameworks without a significant drop in performance. Generalizability refers to the model's ability to perform well AFs with graph properties not seen during training (i.e. out-of-domain AFs).

A PBBG dataset consists of AFs generated by generators from ICCMA 2020 [3] as used by

Craandijk and Bex [4], namely *AFBenchGen2*, *AFGen Benchmark Generator*, *GroundedGenerator*, *SccGenerator*, *StableGenerator*. This set of generators can generate AFs of various sizes, making them suitable to evaluate in-domain scalability. The KWT dataset is tailored by Kuhlmann et al. [15] towards generating abstract argumentation frameworks that are particularly hard for tasks related to deciding preferred acceptance by avoiding as much as possible the easy cases (where the accepted arguments are (almost) similar to the grounded extension). These AFs are particularly suited to test out-of-domain generalization. The ICCMA dataset consists of AFs that are part of the ICCMA 2017 benchmark competition.[1] These are large AFs with on average around 650 AFs. Notably, a part of this set is generated by generators not in PBBG, namely *ABA2AF*, *admbuster*, *Planning2AF*, *sembuster*, and *traffic*. These AFs are suitable to evaluate both scalability as generalization.

5. Experimental analysis

To assess the performance of different Graph Neural Network (GNN) aggregation functions in predicting sceptical argument acceptance under preferred semantics, we conduct a series of experiments. We focus on this particular problem due to its computational complexity and the fact that the KWT dataset, specifically designed for this task, presents a challenging scenario for GNNs to solve [15]. We create a training dataset, PBBG-train, comprising 100,000 argumentation frameworks (AFs) with argument counts ranging from 5 to 25. Utilizing the same set of generators, we generate three additional datasets: PBBG-val and PBBG-test, each containing 1,000 AFs with exactly 25 arguments, and PBBG-scale, which consists of 1,000 AFs with 100 arguments. PBBG-val is used as a validation set, PBBG-test serves to test the model's learning efficacy, while PBBG-scale evaluates in-domain scalability. Lastly, we use the same 1,000 KWT instances as previously employed by Kuhlmann and Thimm [15] to test out-of-domain generalization and the same 450 AFs from the ICCMA 2017 competition to assess both scalability and out-of-domain generalization capabilities. We adopt the training procedure [2] as described by Craandijk and Bex [4]. We evaluate with the Matthews correlation coefficient (MCC) as it is regarded as a balanced measure, even when classes are unequally distributed. [16]. The MCC is a correlation coefficient value between -1 and +1, where +1 represents a perfect prediction, 0 an average random prediction and -1 an inverse prediction.

Aggregator	PBBG-test	PBBG-scale	KWT	ICCMA
Sum	0.99	0.81	0.57	0.25
Mean	0.98	0.94	0.61	0.23
Max	0.95	0.92	0.72	0.58
Attention	0.99	0.93	0.88	0.58
Convolution	0.98	0.96	0.62	0.30

Table 1
Matthews correlation coefficient scores for sceptical acceptance under the preferred semantics

[1] http://argumentationcompetition.org/2017/
[2] https://github.com/DennisCraandijk/DL-abstract-argumentation

As Table 1 illustrates, all aggregation functions are able to capture the characteristics of the PBBG-train data as demonstrated by the PBBG-test dataset. When scaling to larger graphs within the same domain, as evidenced by the PBBG-scale dataset, all aggregation functions generalize well. Only the performance of the Sum aggregator drops, which can be caused by unstable node embedding as mentioned in Section 3. A disparity between aggregators emerges on the KWT dataset. Here, the Sum, Mean, and Convolution aggregators struggle to generalize beyond their training data, yielding MCCscores below 0.65. Conversely, the Max and Attention aggregators excel, likely due to their ability to concentrate on specific features within a node's neighborhood rather than merging all features indiscriminately. Despite not being the main goal of this work, both aggregators even surpass the previous state-of-the-art result, as reported by Kuhlmann et al. [15], in this training regime by a large margin. This effect is also visible on the ICCMA dataset, which however still proves to be challenging for all GNN variants as scores top at 0.58 MCC.

The Max and Attention aggregators emerge as optimal choices for GNN applications in abstract argumentation. Contrary to the Mean, Sum, and Convolution aggregators, which aggregate neighbor features uniformly, the Max and Attention mechanisms empower GNNs to selectively hone in on specific information. This selective focus allows GNNs to capitalize on the most significant interactions between arguments and counterarguments, thereby enhancing their performance in abstract argumentation tasks.

6. Conclusion

In this paper, we explored the effects of different aggregation functions on the generalizability of GNN's for solving abstract argumentation semantics. Through a comprehensive experimental analysis, we demonstrated that the choice of aggregation function plays a central role in determining the performance and robustness of GNNs in this context. Our findings highlight that while most aggregation functions perform similarly in terms of in-domain scalability, significant differences emerge when evaluating out-of-domain generalization. Specifically, the Max and Attention aggregation functions show better performance in handling AFs with graph properties not seen during training, indicating their potential to capture the dynamics between arguments from the argumentation frameworks. Future research could investigate the impact of other architectural design choices and training strategies on model performance and generalizability, especially on the challenging ICCMA dataset.

References

[1] K. Atkinson, P. Baroni, M. Giacomin, A. Hunter, H. Prakken, C. Reed, G. R. Simari, M. Thimm, S. Villata, Towards artificial argumentation, AI Mag. 38 (2017) 25–36.

[2] G. Charwat, W. Dvorák, S. A. Gaggl, J. P. Wallner, S. Woltran, Methods for solving reasoning problems in abstract argumentation - A survey, Artificial Intelligence 220 (2015) 28–63.

[3] S. A. Gaggl, T. Linsbichler, M. Maratea, S. Woltran, Design and results of the second international competition on computational models of argumentation, Artif. Intell. 279 (2020).

[4] D. Craandijk, F. Bex, Deep learning for abstract argumentation semantics, in: Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020, ijcai.org, 2020, pp. 1667–1673.
[5] I. Kuhlmann, M. Thimm, Using graph convolutional networks for approximate reasoning with abstract argumentation frameworks: A feasibility study, in: N. B. Amor, B. Quost, M. Theobald (Eds.), Scalable Uncertainty Management - 13th International Conference, SUM 2019, Compiègne, France, December 16-18, 2019, Proceedings, volume 11940 of *Lecture Notes in Computer Science*, Springer, 2019, pp. 24–37.
[6] L. Malmqvist, T. Yuan, P. Nightingale, S. Manandhar, Determining the acceptability of abstract arguments with graph convolutional networks, in: S. A. Gaggl, M. Thimm, M. Vallati (Eds.), Proceedings of the Third International Workshop on Systems and Algorithms for Formal Argumentation co-located with the 8th International Conference on Computational Models of Argument (COMMA 2020), September 8, 2020, volume 2672 of *CEUR Workshop Proceedings*, CEUR-WS.org, 2020, pp. 47–56.
[7] P. Cibier, J. Mailly, Graph convolutional networks and graph attention networks for approximating arguments acceptability - technical report, CoRR abs/2404.18672 (2024).
[8] P. M. Dung, On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games, Artificial Intelligence 77 (1995) 321–358.
[9] W. L. Hamilton, R. Ying, J. Leskovec, Representation learning on graphs: Methods and applications, IEEE Data Engineering Bulletin 40 (2017) 52–74.
[10] K. Xu, W. Hu, J. Leskovec, S. Jegelka, How powerful are graph neural networks?, in: 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019, 2019.
[11] C. K. Joshi, Q. Cappart, L. Rousseau, T. Laurent, X. Bresson, Learning TSP requires rethinking generalization, CoRR abs/2006.07054 (2020). arXiv:2006.07054.
[12] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, Y. Bengio, Graph Attention Networks, International Conference on Learning Representations (2018).
[13] T. N. Kipf, M. Welling, Semi-supervised classification with graph convolutional networks, in: 5th International Conference on Learning Representations), 2017.
[14] D. Craandijk, F. Bex, Enforcement heuristics for argumentation with deep reinforcement learning, in: Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022, AAAI Press, 2022, pp. 5573–5581.
[15] I. Kuhlmann, T. Wujek, M. Thimm, On the impact of data selection when applying machine learning in abstract argumentation, in: F. Toni, S. Polberg, R. Booth, M. Caminada, H. Kido (Eds.), Computational Models of Argument - Proceedings of COMMA 2022, Cardiff, Wales, UK, 14-16 September 2022, volume 353 of *Frontiers in Artificial Intelligence and Applications*, IOS Press, 2022, pp. 224–235.
[16] D. Powers, Evaluation: From precision, recall and f-measure to roc, informedness, markedness & correlation, Journal of Machice Learning Technology 2 (2011) 2229–3981.