Determining the Acceptability of Abstract Arguments with Graph Convolutional Networks

Lars MALMQVIST [a,1], Tommy YUAN [a], Peter NIGHTINGALE [a], and Suresh MANANDHAR [b]

[a] *University of York, York, UK*
[b] *NAAMI, Kathmandu, Nepal*

Abstract. This paper presents a new deep learning approach to sceptical and credulous acceptance of arguments, two key problems in Abstract Argumentation that are most commonly solved using exact methods often by reduction to SAT. We train a Graph Convolutional Neural Network with a randomised training regime, dynamic balancing of training data, and improved residual connections and achieve up to 97.15% accuracy improving on past studies by 30 percentage points. The new approach is applied to problems from the ICCMA 2017 competition and achieves a new state of the art for this type of architecture. Additionally, the training regime used for this study has potential wider applicability in deep learning for graph structured NP-hard problems.

Keywords. argumentation, abstract argumentation, solvers, graph convolutional networks, deep learning

1. Introduction

Abstract Argumentation is a formalism for non-monotonic reasoning that bases its representation on the modelling of conflict. It is typically represented in the form of a directed graph in which vertices represent arguments and edges a relation of attack. This gives rise to a range of reasoning problems that determine the acceptability of arguments or the joint acceptability of sets of arguments. The majority of these reasoning problems are known to be NP-hard[1,2].

The vast majority of solution approaches in Abstract Argumentation are exact and complete, often using reduction to some other formalism such as SAT [3] or ASP [4] to solve the problem. In general, SAT based approaches have achieved the best solver performance, but this type of approach does not necessarily provide the easiest fit with real world deployment scenarios and very large scale datasets. There has been some research into probabilistic methods [5], which unfortunately is not directly comparable to this study in the evaluation criteria, and a single feasibility study into using Graph Con-

[1] Corresponding Author: Lars Malmqvist, University of York, E-mail: lm1775@york.ac.uk.

volutional Networks (GCN) [6] that had mixed results depending on the dataset under consideration.

There are, however, major benefits to approximate approaches such as Graph Neural Networks in this area that makes them worth considering in some cases. In particular, they have greatly superior run-time performance and are more easily integrated into heavily parallelised Cloud-based architectures, although this comes at the cost of large memory consumption and GPU support. In this paper, we compare only solver accuracy and runtime performance as memory consumption metrics are not available.

This paper contributes to realising this goal by setting a new and improved state of the art for the use of Deep Neural Networks to solve abstract argumentation problems. This improvement applies both to the overall accuracy and to the balance between positive and negative inference. Furthermore, this paper is the first to consider both credulous and sceptical acceptability. Finally, this paper introduces a new randomised training approach that may have broader applicability to improving the training of GCN architectures.

The rest of the paper will be structured as follows. First, the relevant background material will be presented. Then we will detail the specific approach and architecture we have used to approach the problem. We will, then, present our results and offer some concluding remarks.

2. Background

2.1. Abstract Argumentation

Abstract Argumentation is a formalism for non-monotonic logic that uses a minimal representation of a set of arguments and a binary relation on these arguments to model conflict-based reasoning. It has its origin in a seminal paper by Dung [7] that laid out the basic theory for the field.

Definition 2.1 (Argumentation Framework) *An argumentation framework is a tuple, $F =< args, atts >$ in which args is a finite set of arguments and atts $\subseteq args \times args$ defines a relation of attack.*

To say that A1 attacks A2 is hence the same as saying that $(A1, A2) \in atts$. If $S \subseteq args$ and $A \in args$ we can extend this nomenclature by saying that A attacks S iff there exists $B \in S$ such that $(A, B) \in atts$. In a parallel manner, we can say that S attacks A iff there exists $B \in S$ such that $(B, A) \in atts$. We can also define a similar notion of defence. An argument $A \in args$ is defended by a set $S \subseteq args$ if, for each $B \in args$ such that $(B, A) \in atts$, there exists a $C \in S$ such that $(C, B) \in atts$.

The notion of acceptability is key to abstract argumentation.

Definition 2.2 (Acceptability) *An argument $A \in args$ is called acceptable with respect to an extension ext $\subseteq args$ iff for every $B \in args$ with B attacks A there is an argument $A' \in ext$ with A' attacks B.*

Extensions are subsets of argumentation frameworks that are collectively acceptable. Extensions are evaluated based on semantics that define rules for what arguments can be accepted together. Almost all semantics require the property of conflict-freeness.

Definition 2.3 (Conflict-Freeness) *Given an argumentation framework $F = (args, atts)$, a given subset, S, of this argumentation framework, is said to be conflict-free iff there does not exist $(A,B) \in atts$ with A, $B \in S$.*

Dung's original paper [7] defined four semantics, but for the current paper, only the preferred semantic concerns us as that is the one used in the experiments.

Definition 2.4 (Preferred Extension) *An extension, $pr \subseteq args$, is a preferred extension iff it has the property of conflict-freeness, all its arguments are acceptable with respect to pr and the extension is maximal with respect to set inclusion.*

2.2. Convolutional Graph Neural Networks

Convolutional graph neural networks (CGNNs) draw on the popularity and success of traditional CNN's in particular in computer vision. There are, however, different ways of defining the convolutional operation when it is applied to graphs, which gives rise to different types of CGNNs. The most common approach bases itself on the digital signal processing where convolution is seen effectively as a noise removal operation. The difference between most variants in this approach including the seminal GCN architecture by Kipf and Welling [8], ChebNet [9], and CaleyNet [10] consists mainly in how they represent, approximate, and simplify the filter operations used in the convolution of the graph to achieve computational improvements. The second main approach to CGNNs stays closer to the conventional CNN definition by considering convolution based on a node's spatial relationships[11]. That means spatial based methods in some way aggregates information from a node's neighbourhood. This can be seen for instance in the Message Passing Neural Network [12] that explicitly defines a framework for looking at graph convolution as a message-passing process.

There has been one previous paper that has applied GCN methods to Abstract Argumentation [6]. In this paper, the authors conducted a feasibility study by applying Kipf's approach to a range of datasets but using random and real world argumentation frameworks with limited success. Graph Neural Networks have also been used more successfully in the related fields of Automated Theorem Proving [13] and the Graph Colouring Problem [14].

3. GCN for Abstract Argumentation

3.1. GCN Architecture

The architecture used in this paper builds and extends on the seminal approach introduced by Kipf and Welling [8], but extends it in a number of areas. In the original formulation, the GCN consisted of an input layer, two hidden layers with RELU nonlinearities inserted in between, and ending with an output layer. Node embeddings were generated using a propagation rule following a first-order approximation of spectral graph convolutions. We follow the same basic pattern, but add a number of features to allow for greater depth and to tailor the approach to abstract argumentation graphs that do not have node-level features.

The core components of the GCN architecture used in this paper include the following elements:

1. Graph Embeddings generated using DeepWalk [15], a random walk based method, fed as input features along with the adjacency matrix of the argumentation framework. For the experiments in this paper the dimension of the embedding was 64
2. An input layer receiving these inputs
3. 4 to 6 repeating blocks of a GCN layer [8] and a Dropout layer [16]. For all experiments the Dropout rate was set to 0.5 and the size of the GCN layers 128
4. Residual connections feeding the original features and the normalised adjacency matrix as additional input at each block
5. A Sigmoid output layer generating a probability for the acceptability of each argument in the framework

The model was trained using Adam [17] with Binary Cross-Entropy as the loss function. The training regime is described in subsequent sections.

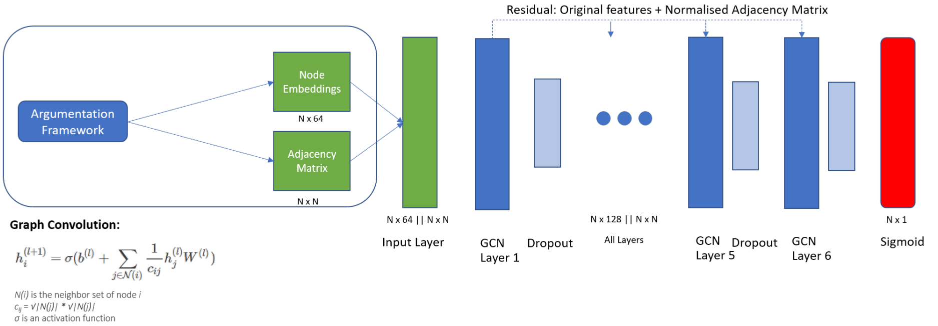
Residual: Original features + Normalised Adjacency Matrix
Argumentation Framework
Node Embeddings
N x 64
Adjacency Matrix
N x N
N x 64 || N x N
Input Layer
GCN Layer 1
Dropout
N x 128 || N x N
All Layers
GCN Layer 5
Dropout
GCN Layer 6
N x 1
Sigmoid
Graph Convolution:
$h_i^{(l+1)} = \sigma(b^{(l)} + \sum_{j \in \mathcal{N}(i)} \frac{1}{c_{ij}} h_j^{(l)} W^{(l)})$
N(i) is the neighbor set of node i
$c_{ij} = \sqrt{|N(j)|} * \sqrt{|N(j)|}$
σ is an activation function

Figure 1. GCN Architecture Overview.

3.2. Randomised Training Batches

Real-world abstract argumentation frameworks tend to have a skewed distribution between acceptable and non-acceptable arguments both for credulous and sceptical acceptance. In particular, there tends to be a large preponderance of non-acceptable arguments. In the argumentation frameworks used for the experiments in this paper, the percentage of non-acceptable arguments ranges from 69.5% to 99.95%. This affects GCN training as the neural net will by default learn to predict a negative outcome even in cases where it is incorrect. This problem was also noted by Kuhlmann and Thimm [6], who generated balanced training data to attempt to address it, but this approach does not seem to have generalised well. Furthermore, a traditional split into fixed training, validation, and test sets for training does not work well for this problem. When using such an approach, the GCN could not effectively generalise to unseen data as confirmed by preliminary experiments.

To overcome these limitations, we have devised a randomised training scheme that generates random training batches at the start of each epoch. The overall training scheme feeds multiple argumentation frameworks to the neural network as a single graph with each argumentation framework represented by a single connected component. To preserve the maximal amount of structural information for learning the entire graph is fed to each network layer. The output layer of the neural network is a prediction of the acceptability of the argument. The randomised training operates by generating a new mask of outcomes to be predicted at the start of each new epoch, essentially asking the neural net to fill in the blanks. This is done to encourage the network to learn to generalise based on structural properties of the graphs.

3.3. Dynamic Balancing and Outlier Exclusion

Two additional measures were taken to address the problems related to imbalanced training data and poor generalisation performance. First, the training mask created to facilitate the generalisation of the neural network was developed to have the option of dynamically balancing the training mask to include equal amounts of acceptable and non-acceptable arguments. This has the intention to avoid the skew caused by unbalanced training data, but also has the unfortunate side effect of reducing the amount of data used for training. This mode is therefore not used in all experiments described in the results section.

A second optimisation was added to handle extremely skewed argumentation frameworks. Some frameworks have no or almost no acceptable arguments and when included tend to skew the training disproportionately. These frameworks have been excluded from the training set using a z-score test with a threshold of 3.5.

3.4. Deep Residual Connections

The original formulation of Graph Convolutional Networks suffers from major performance degradation with an increase of depth beyond a certain limit. Kipf's [8] original GCN, for instance, used only 2-layers in the model. In practice, as the depth of the GCN increases beyond this limit the model stops responding to training data and instead converges to a fix-point. This problem is known as the suspended animation problem and the limit as the suspended animation limit.

Several approaches have been applied to overcome this limit and allow greater depth in GCN architectures. Among the most fruitful approaches have been those that adapt the notion of residual connections to the GCN context[18] by feeding in the graph structure and node features across layers in a variety of ways.

In this paper, we follow a similar approach by adapting the graph-raw residual defined by Zhang and Meng[18]. They define the residual term as the multiplication of the normalised adjacency matrix and the raw input features. This residual term is fed as input to each layer in the model, which achieves the aim of extending the suspended animation limit.

The only difference in our approach is that the normalised adjacency matrix and raw input features are fed to each layer separately rather than as a unit, largely for reasons related to the implementation approach.

4. Experiments

4.1. Dataset and Experimental Setup

The experiments were run against a dataset consisting of 900 argumentation frameworks selected from the ICCMA 2017[2] Benchmark datasets. The selection includes frameworks from benchmark sets A, B, and C including all 5 difficulty categories. Except for the exclusion of a small number of very large argumentation frameworks that could not be processed in the computational environment used for the experiment because of memory limitations, no systematic exclusions were made. Input frameworks contained up to 15,605 arguments and 6,250,500 attacks with an average number of arguments of 1,595 and average number of attacks of 187,542. We divided the dataset into 9 folds to facilitate the training process and the neural net was trained sequentially on each fold. Training and validation sets were generated using a random mask as described above and 10% of graphs were held out as a test set for final evaluation.

The GCN model has been evaluated using two separate tasks: credulous acceptability and sceptical acceptability under the preferred semantic. For an argument to be credulously acceptable in the context of a given argumentation framework it must belong to one of the extensions of that argumentation framework; to be sceptically acceptable, it must belong to all. To be able to frame the problem for supervised learning, we generated ground truth solutions using the Pyglaf argumentation solver [19] for all argumentation frameworks in the dataset.

The experiments were run on a single K80 GPU instance with 12GB ram. Each model took between 4 and 12 hours to train. Separate models were trained for each experiment listed in the results below. The following table details what elements were used for each model.

Table 1. Models Trained

Model	**Layers**	**Balanced**	**Randomised**	**Residuals**
4-Layers Modified GCN	4	No	Yes	Yes
5-Layers Modified GCN	5	No	Yes	Yes
6-Layers Modified GCN	6	No	Yes	Yes
Mod GCN with Balanced Data	4	Yes	Yes	Yes
Mod GCN with Fixed Batches	4	No	No	Yes

4.2. Results

4.2.1. Credulous Acceptability

The results for Credulous Acceptability improve by 30 percentage points on the previous baseline set by Kuhlmann and Thimm. While the best performing model in terms of overall accuracy is the 4-Layer Modified GCN, the better performance on positive acceptability coupled with the very slight decrease in overall accuracy means that the 5-Layer Modified GCN is the overall best performing model.

Excepting the Modified GCN with Balanced Data, all the other models have a disparity in favour of negative acceptability, which is consistent with the previous findings

[2] http://argumentationcompetition.org/2017/

by Kuhlmann and Thimm. The Modified GCN with Balanced Data inverts the pattern with a similar disparity between positive and negative in favour of the positive side, which demonstrates that there is nothing inherent about negative acceptability that makes it easier to learn. If anything, the opposite would seem to be the case as the Modified GCN with Balanced Data will see roughly equal amounts of positive and negative training instances. The poorer overall performance of this model could be due to the lower amounts of training instances seen because of dynamic balancing.

The extremely poor performance when predicting positive acceptability of the Modified GCN with Fixed Batches demonstrates the key contribution of the randomised training regime in obtaining good results in this domain.

Table 2. Results - Credulous Acceptability

Model	**Accuracy**		
	Overall	**Yes**	**No**
4-Layers Modified GCN	92,68%	69,33%	93,54%
5-Layers Modified GCN	92,26%	73,56%	92,95%
6-Layers Modified GCN	91,63%	71,81%	92,37%
Modified GCN with Balanced Data	81,20%	91,20%	71,00%
Modified GCN with Fixed Batches	96,40%	7,00%	99,70%
Kuhlmann and Thimm 2019 - Unbalanced	62,00%	10,00%	97,00%
Kuhlmann and Thimm 2019 - Balanced	63,00%	17,00%	93,00%

4.2.2. Sceptical Acceptability

The results for Sceptical Acceptability seem to reflect the much starker imbalance between negative and positive instances in this setting. By the nature of the problem there will tend to be much fewer positive instances in the sceptical setting. This is reflected in all the models having a large discrepancy between positive and negative accuracy.

Similar to the credulous results, the Modified GCN with Fixed Batches is the worst performer. It is, however, even more extreme for the sceptical case, possibly for the same reason as before. For all intents and purposes the model is incapable of correctly predicting positive acceptability and only gets good overall accuracy from the extreme skew of the underlying dataset.

In contrast to the credulous setting, the Modified GCN with Balanced Data is the clear overall winner under the sceptical setting. It has both the best overall accuracy and is vastly superior in predicting positive acceptability with a relatively minor hit to negative accuracy.

Table 3. Results - Sceptical Acceptability

Model	**Accuracy**		
	Overall	**Yes**	**No**
4-Layers Modified GCN	96,21%	24,04%	97,10%
5-Layers Modified GCN	96,20%	22,92%	97,11%
6-Layers Modified GCN	96,24%	22,69%	97,15%
Modified GCN with Balanced Data	97,15%	46,35%	94,39%
Modified GCN with Fixed Batches	98,44%	0,33%	99,66%

4.3. Ablation Studies

4.3.1. Effects of Depth

For both credulous and sceptical acceptability, three separate models were trained differing only in the number of layers they possess[3]. Based on these results the effect of depth is inconclusive, but at best minor. Under the credulous setting the 5-Layer Modified GCN outperforms the other two slightly. Under the sceptical setting the results are not significantly different enough to assess which model is superior.

The reasons why greater depth does not seem to lead to better performance are unclear at this stage and will require further research. Four layers may already be adequate to capture what can be learned from the training data. It may also be related to the well known issues experienced by other GCN models that have tried to increase depth. However, unlike some of these models, we are not seeing a decline in performance with the addition of layers, but a more or less stationary set of results.

Table 4. Results - By Depth

Model	**Accuracy**		
	Overall	**Yes**	**No**
4-Layers Modified GCN (Credulous)	92,68%	69,33%	93,54%
5-Layers Modified GCN (Credulous)	92,26%	73,56%	92,95%
6-Layers Modified GCN (Credulous)	91,63%	71,81%	92,37%
4-Layers Modified GCN (Sceptical)	96,21%	24,04%	97,10%
5-Layers Modified GCN (Sceptical)	96,20%	22,92%	97,11%
6-Layers Modified GCN (Sceptical)	96,24%	22,69%	97,15%

4.3.2. Effects of Randomisation and Balancing

The largest and most substantial difference in this study is between the models that include the randomised training regime and those that do not, including both models from Kuhlmann and Thimm and the Modified GCN with Fixed Batches for both the credulous and the sceptical case. That points towards this training regime being the largest contributor to increased model performance.

The effects of balancing are more mixed, but also substantial. Kuhlmann and Thimm found a small improvement from using a balanced dataset. In our model under the credulous setting, the Model GCN with Balanced Data performed substantially worse than the other models that also use the randomised training regime. However, for the sceptical setting it was hands-down the best performing model. This may be due to a relationship with the extent to which the underlying training data is skewed. One could speculate that the more skewed the training data, the greater the value of dynamic balancing.

[3] Note that the tables for the ablation studies do not contain additional results, but are constructed based on the previous models.

Table 5. Results - By Training Regime

Model	**Accuracy**		
	Overall	**Yes**	**No**
Modified GCN with Balanced Data (Credulous)	81,20%	91,20%	71,00%
Modified GCN with Fixed Batches (Credulous)	96,40%	7,00%	99,70%
Modified GCN with Balanced Data (Sceptical)	97,15%	46,35%	94,39%
Modified GCN with Fixed Batches (Sceptical)	98,44%	0,33%	99,66%
Kuhlmann and Thimm 2019 - Unbalanced	62,00%	10,00%	97,00%
Kuhlmann and Thimm 2019 - Balanced	63,00%	17,00%	93,00%

4.4. Runtime Performance

Kuhlmann and Thimm [6] report an average runtime of 0.000007 seconds per argument measured over their entire test set compared against a mean runtime for their reference solver CoQuiAAS of 0.119561 seconds. Our runtime performance is consistent with this achieving an average runtime of 0.000005 seconds per argument over the complete test set.

5. Conclusion

The results in this article improve substantially on the previous baseline and while the accuracy achieved is perhaps not yet sufficient for most real-world applications, it does in some cases come close enough for consideration. Problems, however, remain around the disparity between positive and negative accuracy, although this in some cases is less marked than others. This problem would need to be addressed prior to any real-world deployment.

The reasonable success of this approach seems to be largely down to the use of the randomised training regime, which may have broader applicability beyond the domain of abstract argumentation especially in addressing other NP-hard graph structured problems. The effects of depth and balancing, while they do contribute in some models, are more mixed.

Finding ways to improve on this baseline may necessitate solving the question of why a deeper model does not in this case lead to better results or alternately combining this model with additional layers of a different type or a different kind of input features to improve learnability.

References

[1] Charwat G, Dvořák W, Gaggl SA, Wallner JP, Woltran S. Methods for solving reasoning problems in abstract argumentation - A survey. Artificial Intelligence. 2015;220:28–63.

[2] Woltran S. Abstract Argumentation – All Problems Solved ? In: ECAI 2014; 2014. p. 1–93.

[3] Cerutti F, Dunne PE, Giacomin M, Vallati M. Computing preferred extensions in abstract argumentation: A SAT-based approach. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). vol. 8306 LNAI; 2014. p. 176–193.

[4] Cerutti F, Gaggl S, Thimm M, Wallner J. Foundations of implementations for formal argumentation. IfCoLog Journal of Logics and their Applications. 2017;4(8):2623–2705. Available from: http://argumentationcompetition.org.

[5] Thimm M. Stochastic local search algorithms for abstract argumentation under stable semantics. Frontiers in Artificial Intelligence and Applications. 2018;305:169–180.

[6] Kuhlmann I, Thimm M. Using Graph Convolutional Networks for Approximate Reasoning with Abstract Argumentation Frameworks: A Feasibility Study. Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics). 2019;11940 LNAI:24–37.

[7] Dung PM. On the acceptability of arguments and its fundamental role in nonmonotonic reasoning, logic programming and n-person games. Artificial Intelligence. 1995:321–357.

[8] Kipf TN, Welling M. Semi-supervised classification with graph convolutional networks. In: 5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings; 2019. Available from: http://arxiv.org/abs/1609.02907.

[9] Monti F, Boscaini D, Masci J, Rodolà E, Svoboda J, Bronstein MM. Geometric deep learning on graphs and manifolds using mixture model CNNs. In: Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017. vol. 2017-Janua; 2017. p. 5425–5434. Available from: https://arxiv.org/pdf/1611.08402.pdf.

[10] Levie R, Monti F, Bresson X, Bronstein MM. CayleyNets: Graph Convolutional Neural Networks with Complex Rational Spectral Filters. IEEE Transactions on Signal Processing. 2019 may;67(1):97–109. Available from: http://arxiv.org/abs/1705.07664.

[11] Wu Z, Pan S, Chen F, Long G, Zhang C, Yu PS. A Comprehensive Survey on Graph Neural Networks. IEEE Transactions on Neural Networks and Learning Systems. 2020:1–21.

[12] Riba P, Fischer A, Lladós J, Fornés A. Learning Graph Distances with Message Passing Neural Networks. In: Proceedings - International Conference on Pattern Recognition. vol. 2018-Augus. IEEE; 2018. p. 2239–2244. Available from: https://ieeexplore.ieee.org/document/8545310/.

[13] Wang M, Tang Y, Wang J, Deng J. Premise selection for theorem proving by deep graph embedding. In: Advances in Neural Information Processing Systems. vol. 2017-Decem; 2017. p. 2787–2797.

[14] Lemos H, Prates M, Avelar P, Lamb L. Graph Colouring Meets Deep Learning: Effective Graph Neural Network Models for Combinatorial Problems. 2019. Available from: http://arxiv.org/abs/1903.04598.

[15] Perozzi B, Al-Rfou R, Skiena S. DeepWalk: Online learning of social representations. Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. 2014:701–710. Available from: https://arxiv.org/pdf/1403.6652.pdf.

[16] Srivastava N, Hinton G, Krizhevsky A, Sutskever I, Salakhutdinov R. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. Journal of Machine Learning Research. 2014;15(56):1929–1958. Available from: http://jmlr.org/papers/v15/srivastava14a.html.

[17] Kingma DP, Ba JL. Adam: A method for stochastic optimization. In: ICLR 2015; 2015. p. 1–15. Available from: https://arxiv.org/pdf/1412.6980.pdf.

[18] Zhang J, Meng L. GResNet: Graph Residual Network for Reviving Deep GNNs from Suspended Animation. 2019;(2016):1–18. Available from: http://arxiv.org/abs/1909.05729.

[19] Alviano M. The pyglaf argumentation reasoner. OpenAccess Series in Informatics. 2018;58:2–5.