Yonas: An Experimental Neural Argumentation Solver

Lars Malmqvist[1]
[1] University of York, York, U.K.
lm1775@york.ac.uk

Abstract

Yonas is an experimental deep reinforcement learning based abstract argumentation solver submitted to the ICCMA 2019 competition. The solver builds on previous work using deep reinforcement learning in fields related to abstract argumentation and in generating solutions to perfect information games in order to provide a new direct approach to reasoning about abstract argumentation graphs. The solver is implemented in Python using PyTorch, and uses a combination of Deep Reinforcement Learning and Monte Carlo Tree Search to solve problems in abstract argumentation.

1 Introduction

Yonas (short for York Neural Argumentation Solver) is an experimental, hybrid abstract argumentation solver that uses Deep Reinforcement Learning (DRL) and Monte Carlo Tree Search (MCTS) to learn how to solve abstract argumentation frameworks. The solver encodes argumentation frameworks in a tree based game structure and implements labelling based reward functions to pre-train a deep reinforcement learning model that is used in MCTS at runtime to guide the tree search in finding solutions.

The solver at this point is capable of finding solutions using a combination of argument games and guided tree-search. It supports the classic semantics (Complete, Preferred, Grounded, Stable), which are implemented using a variety of algorithms that combine to provide the complete solver. The solver is currently highly experimental and has empirically observed limitations that mean some argumentation graphs cannot be solved within a reasonable timespan. However, it provides a proof-of-concept for applying machine learning methods to the solution of argumentation frameworks.

2 Deep Reinforcement Learning and Monte Carlo Tree Search for Abstract Argumentation

The solution to abstract argumentation frameworks can be thought of as a dialogue game where two opponents take turns making arguments and counterarguments until a resolution is found or it is determined that no resolution is possible. The process of making arguments and counterarguments induces a dispute tree that can be analysed by an appropriate algorithm to solve argumentation problems (Modgil and Caminada 2009). Yonas takes this idea as a point of departure, but instead of using previous approaches to solving dialogue games using an explicit hand-coded algorithm, Yonas learns to solve argumentation frameworks through reinforcement learning.

Combining Deep Reinforcement Learning and Monte Carlo Tree Search has become a dominant approach to the solution of many types of problems that have game like features especially following the success of AlphaGo Zero (Silver et al. 2017). In this approach, the tree search conducted by MCTS is guided by a set of probabilities computed by the deep neural net and the moves taken by MCTS feed back into the training of the neural net. Recent work has seen this or similar approaches applied to a areas such as proof search for automated theorem proving (Kaliszyk et al. 2018) or the solution of Rubik's Cubes (McAleer et al. 2018) that share several important characteristics with the solution of abstract argumentation frameworks.

Within argumentation research, "classic" reinforcement learning has been used to solve simple argumentation problems (Alahmari et al. 2017). However, this previous work was not aimed at providing a complete solver and didn't deploy any DRL methods. Yonas attempts to bridge this gap and take a step towards a fully learned DRL based abstract argumentation solver, although admittedly it is still at an early stage.

3 Implementation

3.1 Design of the Solver

The solver is implemented in Python using the PyTorch[1] deep learning framework. There are two important parts to the solver:

1. **Pre-training.** During pre-training a DRL model is trained on a set of benchmark argumentation frameworks taken from a past ICCMA competition. The network architecture is a simple feed-forward network augmented with a graph embedding.
2. **Runtime.** At runtime the solver uses MCTS to search for solutions to a specific argumentation framework using the pre-trained DRL model to provide the heuristic guiding the tree search.

Pre-training is done using Deep Q-Learning (Mnih et al. 2015) on a set of argumentation frameworks taken from the ICCMA 2017 competition[2]. By exploring the structure of these frameworks using a hand-coded reward function that is based on evaluating a labelling of the argumentation graph, the deep neural net learns to rank potential actions in the search space as more or less likely to lead to a solution. The quality of the learning is evaluated post-hoc by comparing to solutions from high-ranking solvers in the 2017 competition.

[1] https://pytorch.org/

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

The runtime algorithm works by conducting a fixed-time tree search in the space of relevant extensions using the neural net to provide a heuristic for the search. Once a suitable base case has been found or the time limit has been reached, the algorithm computes the answer to the specific query using a specialised approach depending on the semantic and the particular problem type under consideration.

Specifically, the following approaches are used for the different reasoning tasks:

- The Grounded Extension is found using a labelling based algorithm adapted from Modgil and Caminada (2009), but optimised to work using mostly tensor-based operations to allow it to perform well with the PyTorch framwork
- Enumeration of all other semantics is done using MCTS guided by a labelling based algorithm that has been adapted to work using the approach described in Schadd et al. that describes an approach for using MCTS with single-player games. The labeling based algorithm has been adapted to exhibit the features of a single-player game.
- For Credulous Acceptance in Complete and Preferred semantics the "Socratic" dialogue game described in Caminada 2014 has been implemented more or less verbatim and a model has been trained using Deep Reinforcement Learning with the PyTorch framework. At runtime, MCTS is run using the pre-trained model to determine the result.
- For Skeptical Acceptance, the solver uses a combination of enumeration and the credulous acceptance algorithm. The solver uses parallel processes to simultaneously prove and disprove the argument under consideration either via complete enumeration in the positive case or via credulous justification of an attacking argument in the negative case.

3.2 Competition Specific Information

The solver implements functionality only for the four classic semantics (Dung 1995): Complete, Grounded, Preferred, and Stable. It does not provide any functionality for Ideal, Semi-Stable, or Stage semantics at this point in time.

All four problem types (SE, EE, DC, and DS) are supported for Complete, Preferred, and Stable Semantics and the two relevant problem types (SE, DC) are supported for Grounded semantics.

The Docker container for the solver is available publicly from Docker hub with the link lmalmqvist/yonas.

References

Alahmari, S., Yuan, T., Kudenko, D. (2017). Reinforcement learning for abstract argumentation: Q-learning approach. In *Adaptive and Learning Agents workshop* (at AAMAS 2017)

Caminada, M., Dvořák, W.,Vesic, S. (2014) Preferred semantics as socratic discussion. *J. Log. Comput*.

Dung, P.M.(1995). On the Acceptability of Arguments and its Fundamental Role in Nonmonotonic Reasoning, Logic Programming and n-Person Games. *Artificial Intelligence 77*, 321–358

Kaliszyk, C., Urban, J., Michalewski, H., Olk, M. (2018). Reinforcement Learning of Theorem Proving. *arXiv preprint arXiv:1805.07563*

McAleer, S., Agostinelli, F., Shmakov, A., and Baldi, P. (2018). Solving the rubik's cube without human knowledge. *arXiv preprint arXiv:1805.07470*

Mnih, V., Kavukcuoglo, K., Silver, D., Rusu, A., Veness, J., Bellemare, M., Graves, A., Riedmiller, M., Fidgeland, A., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature 518*, 529–533

Modgil, S., & Caminada, M. (2009). Proof Theories and Algorithms for Abstract Argumentation. In *Argumentation for Artificial Intelligence* (pp. 105-130). New York: Springer.

Schadd, M.. Winands, M., van den Herik, H. , Chaslot, G., Uiterwijk, J. (2008) Single-player Monte-Carlo tree search, *Proc. Comput. Games*, 1-12

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., van den Driessche, G., Graepel, T., Hassabis, D. (2017). Mastering the game of go without human knowledge. *Nature 550*, 354–359