A Roadmap for Neuro-argumentative Learning

Maurizio Proietti¹*, Francesca Toni²*

¹IASI-CNR, Rome, Italy
²Department of Computing, Imperial College London, UK

Abstract

Computational argumentation (CA) has emerged, in recent decades, as a powerful formalism for knowledge representation and reasoning in the presence of conflicting information, notably when reasoning non-monotonically with rules and exceptions. Much existing work in CA has focused, to date, on reasoning with given argumentation frameworks (AFs) or, more recently, on using AFs, possibly automatically drawn from other systems, for supporting forms of XAI. In this short paper we focus instead on the problem of learning AFs from data, with a focus on neuro-symbolic approaches. Specifically, we overview existing forms of neuro-argumentative (machine) learning, resulting from a combination of neural machine learning mechanisms and argumentative (symbolic) reasoning. We include in our overview neuro-symbolic paradigms that integrate reasoners with a natural understanding in argumentative terms, notably those capturing forms of non-monotonic reasoning in logic programming. We also outline avenues and challenges for future work in this spectrum.

Keywords
Computational argumentation, Artificial neural networks, Non-monotonic reasoning

1. Introduction

Computational argumentation (CA) has emerged, since the nineties, as a powerful formalism for knowledge representation and reasoning in the presence of conflicting information (see [1, 2] for recent overviews). It has been shown to capture and generalise, in particular, several forms of non-monotonic reasoning [3, 4], notably required when operating with rules and exceptions naturally giving rise to conflicts (e.g. both the rule “birds fly” and the exception “penguins do not fly” apply to the same individual tweety – a penguin and thus a bird). Also, it is being widely deployed to support forms of explainable AI (XAI) (e.g. see overview in [5]), given the appeal of argumentation in explanations amongst humans, e.g. as in [6], within the broad view that XAI should take findings from the social sciences into account [7].

To date, the bulk of work in CA amounts to defining so-called argumentation frameworks (AFs), which are symbolic representations equipped with semantics/tools for reasoning towards the resolution of conflicts and drawing (argumentatively acceptable) conclusions. In addition, increasingly more attention has been given over the years to combining CA and machine
learning (e.g. as overviewed in [8]). Here, we focus specifically on methods combining CA and machine learning with artificial neural networks (NNs), focusing on what we term neuro-argumentative (machine) learning (NAL), resulting from a combination of neural (machine learning) mechanisms and (symbolic) argumentative reasoning using methods in CA. Specifically, in Section 2 we overview existing forms of NAL, including neuro-symbolic paradigms integrating forms of logic programming with a natural understanding in CA terms (see below). Finally, in Section 3 we outline avenues and challenges for future work in the broad spectrum of NAL.

**Background on CA.** The simplest form of AFs is given by abstract AFs (AAFs) [3], which boil down to directed graphs whose nodes are arguments (e.g. “tweety flies as it is a bird” and “tweety does not fly as it is a penguin”) and whose edges are attacks between them (e.g., the latter argument for tweety attacks the former). They are equipped, e.g., with the semantics of stable extensions, which are conflict-free sets of arguments attacking every argument they do not contain (leading to accepting the second argument for tweety and inferring that it does not fly).

Several other forms of CA have been proposed and play a role in existing works in NAL, notably bipolar AFs (BAFs) [9], i.e. directed graphs with two types of edges: attacks as in AAFs and supports (e.g. between “tweety has wings” and the earlier “tweety flies as it is a bird”), variations of AAFs and BAFs such as quantified BAFs (where arguments are equipped with base scores and dialectical strengths obtained using gradual semantics [10]), weighted (quantified) BAFs (where edges are weighted, e.g. see [11]), and forms of structured CA [12], where arguments and attacks are drawn from logical formalisms. For example, in Assumption-Based AFs (ABAFs) [4, 13, 14, 15], arguments are drawn from “rules” and are supported by “assumptions”, and attacks are targeted at the assumptions, by arguments for their “contraries”.

CA and non-monotonic reasoning. CA is naturally non-monotonic and indeed there are some close connections between (forms of) CA and several formalisms for non-monotonic reasoning [3, 4], including normal logic programs (LPs in short) with negation as failure (NAF) and answer set programs (ASPs in short). In particular, ABAFs admit LPs/ASPs as instances. For illustration, consider the LP/ASP \( P \cup F \) with rules \( P = \{ \text{flies}(X) \leftarrow \text{bird}(X), \neg\text{flies}(X), \neg\text{flies}(X) \leftarrow \text{penguin}(X) \} \) (with \( \neg \) interpreted syntactically, and \( \neg \) denoting NAF) and facts \( F = \{ \text{penguin(tweety)}, \text{bird(tweety)} \} \). This LP/ASP corresponds to an ABAF with

- rules \( P \cup F \),
- all NAF literals in the vocabulary of the LP as assumptions,
- contraries of assumptions \( \neg l \) given by \( l \).

Note that, in particular, assumptions include \( \neg\neg\text{flies(tweety)} \), with contrary \( \neg\text{flies(tweety)} \). The semantics of the original LP/ASP is exactly captured by the semantics of the ABAF [4]. For example, stable extensions of ABAFs correspond exactly to stable models of LPs/ASPs [16].

---

1We will refer to several standard NN architectures, notably multi-layer perceptrons (MLPs), convolutional neural networks (CNNs), Long-Short-Term-Memory NNs (LSTMs), and AutoEncoders (AEs).
2. Overview of existing work in NAL

Here, we focus on works at the intersection of CA and NNs positioned as NAL, while ignoring works combining CA and NNs for supporting CA itself, e.g.: NNs performing argumentative reasoning, as in [17, 18, 19]; works on argument mining, using NNs to extract arguments and/or AFs, as in [20]; methods using NNs for computing semantics of AFs, such as [21, 22, 23]; works on the correspondence between AFs of various types and NNs, as in [24, 25].

We summarise existing works on NAL in Table 1 and related works in neuro-symbolic learning with LP/ASP in Table 2, and describe their main characteristics for our purposes below.

Translation of NNs to AFs. A line of work includes methods that take a trained NN as an input and translate it into an AF of some kind. Amongst these methods, DAX [26] envisages generic mappings into generalised AFs, with any number of dialectical relations. Several concrete instantiations of this approach focus on mapping NNs (e.g. MLPs or CNNs) into BAFs [27, 28]. Further, SpArX [29] relies upon a provable one-to-one-correspondence between MLPs and weighted BAFs [11] to translate MLPs into sparse weighted BAFs, where hidden neurons are clustered, for the purpose of rendering the MLP sparser and more interpretable. These methods aim at using the AFs resulting from the translation for explaining the NNs.

Pipeline methods. Some other methods use NNs to extract AFs, and then apply symbolic, argumentative reasoning on the AFs to support some downstream task. For example, ADA [30] uses neural classifiers and LSTMs to mine quantified BAFs from textual reviews, and then reasons with them (under gradual semantics) to provide recommendations for movies, evaluated against review aggregations measures. Also, DeAr [31] uses AutoEncoders for feature selection from tabular data to extract AAFs, and then reasons with them (under extension-based semantics) to provide explainable classifications from tabular inputs. Further, Local-HDP-ABL [32] deploys (possibly neural) feature selection methods with images to extract BAFs for reasoning over the images for explainable classification. In these methods the two components (AF extraction and symbolic, argumentative reasoning) are decoupled within a pipeline, and the argumentative reasoning provides “delayed” feedback to the AF extraction by the NN.

Integrated methods. Finally, argumentative reasoning over AFs has been integrated within NNs as a form of inductive bias, to guide the learning within the NNs. Specifically, NSAM [33]

<table>
<thead>
<tr>
<th>Paper</th>
<th>Formalism</th>
<th>Learning Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>DAX [26, 27, 28]</td>
<td>BAFs</td>
<td>Translation of NNs (MLPs, CNNs) to AFs</td>
</tr>
<tr>
<td>SpArX [11, 29]</td>
<td>Weighted BAFs</td>
<td>Translation of MLPs to AFs</td>
</tr>
<tr>
<td>ADA [30]</td>
<td>Quantified BAFs</td>
<td>AFs learnt by NNs → argumentative reasoning</td>
</tr>
<tr>
<td>DeAr [31]</td>
<td>AAFs</td>
<td>AFs learnt by AEs → argumentative reasoning</td>
</tr>
<tr>
<td>Local-HDP-ABL [32]</td>
<td>BAFs</td>
<td>AFs learnt by NNs → argumentative reasoning</td>
</tr>
<tr>
<td>NSAM [33]</td>
<td>probabilistic semi-AAFs</td>
<td>Reasoning with AFs as inductive bias</td>
</tr>
</tbody>
</table>

Table 1

NAL approaches based on argumentation frameworks (AFs) of various kinds
defines argumentation Boltzmann machines (a form of NNs capturing argumentative knowledge, in the form of probabilistic semi-abstract AFs), trained on instances of the argumentative knowledge applicable to given data, to make predictions that can be explained argumentatively.

Related work in logic programming. We include neuro-symbolic paradigms integrating forms of non-monotonic LPs and ASPs, as those have a natural understanding in CA terms. We omit instead neuro-symbolic methods focusing on positive logic programs (e.g. see overview in [34]), as they support monotonic reasoning only and are not relevant to an argumentative viewpoint. The LP works are summarised in Table 2. CIL$^2$P [35] extracts LPs from MLPs, for the purposes of explainability. Some methods follow a pipeline approach, learning LPs from NNs pre-trained for feature extraction on images: NeSyFOLD [36] generates stratified LPs from CNNs, and FFNSL [37] learns ASPs. Other methods integrate reasoning with LPs/ASPs provided by humans, for the purpose of training NNs end-to-end (e2e) to benefit from the knowledge represented in the LPs/ASPs and learn from it as well as from unstructured data such as images: NeurASP [38] represents knowledge in the form of probabilistic ASPs, DeepProbLog [39] uses probabilistic stratified LPs (i.e. ProbLog programs), NeuroLog [40] makes use of abductive LP to supervise the e2e training process, and embed2sym [41] integrates clustering for feature extraction with ASPs. Finally, pix2rule [42] learns ASPs e2e, through a training regime that processes both images and ASPs, by including a differentiable layer in NNs from which ASP rules can be extracted.

3. Challenges

We conclude by discussing the role that NAL could have in the future development of neuro-symbolic systems in three settings: 1) the NN component is pre-trained, AFs are learnt; 2) AFs are predefined, the NN component is learnt; 3) both components are learnt e2e.

NNs pre-trained, AFs learnt. The simplest way in which CA could be used is within a simple pipeline architecture: an AF is extracted (or learnt) from a pre-trained NN. This approach is very close to standard CA-based forms of XAI as discussed earlier [30, 31, 32]. A more advanced variant of this pipeline architecture could be implemented by representing pre-trained NN modules symbolically as (probabilistic) neural predicates, similarly to [36, 39]. With this

<table>
<thead>
<tr>
<th>Paper</th>
<th>Formalism</th>
<th>Learning Setting</th>
</tr>
</thead>
<tbody>
<tr>
<td>CIL$^2$P[35]</td>
<td>Propositional LP</td>
<td>Translation of NNs to LP</td>
</tr>
<tr>
<td>NeSyFOLD[36]</td>
<td>Stratified LP</td>
<td>LP learnt from pre-trained CNN</td>
</tr>
<tr>
<td>FFNSL[37]</td>
<td>ASP</td>
<td>LP learnt from pre-trained NNs</td>
</tr>
<tr>
<td>NeurASP[38]</td>
<td>Probabilistic ASP</td>
<td>e2e training of NNs via user-defined rules</td>
</tr>
<tr>
<td>DeepProbLog[39]</td>
<td>Probabilistic Stratified LP</td>
<td>e2e training of NNs via user-defined rules</td>
</tr>
<tr>
<td>NeuroLog[40]</td>
<td>Abductive LP</td>
<td>e2e training of NNs via user-defined rules</td>
</tr>
<tr>
<td>embed2sym[41]</td>
<td>ASP</td>
<td>e2e training of NNs via user-defined rules</td>
</tr>
<tr>
<td>pix2rule[42]</td>
<td>ASP</td>
<td>e2e learning of ASP</td>
</tr>
</tbody>
</table>

Table 2

Neuro-symbolic approaches based on non-monotonic rules (mappable to AFs)
representation, together with suitable background knowledge by domain experts, we could learn symbolic concepts as ABAFs (e.g. as suggested in [43]) or as probabilistic ABAFs [44].

**Predefined AFs, NNs learnt.** In this type of systems NNs are trained e2e: the input to NNs is labeled by symbolic concepts for which we provide a suitable background knowledge defined by means of AFs. In this context, ABA frameworks could be used to guide the training of NNs, again extending the work done in the area of LP and ASP [36, 39]. The ability of ABAFs to formalise various kinds of knowledge/reasoning (e.g. default theories, ontologies, and temporal logics) would be a plus for achieving better accuracy and reliability with smaller data sets.

**NNs and AFs both learnt.** The most challenging task is to construct systems that are composed of neural modules and symbolic modules where both components are learnt at the same time, while training is done e2e on their composition. A critical sub-task is learning *latent concepts*, that is, the symbolic concepts associated with the output of the neural component. Initial work in this direction, for LP/ASP, could provide a starting point. For example, [45] design an approach based on so-called *policy functions*, similar to reinforcement learning, to learn symbolic knowledge which could be seen as a collection of facts (and thus *monotonic*). While this work makes quite strong hypotheses on the form of the symbolic knowledge to be learnt, it would be interesting to explore whether the approach could be generalised to learn (non-monotonic) AFs. Furthermore, the aforementioned [42] – proposing a specific approach for e2e learning of relations and rules from images – could provide a fruitful starting point to learn AFs.

Although CA-based approaches have not been considered in this setting, they might be advantageous. First, the features of ABAFs useful for learning the two components separately can also be useful for their combined learning. For example, in this more complex scenario we could exploit ABAFs, following an iterative approach, for a variety of tasks such as: represent rich background knowledge, use that knowledge to generate suitable neural model templates, to train them, and extract new knowledge from trained neural models. Further, for this more complex task, we need representation and reasoning mechanisms that cope with the non-monotonicity of knowledge extraction and learning. To this aim, the ability of CA to support various forms of non-monotonic reasoning and to represent the learnt symbolic knowledge in the form of defeasible rules, could play a key role. Notably, we believe that abduction, which can be realised naturally in CA, could be useful for learning latent concepts.

### 4. Acknowledgments

We would like to thank the anonymous reviewers for constructive criticism. We also thank support from the Royal Society, UK (IEC\R2\222045 - International Exchanges 2022 Cost Share). M. Proietti is a member of the INDOM-GNCS research group, Italy. F. Toni was partially funded by the European Research Council (ERC) under the European Union’s Horizon 2020 research and innovation programme (grant agreement No. 101020934) and by J.P. Morgan and the Royal Academy of Engineering under the Research Chairs and Senior Research Fellowships scheme.
References


