Investigating Logic Tensor Networks for Neural-Symbolic Argument Mining

Anonymous ACL submission

Abstract

We present an application of neural-symbolic learning to argument mining. We use Logic Tensor Networks to train neural models to jointly fit the data and satisfy specific domain rules. Our experiments on a corpus of scientific abstracts indicate that including symbolic rules during the training process improves classification performance, compliance with the rules, and robustness of the results.

1 Introduction

Argument Mining (AM) stemmed from Natural Language Processing (NLP) and Knowledge Representation and Reasoning (Cabrio and Villata, 2018), with the goal of automatically extracting arguments and their relations from natural language texts (Lippi and Torroni, 2016). Like in most areas of NLP, deep learning has recently pushed the envelope also in AM. Yet, many challenges still stand open, as argumentation involves tasks such as reasoning, debate and persuasion, which cannot be easily addressed by deep architectures alone, sophisticated as they may be. For that reason, Galassi et al. (2019) argue that a combination of symbolic and sub-symbolic approaches could leverage significant advances in AM. They illustrate the point using two neural-symbolic (NeSy) frameworks, DEEPPROBLOG (Manhaeve et al., 2021) and Grounding-Specific Markov Logic Networks (Lippi and Frasconi, 2009), albeit without empirical evaluations.

Unfortunately, many of the existing frameworks are under continuous development and their applications are often limited to a few case studies in a single domain. Pacheco and Goldwasser (2021) analyze existing NeSy frameworks, observing that they are not specifically designed to support a variety of NLP tasks, and critically lack of a series of important features. We shall add to the list of shortcomings a lack of support for *collective classification* (Sen et al., 2008). This is a fundamental feature for AM, since argument analysis is typically context-dependent, meaning that classifying each argumentative component (or relation) requires considering not only the attributes of that component or relation, but also those of other connected components and relations. To address these limitations, Pacheco and Goldwasser introduce a NeSy framework based on factor graphs called DRAIL and show its application in the AM domain. To the best of our knowledge, no other NeSy approaches to AM have been investigated so far.

The present study focuses on neural-symbolic methods for AM. Besides the in-depth contribution, its aim is to pave the way for a broader application of such methods in the NLP domain. We address AM using a different NeSy framework, namely Logic Tensor Networks. We focus on the classification of argumentative component and prediction of links between component pairs. Importantly, LTNs allow to easily decouple the symbolic and sub-symbolic parts of the model, and enable collective classification during training. Our results indicate that the introduction of logic rules improves classification performance, compliance with the rules, and robustness of the results. To the best of our knowledge, this is the first application of LTNs to NLP.

2 Logic Tensor Networks (LTNs)

Logic Tensor Networks (LTNs) (Serafini and d'Avila Garcez, 2016; Donadello et al., 2017) integrate first-order many-valued logical reasoning (Bergmann, 2008) with tensor networks (Socher et al., 2013). The framework is implemented in TensorFlow (Abadi et al., 2016). LTNs belong to the "tensorization" class of undirect NeSy approaches (De Raedt et al., 2020) which embed First-Order Logic (FOL) entities, such as constants and facts, into real-valued tensors. The framework enables to combine data-driven machine learning with background knowledge expressed through first-order fuzzy logic represen-

tations. Therefore, one can use FOL to impose soft constraints at training time and investigate properties at test time. Once trained, neural architectures can be used independently of the framework. One can also use LTNs as a verification tool, to assess the ability of any given network to respect any given property, expressed as an FOL query.

LTN *variables* are an abstract representation of data. They must be linked to a set of real-valued vectors, which are all the possible groundings of that variable. A single data point of this set can be represented using LTN *constants*. LTN *functions* represent operations over variables and produce real-valued vectors. The evaluation is done by a set of TensorFlow operations, e.g., a neural network, defined together with the function. LTN *predicates* are a special class of functions whose output is a single real value between 0 and 1, which represents the degree of truth of the predicate. They can be used to represent classes of objects as well as properties that may hold between multiple objects. The learning setting is defined in terms of LTN *axioms*, i.e., formulas that specify logic conditions in terms of predicates, functions, and variables and can be used to assign labels to data and to specify soft constraints. Axioms can include logical connectives ($\wedge$, $\vee$, $\sim$, $\Rightarrow$)[1] and quantifiers ($\forall$, $\exists$).

Reasoning is performed in the form of *approximate satisfiability*, which means that the optimization process aims to maximize the level of satisfiability of a grounded theory, by minimizing the loss function (Serafini and d'Avila Garcez, 2016).

3 Argument Mining with LTNs

We frame component classification and link prediction as two classification tasks. To address them, we define two neural networks: NNCOMP and NNLINK. The first network takes a component and produces a probability distribution over the possible component classes. The second one receives two components and outputs a single value between 0 and 1, which represents the probability of there being an argumentative link between them.

Data-driven optimization is defined through three elements for each class of both tasks: a variable, a predicate, and an axiom. The variable is associated with all the data of the training set that belong to that class. The predicate is linked to the corresponding output of our networks. The axiom combines the previous elements and defines the optimization objective.

The rule-driven optimization is defined via variables linked to all the training data and through specific axioms that express the rules. For example, to enforce the antisymmetric property of links we define two variables (x and y), associate them with all the components of the training set, and specify the following axiom:

$$\forall x, y : LINK(x, y) \Rightarrow \sim LINK(y, x) \quad (1)$$

4 Experimental Setting

Before we describe our experimental setup, a word is in order about the implementation of LTNs we used, which does not expose APIs to easily configure some aspects of the training procedure. In the current implementation, when a predicate is defined in LTN over a set of variables, all the possible groundings of such variables are used as part of the same batch. This is necessary in order for the LTN to evaluate the predicate's truth degree. To clarify: given two components A and B, suppose one wants to determine if A and B are linked. This means evaluating LINK(A,B). In the current implementation, A and B need to belong to the same batch. Now, if we take a third component C and we want to determine LINK(A,C), A and C need to belong to the same batch. Also, Equation 2 creates a dependency between CLAIM and LINK, thus the optimization step must also consider the value of CLAIM(A), CLAIM(B) and CLAIM(C) alongside the value of the two LINK predicates. Since this applies to any pair of components, eventually all the data need to belong to the same batch. Accordingly, one cannot use mini-batches during training, which limits the scalability of the approach. Although this is not a theoretical limitation, it had a practical impact on our experimental setting, since it forced us to experiment with small-sized corpus, sentence embeddings, and neural architectures.

4.1 Data

The AbstRCT Corpus (Mayer et al., 2020, 2021) consists of 659 abstracts of scientific papers regarding randomized control trials for the treatment of specific diseases. The corpus includes three topical datasets: neoplasm, glaucoma, and mixed. The first one is divided into training, test, and validation splits, while the others are designed to be tests sets. The corpus contains about 4,000 argumentative components divided into two classes: EVIDENCE and CLAIM. Out of nearly 25,000 possi-

[1]The symbol $\sim$ stands for logical negation.

ble pairs of components that belong to the same document, about 10% are connected through a direct link. Claims only point to other claims. See Appendix A for further details.

Sentence embeddings are created from 25-size pre-trained GloVe embeddings (Pennington et al., 2014), by averaging over the words of the sentence. This simple method yields a low-dimensional representation with no need to train new embeddings or to rely on dimensionality reduction techniques.

4.2 Method

To evaluate whether the use of symbolic rules within a neural model benefits argument mining tasks, which is the aim of this work, we compare the results obtained by two different models, that differ only in the way they are trained. NEURAL is the model trained in the usual way, i.e., by only exploiting its sub-symbolic component. NESY is the model obtained by training the same architecture using also LTN axioms. We did not include comparisons with other state-of-the-art neural-symbolic architectures, because we could find none that could be taken off-the-shelf and used in our experiments.

For the NEURAL approach, we use three predicates, corresponding to the classes of the dataset: $LINK$, $EVIDENCE$, and $CLAIM$. For the NESY approach, we include axioms reflecting properties of the corpus, stipulating that (i) no symmetric link can exist (Eq. 1), and (ii) claims can be linked only to other claims (Eq. 2). The latter axiom connects the two tasks, thus inducing a joint-learning setting. Appendix B lists in detail all the LTN entities involved in our training setting.

$$\forall x, y : LINK(x, y) \wedge CLAIM(x) \Rightarrow CLAIM(y) \quad (2)$$

To avoid overfitting, we early-stop the process using the F1 score of link prediction on the validation set, with a patience of 1,000 epochs. We focus on link prediction because it is considered the most challenging task, and arguably the one that would benefit the most from the introduction of rules.

We evaluate the two models along the following dimensions:

- **Performance**: we measure the F1 metrics regarding link prediction and component classification, to assess whether the rules improve the performance of the models;
- **Robustness**: we compute the degree of agreement between the networks, to assess if the use of rules increases robustness against the intrinsic randomness of the training process;
- **Compliance**: we test whether the prediction of the models respects the desired properties.

4.3 Architecture

The aforementioned issues with the current LTNs implementation and our limited computational resources prevented us from integrating LTNs with NLP state-of-the-art models. However, we can still operate a meaningful comparison between NEURAL and NESY, *all else being equal*. Accordingly, we define a simple network composed of three stacked fully-connected layers followed by a softmax classification layer. To obtain more robust results with respect to the non-deterministic elements of the training procedure (Goodfellow et al., 2016), we follow Galassi et al. (2021) and train an ensemble of 20 networks both for NNCOMP and NNLINK, and evaluate the aggregated output. We implemented and compared two aggregation methods. Majority voting (MAJ) is a common one. However, it provides a categorical output, preventing a probabilistic interpretation of the prediction. Our alternative method is the average of the output of the networks (AVG). That, however, is known to be vulnerable to outliers.

4.4 Results

Table 1 summarizes the results of our experiments.[2] For the classification tasks, we report the macro-F1 score for component classification and the F1 score for the link class. Agreement is measured by Krippendorff's α, while the degree of truth of the properties is given as the ratio between the number of instances where the clause holds and the number of instances where only its left-hand side holds.

As far as the AM tasks, the difference between the MAJ and AVG approaches is negligible in the NESY setting, while it is more evident in the NEURAL setting for link prediction, where the majority voting achieves better performance.

As expected, rules seems to especially benefit *link prediction*, where the networks trained with rules perform consistently better than those trained without. Conversely, the latter perform marginally

[2] Since our focus is on evaluating the effect of rules, in our tables we did not include the performance results of state-of-the-art approaches, as these figures would be misleading.

Dataset	Split	Approach	Classification Comp.	Classification Link	Agreement Comp.	Agreement Link	Properties Eq. 1	Properties Eq. 2
Neoplasm	Val.	NEURAL	83 - 84	42 - 41	77	66	88 - 84	92 - 83
		NESY	84 - **85**	**44** - 43	**81**	**71**	**99** - 98	**99** - **99**
Neoplasm	Test	NEURAL	79 - **80**	34 - 31	77	64	87 - 81	96 - 85
		NESY	79 - 78	**35** - **35**	**79**	**70**	**99** - 96	**99** - 94
Glaucoma	Test	NEURAL	**82** - **82**	45 - 43	**75**	66	93 - 90	89 - 74
		NESY	81 - **82**	**47** - 45	**75**	**71**	≈**100** - 98	**99** - 90
Mixed	Test	NEURAL	**81** - **81**	38 - 34	75	64	89 - 85	95 - 86
		NESY	**81** - 80	39 - **40**	**76**	**69**	≈**100** - 97	**97** - 96

Table 1: Percentage scores obtained on the AbstRCT corpus. For classification and compliance, we report both the result obtained by the MAJ approach (before the dash) and by the AVG approach (after the dash).

better on component classification, in a few cases. The results are, however, comparable.

The use of rules clearly benefits *robustness*, boosting the agreement by at least 5 points for link prediction and a few points for component classification. This is also confirmed by the smaller difference between AVG and MAJ.

The greatest improvement regards the *compliance* with the rules. The NESY approach satisfies the properties almost perfectly in the MAJ setting, and achieves results above 90% in the AVG one. The baseline is consistently less compliant, and performs significantly worse in the AVG setting.

All these results hold for the three test sets.

5 Discussion

We presented the first application of LTN to NLP, and one of the few applications of NeSy approaches to AM. In our opinion, there are several advantages in such an approach.

From an *analysis/interpretation* perspective, logical rules play an active role not only during training but also at inference time, offering a means to investigate the behavior of the models.

From a *user* perspective, the definition of training rules and queries requires only a basic knowledge of FOL, which may contribute to reducing the divide between system architects and domain experts, who do not need to be also experts in machine learning, NeSy systems, or deep networks.

From an *architectural* perspective, the decoupling between symbolic and neural components allows changing either of them without any direct impact on the other, except for the definition of key concepts such as the predicates/labels of the problem. Such a modularity may be highly beneficial in the context of AM, where one could use the same neural architecture with different corpora by expressing different symbolic rules. Indeed, the structural diversity of datasets and labeling schemes is a known issue in AM research, often leading to tailored solutions (Lippi and Torroni, 2016).

Performance-wise, the introduction of two symbolic rules increased link prediction performance without hindering component classification performance, whereas it boosted robustness and largely improved compliance. While the networks used in our experiments are much simpler than state-of-the-art models, and clearly they do not achieve comparable performance, we speculate that rules may benefit advanced models as well. However, it is difficult to determine how much improvement can carry over from weak baselines to stronger models (Denkowski and Neubig, 2017).

On the down side, we shall remark that one major challenge for this kind of approaches is *scalability* to larger domains, and the fact that they are not specifically designed for NLP tasks, so their development is yet in its infancy.

As future work, we are considering the weighting of soft rules, so as to distinguish between rules expressing preferences (or theories) and those expressing constraints. Another direction regards the recognition of properties that are not explicit in the training data but can be defined through logical rules. This could allow the network to infer information regarding components or relations without labeled training data: for example, finding which claim is the major claim of a document, or which components agree with each other.

References

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2016. TensorFlow: Large-scale machine learning on heterogeneous systems. *CoRR*, abs/1603.04467.

Merrie Bergmann. 2008. *An introduction to many-valued and fuzzy logic: semantics, algebras, and derivation systems*. Cambridge University Press.

Elena Cabrio and Serena Villata. 2018. Five years of argument mining: a data-driven analysis. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, pages 5427–5433. International Joint Conferences on Artificial Intelligence Organization.

Luc De Raedt, Sebastijan Dumancic, Robin Manhaeve, and Giuseppe Marra. 2020. From statistical relational to neuro-symbolic artificial intelligence. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, pages 4943–4950. ijcai.org.

Michael J. Denkowski and Graham Neubig. 2017. Stronger baselines for trustable results in neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation, NMT@ACL 2017, Vancouver, Canada, August 4, 2017*, pages 18–27. Association for Computational Linguistics.

Ivan Donadello, Luciano Serafini, and Artur D'Avila Garcez. 2017. Logic tensor networks for semantic image interpretation. In *IJCAI*, page 1596–1602. AAAI Press.

Andrea Galassi, Kristian Kersting, Marco Lippi, Xiaoting Shao, and Paolo Torroni. 2019. Neural-symbolic argumentation mining: An argument in favor of deep learning and reasoning. *Frontiers in Big Data*, 2:52.

Andrea Galassi, Marco Lippi, and Paolo Torroni. 2021. Multi-task attentive residual networks for argument mining. *CoRR*, abs/2102.12227.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press.

Marco Lippi and Paolo Frasconi. 2009. Prediction of protein β-residue contacts by markov logic networks with grounding-specific weights. *Bioinformatics*, 25(18):2326–2333.

Marco Lippi and Paolo Torroni. 2016. Argumentation mining: State of the art and emerging trends. *ACM Trans. Internet Technol.*, 16(2):10:1–10:25.

Robin Manhaeve, Sebastijan Dumančić, Angelika Kimmig, Thomas Demeester, and Luc De Raedt. 2021. Neural probabilistic logic programming in DeepProbLog. *Artificial Intelligence*, 298:103504.

Tobias Mayer, Elena Cabrio, and Serena Villata. 2020. Transformer-based argument mining for healthcare applications. In *ECAI 2020 - 24th European Conference on Artificial Intelligence*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, pages 2108–2115. IOS Press.

Tobias Mayer, Santiago Marro, Elena Cabrio, and Serena Villata. 2021. Enhancing evidence-based medicine with natural language argumentative analysis of clinical trials. *Artificial Intelligence in Medicine*, page 102098.

Maria Leonor Pacheco and Dan Goldwasser. 2021. Modeling content and context with deep relational learning. *Trans. Assoc. Comput. Linguistics*, 9:100–119.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543. Association for Computational Linguistics.

Prithviraj Sen, Galileo Namata, Mustafa Bilgic, Lise Getoor, Brian Gallagher, and Tina Eliassi-Rad. 2008. Collective classification in network data. *AI Mag.*, 29(3):93–106.

Luciano Serafini and Artur S. d'Avila Garcez. 2016. Learning and reasoning with logic tensor networks. In *AI*IA 2016 Advances in Artificial Intelligence*, pages 334–348, Cham. Springer International Publishing.

Richard Socher, Danqi Chen, Christopher D Manning, and Andrew Ng. 2013. Reasoning with neural tensor networks for knowledge base completion. In *Advances in Neural Information Processing Systems*, volume 26, pages 926–934. Curran Associates, Inc.

Dataset	Neoplasm			Glaucoma	Mixed
Split	Train	Valid.	Test	Test	Test
Documents	350	50	100	100	100
Components	2,267	326	686	594	600
Evidence	1,537	218	438	404	338
Claim	730	108	248	190	212
Couples	14,286	2,030	4,380	3,332	3,332
Links	1,418	219	424	367	329

Table 2: AbstRCT dataset composition.

A Corpus and architectural choices

The scalability issues have influenced most of our choices in terms of experimental setting.

- Needing low-dimensionality features, we used GloVe embeddings rather than more advanced methods because it offers low-dimensional (25-size) pre-trained word embedding.
- Due to the limited computational power we could afford, we had to rule out state-of-the-art architectures with millions of parameters. We have thus used simple neural architectures with a much lower number of parameters.
- The choice of the AbstRCT corpus was motivated by the following reasons: 1) it comes from real-world documents, as opposed to data elicited or subject to guidelines; 2) it offers multiple test sets, allowing a general evaluation; 3) all the documents have an argumentative structure; 4) the argumentative model presents a specific domain-related rule; 5) it fits our computational requirements because the training set contains a limited number of documents; 6) it has been presented and published in two high-quality venues.

The AbstRCT corpus is available at https://gitlab.com/tomaye/abstrct. Its composition is reported in Table 2. Some of the documents of the neoplasm and glaucoma test set are also included into the mixed set.

We applied GloVe embeddings, downloaded from https://nlp.stanford.edu/projects/glove/, directly on the words of the documents, without pre-processing. We used random embeddings for OOV words.

We use a neural network made of three stacked fully-connected layers of size 10, 20, and 10, followed by a softmax classification layer with two outputs: CLAIM or EVIDENCE for NNCOMP and LINK or NOLINK for NNLINK. We use ReLU as activation function, and employ dropout with probability $p = 0.4$ after each layer. The two models have 712 (NNCOMP) and 962 (NNLINK) trainable parameters.

B LTN training entities

Figure 1 lists the LTN entities involved in our training setting.

C Infrastructure and Runtime Details

All our experiments run on an ASRock Z370 Pro4 motherboard, GeForce GTX 1080 Ti GPU, Intel Core i7-8700K @ 3.70GHz CPU.

Using the baseline approach, the average training time for each network is less than one minute. Using our NeSy approach, the average training time for each network is 14 minutes, with a standard deviation of about 3 minutes. Inference on the whole ensemble of 20 networks in all the considered test datasets and approaches requires less than 30 seconds.

D Deeper Architecture

To better investigate whether the application of LTNs may impact also more complex architectures, we have repeated the experiments using deeper networks with 11 layers instead of 3, and more than 3,000 parameters instead of less than 1,000. Due to the aforementioned scalability limitations, we used only roughly 60% of the training documents.

It is important to remark that the less training data may heavily impact performance, possibly favoring the NESY setting, and therefore it is not possible to draw definitive conclusions based on these results. It is also important to underline that these architectures are nowhere near the complexity of Transformers-based architectures, such as the 100M parameters of BERT models.

The results are reported in Table 3. Overall, the use of the NESY setting has a positive impact on the networks, both for the classification task and the respect of the properties, coherently with our previous experiments. The main differences concerns robustness, which is negatively impacted both for component classification and link prediction. The other difference is limited to the Glaucoma test set, where the NEURAL setting achieves better scores for Eq. 1.

```
VARIABLES: # in LTN variable names are preceded by "?"
    ?train_claim # components belonging to the claim class
    ?train_premise # components belonging to the premise class
    ?train_data # all the training components
    ?train_data2 # all the training components
    ?train_links # pairs of components that are linked
    ?train_nolinks # pairs of components that are not linked

FUNCTIONS:
    concat # given two components, merges them to obtain a pair

PREDICATES:
    CLAIM
    PREMISE
    LINK

DATA-BASED AXIOMS: # used in all the setting
    forall ?train_claim: CLAIM(?train_claim)
    forall ?train_premise: PREMISE(?train_premise)
    forall ?train_links: LINK(?train_links)
    forall ?train_nolinks: ~LINK(?train_nolinks)

RULE-BASED AXIOMS: # used only in the NeSy setting
    forall ?train_data, ?train_data2 :
        LINK(concat(?train_data, ?train_data2)) ->
        ~LINK(concat(?train_data2, ?train_data))
    forall ?train_data, ?train_data2 :
        LINK(concat(?train_data, ?train_data2)) & CLAIM(?train_data) ->
        CLAIM(?train_data2)
```

Figure 1: LTN objects involved in our training setting.

			Classification		Agreement		Properties	
Dataset	Split	Approach	Comp.	Link	Comp.	Link	Eq. 1	Eq. 2
Neoplasm	Val.	NEURAL	80 - 74	41 - 35	**58**	**46**	88 - 68	96 - 87
		NESY	**81** - 80	**46** - 37	53	39	**100** - 58	**100** - 92
Neoplasm	Test	NEURAL	78 - 74	**33** - 26	**55**	**41**	87 - 67	93 - 87
		NESY	**79** - 77	**33** - 28	52	35	**94** - 64	≈**100** - 90
Glaucoma	Test	NEURAL	**80** - 76	39 - 34	**55**	**45**	**86** - 58	98 - 91
		NESY	**80** - 75	**42** - 37	52	39	81 - 44	≈**100** - 95
Mixed	Test	NEURAL	80 - 77	35 - 30	**57**	**44**	88 - 59	96 - 87
		NESY	**82** - 80	**39** - 29	54	37	**97** - 56	**100** - 96

Table 3: Percentage scores obtained on the AbstRCT corpus using a deeper architecture and a smaller training set. For classification and compliance, we report both the result obtained by the MAJ approach (before the dash) and by the AVG approach (after the dash).