Investigating Logic Tensor Networks for Neural-Symbolic Argument Mining

Andrea Galassi,[1] **Marco Lippi,** [2] **Paolo Torroni** [1]

[1] Department of Computer Science and Engineering (DISI), University of Bologna, Bologna, Italy
[2] Department of Sciences and Methods for Engineering (DISMI), University of Modena and Reggio Emilia, Modena, Italy
a.galassi@unibo.it, marco.lippi@unimore.it, paolo.torroni@unibo.it

Abstract

We present an application of neural-symbolic learning to the task of argument mining, where argument components and their relations are extracted from unstructured textual corpora. We use the framework of 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 to the rules, and robustness of the results. As in the case of other neural-symbolic applications, we further discuss how scalability remains a crucial issue.

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).

Argumentation is an ancient discipline that has its roots in logic and philosophy, aimed to study the way in which humans debate and reason. Inspired by the seminal work of Dung (1995), the application of computer science to the domain of argumentation has brought to the development of a fertile research area named computational argumentation. While several definitions of argument exist in the literature, one of the most intuitive has been given by Douglas Walton (2009): an argument is defined as a statement about a topic, usually named claim, possibly supported by a set of premises. The discipline of AM aims to extract such argument components from textual corpora, as well as the relations between them, which can be, for example, a supporting relation between a premise and a claim, or an attacking relation between two different claims.

Like in most NLP applications, deep learning has recently pushed the state-of-the-art also in AM. Yet, many challenges still stand open, as argumentation involves tasks such as reasoning, debate and persuasion that cannot be easily addressed by deep architectures only, 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 by exploiting domain knowledge and the constraints imposed by the underlying argument model. They illustrate that idea using two neural-symbolic (NeSy) frameworks, DEEPPROBLOG (Manhaeve et al. 2021) and Grounding-Specific Markov Logic Networks (Lippi and Frasconi 2009), but do not offer empirical evaluations. Unfortunately, many of the existing NeSy frameworks are under continuous development and their applications are often limited to a single domain and a few case studies. In particular, NLP tasks are seldom considered, though we believe they represent important and challenging benchmarks.

In this vein, 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 exquisitely context-dependent, and the task of classifying a single argumentative component (or relation) should be carried out by considering not only the attributes of that component or relation, but also of the attributes of other connected components and relations. To address these limitations, Pacheco and Goldwasser (2021) introduce the DRAIL NeSy framework and show its application in the AM domain. To the best of our knowledge, no other NeSy approach to AM has been investigated so far.

In this work, we address AM using a different NeSy framework, namely Logic Tensor Networks (Serafini and d'Avila Garcez 2016). We focus on the classification of argumentative component and prediction of links between component pairs. Importantly, LTNs allow us 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 to the rules, and robustness of the results. To the best of our knowledge, this is also the first application of LTNs to NLP.

Logic Tensor Networks (LTNs)

Logic Tensor Networks (LTNs) (Serafini and d'Avila Garcez 2016; Donadello, Serafini, and Garcez 2017; Badreddine et al. 2020) are a framework that integrates first-order many-valued logical reasoning (Bergmann 2008) with tensor networks (Socher et al. 2013), 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 representations. 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 from the framework.

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 ($\forall$, $\exists$).

LTNs, similarly to DEEPPROBLOG, enable the creation of *vertical-hybrid* learning systems, where high-level logic is placed on top of deep networks, as opposed to *horizontal-hybrid* learning (e.g., the work of Hu et al. (2016)), where the symbolic knowledge is encoded into the networks themselves (d'Avila Garcez et al. 2019). The idea behind the design of these systems is that the symbolic part must influence the behavior of the neural part and provide means to interpret their results.

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). Inference follows a model-theoretic perspective, which means that learning is done via shared parameters over the ground model, whereas inference is based on possible groundings of the model (De Raedt et al. 2020). After training, it is possible to evaluate queries expressed in FOL, as a means to assess the performance of the neural networks as well as to verify the degree of truth of a property.

Argument Mining with LTNs

We frame both component classification and link prediction as classification tasks. To address them, we define two neural networks, NNCOMP and NNLINK. The first network takes a component in input and produces a probability distribution over the possible component classes. The second receives two components and outputs a single value between 0 and 1, which represents the probability of there being an argumentative link between them.[2]

Data-driven optimization is defined through three elements for each class of both tasks: a variable, a predicate, and an axiom. The predicate is linked to the respective output of our networks, whereas the variable is associated to all the data of the training set that belong to that class, and the axiom combines the previous elements and defines the optimization objective. For example, given a class "CLAIM" of components, we define the variable $vClaim$, the predicate $CLAIM$ and the following axiom:

$$\forall\, vClaim : CLAIM(vClaim) \quad (1)$$

The rule-driven optimization is defined through 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 (vC_1 and vC_2) and associate them to all the components of the training set, and specify the following axiom:

$$\forall\, vC_1, vC_2 : LINK(vC_1, vC_2) \Rightarrow \sim LINK(vC_2, vC_1) \quad (2)$$

Experimental Setting

The implementation of LTNs we used does not expose APIs to easily configure some aspects of the training procedure.[3] Indeed, to guarantee the consistency of the tensor network, the training procedure employed in our experiments does not use mini-batches, which unfortunately has repercussions on the computational resources required. This limit is neither theoretical nor methodological, but it derives from the current implementation of the framework: when a predicate is defined in an LTN over a set of variables, all the possible groundings of such variables are used as part of the same batch. This is necessary for the LTN to evaluate the degree of truth of the predicate. Unfortunately, there is no way to easily construct a distinct mini-batch for each different document, therefore even a few simple rules that connect multiple entities, such as the rule shown in Equation 2, being necessarily applied to any component pair in the corpus, are sufficient to make all the data belong to the same batch.

Due to this scalability issue, we have chosen to experiment on the AbstRCT corpus, which has a limited number of documents, to represent sentences using sentence embeddings of small size, and to use neural architectures with a reduced number of trainable parameters.

Data

The AbstRCT Corpus (Mayer, Cabrio, and Villata 2020; Mayer et al. 2021) consists of 659 abstracts of scientific papers regarding randomized control trials for the treatment of specific diseases.[4] The corpus is divided into three topical

[1] The symbol $\sim$ indicates logical negation.

[2] This definition assumes that there is a single type of link. Otherwise, one should simply augment the number of output neurons of `nnLink` in order to match the number of possible relations.

[3] We used version 1.0 of the framework.

[4] The corpus is available at https://gitlab.com/tomaye/abstrct.

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 1: AbstRCT dataset composition.

datasets: neoplasm, glaucoma, and mixed. Neoplasm contains 500 abstracts divided into training (350), test (100), and validation (50) splits. The other two datasets contain 100 abstract each and are designed to be test sets.[5] To the best of our knowledge, this is the only corpus for AM that offers three test sets, allowing general evaluation.

The corpus contains about 4,000 argumentative components divided into two classes: EVIDENCE (2,808) and CLAIM (1,390). Out of the almost 25,000 possible pairs of components that belong to the same document, about 10% are connected through a direct link. Its composition is reported in Table 1. The argumentative model chosen for annotation enforces only one constraint: claims can have an outgoing link only to other claims.

Sentence embeddings are created using pre-trained GloVe embeddings of size 25 (Pennington, Socher, and Manning 2014), by averaging over the words of the sentence.[6] Such a simple method yields a low-dimensional representation without requiring to train new embeddings or rely on dimensionality reduction techniques. In the future we want to investigate more advanced sentence embeddings such as those presented by Reimers and Gurevych (2019) and Cer et al. (2018).

Architecture

For what concerns the neural architecture, we rely on a simple network. The aforementioned scalability issues have prevented us to experiment with NLP state-of-the-art models such as the Transformer (Vaswani et al. 2017) or BERT-based models (Devlin et al. 2019).

Our architecture is made of three stacked fully-connected layers of size 10, 20, and 10, followed by a softmax classification layer. 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. To obtain more robust results with respect to the non-deterministic elements of the training procedure (Goodfellow, Bengio, and Courville 2016), we follow Galassi, Lippi, and Torroni (2021) and train an ensemble of 20 networks both for NNCOMP and NNLINK, and evaluate the aggregated output. Majority voting (MAJ) could be a simple aggregation method. However, that would provide a categorical output, losing the probabilistic semantic of the prediction. That could be a drawback. An alternative would be to use the average of the output of the networks (AVG). That, however, would be vulnerable to outliers. We have therefore decided to try both approaches, so as to better evaluate the options from multiple perspectives. The MAJ and AVG approaches are two among the commonest aggregation methods: there are of course others. In particular, a possibility we plan to explore in future work is to represent the probability score assigned to a class as the percentage of networks that give it the highest probability. Such a method should guarantee robustness while fitting the fuzzy logic semantic of the framework.

Method

To properly evaluate whether the use of symbolic rules within the model yields positive results, we compare against a baseline model where only the sub-symbolic component is exploited. In the NeSy model, we include two LTN axioms based on characteristic properties of the corpus: (i) no symmetric link can exist, and (ii) claims can be linked only to other claims. For the purely data-driven approach, we make use of three predicates, corresponding to the classes of the dataset: $LINK$, $EVIDENCE$, and $CLAIM$. For our NeSy approach we include the axioms reported in Equations 2 and 3. In particular, the latter axiom connects the two tasks, thus inducing a joint-learning setting.

$$\forall\, vC_1, vC_2 : \; LINK(vC_1, vC_2) \wedge CLAIM(vC_1) \Rightarrow CLAIM(vC_2) \quad (3)$$

To avoid overfitting, we early-stop the process by monitoring the F1 score of link prediction on the validation set, using patience of 1,000 epochs. We intentionally 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 (neural baseline and NeSy) along the following dimensions:

- **Performance**: we measure the F1 score for the tasks of link prediction and component classification, to assess whether the injection of rules contributes to improve the performance of the models;

[5] Some of the documents of the neoplasm and glaucoma test set are also included into the mixed set.

[6] GloVe word embeddings can be downloaded at https://nlp.stanford.edu/projects/glove/.

- **Compliance**: we test whether the models respect the two desired properties, through LTN queries performed on the AVG ensemble;
- **Robustness**: we compute the degree of agreement between the networks related to the predictions of the MAJ ensemble, to assess whether the use of rules increase robustness against the intrinsic randomness of the training process.

Our experimental evaluation does not include a comparison with other neuro-symbolic methods, since no software can be easily used as an off-the-shelf competitor representing neuro-symbolic state-of-the-art approaches.

As for NLP state-of-the-art models, we decided to not include results obtained by previous neural approaches in our tables, considering them misleading for the purpose of this work. Indeed, our aim is to show the positive effect of the use of a neuro-symbolic framework over a plain neural architecture for a very challenging NLP task such as AM, by introducing symbolic rules as a driving element of the training procedure.

State-of-the-art models would have an inherent advantage against ours due to their large size (millions of parameters against less than one thousand), and therefore the comparison would not provide any useful information. In the future, after addressing the scalability issues, we aim to include such architectures in our experiments, by comparing their standard training mechanism against their training using LTNs.

Infrastructure and Runtime Details

We have performed all our experiments on the following infrastructure: 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 can be performed on the whole ensemble of 20 networks in less than 30 seconds in all the considered test datasets and for all the considered approaches.

Results

Table 2 summarizes the results of our experiments. For the classification tasks, we report the macro-F1 score for component classification and the F1 score for the link class. The agreement is measured as Krippendorff's α, while the degree of truth of the properties is evaluated through LTN queries. For what concerns the AM tasks, the difference between the MAJ and AVG approaches is negligible in the rule-based setting, while it is more evident in the no-rules setting for link prediction, where the majority voting achieves better performance.

The use of rules seems to be beneficial especially for the task of *link prediction*, where the networks perform consistently better than those trained without rules. Conversely, in a few cases the latter perform marginally better on component classification. 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 benefit is confirmed also by the smaller difference between the AVG and the MAJ approaches for classification.

Finally, as far as compliance to the rules, we observe that a purely data-driven approach already satisfies the properties almost completely. However, the introduction of rules during training further improves compliance, pushing it very close to 100%. All these results appear to be consistent across the three test sets.

The high value of compliance may seem unusual, but it can be easily explained. A logic clause $A \Rightarrow B$ is considered true when both A and B are true or when A is false. The high value obtained by the base method are partially due to the latter case. Indeed, Eqs. 2 and 3 have a LINK predicate in their left part, so for every pair of components the equation will result true for every case where they are not linked. Since in each test set the number of linked pairs amount to about 10%of the total, the lower bound in respecting the rules is around 90%. Finally, we remark that an improvement of 1-2 percentage points corresponds to 30-40 pairs of components, which we believe to be not negligible.

Discussion

We presented the first application of LTNs to a challenging NLP task, 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. For example, we could easily measure compliance.

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 did not negatively affect component classification performance and it increased link classification performance, while at the same time boosting robustness and compliance. It is worthwhile noticing that the networks used in our experiments are much simpler than state-of-the-art models, and obviously they do not achieve comparable performance, but we speculate that the impact of rules may hold even for more advanced models.

Dataset	Split	Approach	Classification Comp.	Classification Link	Agreement Comp.	Agreement Link	Properties Eq. 2	Properties Eq. 3
Neoplasm	Validation	Data	83 - 84	42 - 41	77	66	98	**100**
		Data + Rules	84 - **85**	**44** - 43	**81**	**71**	**100**	**100**
Neoplasm	Test	Data	79 - **80**	34 - 31	77	64	98	**100**
		Data + Rules	79 - 78	**35 - 35**	**79**	**70**	**100**	**100**
Glaucoma	Test	Data	**82 - 82**	45 - 43	**75**	66	99	**100**
		Data + Rules	81 - **82**	**47** - 45	**75**	**71**	**100**	**100**
Mixed	Test	Data	**81 - 81**	38 - 34	75	64	98	**100**
		Data + Rules	**81** - 80	39 - **40**	**76**	**69**	**100**	**100**

Table 2: Results of NeSy AM on AbstRCT against the data-driven baseline. For component classification, we report both the result obtained by the MAJ approach (before the dash) and by the AVG approach (after the dash). Scores are reported as percentage values.

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.

Once the scalability issue will be solved, we plan to experiment with larger corpora, more advanced embeddings, and deeper neural architectures. Moreover, it will be interesting to define rules that apply only to a subsets of entities. For example, in our benchmark, the argumentation graphs link only entities that belong to the same document, hence the collective classification may be performed document-wise, rather than dataset-wise, with the consequence that rules will be applied only between elements of the same document. Such a consequence may be a desired property or an unwanted drawback, according to the specific context. A collective classification on the whole corpus would be beneficial in applications where the argumentation spans across multiple domains and the aim is to find relations between components that belong to different documents. This approach suits contexts such as mining argumentation in social networks (Bosc, Cabrio, and Villata 2016) or retrieving arguments related to a specific topic (Ein-Dor et al. 2020).

Another direction regards the recognition of properties that are not explicit in the training data but can be defined through logical rules. In the context of our setting, it would be possible to define a predicate that represents a property without the need to provide any grounding example for it. This could be achieved by creating axioms that involve such a predicate and other grounded predicates, so as to train the neural network associated with the new predicate along with the ones for which the grounding is provided. 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

Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G. S.; Davis, A.; Dean, J.; Devin, M.; Ghemawat, S.; Goodfellow, I.; Harp, A.; Irving, G.; Isard, M.; Jia, Y.; Jozefowicz, R.; Kaiser, L.; Kudlur, M.; Levenberg, J.; Mané, D.; Monga, R.; Moore, S.; Murray, D.; Olah, C.; Schuster, M.; Shlens, J.; Steiner, B.; Sutskever, I.; Talwar, K.; Tucker, P.; Vanhoucke, V.; Vasudevan, V.; Viégas, F.; Vinyals, O.; Warden, P.; Wattenberg, M.; Wicke, M.; Yu, Y.; and Zheng, X. 2016. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. *CoRR* abs/1603.04467. URL http://arxiv.org/abs/1603.04467.

Badreddine, S.; d'Avila Garcez, A.; Serafini, L.; and Spranger, M. 2020. Logic Tensor Networks. *CoRR* abs/2012.13635. URL https://arxiv.org/abs/2012.13635.

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

Bosc, T.; Cabrio, E.; and Villata, S. 2016. Tweeties Squabbling: Positive and Negative Results in Applying Argument Mining on Social Media. In Baroni, P.; Gordon, T. F.; Scheffler, T.; and Stede, M., eds., *COMMA 2016*, volume 287 of *Frontiers in Artificial Intelligence and Applications*, 21–32. IOS Press. doi:10.3233/978-1-61499-686-6-21. URL https://doi.org/10.3233/978-1-61499-686-6-21.

Cabrio, E.; and Villata, S. 2018. Five Years of Argument Mining: a Data-driven Analysis. In *Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, IJCAI-18*, 5427–5433. International Joint Conferences on Artificial Intelligence Organization. doi:10.24963/ijcai.2018/766. URL https://doi.org/10.24963/ijcai.2018/766.

Cer, D.; Yang, Y.; Kong, S.-y.; Hua, N.; Limtiaco, N.; St. John, R.; Constant, N.; Guajardo-Cespedes, M.; Yuan, S.; Tar, C.; Strope, B.; and Kurzweil, R. 2018. Universal Sentence Encoder for English. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 169–174. Brussels, Belgium: Association for Computational Linguistics. doi:10.18653/v1/D18-2029. URL https://www.aclweb.org/anthology/D18-2029.

d'Avila Garcez, A. S.; Gori, M.; Lamb, L. C.; Serafini, L.; Spranger, M.; and Tran, S. N. 2019. Neural-symbolic Computing: An Effective Methodology for Principled Integration of Machine Learning and Reasoning. *FLAP* 6(4): 611–632. URL https://collegepublications.co.uk/ifcolog/?00033.

De Raedt, L.; Dumancic, S.; Manhaeve, R.; and Marra, G. 2020. From Statistical Relational to Neuro-Symbolic Artificial Intelligence. In Bessiere, C., ed., *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*, 4943–4950. ijcai.org. doi:10.24963/ijcai.2020/688. URL https://doi.org/10.24963/ijcai.2020/688.

Devlin, J.; Chang, M.; Lee, K.; and Toutanova, K. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *NAACL-HLT (1)*, 4171–4186. ACL.

Donadello, I.; Serafini, L.; and Garcez, A. D. 2017. Logic Tensor Networks for Semantic Image Interpretation. In *IJCAI*, 1596–1602. AAAI Press. ISBN 9780999241103. doi:10.24963/ijcai.2017/221. URL https://doi.org/10.24963/ijcai.2017/221.

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(2): 321–358. doi:10.1016/0004-3702(94)00041-X. URL http://www.sciencedirect.com/science/article/pii/000437029400041X.

Ein-Dor, L.; Shnarch, E.; Dankin, L.; Halfon, A.; Sznajder, B.; Gera, A.; Alzate, C.; Gleize, M.; Choshen, L.; Hou, Y.; Bilu, Y.; Aharonov, R.; and Slonim, N. 2020. Corpus Wide Argument Mining - A Working Solution. In *AAAI*, 7683–7691. AAAI Press. URL https://aaai.org/ojs/index.php/AAAI/article/view/6270.

Galassi, A.; Kersting, K.; Lippi, M.; Shao, X.; and Torroni, P. 2019. Neural-Symbolic Argumentation Mining: An Argument in Favor of Deep Learning and Reasoning. *Frontiers in Big Data* 2: 52. doi:10.3389/fdata.2019.00052. URL https://doi.org/10.3389/fdata.2019.00052.

Galassi, A.; Lippi, M.; and Torroni, P. 2021. Multi-Task Attentive Residual Networks for Argument Mining. *CoRR* abs/2102.12227. URL https://arxiv.org/abs/2102.12227.

Goodfellow, I.; Bengio, Y.; and Courville, A. 2016. *Deep Learning*. MIT Press. URL http://www.deeplearningbook.org.

Hu, Z.; Ma, X.; Liu, Z.; Hovy, E.; and Xing, E. 2016. Harnessing Deep Neural Networks with Logic Rules. In *ACL (1)*, 2410–2420. Berlin, Germany: Association for Computational Linguistics. doi:10.18653/v1/P16-1228. URL https://www.aclweb.org/anthology/P16-1228.

Lippi, M.; and Frasconi, P. 2009. Prediction of protein β-residue contacts by Markov logic networks with grounding-specific weights. *Bioinformatics* 25(18): 2326–2333. ISSN 1367-4803. doi:10.1093/bioinformatics/btp421. URL https://dx.doi.org/10.1093/bioinformatics/btp421.

Lippi, M.; and Torroni, P. 2016. Argumentation Mining: State of the Art and Emerging Trends. *ACM Trans. Internet Technol.* 16(2): 10:1–10:25. ISSN 1533-5399. doi:10.1145/2850417. URL http://doi.acm.org/10.1145/2850417.

Manhaeve, R.; Dumančić, S.; Kimmig, A.; Demeester, T.; and De Raedt, L. 2021. Neural probabilistic logic programming in DeepProbLog. *Artificial Intelligence* 298: 103504. ISSN 0004-3702. doi:https://doi.org/10.1016/j.artint.2021.103504. URL https://www.sciencedirect.com/science/article/pii/S0004370221000552.

Mayer, T.; Cabrio, E.; and Villata, S. 2020. Transformer-Based Argument Mining for Healthcare Applications. In Giacomo, G. D.; Catalá, A.; Dilkina, B.; Milano, M.; Barro, S.; Bugarín, A.; and Lang, J., eds., *ECAI 2020 - 24th European Conference on Artificial Intelligence*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, 2108–2115. IOS Press. doi:10.3233/FAIA200334. URL https://doi.org/10.3233/FAIA200334.

Mayer, T.; Marro, S.; Cabrio, E.; and Villata, S. 2021. Enhancing Evidence-Based Medicine with Natural Language Argumentative Analysis of Clinical Trials. *Artificial Intelligence in Medicine* 102098. ISSN 0933-3657. doi:https://doi.org/10.1016/j.artmed.2021.102098. URL https://www.sciencedirect.com/science/article/pii/S0933365721000919.

Pacheco, M. L.; and Goldwasser, D. 2021. Modeling Content and Context with Deep Relational Learning. *Trans. Assoc. Comput. Linguistics* 9: 100–119. URL https://transacl.org/ojs/index.php/tacl/article/view/2381.

Pennington, J.; Socher, R.; and Manning, C. 2014. Glove: Global Vectors for Word Representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543. Association for Computational Linguistics. doi:10.3115/v1/D14-1162. URL http://aclweb.org/anthology/D14-1162.

Reimers, N.; and Gurevych, I. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 3982–3992. Hong Kong, China: Association for Computational Linguistics. doi:10.18653/v1/D19-1410. URL https://www.aclweb.org/anthology/D19-1410.

Sen, P.; Namata, G.; Bilgic, M.; Getoor, L.; Gallagher, B.; and Eliassi-Rad, T. 2008. Collective Classification in Network Data. *AI Mag.* 29(3): 93–106. doi:10.1609/aimag.v29i3.2157. URL https://doi.org/10.1609/aimag.v29i3.2157.

Serafini, L.; and d'Avila Garcez, A. S. 2016. Learning and Reasoning with Logic Tensor Networks. In Adorni, G.; Cagnoni, S.; Gori, M.; and Maratea, M., eds., *AI*IA 2016 Advances in Artificial Intelligence*, 334–348. Cham: Springer International Publishing. ISBN 978-3-319-49130-1.

Socher, R.; Chen, D.; Manning, C. D.; and Ng, A. 2013. Reasoning With Neural Tensor Networks for Knowledge Base Completion. In Burges, C. J. C.; Bottou, L.; Welling, M.; Ghahramani, Z.; and Weinberger, K. Q., eds., *Advances in Neural Information Processing Systems*, volume 26, 926–934. Curran Associates, Inc. URL https://proceedings.neurips.cc/paper/2013/file/b337e84de8752b27eda3a12363109e80-Paper.pdf.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L. u.; and Polosukhin, I. 2017. Attention is All you Need. In *NIPS*, 5998–6008. Curran Associates, Inc.

Walton, D. 2009. Argumentation Theory: A Very Short Introduction. In Simari, G.; and Rahwan, I., eds., *Argumentation in Artificial Intelligence*, 1–22. Springer US. ISBN 978-0-387-98196-3. doi:10.1007/978-0-387-98197-0_1. URL http://link.springer.com/chapter/10.1007/978-0-387-98197-0_1.