Argumentative Link Prediction using Residual Networks and Multi-Objective Learning

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

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

Abstract

We explore the use of residual networks for argumentation mining, with an emphasis on link prediction. The method we propose makes no assumptions on document or argument structure. We evaluate it on a challenging dataset consisting of user-generated comments collected from an online platform. Results show that our model outperforms an equivalent deep network and offers results comparable with state-of-the-art methods that rely on domain knowledge.

1 Introduction

Argumentation mining is a growing sub-area of artificial intelligence and computational linguistics whose aim is to automatically extract arguments from generic textual corpora (Lippi and Torroni, 2016a). The problem is typically broken down into focused sub-problems such as the identification of sentences containing argument components like claims and premises, of the boundaries of argument components within such sentences, and the prediction of the argumentative structure of the document at hand.

In spite of significant results achieved in component identification tasks, such as claim/evidence detection (Rinott et al., 2015; Lippi and Torroni, 2015; Park and Cardie, 2014; Park et al., 2015b; Stab and Gurevych, 2014), classification (Eckle-Kohler et al., 2015; Niculae et al., 2017) and boundary detection (Sardianos et al., 2015; Levy et al., 2014; Lippi and Torroni, 2016b; Habernal and Gurevych, 2017), comparatively less progress has been made in the arguably more challenging argument structure prediction task (Cabrio and Villata, 2012; Stab and Gurevych, 2014).

Again due to the challenging nature of the general argumentation mining problem, solutions have typically addressed a specific genre or application domain, such as legal texts (Mochales Palau and Moens, 2011), persuasive essays (Stab and Gurevych, 2017), or Wikipedia articles (Levy et al., 2014; Rinott et al., 2015) and have heavily relied on domain knowledge. One particular aspect of the domain is the argument model. While argumentation as a discipline has developed rather sophisticated argument models, such as Toulmin's (1958), the majority of the available argumentation mining data sets refer to ad-hoc, usually simpler argument models, often in an effort to obtain a reasonable inter-annotator agreement. Another crucial aspect is the document structure. For instance, in some domains, certain argument components occupy a specific position in the document.

Moreover, until recently, approaches have mostly used traditional methods such as support vector machines, logistic regression and naive Bayes classifiers. Only in the last couple of years the field has started to look more systematically into neural network-based architectures, such as long short-memory networks and convolutional neural networks, and structured output classifiers.

The aim of this study is to investigate the application of residual networks–a deep neural network architecture not previously applied to this domain–to a challenging structure prediction task, namely link prediction. Our ambition is to define a model that does not exploit domain-specific, highly engineered features, or information on the underlying argument model, and could thus be, at least in principle, of general applicability. Our results match those of state-of-the-art methods that rely on domain knowledge, but use much less a-priori information.

The next section reviews recent applications of neural networks to argumentation mining. Section 3 presents our model, Section 4 the benchmark, and Section 5 discusses results. Section 6 concludes.

Proceedings of the 5th Workshop on Argument Mining, pages 1–10
Brussels, Belgium, November 1, 2018. ©2018 Association for Computational Linguistics

2 Related work

The application of neural network architectures in argumentation mining is relatively recent. A study most closely related to ours was presented by Niculae et al. (2017) and will be described in greater detail in Section 4. The authors propose a structured learning framework based on factor graphs. Their approach imposes constraints to the graph according to the underlying argument model, and it includes a joint optimization method based on the AD3 algorithm (Martins et al., 2015), structured Support Vector Machines (Tsochantaridis et al., 2005) and Recurrent Neural Networks (Rumelhart et al., 1986). Link prediction and argument component classification are performed jointly, reaching state-of-the-art results on two distinct corpora. In contrast to our method, Niculae et al.'s heavily relies on a-priori knowledge.

In the domain of persuasive essays, Eger et al. (2017) consider several sub-tasks of argumentation mining, making use of various neural architectures. These include neural parsers (Dyer et al., 2015; Kiperwasser and Goldberg, 2016), LSTMs for joint entity and relation extraction (LSTM-ER) (Miwa and Bansal, 2016), and Bidirectional LSTM coupled with Conditional Random Fields and Convolutional Neural Networks (BLCC) (Ma and Hovy, 2016) in a multi-task learning framework (Søgaard and Goldberg, 2016). Eger et al. conclude that neural networks can outperform feature-based techniques in argumentation mining tasks.

Convolutional Neural Networks and LSTMs have been used by Guggilla et al. (2016) to perform claim classification, whereas bidirectional LSTMs have been exploited by Habernal and Gurevych (2016) to assess the persuasiveness of arguments. More recently, neural networks have been applied to the task of topic-dependent evidence detection (Shnarch et al., 2018), improving the performance on a manually labelled corpus through the use of unsupervised data. Potash et al. (2017) have applied Pointer Networks (Vinyals et al., 2015) to argumentation mining.

Looking beyond argumentation mining, Lei et al. (2018) reviews the application of several deep learning techniques for sentiment analysis, while Conneau et al. (2017) for the first time applies very deep residual networks to NLP-related task and successfully performs text classification at the character level. Small residual convolutional networks have been successfully applied by Zhang et al. (2018) to multi-label classification on medical notes and by Huang and Wang (2017) to distantly-supervised relation extraction, where a knowledge base is used to generate a noisy set of positive relations among unlabeled data.

3 Residual networks for argument mining

Residual networks (He et al., 2016a,b) are a recent family of deep neural networks that achieved outstanding results in many machine learning tasks, in particular in computer vision applications such as medical imaging (Yu et al., 2017), computational linguistics (Bjerva et al., 2016), crowd flow prediction (Zhang et al., 2017), and game playing (Cazenave, 2018; Chesani et al., 2018).

The core idea behind residual networks, illustrated by Figure 1, is to create shortcuts that link neurons belonging to distant layers, whereas standard feed-forward networks typically link neurons belonging to subsequent layers only. This kind of architecture usually results in a speedier training phase, and it usually allows to train networks with a very large number of layers. The original architecture exploits convolutional layers, but it can be generalized to dense (fully-connected) layers. The motivation behind residual networks is that if multiple non-linear layers can asymptotically approximate a complex function $H(x)$, they can also asymptotically approximate its residual function $F(x) = H(x) - x$. The original function is therefore obtained by simply adding back the residual value: $H(x) = F(x) + x$.

The architecture we propose in this paper makes use of the dense residual network model, along with an LSTM (Hochreiter and Schmidhuber, 1997), to jointly perform link prediction and argument component classification. More specifically, our approach works at a local level on pairs of sentences, without any document-level global optimization, and without imposing model constraints induced, e.g., by domain-specific or genre-specific hypotheses. For that reason, it lends itself to integration with more complex systems.

3.1 Model description

One of our aims is to propose a method that abstracts away from a specific argument model. We thus reason in terms of abstract entities, such as

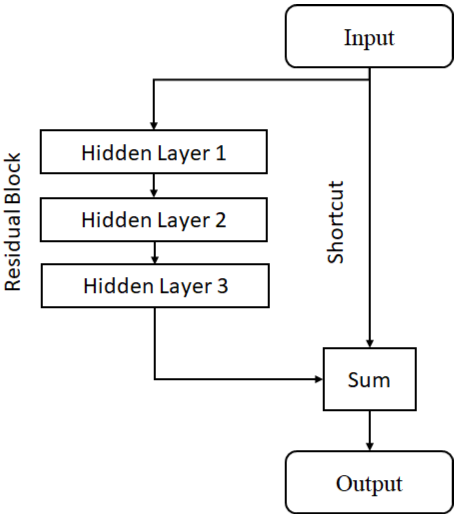


Figure 1: General schema of a residual network with a single residual block with three hidden layers.

argumentative propositions and the links among them. Such abstract entities are instantiated into concrete categories, such as claims and premises, supports or attacks, as soon as we apply the method to a domain described by a specific dataset whose annotations follow a concrete argument model. In particular, in this work we instantiate our model with the categories proposed by Niculae et al. (2017) for the annotation of the CDCP corpus.

In general, a document D is a sequence of *tokens*, i.e., words and punctuation marks. An argumentative proposition a is a sequence of contiguous tokens within D, which represents an argument, or part thereof. A labeling of propositions is induced by the chosen argument model. Such a labeling associates each proposition with the corresponding category of the argument component it contains.

Given two propositions a and b belonging to the same document, a directed relation from the former (*source*) to the latter (*target*) is represented as $a \to b$. Reflexive relations ($a \to a$) are not allowed.

Each relation $a \to b$ is characterized by two labels: a (Boolean) *link label*, $L_{a \to b}$, and a *relation label*, $R_{a \to b}$. The link label indicates the presence of a link, and is therefore *true* if there exists a directed link from a to b, and *false* otherwise. The relation label instead contains information on the nature of the link connecting a and b. In particular, it represents the direct or inverse relation between the two propositions, according to the links that connect a to b or b to a. In other words, its domain is composed, according to the underlying argument model, not only by all the possible link types (e.g., *attack* and *support*), but also by their opposite types (e.g., *attackedBy* and *supportedBy*) as well as by a category, *none*, meaning absence of link in either direction.[1]

One objective is to establish the value of the link label $L_{a \to b}$ for each possible input pair of propositions (a, b) belonging to the same document D. Such a *link prediction* task can be considered as a sub-task of argument structure prediction. Another objective is the *classification* of propositions and relations, i.e., the prediction of labels P_a, P_b, $R_{a \to b}$. That is also jointly performed, as in (Niculae et al., 2017). Notice, however, that Niculae et al. do not predict $R_{a \to b}$ relations, but only link and proposition labels.

3.2 Embeddings and features

Since the purpose of this work is to evaluate deep residual networks as an instrument for argumentation mining, without resorting to domain- or genre-specific information, the system relies on a minimal set of features that do not require elaborate processing.

Any input token is transformed into a 300-dimensional embedding by exploiting the GloVe pre-trained vocabulary (Pennington et al., 2014). Input sequences are zero-padded to the length of the longest sequence (153 tokens). The distance between two propositions could also be relevant to establishing whether two components are linked. We thus employed the number of propositions that separate two given propositions as an additional feature. Following previous works in the game domain, where scalar values have been encoded in binary form (Silver et al., 2016; Cazenave, 2018; Chesani et al., 2018), we represented distance using as a 10-bit array, where the first 5 bits are used in case that the source precedes the target, and the last 5 bits are used in the opposite case. In both cases, the number of consecutive "1" values encodes the value of the distance (distances are capped by 5). For example, if the target precedes the source by two sentences, the distance is -2, which produces encoding 0001100000; if the source precedes the target by three sentences, the distance is 3, with encoding 0000011100. In this

[1] Given the *none* category, label $L_{a \to b}$ could, in principle, be induced by label $R_{a \to b}$, but it is still convenient to keep both during the optimization process.

way, the Hamming distance between two distance value encodings is equal to the difference between the two distance values.

3.3 Residual Network Architecture

The network architecture is illustrated in Figure 2. It is composed by the following macro blocks:

- two deep embedders, one for sources and one for targets, that manipulate token embeddings;
- a dense encoding layer for feature dimensionality reduction;
- an LSTM to process the input sequences;
- a residual network;
- the final-stage classifiers.

Source and target propositions are encoded separately by the first three blocks, then they are concatenated together, along with the distance, and given as input to the residual network.

The deep embedders refine the token embeddings, thus creating new, more data-specific embeddings. Relying on deep embedders instead of on pre-trained autoencoders, aims to achieve a better generality, at least in principle, and avoid excessive specialization, thus limiting overfitting. The dimensionality reduction operated by the dense encoding layer allows to use an LSTM with fewer parameters, which has two positive effects: it reduces the time needed for training, and again it limits overfitting.

The deep embedders are residual networks composed by a single residual block, composed by 4 pre-activated time-distributed dense layers. Accordingly, each layer applies the same transformation to each embedding, regardless of their position inside the sentence. All the layers have 50 neurons, except the last one, which has 300 neurons.

The dense encoding layer reduces the size of the embedding sequences by applying a time-distributed dense layer, which reduces the embedding size to 50, and a time average-pooling layer (Collobert et al., 2011), which reduces the sequence size to $1/10$ of the original. The resulting sequences are then given as input to a single bidirectional LSTM, producing a representation of the proposition of size 50. Thus, for each proposition, 153 embeddings of size 300 are transformed first into 153 embeddings of size 50, then into 15 embeddings of size 50, and finally in a single feature of size 50.

Source and target features, computed this way, alongside with the distance encoding, are then concatenated together and given as input to the residual network. The first level of the network is a dense encoding layer with 20 neurons, while the residual block is composed by a layer with 5 neurons and one with 20 neurons. The sums of the results of the first and the last layers of the residual networks are provided as input to the classifiers.

The final layers of the system are three independent softmax classifiers used to predict the source, the target, and the relation labels. The output of each classifier is a probability distribution along all the possible classes of that label. The predicted class is the one with the highest score. All these three classifiers, which predict labels for two different tasks, contribute simultaneously to our learning model. The link classifier is obtained by summing the relevant scores produced by the relation classifier.[2]

All the dense layers use the rectifier activation function (Glorot et al., 2011), and they randomly initialize weights with He initialization (He et al., 2015). The application of all non-linearity functions is preceded by batch-normalization layers (Ioffe and Szegedy, 2015) and by dropout layers (Srivastava et al., 2014), with probability $p = 0.1$.

4 Benchmark

4.1 Dataset

We evaluated our model against the Cornell eRulemaking Corpus (CDCP) (Niculae et al., 2017). This consists of 731 user comments from a eRulemaking website, for a total of about 4,700 propositions, all considered to be argumentative.[3] The argument model adopted is the one proposed by Park et al. (2015a), where links are constrained to form directed graphs. Propositions are divided into 5 classes: POLICY (17%), VALUE (45%), FACT

[2] For instance, if our model considers *attack* and *support* relations as the only possible links, and the relation classifier scores are *attack* $= 0.15$, *support* $= 0.2$, *attackedBy* $= 0.1$, *supportedBy* $= 0.05$, *none* $= 0.5$, then the link classifier scores are: *true* $= 0.35$, *false* $= 0.65$.

[3] In an effort to obtain comparable results, we applied same preprocessing steps described in (Niculae et al., 2017), enforcing transitive closure and removing nested proposition, even though our approach does not take into account the argumentation model, nor its properties.

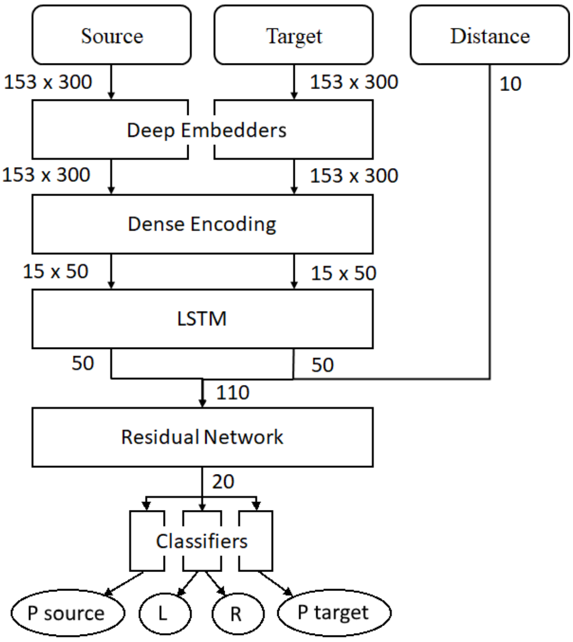


Figure 2: A block diagram of the proposed architecture. The figure shows, next to each arrow, the dimensionality of the data involved, so as to clarify the size of the inputs and the outputs of each block.

(16%), TESTIMONY (21%) and REFERENCE (1%). Links are divided between REASON (97%) and EVIDENCE (3%). Figure 3 shows an annotated document from the CDCP corpus.

Link prediction is a particularly difficult task in the CDCP dataset, where only 3% of all the possible proposition pairs (more than 43,000) are linked. A preliminary analysis of the data suggests that the number of propositions separating source and target (*distance*) could be a relevant feature, since most linked propositions are not far from each other. Indeed, as Figure 4 shows, around 70% of links are between adjacent propositions.

We tokenized documents using a hand-crafted parser based on the progressive splitting of the tokens and search within the GloVe vocabulary. We preferred not to use existing tools because of the nature of the data, since the CDCP documents often do not follow proper writing conventions (such as the blank space after the period mark), leading in some cases to a wrong tokenization. As a result, the number of tokens not contained in the GloVe dictionary dramatically reduced from 384, originally obtained with the software provided by Niculae et al. (2017), to 84. Each of these tokens was mapped into a randomly-generated embedding.

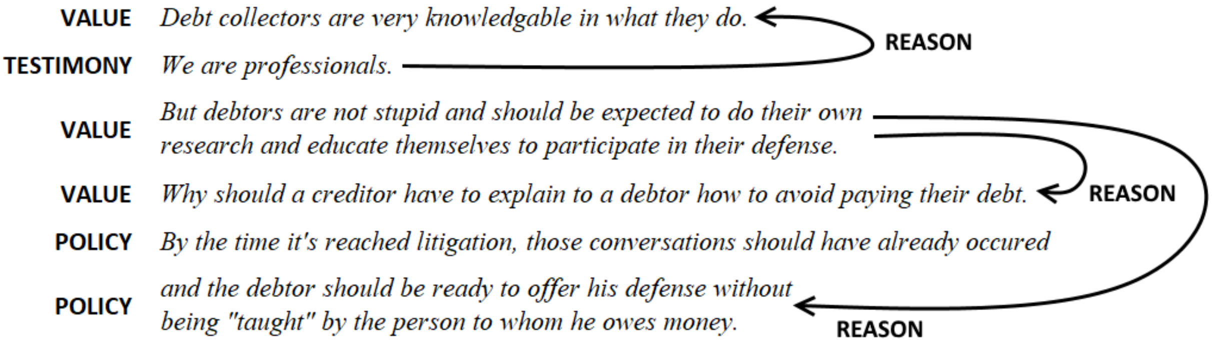
Table 1: Experimental dataset composition.

Split	Train	Valid.	Test	Total
Documents	513	68	150	731
Propositions	3,338	468	973	4,779
Values	1438	231	491	2160
Policies	585	77	153	815
Testimonies	738	84	204	1026
Facts	549	73	124	746
References	28	3	1	32
Couples	30,056	3,844	9,484	43,384
Links	923	143	272	1,338
Reasons	888	139	265	1292
Evidences	35	4	7	46

4.2 Structured Learning

The state of the art for the CDCP corpus is the work described by the corpus authors themselves (Niculae et al., 2017). They use a structured learning framework to jointly classify all the propositions in a document and determine which ones are linked together. To perform the classification, the models can rely on many factors and constraints. The unary factors represent the model's belief in each possible class for each proposition or link, without considering any other proposition or link. For each link between two propositions, the compatibility factors influence link classification according to the proposition classes, taking into account adjacency between propositions and precedence between source and target. The second-order factors influence the classification of pairs of links that share a common proposition, by modeling three local argumentation graph structures: grandparent, sibling and co-parent. Furthermore, constraints are introduced to enforce adherence to the desired argumentation structure, according to the argument model and domain characteristics.

The authors discuss experiments with 6 different models, which differ by complexity (the type of factors and constraints involved) and by how they model the factors (SVMs and RNNs). The RNN models compute sentence embeddings, by exploiting initialization with GloVe word vectors, while the SVMs models rely on many specific features. The first-order factors rely on the same features used by Stab and Gurevych (2017), both for the propositions and the links. These are, among the others, unigrams, dependency tuples, token statistics, proposition statistics, propo-



Debt collectors are very knowledgable in what they do.

We are professionals.

But debtors are not stupid and should be expected to do their own research and educate themselves to participate in their defense.

Why should a creditor have to explain to a debtor how to avoid paying their debt.

By the time it's reached litigation, those conversations should have already occured

and the debtor should be ready to offer his defense without being "taught" by the person to whom he owes money.

Figure 3: Argumentation structure in one of the documents of the CDCP corpus.

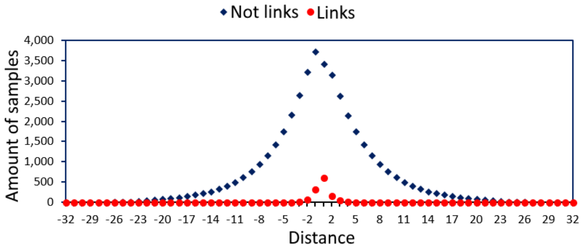


Figure 4: Link distribution in the CDCP dataset with respect to distance. The distance is considered positive when the source precedes the target, negative otherwise.

sition location, indicators from hand-crafted lexicons and handcrafted ones, shared phrases, subclauses, depth of the parse tree, tense of the main verb, modal verbs, POS, production rules, type probability, discourse triplets (Lin et al., 2014), and average GloVe embeddings. The higher-order factors exploit the following features between all three propositions and between each pair: same sentence indicators, proposition order, Jaccard similarity, presence of any shared nouns, and shared noun ratios. The overall feature dimensionality is reportedly 7000 for propositions and 2100 for links, not counting 35 second-order features.

5 Results

5.1 Experimental setting

We created a validation set by randomly selecting documents from the original training split with 10% probability. We used the remaining documents as training data and the original test split as is. Table 1 reports the statistics related to the three splits.

We defined the learning problem as a multi-objective optimization problem, whose loss function is given by the weighted sum of four different components: the categorical cross-entropy on three labels (source and target categories, link relation category) and an L_2 regularization on the network parameters. The weights of these components were, respectively, 1, 1, 10, 10^{-4}.

We performed mini-batch optimization using Adam (Kingma and Ba, 2014) with parameters $b_1 = 0.9$ and $b_2 = 0.9999$, and by applying proportional decay of the initial learning rate $\alpha_0 = 5 \times 10^{-3}$. Training was early-stopped after 200 epochs with no improvements on the validation data. We chose the numerous hyper-parameters of the architecture and of the learning model after an initial experimental setup phase, based on the performance on the validation set for the link prediction task. Results obtained in this phase confirmed that the presence of the deep embedder block and of the distance feature lead to better results.

We compared the results of the residual network model against an equivalent deep network with the same number of layers and the same hyper-parameters, but without the shortcut that characterize the residual network block. We applied two different training procedures for both this deep network baseline and the residual network. In particular, as the criterion for early stopping we used once the error on link prediction and once the error on proposition classification. In the presentation of our results we will refer to these two models as link-guided (LG) and proposition-guided (PG).

Following (Niculae et al., 2017), we measured the performance of the models by computing the F_1 score for links, propositions, and the average between the two, in order to provide a summary evaluation. More specifically, for the links we measured the F_1 of the positive classes (as the harmonic mean between precision and recall), whereas for the propositions we used the score of each class and then we computed the macro-average. We also reported the F_1 score for each direct relation class, alongside with their macro-average.

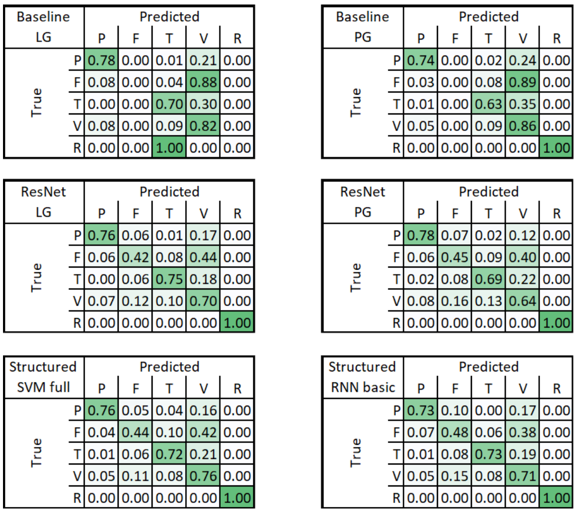
Since each proposition is involved in many pairs, both as a source and as a target, its classification is performed multiple times. To classify it uniquely, we considered the average probability score assigned to each class and we have assigned the most probable class. That is of course not the only option. Another possibility could be to assign the class that results to be the most probable in most of the cases, thus relying on a majority vote. A further option could be to simply consider the label with highest confidence. However, this procedure might be more sensitive to outliers, because the misclassification of a sentence in just one pair would lead to the misclassification of the sentence, regardless of all the other pairs. A deeper analysis of different techniques to address this issues is left to future research.

5.2 Discussion and analysis

Table 2 summarizes the evaluation of baselines and residual networks,[4] also showing the best scores obtained by the structured learning configurations presented in (Niculae et al., 2017).

Results highlight how the proposed approach based on residual networks outperforms the state of the art for what concerns link prediction. In addition, residual link-guided network training consistently performs better than both deep networks baselines in all the three tasks.

As for proposition label prediction, the results obtained through structured approaches still maintain a slight advantage over residual networks. This could be partially explained by the fact that hyper-parameter tuning was done with the aim to select the best model for link prediction. It should also be considered that we perform proposition classification relying on the merging of labels obtained through local optimization, while the structured learning approach exploits a global optimization. Nonetheless, the average score of residual networks is better than that of structured RNNs, thus proving the generality of the approach.



Baseline LG		Predicted				
		P	F	T	V	R
True	P	0.78	0.00	0.01	0.21	0.00
	F	0.08	0.00	0.04	0.88	0.00
	T	0.00	0.00	0.70	0.30	0.00
	V	0.08	0.00	0.09	0.82	0.00
	R	0.00	0.00	1.00	0.00	0.00

Baseline PG		Predicted				
		P	F	T	V	R
True	P	0.74	0.00	0.02	0.24	0.00
	F	0.03	0.00	0.08	0.89	0.00
	T	0.01	0.00	0.63	0.35	0.00
	V	0.05	0.00	0.09	0.86	0.00
	R	0.00	0.00	0.00	0.00	1.00

ResNet LG		Predicted				
		P	F	T	V	R
True	P	0.76	0.06	0.01	0.17	0.00
	F	0.06	0.42	0.08	0.44	0.00
	T	0.00	0.06	0.75	0.18	0.00
	V	0.07	0.12	0.10	0.70	0.00
	R	0.00	0.00	0.00	0.00	1.00

ResNet PG		Predicted				
		P	F	T	V	R
True	P	0.78	0.07	0.02	0.12	0.00
	F	0.06	0.45	0.09	0.40	0.00
	T	0.02	0.08	0.69	0.22	0.00
	V	0.08	0.16	0.13	0.64	0.00
	R	0.00	0.00	0.00	0.00	1.00

Structured SVM full		Predicted				
		P	F	T	V	R
True	P	0.76	0.05	0.04	0.16	0.00
	F	0.04	0.44	0.10	0.42	0.00
	T	0.01	0.06	0.72	0.21	0.00
	V	0.05	0.11	0.08	0.76	0.00
	R	0.00	0.00	0.00	0.00	1.00

Structured RNN basic		Predicted				
		P	F	T	V	R
True	P	0.73	0.10	0.00	0.17	0.00
	F	0.07	0.48	0.06	0.38	0.00
	T	0.01	0.08	0.73	0.19	0.00
	V	0.05	0.15	0.08	0.71	0.00
	R	0.00	0.00	0.00	0.00	1.00

Figure 5: Confusion matrix for proposition prediction. Top: baseline networks; middle: residual networks; bottom: structured prediction by (Niculae et al., 2017).

We shall also remark that our approach can achieve such results without exploiting any specific hypothesis or a-priori knowledge of the genre or domain. This could be an added value in contexts where arguments may be laid out freely, without following a pre-determined argument model, yet it would be interesting to uncover the underlying argumentation's structure.

Results also indicate that the most common mistake regards the prediction of facts as values (see Figure 5). That should come as no surprise, since VALUE is by far the largest class in the corpus, and it is therefore also affected by many false positives. Interestingly, baselines completely avoid to classify any proposition as a FACT.

As far as relation label prediction is concerned, this model apparently fails to predict the EVIDENCE relation. That negative result was also to be expected, since such a class is scarcely present in the whole dataset (less than 1%).

6 Conclusion and future work

We presented the first application of residual networks in the argumentation mining domain. We proposed a model that outperforms an equivalent deep network and competes with state-of-the-art techniques in a challenging dataset.

Considering that the model makes use of only one simple feature – the argumentative distance between two proposition – a natural extension of

[4] We report the results obtained on just one trained model. As explained in (Reimers and Gurevych, 2017), due to the non-deterministic behavior of the neural networks, this scores are influenced by the random seed of the training. Evaluating the same model trained many times with different seeds, and reporting the average scores would clearly yield a more robust evaluation.

Table 2: F_1 scores computed on the test set. For each class, the number of instances is reported in parenthesis. For the comparison with structured learning, the best scores obtained by any of the structured configurations are reported.

	Deep Baseline		Deep Residual		Structured	
Metric	LG	PG	LG	PG	SVM	RNN
Average (Link and Proposition)	33.18	42.88	47.28	46.37	**50.0**	43.5
Link (272)	22.56	22.45	**29.29**	20.76	26.7	14.6
Proposition (973)	43.79	63.31	65.28	71.99	**73.5**	72.7
VALUE (491)	73.77	74.45	72.19	73.24	76.4	73.7
POLICY (153)	73.85	76.09	74.36	76.43	77.3	76.8
TESTIMONY (204)	71.36	65.98	72.86	68.63	71.7	75.8
FACT (124)	0	0	40.31	41.64	42.5	42.2
REFERENCE (1)	0	100	66.67	100	100	100
Relation (272)	11.68	11.52	**15.01**	10.31		
REASON (265)	23.35	23.04	30.02	20.62		
EVIDENCE (7)	0	0	0	0		

this study would be its integration in a more structured and constrained argumentation framework.

Since in argumentation it is often the case that single propositions cannot contain all the relevant information to predict argument components and relations, it could be useful to provide also the context of argumentation as an input. Hence, another interesting direction of investigation could be the integration of the whole document text in the model.

References

Johannes Bjerva, Barbara Plank, and Johan Bos. 2016. Semantic tagging with deep residual networks. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3531–3541.

Elena Cabrio and Serena Villata. 2012. Combining textual entailment and argumentation theory for supporting online debates interactions. In *Proceedings of the 50th annual meeting of the Association for Computational Linguistics (ACL 2012)*, pages 208–212, Jeju, Korea. Association for Computational Linguistics.

T. Cazenave. 2018. Residual networks for computer go. *IEEE Transactions on Games*, 10(1):107–110.

F. Chesani, A. Galassi, M. Lippi, and P. Mello. 2018. Can deep networks learn to play by the rules? a case study on nine men's morris. *IEEE Transactions on Games*, pages 1–1.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2017. Very deep convolutional networks for text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, volume 1, pages 1107–1116.

Chris Dyer, Miguel Ballesteros, Wang Ling, Austin Matthews, and Noah A Smith. 2015. Transition-based dependency parsing with stack long short-term memory. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, volume 1, pages 334–343.

Judith Eckle-Kohler, Roland Kluge, and Iryna Gurevych. 2015. On the role of discourse markers for discriminating claims and premises in argumentative discourse. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, page to appear, Lisbon, Portugal. Association for Computational Linguistics.

Steffen Eger, Johannes Daxenberger, and Iryna Gurevych. 2017. Neural end-to-end learning for computational argumentation mining. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 11–22.

Xavier Glorot, Antoine Bordes, and Yoshua Bengio. 2011. Deep sparse rectifier neural networks. In *Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics*, volume 15 of *Proceedings of Machine Learning Research*, pages 315–323, Fort Lauderdale, FL, USA. PMLR.

Chinnappa Guggilla, Tristan Miller, and Iryna Gurevych. 2016. CNN-and LSTM-based claim classification in online user comments. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 2740–2751.

Ivan Habernal and Iryna Gurevych. 2016. Which argument is more convincing? analyzing and predicting convincingness of web arguments using bidirectional lstm. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1589–1599.

Ivan Habernal and Iryna Gurevych. 2017. Argumentation mining in user-generated web discourse. *Computational Linguistics*, 43(1):125–179.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pages 1026–1034.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016a. Deep residual learning for image recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 770–778.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2016b. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, pages 630–645. Springer.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*, 9(8):1735–1780.

YiYao Huang and William Yang Wang. 2017. Deep residual learning for weakly-supervised relation extraction. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1803–1807.

Sergey Ioffe and Christian Szegedy. 2015. Batch normalization: Accelerating deep network training by reducing internal covariate shift. In *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 448–456, Lille, France. PMLR.

Diederik Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Eliyahu Kiperwasser and Yoav Goldberg. 2016. Simple and accurate dependency parsing using bidirectional LSTM feature representations. *Transactions of the Association for Computational Linguistics*, 4:313–327.

Zhang Lei, Wang Shuai, and Liu Bing. 2018. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 0(0):e1253.

Ran Levy, Yonatan Bilu, Daniel Hershcovich, Ehud Aharoni, and Noam Slonim. 2014. Context dependent claim detection. In *COLING 2014, Dublin, Ireland*, pages 1489–1500. ACL.

Ziheng Lin, Hwee Tou Ng, and Min-Yen Kan. 2014. A pdtb-styled end-to-end discourse parser. *Natural Language Engineering*, 20(2):151–184.

Marco Lippi and Paolo Torroni. 2015. Context-independent claim detection for argument mining. In *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, IJCAI 2015, Buenos Aires, Argentina, July 25-31, 2015*, pages 185–191. AAAI Press.

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

Marco Lippi and Paolo Torroni. 2016b. Margot. *Expert Syst. Appl.*, 65(C):292–303.

Xuezhe Ma and Eduard Hovy. 2016. End-to-end sequence labeling via bi-directional LSTM-CNNs-CRF. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 1064–1074.

André FT Martins, Mário AT Figueiredo, Pedro MQ Aguiar, Noah A Smith, and Eric P Xing. 2015. Ad 3: Alternating directions dual decomposition for map inference in graphical models. *The Journal of Machine Learning Research*, 16(1):495–545.

Makoto Miwa and Mohit Bansal. 2016. End-to-end relation extraction using LSTMs on sequences and tree structures. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Raquel Mochales Palau and Marie-Francine Moens. 2011. Argumentation mining. *Artificial Intelligence and Law*, 19(1):1–22.

Vlad Niculae, Joonsuk Park, and Claire Cardie. 2017. Argument mining with structured SVMs and RNNs. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics, ACL 2017, Vancouver, Canada, July 30 - August 4, Volume 1: Long Papers*, pages 985–995. Association for Computational Linguistics.

Joonsuk Park, Cheryl Blake, and Claire Cardie. 2015a. Toward machine-assisted participation in eRulemaking: An argumentation model of evaluability. In *Proceedings of the 15th International Conference on Artificial Intelligence and Law*, pages 206–210. ACM.

Joonsuk Park and Claire Cardie. 2014. Identifying appropriate support for propositions in online user comments. In *Proceedings of the First Workshop on Argumentation Mining*, pages 29–38, Baltimore, Maryland. Association for Computational Linguistics.

Joonsuk Park, Arzoo Katiyar, and Bishan Yang. 2015b. Conditional random fields for identifying appropriate types of support for propositions in online user comments. In *Proceedings of the Second Workshop on Argumentation Mining*. Association for Computational Linguistics.

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.

Peter Potash, Alexey Romanov, and Anna Rumshisky. 2017. Here's my point: Joint pointer architecture for argument mining. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1364–1373.

Nils Reimers and Iryna Gurevych. 2017. Reporting score distributions makes a difference: Performance study of lstm-networks for sequence tagging. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 338–348.

Ruty Rinott, Lena Dankin, Carlos Alzate Perez, Mitesh M. Khapra, Ehud Aharoni, and Noam Slonim. 2015. Show me your evidence - an automatic method for context dependent evidence detection. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing, EMNLP 2015, Lisbon, Portugal, September 17-21, 2015*, pages 440–450. The Association for Computational Linguistics.

David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. 1986. Learning representations by back-propagating errors. *Nature*, 323(6088):533.

Christos Sardianos, Ioannis Manousos Katakis, Georgios Petasis, and Vangelis Karkaletsis. 2015. Argument extraction from news. pages 56–66.

Eyal Shnarch, Carlos Alzate, Lena Dankin, Martin Gleize, Yufang Hou, Leshem Choshen, Ranit Aharonov, and Noam Slonim. 2018. Will it blend? blending weak and strong labeled data in a neural network for argumentation mining. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 599–605.

David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al. 2016. Mastering the game of go with deep neural networks and tree search. *nature*, 529(7587):484.

Anders Søgaard and Yoav Goldberg. 2016. Deep multi-task learning with low level tasks supervised at lower layers. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 231–235.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(1):1929–1958.

Christian Stab and Iryna Gurevych. 2014. Identifying argumentative discourse structures in persuasive essays. In *EMNLP 2014, Doha, Qatar*, pages 46–56. ACL.

Christian Stab and Iryna Gurevych. 2017. Parsing argumentation structures in persuasive essays. *Computational Linguistics*, 43(3):619–659.

Stephen Edelston Toulmin. 1958. *The Uses of Argument*. Cambridge University Press.

Ioannis Tsochantaridis, Thorsten Joachims, Thomas Hofmann, and Yasemin Altun. 2005. Large margin methods for structured and interdependent output variables. *Journal of machine learning research*, 6(Sep):1453–1484.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*, pages 2692–2700.

L. Yu, H. Chen, Q. Dou, J. Qin, and P. A. Heng. 2017. Automated melanoma recognition in dermoscopy images via very deep residual networks. *IEEE Transactions on Medical Imaging*, 36(4):994–1004.

Junbo Zhang, Yu Zheng, and Dekang Qi. 2017. Deep spatio-temporal residual networks for citywide crowd flows prediction. In *AAAI*, pages 1655–1661.

Xinyuan Zhang, Ricardo Henao, Zhe Gan, Yitong Li, and Lawrence Carin. 2018. Multi-label learning from medical plain text with convolutional residual models. *arXiv preprint arXiv:1801.05062*.