Randomized Deep Structured Prediction for Discourse-Level Processing

Manuel Widmoser[*1] **Maria Leonor Pacheco**[*2] **Jean Honorio**[2] **Dan Goldwasser**[2]
[1]University of Salzburg, Austria
[2]Purdue University, USA
mwidmoser@cs.uni-salzburg.at,
{pachecog, jhonorio, dgoldwas}@purdue.edu

Abstract

Expressive text encoders such as RNNs and Transformer Networks have been at the center of NLP models in recent work. Most of the effort has focused on sentence-level tasks, capturing the dependencies between words in a single sentence, or pairs of sentences. However, certain tasks, such as argumentation mining, require accounting for longer texts and complicated structural dependencies between them. Deep structured prediction is a general framework to combine the complementary strengths of expressive neural encoders and structured inference for highly structured domains. Nevertheless, when the need arises to go beyond sentences, most work relies on combining the output scores of independently trained classifiers. One of the main reasons for this is that constrained inference comes at a high computational cost. In this paper, we explore the use of randomized inference to alleviate this concern and show that we can efficiently leverage deep structured prediction and expressive neural encoders for a set of tasks involving complicated argumentative structures.

1 Introduction

Many discourse-level NLP tasks require modeling complex interactions between multiple sentences, paragraphs or even documents. For example, analyzing opinions in online conversations (Hasan and Ng, 2013; Sridhar et al., 2015) requires modeling the dependencies between the opinions in individual posts, the disagreement between posts in long conversational threads and the overall view of users, given all their posts.

Learning in these settings is extremely challenging. It requires highly expressive models that can capture the claims made in each document, either by using a rich, manually crafted feature set, or by using neural architectures to learn an expressive representation (Ji and Eisenstein, 2014; Niculae et al., 2017). In addition, reasoning about the interaction between these decisions is often computationally challenging, as it requires incorporating domain-specific constraints into the search procedure, making exact inference intractable. As a result, most current work relies on highly engineered solutions, which are difficult to adapt. Instead of training structured predictors that model the interaction between decisions during training, they combine locally trained classifiers at test time (Stab and Gurevych, 2017).

Our goal in this paper is to study realistic settings, in which discourse-level problems can be learned efficiently when leveraging *deep structured prediction*, a framework for combining rich neural representation with an inference-layer, forcing consistency between them (Zheng et al., 2015). These models were applied successfully to simpler NLP tagging tasks (Lample et al., 2016), in which inference is tractable. However, as shown in a recent argumentation mining work (Niculae et al., 2017), their applicability to more complex learning tasks is not guaranteed.

Randomized inference algorithms have been proposed for structured NLP tasks, such as tagging and dependency parsing, in the context of linear models (Zhang et al., 2014, 2015; Ma et al., 2019). This approach offers an efficient alternative to exact inference. Instead of finding the optimal output state, the algorithm makes greedy updates to a randomly initialized (or locally initialized) output assignment state. Our main contribution is to explore these ideas in the context of deep structured models composed of expressive text encoders, where theoretical guarantees are weak or nonexistent. Moreover, we do this for discourse-level tasks involving a rich set of domain constraints. To do this, we consider two variations of this approach. In the first, the

[*] Contributed equally to this work as first authors

Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics, pages 1174–1184
April 19 - 23, 2021. ©2021 Association for Computational Linguistics

algorithm samples and traverses only legal states (i.e., consistent with the constraints imposed by domain knowledge). In the second, these restrictions are ignored and only applied at test time. Adapting the sampling procedure to the specific constraints imposed by each domain is difficult, motivating the second approach as a generic alternative.

We focus on two discourse-level tasks, stance prediction in discussion forums, described above, and parsing argumentation structures in essays (Stab and Gurevych, 2017). The latter consists of constructing an argumentation tree that represents the type-of, and relation-between, the arguments made in the essay. Models for both tasks typically employ declarative inference for incorporating domain knowledge. Our experiments are designed to quantify the trade-off between different modeling choices, both in terms of task performance and computational cost. We compare exact ILP models, approximate inference based on the popular AD^3 algorithm (Martins et al., 2015) and the two randomized inference algorithms. Our experiments show that in all cases, deep structured prediction outperforms traditional shallow approaches, structured learning outperforms inference over locally trained models, and generic randomized inference performs competitively to exact inference.

2 Related Work

Using deep structured prediction for NLP has been studied in previous work, typically on traditional sentence-level tasks such as dependency parsing (Chen and Manning, 2014; Weiss et al., 2015), transition systems (Andor et al., 2016), named entity recognition (Lample et al., 2016) and sequence labeling systems (Ma and Hovy, 2016). In most of these cases, inference is tractable. More recently, some efforts have started to look at incorporating deep structured prediction to discourse tasks such as argument mining (Niculae et al., 2017), event and temporal relation extraction (Han et al., 2019) and discourse representation parsing (Liu et al., 2019). In all of these cases, constrained inference is formulated as an integer linear program and solved either using off-the-shelf optimizers or approximation algorithms like AD^3 (Martins et al., 2015).

Randomized approximation has been introduced as an alternative to exact inference. Zhang et al. (2014) suggest a simple randomized greedy inference algorithm and empirically demonstrate its effectiveness for dependency parsing and other traditional NLP tasks (Zhang et al., 2015). The theoretical results in (Honorio and Jaakkola, 2016), based on the probably approximately correct Bayes framework, characterize these findings by providing generalization bounds. More recently, Ma et al. (2019) extended the work of (Zhang et al., 2014, 2015) to structured prediction tasks with large structured outputs by leveraging local classifiers to find good starting solutions and improve the accuracy of search. All of these methods were evaluated on linear structured models.

In this paper, we focus on two tasks: mining argumentative structures in essays and stance prediction in online debates. Stab and Gurevych (2017) approach argumentative essays using an exhaustive set of hand-crafted features, linear local classifiers and ILP at test time. Niculae et al. (2017) jointly learn to score multiple decisions while enforcing domain constraints. They explore structured SVMs and RNNs, using the AD^3 inference algorithm (Martins et al., 2015). On the other hand, there are several works on predicting user stances in online debates. Some approaches model the problem as a text classification task (Somasundaran and Wiebe, 2010; Sun et al., 2018), while other approaches take a collective approach to model user behavior and interactions (Walker et al., 2012; Hasan and Ng, 2013; Sridhar et al., 2015; Li et al., 2018). In the latter case, inference procedures include MaxCut, ILP and probabilistic soft logic (Bach et al., 2017).

3 Modeling

We look at two challenging structured prediction problems that deal with long texts where dependencies span across different paragraphs, documents and authors. To deal with these setups, we define neural factor graphs $G = \{\Psi\}$ where each decision $\psi_i \in \Psi$ is associated with a neural architecture ρ_i and a set of parameters $\boldsymbol{\theta}_i$. In this section, we introduce the tasks in detail.

3.1 Argument Mining

This task aims to identify argumentative structures in essays. Each argumentative structure forms a tree, and there is a forest per document. Nodes correspond to spans of text in the document and they can be labeled either as *claims, major claims* or *premises*. Edges correspond to stances (i.e., support/attack relations between nodes). The spans of texts are given, and we need to label nodes, predict which pairs of nodes are connected by an edge and label the edges. Domain knowledge can be exploited as there are only valid edges between pairs of premises, a premise and a claim, or a claim and a major claim. At the same time, all trees are rooted at major claims. Similarly to previous work, we model second order relationships: **grandparent** ($a \rightarrow b \rightarrow c$) and **co-parent** ($a \rightarrow b \leftarrow c$) (Martins and Almeida, 2014; Niculae et al., 2017).

Figure 1 has a visual representation of the structure. In this problem, each forest defines a factor graph Ψ and G is the collection of all documents. We define a set of five neural architectures corresponding to the five types of decisions that we need to make: $NN = \{\rho_{\text{node}}, \rho_{\text{link}}, \rho_{\text{stance}}, \rho_{\text{grandparent}}, \rho_{\text{coparent}}\}$, each with its own set of parameters $\boldsymbol{\theta} = \{\theta_{\text{node}}, \theta_{\text{link}}, \theta_{\text{stance}}, \theta_{\text{grandparent}}, \theta_{\text{coparent}}\}$. Note that in principle, we can substitute each (ρ_i, θ_i) with any neural architecture. We include details about the architectures in the experimental section.

3.2 Stance Prediction

Given a debate thread on a specific political issue, the task is to predict the stance of each post w.r.t. the issue (e.g., pro-life or pro-choice on abortion) (Walker et al., 2012). Following previous work, we model the problem as a collective classification task and consider all posts in a given thread. To do this, we add the task of predicting stance agreement between consecutive posts. As observed in Figure 1, each thread forms a tree, where users participate and respond to each other's posts. For a thread labeling to be valid, we need to enforce consistency between the node and edge labels.

In this case, each discussion thread defines a factor graph Ψ and G is the collection of threads. We define two neural architectures $NN = \{\rho_{\text{stance}}, \rho_{\text{agreement}}\}$, each with its own set of parameters $\boldsymbol{\theta} = \{\theta_{\text{stance}}, \theta_{\text{agreement}}\}$. As in the previous setup, each (ρ_i, θ_i) can be substituted by any neural architecture, more details are outlined in the experimental section.

4 Learning

We learn a joint neural model that uses inference during training to ensure consistency across all decisions. Let Ψ be a factor graph with potentials $\psi_i \in \Psi$ over all possible structures Y. Let $\boldsymbol{x_i}$ be the input vector to potential ψ_i. Let $\boldsymbol{\theta} = \{\theta^i\}$ be

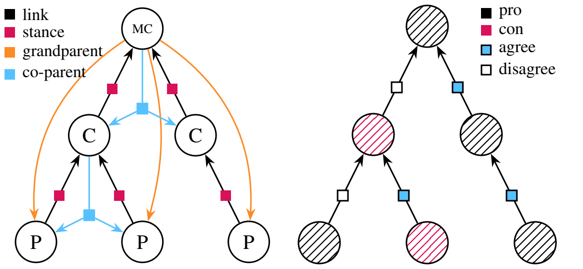


Figure 1: Argument Mining (left), Stance Prediction (right)

a set of parameter vectors associated with a set of neural networks $\boldsymbol{\rho} = \{\rho_i\}$, and $\rho_i(\boldsymbol{x_i}, \boldsymbol{y_i}; \theta^i)$ is the score for potential ψ_i resulting from a forward pass. Here $\boldsymbol{y} \in Y$ corresponds to the gold structure and $\boldsymbol{\hat{y}} \in Y$ to the prediction resulting from the MAP inference procedure:

$$\begin{gathered}\arg\max_{\boldsymbol{y}\in Y} \sum_{\psi_i \in \Psi} \rho_i(\boldsymbol{x_i}, \boldsymbol{y_i}; \theta^i) \\ s.t.\ c(\boldsymbol{x_c}, \boldsymbol{y_c}) \ \ \forall c \in C\end{gathered} \tag{1}$$

Where C is a set of domain-specific constraints defined over the factor graph Ψ, and $\boldsymbol{x_c}, \boldsymbol{y_c}$ indicates inputs and variables relevant to the constraints. In this work, we experiment with different algorithms to obtain or approximate the $\arg\max$, including the randomized procedures outlined in Section 5.

To learn $\boldsymbol{\theta}$, we use the structured hinge loss $L(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\hat{y}}; \boldsymbol{\theta})$ defined as:

$$\begin{aligned}\max\Big(0, \max_{\boldsymbol{\hat{y}}\in Y}\big(\Delta(\boldsymbol{y}, \boldsymbol{\hat{y}}) + \sum_{\psi_i \in \Psi} \rho_i(\boldsymbol{x_i}, \boldsymbol{\hat{y}_i}; \theta^i)\big) \\ - \sum_{\psi_i \in \Psi} \rho_i(\boldsymbol{x_i}, \boldsymbol{y_i}; \theta^i)\Big)\end{aligned} \tag{2}$$

Where $\Delta(\boldsymbol{y}, \boldsymbol{\hat{y}})$ is the Hamming loss. To introduce the Hamming loss into the objective, we perform loss augmented inference. The pseudo-code for the structured learning procedure can be observed in Algorithm 1. We implemented our models using DRaiL (Pacheco and Goldwasser, 2020), a declarative deep structured prediction framework built on PyTorch, and extended it to support our randomized inference procedures[1].

5 Randomized Inference

In this section, we describe the randomized inference procedure used for each task. We define the

[1] The source code for this paper is available on https://www.gitlab.com/purdueNlp/DRaiL

Algorithm 1 Deep Structured Prediction

```
$p \leftarrow 0$
loss$_{\text{best}} \leftarrow \infty$
$\boldsymbol{\theta} \leftarrow \boldsymbol{\theta}^{\text{local}}$
$\boldsymbol{\theta}^{\text{ret}} \leftarrow \boldsymbol{\theta}$
while $p <$ patience do
    for each $\Psi \in G_{\text{train}}$ do
        for each $\psi_i \in \Psi$ do
            $w_i \leftarrow \rho_i(\boldsymbol{x_i}, \boldsymbol{y_i}; \theta^i)$          //forward pass
        $\hat{\boldsymbol{y}} \leftarrow \arg\max_{\boldsymbol{y} \in Y} \sum_{\psi_i \in \Psi} w_i \psi_i(\boldsymbol{x_i}, \boldsymbol{y_i})$
        loss $\leftarrow L(\boldsymbol{x}, \boldsymbol{y}, \hat{\boldsymbol{y}}; \boldsymbol{\theta})$
        backpropagate loss
    loss$_{\text{dev}} \leftarrow 0$
    for each $\Psi \in G_{\text{dev}}$ do
        for each $\psi_i \in \Psi$ do
            $w_i \leftarrow \rho_i(\boldsymbol{x_i}, \boldsymbol{y_i}; \theta^i)$          //forward pass
        $\hat{\boldsymbol{y}}_{\text{dev}} \leftarrow \arg\max_{\boldsymbol{y} \in Y} \sum_{\psi_i \in \Psi} w_i \psi_i(\boldsymbol{x_i}, \boldsymbol{y_i})$
        loss$_{\text{dev}} \leftarrow$ loss$_{\text{dev}} + L(\boldsymbol{x}, \boldsymbol{y}_{\text{dev}}, \hat{\boldsymbol{y}}_{\text{dev}}; \boldsymbol{\theta})$
    if loss$_{\text{dev}} <$ loss$_{\text{best}}$ then
        loss$_{\text{best}} \leftarrow$ loss$_{\text{dev}}$
        $\boldsymbol{\theta}^{\text{ret}} \leftarrow \boldsymbol{\theta}$
        $p \leftarrow 0$
    else
        $p \leftarrow p + 1$
return $\boldsymbol{\theta}^{\text{ret}}$
```

relevant domain constraints for each case, and explain how we sample solutions that respect them. Finally, we include a discussion about the theoretical bounds for the linear case.

5.1 Argument Mining

For randomized inference on argument mining, we adapt the randomized greedy algorithm proposed by Zhang et al. (2014). Algorithm 2 outlines the overall procedure. We will consider that each paragraph $p \in P$ of an essay contains a single tree. We obtain a local optimum tree $\hat{y}$ by using the hill climbing algorithm, which is further described below. After that, $\hat{y}$ is labeled and added to the forest Y. We iterate over each paragraph (line 4) and subsequently score the forest as:

$$\bar{\mathcal{S}}(Y) = \sum_{\hat{y} \in Y} \mathcal{S}(\hat{y}) = \sum_{\hat{y} \in Y} w + h \, \|y - \hat{y}\|_1 \quad (3)$$

Where $w = \sum_{\psi_i \in \Psi} \rho_i(\boldsymbol{x_i}, \hat{\boldsymbol{y}}_{\boldsymbol{i}}; \theta^i)$ is the sum of the scores of the potentials for the predicted tree $\hat{y}$. We add a weighted Hamming distance term to the scoring function in order to additionally penalize the score the more the tree structure differs from the gold structure. $h \, \|y - \hat{y}\|_1$ gets close to w if the distance is low, and close to zero if it is high.

Algorithm 2 Randomized Inference

```
 1: $\hat{Y} \leftarrow \{\}$
 2: for number of restarts do
 3:     $Y \leftarrow \{\}$
 4:     for each $p \in P$ do
 5:         $\hat{y} \leftarrow$ hill_climbing($p$)
 6:         label($\hat{y}$)
 7:         $Y \leftarrow Y \cup \{\hat{y}\}$
 8:     if $\bar{\mathcal{S}}(Y) > \bar{\mathcal{S}}(\hat{Y})$ then
 9:         $\hat{Y} \leftarrow Y$
10: return $\hat{Y}$
```

More specifically, let $\|y - \hat{y}\|_1$ be in $[0, 1]$, e.g., by dividing the number of node and edge differences by the total number of nodes and edges. In its simplest form, h can be assigned to $-w$, and thus $\mathcal{S}(\hat{y})$ would become w if $y = \hat{y}$ or 0 if they differ in every node and edge. Whenever the score of the locally improved forest is better than the forest found so far, Y becomes the new currently best scoring forest $\hat{Y}$. Since hill climbing might get stuck in a local optimum, we repeat line 3-9 for a constant number of restarts.

Algorithm 3 Hill Climbing

```
$\hat{y}_0 \leftarrow$ initialize tree randomly for paragraph $p$
label($\hat{y}_0$)
$\hat{y} \leftarrow \hat{y}_0$
$t \leftarrow 0$
repeat
    $\mathcal{L} \leftarrow$ top-down level node list of $\hat{y}$
    for $i = 1, ..., |\mathcal{L}|$ do
        for $j = i - 1, ..., 0$ do
            $\hat{y}_{t+1} \leftarrow$ connect subtree of $\mathcal{L}_i$ to $\mathcal{L}_j$
            label($\hat{y}_{t+1}$)
            if $\mathcal{S}(\hat{y}_{t+1}) > \mathcal{S}(\hat{y})$ then
                $\hat{y} \leftarrow \hat{y}_{t+1}$
            $t \leftarrow t + 1$
until no improvement in this iteration
return $\hat{y}$
```

Algorithm 3 describes the hill climbing procedure. It initially draws uniformly a tree $\hat{y}_0$ at random. Then the greedy algorithm applies local updates on $\hat{y}_t$ and attempts to achieve a better scoring tree $\hat{y}_{t+1}$. This is done by iterating through a top-down level node list $\mathcal{L}$ of $\hat{y}_t$. Denote i as the current position in the list, then the entire subtree of $\mathcal{L}_i$ is connected to the node $\mathcal{L}_j$, whereas $j = i - 1, i - 2, \ldots, 0$. If the score of $\hat{y}_{t+1}$ is higher than the score of $\hat{y}_t$, the newly generated tree is kept. The algorithm continues until the score can no longer be improved and therefore yields a local optimum tree. Figure 2 depicts how such local

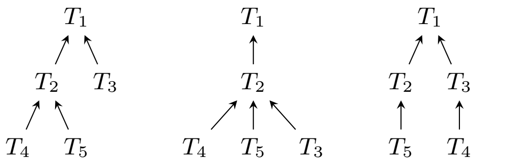


Figure 2: Greedy local update of a tree $\hat{y}_t$ (left) to $\hat{y}_{t+1}$ and $\hat{y}_{t+2}$ without score improvement

updates are performed, $\mathcal{L} = (T_1, T_2, T_3, T_4, T_5)$.

It might be the case that a paragraph contains more than a single tree, therefore, when a tree is initially drawn at random, we introduce an additional *phantom node* which serves as the new root. This modification no longer restricts hill climbing on trees only. Moreover, it allows us having multiple roots and we treat the second layer of the tree like the top layer.

Domain Specific Constraints: For node labeling, we exploit domain knowledge. *Major claims* can only occur in the first or last paragraph, and there has to be at least one major claim in each essay. A root gets labeled as major claim with some fixed probability depending on the paragraph (first or last), holding the condition that there has to exist at least one. Any other root is labeled as a *claim* in each paragraph. Nodes having an edge to a major claim are labeled as claims as well. All remaining nodes are *premises*. An edge can have either the label *support* or *attack* and we draw all edge labels randomly with a probability of 0.9 being a support label. The node and edge labels are determined after each iteration since scoring depends on both, links and labels.

In Section 6, we evaluate our models using randomized inference with and without domain specific constraints. In the latter case, all labels are chosen at random.

5.2 Stance Prediction

A debate thread provides a fixed structure, thus nodes and links are predefined and no improvement of the tree structure needs to be done. However, nodes and edges still need to be labeled and can be improved. Initially, we pick the node labels, which can either be *pro* or *con*. Following the observations made by Ma et al. (2019), we leverage local classifiers and greedily chose the label with the highest score for each node.

Domain Specific Constraints: To respect the dependencies between node and edges labels, we use the following heuristic: If two consecutive nodes u and v have different stances, the edge (u, v) receives a *disagreement* label, if they share the same stance, (u, v) gets an *agreement* label. When author constraints are considered as well, we additionally force stances of posts to be equal when written by the same author.

We attempt to improve node labels by flipping them randomly and subsequently adjust the edge labels. This is done until an iteration no longer improves the overall score. We restart the algorithm for a constant number of times in order to increase the chance of achieving a global optimum. In the experiments, we evaluate our models using randomized inference with and without domain specific constraints. When constraints are not used, a random node is flipped and its adjacent edge adjusted, without enforcing consistency in the whole tree.

The error of the *constrained* randomized algorithms can be bound for the linear case. Let's define the norm of the set of parameter vectors $\boldsymbol{\theta}$ as follows: $\|\boldsymbol{\theta}\| = \sqrt{\sum_{\theta^i \in \boldsymbol{\theta}} \|\theta^i\|^2}$, where $\|\theta^i\|$ is the Euclidean norm of the parameter vector θ^i. Let n be the number of training samples. From Theorem 2 and Claim ii in (Honorio and Jaakkola, 2016), for $\rho_i(\boldsymbol{x_i}, \boldsymbol{y_i}; \theta^i)$ linear in θ^i, the generalization bound (i.e., the difference between the test error and training error) is on the order of $\|\boldsymbol{\theta}\|^2/n + \|\boldsymbol{\theta}\|/\sqrt{n} + \max(1/\log 2, \|\boldsymbol{\theta}\|^2) \log^{3/2} n/\sqrt{n}$. The above generalization bound is decreasing in n, and increasing in $\|\boldsymbol{\theta}\|$, which suggests the use of a large training set, and the penalization of the norm $\|\boldsymbol{\theta}\|$ during learning. In our experiments, we show that in practice we can obtain competitive results by implementing the randomized algorithms for the non-linear case.

6 Experiments

We learn our models using four different inference procedures: (1) **ILP** defines the inference problem as an integer linear program and uses the Gurobi solver[2] to perform exact inference, (2) **AD^3/ILP** translates the ILP program into an AD^3 instance to perform approximate inference, (3) **Rand-C** uses the randomized method with domain constraints, and (4) **Rand** uses the randomized method without domain constraints. Note that we always use exact inference to evaluate on both the development and test sets. For completeness, we add an entry **AD^3**

[2] https://www.gurobi.com/products/gurobi-optimizer

where we use AD^3 for both training and testing. When using ILP or AD^3, the domain constraints are expressed declaratively.

All experiments were run on a 32 core 3.2Ghz Intel Xeon CPU machine with 128GB RAM and an NVIDIA GeForce GTX 1080 Ti 11GB GDDR5X GPU. We performed an exhaustive search for hyperparameters on the development set. We tuned the learning rate (lr $\in \{$1e-6, 2e-6, 5e-6, 1e-5, 2e-5, 5e-5, 1e-4, 2e-4, 5e-4, 1e-3, 2e-3, 5e-3, 1e-2, 2e-2, 5e-2, 1e-1$\}$), patience (p $\in \{5, 10, 15, 20\}$), and number of restarts (r $\in \{1, 5, 10, 15, 20, 30, 50, 100\}$). The weight decay was fixed at 1e-5 (PyTorch's default). We found that results were stable for local and global models, for different sets of constraints and across inference algorithms.

6.1 Argument Annotated Persuasive Essays

Dataset: We used the UKP dataset (Stab and Gurevych, 2017), consisting of 402 documents, with a total of 6,100 propositions and 3,800 links (17% of pairs). We use the train/dev/test splits used by Niculae et al. (2017), and report macro F1 for components and positive F1 for relations.

Learning and Representation: We did 5 repetitions and reported the average performance. Each repetition used a different seed to initialize the model parameters. For training, we used stochastic gradient descent, a patience of 10, weight decay of 1e-5, and 5 restarts for randomized inference. For local models, we used a learning rate of 0.05 and for structured learning we used a learning rate of 1e-4. Similarly to previous work on deep structured prediction (Han et al., 2019), we obtained better results by performing structured learning over locally trained models, instead of training them from scratch. To represent the component and the essay, we used a BiLSTM over the words, initialized with GloVe embeddings (Pennington et al., 2014), concatenated with a feature vector following Niculae et al. (2017), without the word features. For representing the relation, we use the components, as well as the relation features used in Niculae et al. (2017). For shallow models, we use a bag-of-words representation for the text and concatenate it with the rest of the features into a single feature vector. Both the feature extraction and the neural implementations are available in the repository.

We test two versions of the model: (1) **Base** includes node labeling, link prediction and link labeling, and (2) **Full** adds grandparent and co-parent factors. Domain constraints are introduced in all models.

Model	Inference	Node	Link	Avg	Stance
Local	–	70.7	52.8	61.7 (60.7)	63.4
	L+I	76.5	56.9	66.7 (66.5)	62.5
Base	ILP	83.0	57.6	70.3 (67.2)	68.0
	AD^3/ILP	**83.2**	58.2	**70.7** (67.2)	**68.4**
	AD^3	83.0	57.6	70.3 (67.2)	68.0
	Rand-C	82.8	58.4	70.6 (67.7)	**68.4**
	Rand	82.9	**58.5**	**70.7** (67.7)	68.0
Full	ILP	83.1	61.2	72.2 (65.3)	**69.2**
	AD^3/ILP	83.7	62.0	72.9 (65.3)	68.5
	AD^3	83.5	61.1	72.3 (65.3)	**69.2**
	Rand-C	**83.8**	62.6	73.2 (66.3)	68.4
	Rand	83.4	**63.2**	**73.3** (65.9)	68.4

Table 1: F1 for argument mining using **deep structured prediction**, Avg results using **shallow** models included in parenthesis

We can analyze the results across three dimensions:

Structured Learning: The advantage of leveraging more structural dependencies can be seen in Table 1. The model gets increasingly better as more dependencies are considered, and using global learning outperforms learning local models and using inference just at prediction time (L+I).

Deep vs. Shallow: There is a consistent trend showing that deep structured models are more expressive than their shallow counterparts, as we can see by comparing average results in Table 1. To obtain good results using linear classifiers, Stab and Gurevych (2017) relied on an exhaustive set of features (Table 2). These numbers cannot be replicated by using just word-features and the feature set suggested by Niculae et al. (2017), as our shallow models and their structured SVM results show. In contrast, deep models and word embeddings are able to leverage this information without additional features. In addition, we find that deep models have a shorter overall training time (3.3x faster for the full model). This can be attributed to the compact embedding representation used in deep models, in contrast to the large sparse one-hot vectors used in linear models. Similarly to previous work (Niculae et al., 2017), we find that higher-order factors and strict constraints are more helpful when using deep structured models than in their shallow counterparts.

Model	Node	Link	Avg
Human upper bound	86.8	75.5	81.2
ILP Joint (Stab and Gurevych, 2017)	82.6	58.5	70.6
Struct RNN strict (Niculae et al., 2017)	79.3	50.1	64.7
Struct RNN full (Niculae et al., 2017)	76.9	50.4	63.6
Struct SVM strict (Niculae et al., 2017)	77.3	56.9	67.1
Struct SVM full (Niculae et al., 2017)	77.6	60.1	68.9
Joint PointerNet (Potash et al., 2017)	84.9	60.8	72.9
Kuribayashi et al. 2019	**85.7**	**67.8**	**76.8**
BERT (Devlin et al., 2019)	71.1	50.8	61.0
BERT-doc	79.5	55.8	67.7
BERT-doc + Inf (Base)	79.9	58.1	69.0
BERT-doc + Structured Prediction (Base)	82.1	60.0	71.1
Deep Full ILP	83.1	61.2	72.2
Deep Full Rand-C	**83.8**	62.6	73.2
Deep Full Rand	83.4	**63.2**	**73.3**

Table 2: Previous work on UKP Dataset

Randomized vs. ILP/AD3: When using deep structured prediction, we did not find a statistically significant difference in the performance of the models that were trained with ILP/AD3 vs. the ones that were trained with constrained and non-constrained randomized inference.

We obtain competitive results with respect to previous work that relies on the same underlying embeddings or features, as observed in Table 2. Recently, Kuribayashi et al. (2019) were able to further improve performance by exploiting contextualized embeddings that look at the whole document, instead of embedding the arguments in isolation, and by making a distinction between argumentative markers and argumentative components. We attempted document-level contextualized embeddings using BERT and were not able to replicate their success[3]. Moreover, we found no significant improvement on the structured prediction models when replacing our BiLSTM encoders with either BERT or document-level BERT. We leave the exploration of an effective way to leverage contextualized embeddings for future work. As for stance prediction, Stab and Gurevych (2017) identify stances over the resulting structure and obtain a macro F1 of 68.0. Our full models obtain commensurate results, 69.2, 68.4 for ILP and randomized inference, respectively.

6.2 Debate Stance Prediction

Dataset: We use a subset of the 4FORUMS dataset from the Internet Argument Corpus (Walker et al., 2012), which consist of a total of 418 discussion threads on four political issues, containing 24,658 posts. We use the same splits as (Li et al., 2018). Most previous work reports accuracy. However, given that labels are highly imbalanced, we also report macro F1.

Learning and Representation: We model the problem as a collective classification task by predicting disagreement between consecutive posts in a given thread. We represented posts using a BERT encoder. For disagreement, we just represented pairs of posts without additional information. We do 5-fold cross validation and report the average performance. For training, we used AdamW, weight decay of 1e-5, a patience of 3, and 50 restarts for randomized inference. For local models, we used a learning rate of 5e-5 and for structured models we used a learning rate of 2e-6. For structured learning, we initialize the parameters using the local models. Note that we keep fine-tuning BERT during training.

Model	Infer.	A		E		GM		GC	
		Acc	F1	Acc	F1	Acc	F1	Acc	F1
Majority		56.8	28.4	65.9	33.0	66.0	33.0	67.9	34.0
Local		66.0	64.3	65.2	54.3	70.0	61.5	68.2	54.1
Base	L+I	71.0	70.4	63.3	59.2	73.6	69.4	66.8	60.2
	ILP	**72.4**	**71.8**	**64.7**	60.6	**75.1**	72.6	70.5	65.8
	AD^3/ILP	**72.4**	**71.8**	63.2	59.4	**75.1**	72.6	69.7	64.3
	AD^3	**72.4**	**71.8**	64.5	**60.7**	**75.1**	72.6	**71.0**	**66.0**
	Rand-C	71.9	71.5	63.1	60.0	75.0	**73.0**	65.4	60.8
	Rand	71.5	71.1	61.3	58.0	74.3	72.0	64.1	60.2
AC	L+I	83.6	84.6	73.3	69.7	84.8	81.9	68.2	60.9
	ILP	87.5	**88.0**	76.1	73.8	**91.2**	**90.3**	74.2	69.9
	AD^3/ILP	86.2	85.8	**76.7**	**73.9**	90.0	89.0	**74.4**	70.7
	AD^3	85.0	84.8	62.7	60.3	87.4	86.3	72.8	67.9
	Rand-C	**87.8**	87.6	**76.7**	73.7	88.9	87.7	73.4	**71.3**
	Rand	86.6	86.4	73.4	70.9	89.9	88.9	72.7	68.8

Table 3: **Post stance** on 4FORUMS. A: Abortion, E: Evolution, GM: Gay Marriage, GC: Gun Control

We test two versions of the model: (1) **Base** includes consistency between node and edge labels, and (2) **AC** adds author constraints enforcing the same stance for all posts by the same author.

Structured Learning: We can also see that the performance of all structured models outperforms learning local models and using inference just at prediction time (L+I), both for post stance (Table 3) and for disagreement (Table 4).

Randomized vs. ILP/AD3: In the case of stance prediction, there is a significant trend in the performance of the different inference methods. Learning with exact inference generally outperforms the randomized constrained procedure, and the latter outperforms its non-constrained version. The differ-

[3] We did not experiment with their extended BoW features, nor AC/AM distinction.

Model	Infer.	A		E		GM		GC	
		Acc	F1	Acc	F1	Acc	F1	Acc	F1
Majority		77.8	38.9	66.4	33.2	73.7	36.9	64.3	32.2
Local		76.0	58.1	63.4	56.0	71.3	56.9	67.0	61.4
Base	L+I	70.8	60.3	62.6	58.4	63.3	58.7	61.4	58.9
	ILP	74.2	**62.9**	63.6	**59.6**	**71.5**	61.4	63.8	59.9
	AD^3/ILP	74.2	**62.9**	62.8	58.8	**71.5**	61.4	64.2	61.5
	AD^3	74.2	**62.9**	**64.3**	59.5	**71.5**	61.4	64.2	59.9
	Rand-C	**76.0**	61.6	64.1	57.3	71.2	60.1	64.8	**61.8**
	Rand	**76.0**	60.7	62.7	58.5	70.4	**61.5**	**65.3**	59.4
AC	L+I	83.2	78.7	72.1	70.0	71.0	68.0	68.8	67.0
	ILP	88.0	82.2	**76.5**	**73.6**	**86.1**	**81.5**	75.4	73.6
	AD^3/ILP	86.3	80.5	74.8	72.4	83.9	79.2	72.8	71.5
	AD^3	84.9	76.4	66.8	61.4	84.4	78.4	68.1	65.8
	Rand-C	**88.2**	**82.4**	74.3	71.7	82.5	78.1	**78.5**	**76.3**
	Rand	87.7	81.4	75.0	71.9	84.1	78.7	75.8	74.4

Table 4: **Disagreement** on 4FORUMS. A: Abortion, E: Evolution, GM: Gay Marriage, GC: Gun Control

ence is more pronounced in the case of **AC** models. However, we find that relative to its simplicity, the randomized procedures obtain highly competitive performance.

Model	A	E	GM	GC	Avg
BERT (Devlin et al., 2019)	66.0	65.2	70.0	68.2	67.4
PSL (Sridhar et al., 2015)	77.0	80.3	80.5	69.1	76.7
Struct. Rep. (Li et al., 2018)*	86.5	**82.2**	87.6	**83.1**	**84.9**
Deep AC ILP	87.5	76.1	**91.2**	**74.2**	**82.3**
Deep AC Rand-C	**87.8**	**76.7**	88.9	73.4	81.7
Deep AC Rand	86.6	73.4	89.9	72.7	80.7

Table 5: Previous work on 4FORUMS (Post Acc)
*Note that (Li et al., 2018) use author profile information in their models, whereas we only look at text

Table 5 compares our models to previous work on this dataset. Sridhar et al. (2015) use probabilistic soft logic (PSL) to learn a global assignment for the post labels. They use local classifiers to obtain the input scores to PSL. The main difference between their approach and ours is that we are able to backpropagate the global error back into the classifiers, and we find that it improves performance considerably. Even though we use BERT encoders in our structured procedure, we can see that BERT alone is not able to solve the task. Lastly, we compare to the structured representation learning method of Li et al. (2018) and find that we are able to improve on abortion and gay marriage only. Note that these two are the issues with more data available (8,000 and 7,000 posts respectively). The main difference with their approach and ours is that they push author profile information into the learned representation. We hypothesize that this is key to obtain good performance for gun control, which contains only 4,000 posts.

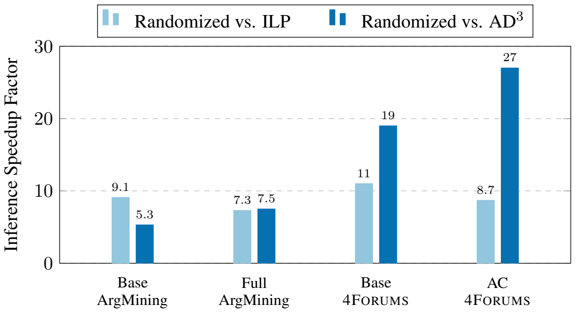


Figure 3: Comparing inference speedup per epoch

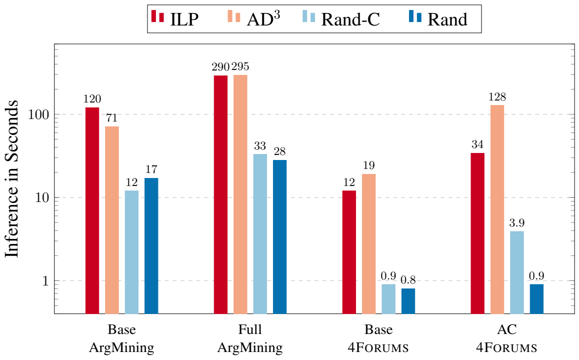
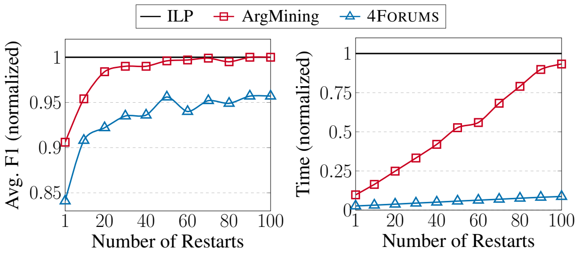


Figure 4: Pure inference runtime in seconds

6.3 Inference Analysis

In our experiments, randomized inference always outperforms ILP and AD^3 in terms of runtime. Figure 3 shows the speedup factor per epoch against ILP and AD^3. In argument mining, AD^3 is faster than ILP, except on our full model, where both perform similarly. We noticed that ILP consumes a lot of time in initialization and encoding. The randomized inference approach is able to predict argumentative structures 9.1x faster than ILP for our base model, and 7.5x faster than AD^3 for our full model. For stance prediction on 4FORUMS, ILP is considerably faster than AD^3, we presume that this is due to the fact that Gurobi is a highly optimized commercial software, and our graphs are small. Randomized inference is 11x faster than ILP on the base model and beats AD^3 by a factor of 27 when author constraints are used.

We also measured pure inference time over five training runs and took the average. Figure 4 shows (in logarithmic scale) plain inference runtime in seconds on the training set for all of our models. We can observe that randomized inference without domain constraints has almost the same performance as the constrained version. Again we find that randomized inference considerably outperforms ILP and AD^3.



(a) (b)

Figure 5: Impact of performance and runtime depending on the number of restarts for randomized inference

Additionally, we evaluated our model at test time by replacing exact inference with randomized inference, incrementally increasing the number of restarts. Figure 5 shows the performance and runtime of the Rand-C algorithm with respect to exact inference $\left(\text{i.e., } \frac{\text{Rand-C}}{\text{ILP}}\right)$. Figure 5a shows that the global optimum is closely approached after just 20 restarts for the argument mining task, as opposed to stance prediction on 4Forums, where a higher number of restarts is required. This is in line with our reported results in Sections 6.1 and 6.2. Figure 5b shows that randomized inference is about twice as fast than ILP when using 50 restarts for the Argument Mining task, and it starts to approach the time needed for ILP after 100 restarts. On the other hand, the randomized algorithm on 4Forums continues to be an order of magnitude faster even when doing 100 repetitions. Note that as the number of restarts keeps increasing, the randomized procedure will eventually surpass the time needed to perform exact inference.

7 Summary

We studied the effectiveness of randomized inference for deep structured prediction and obtained positive results for two challenging discourse-level tasks. We showed that, in practice, we can train complex structured models, using expressive neural architectures, and get competitive results at a lower computational cost. Moreover, we saw that combining expressive representations and inference is a promising direction for modeling discourse-level structures. Future directions include expanding the discussion to other tasks involving more complex structures, as well as exploring shared representations across different sub-tasks.

References

Daniel Andor, Chris Alberti, David Weiss, Aliaksei Severyn, Alessandro Presta, Kuzman Ganchev, Slav Petrov, and Michael Collins. 2016. Globally normalized transition-based neural networks. 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*.

Stephen H. Bach, Matthias Broecheler, Bert Huang, and Lise Getoor. 2017. Hinge-loss markov random fields and probabilistic soft logic. *J. Mach. Learn. Res.*, 18:109:1–109:67.

Danqi Chen and Christopher D. Manning. 2014. A fast and accurate dependency parser using neural networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 740–750.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Rujun Han, Qiang Ning, and Nanyun Peng. 2019. Joint event and temporal relation extraction with shared representations and structured prediction. 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 2019, Hong Kong, China, November 3-7, 2019*, pages 434–444.

Kazi Saidul Hasan and Vincent Ng. 2013. Stance classification of ideological debates: Data, models, features, and constraints. In *Sixth International Joint Conference on Natural Language Processing, IJCNLP 2013, Nagoya, Japan, October 14-18, 2013*, pages 1348–1356.

Jean Honorio and Tommi S. Jaakkola. 2016. Structured prediction: From gaussian perturbations to linear-time principled algorithms. In *Proceedings of the Thirty-Second Conference on Uncertainty in Artificial Intelligence, UAI 2016, June 25-29, 2016, New York City, NY, USA*.

Yangfeng Ji and Jacob Eisenstein. 2014. Representation learning for text-level discourse parsing. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics, ACL 2014, June 22-27, 2014, Baltimore, MD, USA, Volume 1: Long Papers*, pages 13–24.

Tatsuki Kuribayashi, Hiroki Ouchi, Naoya Inoue, Paul Reisert, Toshinori Miyoshi, Jun Suzuki, and Kentaro Inui. 2019. An empirical study of span representations in argumentation structure parsing. In

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4691–4698, Florence, Italy. Association for Computational Linguistics.

Guillaume Lample, Miguel Ballesteros, Sandeep Subramanian, Kazuya Kawakami, and Chris Dyer. 2016. Neural architectures for named entity recognition. In *NAACL HLT 2016, The 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, San Diego California, USA, June 12-17, 2016*, pages 260–270.

Chang Li, Aldo Porco, and Dan Goldwasser. 2018. Structured representation learning for online debate stance prediction. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 3728–3739.

Jiangming Liu, Shay B. Cohen, and Mirella Lapata. 2019. Discourse representation parsing for sentences and documents. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, pages 6248–6262.

Chao Ma, F. A. Rezaur Rahman Chowdhury, Aryan Deshwal, Md Rakibul Islam, Janardhan Rao Doppa, and Dan Roth. 2019. Randomized greedy search for structured prediction: Amortized inference and learning. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI 2019, Macao, China, August 10-16, 2019*, pages 5130–5138.

Xuezhe Ma and Eduard H. 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, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

André F. T. Martins and Mariana S. C. Almeida. 2014. Priberam: A turbo semantic parser with second order features. In *Proceedings of the 8th International Workshop on Semantic Evaluation, SemEval@COLING 2014, Dublin, Ireland, August 23-24, 2014*, pages 471–476.

André F. T. Martins, Mário A. T. Figueiredo, Pedro M. Q. Aguiar, Noah A. Smith, and Eric P. Xing. 2015. Ad^3: alternating directions dual decomposition for MAP inference in graphical models. *J. Mach. Learn. Res.*, 16:495–545.

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.

Maria Leonor Pacheco and Dan Goldwasser. 2020. Modeling content and context with deep relational learning. *Computing Research Repository*, arXiv:2010.10453.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, 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, Copenhagen, Denmark. Association for Computational Linguistics.

Swapna Somasundaran and Janyce Wiebe. 2010. Recognizing stances in ideological on-line debates. In *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, pages 116–124, Los Angeles, CA. Association for Computational Linguistics.

Dhanya Sridhar, James R. Foulds, Bert Huang, Lise Getoor, and Marilyn A. Walker. 2015. Joint models of disagreement and stance in online debate. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 116–125.

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

Qingying Sun, Zhongqing Wang, Qiaoming Zhu, and Guodong Zhou. 2018. Stance detection with hierarchical attention network. In *Proceedings of the 27th International Conference on Computational Linguistics, COLING 2018, Santa Fe, New Mexico, USA, August 20-26, 2018*, pages 2399–2409.

Marilyn A. Walker, Pranav Anand, Rob Abbott, and Ricky Grant. 2012. Stance classification using dialogic properties of persuasion. In *Human Language Technologies: Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, June 3-8, 2012, Montréal, Canada*, pages 592–596.

David Weiss, Chris Alberti, Michael Collins, and Slav Petrov. 2015. Structured training for neural network transition-based parsing. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing of the Asian Federation of Natural Language Processing, ACL 2015, July 26-31, 2015, Beijing, China, Volume 1: Long Papers*, pages 323–333.

Yuan Zhang, Tao Lei, Regina Barzilay, and Tommi S. Jaakkola. 2014. Greed is good if randomized: New inference for dependency parsing. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1013–1024.

Yuan Zhang, Chengtao Li, Regina Barzilay, and Kareem Darwish. 2015. Randomized greedy inference for joint segmentation, POS tagging and dependency parsing. In *NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Denver, Colorado, USA, May 31 - June 5, 2015*, pages 42–52.

Shuai Zheng, Sadeep Jayasumana, Bernardino Romera-Paredes, Vibhav Vineet, Zhizhong Su, Dalong Du, Chang Huang, and Philip H. S. Torr. 2015. Conditional random fields as recurrent neural networks. In *2015 IEEE International Conference on Computer Vision, ICCV 2015, Santiago, Chile, December 7-13, 2015*, pages 1529–1537.