Inference-Masked Loss for Deep Structured Output Learning

Quan Guo[1] , **Hossein Rajaby Faghihi**[1] , **Yue Zhang**[1] ,
Andrzej Uszok[2] and **Parisa Kordjamshidi**[1*]
[1]Michigan State University
[2] Florida Institute for Human and Machine Cognition
{guoquan, rajabyfa, zhan1624}@msu.edu, auszok@ihmc.us, kordjams@msu.edu

Abstract

Structured learning algorithms usually involve an inference phase that selects the best global output variables assignments based on the local scores of all possible assignments. We extend deep neural networks with structured learning to combine the power of learning representations and leveraging the use of domain knowledge in the form of output constraints during training. Introducing a non-differentiable inference module to gradient-based training is a critical challenge. Compared to using conventional loss functions that penalize every local error independently, we propose an inference-masked loss that takes into account the effect of inference and does not penalize the local errors that can be corrected by the inference. We empirically show the inference-masked loss combined with the negative log-likelihood loss improves the performance on different tasks, namely entity relation recognition on CoNLL04 and ACE2005 corpora, and spatial role labeling on CLEF 2017 mSpRL dataset. We show the proposed approach helps to achieve better generalizability, particularly in the low-data regime.

1 Introduction

Structured learning considers learning to predict output variables that are interdependent and need to obey some structural constraints. Structured learning approaches involve an inference phase after computing local predictions along with their probabilities/scores which can be generally formulated by a Maximum A Posteriori (MAP) inference problem. On one hand, structured learning are powerful in its flexibility to incorporate domain knowledge and requirement of fewer data. On the other hand, deep neural networks have achieved significant results by using large number of parameters and examples. Exploiting structured output in deep neural networks can help to combine the power of learning representations and leveraging the use of output structure during training. It will enable deep neural networks in imposing explicit structural constraints on their predictions by using domain knowledge on top of the data.

Current neural network architectures are able to represent structures such as sequences [Cho *et al.*, 2014] and graphs [Wu *et al.*, 2019]. However, the structure is mostly considered in the input side based on processing real-valued tensors and there is a lack of systematic way to predict discrete structured outputs. Neural networks are data-hungry. If we rely on the model to capture the global structure without explicitly modeling them, the need to the large data is even more critical. While providing explicit structural information to the model can reduce the need to large parameters and examples of deep neural networks, introducing the non-differentiable inference to gradient-based training is still a critical challenge. Deep neural networks are trained by back-propagation algorithms that calculate the gradient of the parameters. In most cases, the inference procedures are non-differentiable and the gradients can not be calculated based on the model output and the loss function.

In the previous studies on combining inference with learning of neural networks, we can observe at least two different types of techniques. One tries to learn local models to generate local predictions and rely on the inference for a global result, for example, structured perceptron [Collins, 2002]. However, the studies are mostly limited to linear models. More studies focus on incorporating the constraints from output structure and domain knowledge in the architecture of neural networks [Li and Srikumar, 2019] or learning algorithms [Chen *et al.*, 2015; Nandwani *et al.*, 2019; Liang *et al.*, 2019] so that the local neural networks learn to generate outputs respecting all the constraints.

In this work, we present a new loss function to extend deep neural network to use inference and exploit the structure of the output. Our proposed Inference-Masked Loss (IML) allows neural networks to keep the local predictions, even if they are false, as much as the inference can correct them. IML belongs to the first class of techniques mentioned above because instead of forcing the local output to respect the constraints, it trains the deep neural networks to rely on the inference. Compared to structured perceptron [Collins, 2002], IML integrates the inference result with the loss in a fully differentiable way and makes it compatible with arbitrary underlying deep neural networks. By relying on inference, the network needs less number of examples compared to train-

*Contact Author

ing without information about the output structure. There are other recent works involving deep neural networks learning with inference. For example, Nandwani *et al.* formulates the learning with inference as a primal-dual problem, solves it by alternating optimization, and addresses the problem by training the neural networks to respect the constraints [Nandwani *et al.*, 2019]. However, to produce the structured output and respect the constraints, it requires the neural networks to use global information even in producing a part of the output. In contrast, IML allows the neural networks to make local predictions without the constraints, and rely on the inference to correct the false predictions. It helps to decompose the dependencies of the output variables and allows the (sub-) neural networks to focus on training for parts of the output locally. The contribution of this paper is as follows:

- We propose the inference-masked loss (IML) that utilizes the global inference not only for prediction but also to train the parameters for structured output learning with deep neural networks.
- We evaluate the proposed method with two different tasks on three datasets under various settings to show the effectiveness of the proposed approach, especially with low training data.

2 Related Work

In structured learning, the inference is usually described by a Maximum A Posteriori (MAP) problem and formulated by a log-linear model, which takes the form of a combination of local feature functions or probabilistic factors [Sutton and McCallum, 2006].

The structured perceptron was proposed to extend the perceptron with structured output [Collins, 2002]. The algorithm updates the model parameters based on the difference between inference assignments and the ground-truth. A recent practice [Weiss *et al.*, 2015] has shown that the algorithm works with deep features. The structured perceptron was applied on top of a deep neural network that generates the features. However, the structured perceptron was trained separately from the deep neural network. Our proposed IML shares the same intuition of structured perceptron, that is, training based on the inference error. Both algorithms are tolerant to the local inconsistencies where there is no error in the inference results. However, the structured perceptron is limited to linear models. IML is a differentiable loss function to train deep neural networks. It can be incorporated with all sophisticated deep neural network architectures and deep learning regularizers.

The recent research regarding training neural networks concerning structural constraints are mainly imposing the constraints either by modifying the architectures, loss functions, or the formulation of the optimization.

Shen *et al.* worked on designing specific architectures and proposed a cumulative softmax to enforce ordering relation among a sequence of neurons. Li and Srikumar proposed a new approach to augment the neural networks with first-order logic by modifying the architecture by applying soft logic on the named neurons inside an existing architecture.

For reformulating the loss function, there is a collection of classical structured loss [Edunov *et al.*, 2018] used in combination with deep features. "SoftmaxMargin" loss [Gimpel and Smith, 2010] augments negative log-likelihood with a cost that penalizes high cost outputs proportional to their cost, which is similar to the "Margin Softmax" loss [Yang *et al.*, 2019; Ilharco *et al.*, 2019]. Researchers also investigated the approaches to incorporate soft and hard constraints with the loss function. Hard constraints are applied by smoothing out the loss function in [Pathak *et al.*, 2015] and by using Krylov subspace methods in [Marquez Neila *et al.*, 2017]. Also, Muralidhar *et al.* proposed to use an added term to the loss function imposing soft constraints. In this approach, hard constraints can be achieved by using a very large cost coefficient.

Training deep neural networks with inference can be formulated by a min-max optimization problem. The constrained maximization, which can be converted to unconstrained form with the Lagrangian multiplier, plays the role of the inference. The minimization updates the weights in the neural networks. To address the problem, Nandwani *et al.* took the dual form of the problems and solved the two optimizations alternatively. Chen *et al.* proposed to use a single step of message passing to estimate the inference and to train the neural networks with the estimated inference. Constraint guided semi-supervised learning algorithm was proposed to generate pseudo-labeled data by the constraints to train the parameters in a model with a supervised learning algorithm [Chang *et al.*, 2012]. Taking a different type of approach, Maes *et al.* formalized the problem of structured prediction as a Reinforcement Learning task and proposed to solve it by learning the policy.

Instead of training neural networks to respect the constraints, our proposed IML allows neural networks to keep local predictions, even if they are false, as long as the global inference can correct them. The model will count on the inference for a global output and tends to learn better decomposition of the task that helps to improve the generalizability of the model.

As mentioned before, MAP inference is usually an essential component of structured learning models. Inference can be performed with various approaches including probabilistic models, classical search algorithms, dynamic programming, or constrained optimization techniques. Some of recent practices integrate predictions of deep neural networks with conditional random fields (CRF) [Ma and Hovy, 2016] based on a chain structure with Viterbi algorithm. Besides, the beam-search algorithm is commonly used to improve the search efficiency of the inference. However, the efficiency of these approaches heavily depends on the independence structure of the underlying probabilistic model in a certain application. On a different thread, using integer linear programming (ILP) approach, one can represent the dependencies with linear constraints that can be solved by efficient off-the-shelf solvers[1]. Roth and Yih propose to model CRF inference as ILP for efficient solving and imposing global constraints over the output

[1]Solving ILP is known to be NP-Hard. However, the derived ILPs are usually very sparse and can be solved in a satisfactory time.

space beyond sequential structures [Roth and Yih, 2005].

One essential but orthogonal aspect to inference problem is the representation of the structure. Researchers have investigated approaches to represent domain knowledge in the form of first-order expressions and using inference over locally trained classifiers [Rizzolo and Roth, 2007], which was extended to the approaches that support joint training and inference [Chang *et al.*, 2012; Kordjamshidi *et al.*, 2015]. Ontologies were also investigated to generate constraints to improve local outputs of deep neural networks by global inference [Guo *et al.*, 2019]. We follow these approaches to represent the output structure and domain knowledge by ontologies and logical expressions, which are converted to linear constraints in the inference problem.

3 Inference-Masked Loss

3.1 Model Formulation

We assume training a deep neural network given a set of examples $\{(X, Y)\}$, that is, pairs of inputs X and outputs Y. Both inputs and outputs can have arbitrary structures. We assume the structure of the output variables can be expressed with a set of linear constraints among them, denoted by $\mathcal{C}(Y) \leq 0$. The constraints can be represented with a set of inequalities without loss of generality [Nandwani *et al.*, 2019]. These constraints originate from our knowledge about the domain and are expressed using logical expressions or in ontologies that are converted to linear constraints –this is detailed in Section 3.4. In this study, we focus on the case where all parts of the output structure are discrete values and can be associated with a collection of binary predicates $q \in \mathbf{Q}$ where $\mathbf{Q}$ is the set of all possible predicates. Each component Y_q of the structured output Y is a binary random variable and indicates whether q is true or not.

We are interested in the joint probability distribution of structured output variables y given input X, $P(y|X)$. The inference is finding the best assignment of y that yields the maximum joint probability. Our framework and loss function are agnostic to the inference method. In this study, we exploit ILP, which solves the problems with constraints that are not satisfying first order Markov property, efficiently [Roth and Yih, 2005]. The joint probability distribution is estimated by a normalized exponential function with logarithm estimated by the total scoring function $g(y, X; \theta)$ that $\log P(y|X) \propto g(y, X; \theta)$. Given the linear constraints, the inference can be represented as an ILP problem as follows.

$$y^* = \underset{y}{\operatorname{argmax}}\, g(y, X; \theta) \quad \text{subject to} \quad \mathcal{C}(y) \leq 0. \tag{1}$$

To train such a model, first, we make local predictions for each component of y. In other words, for each predicate $q \in \mathbf{Q}$, we make a local prediction $f_q(X; \theta)$ with deep neural networks, where f_q is a local network and θ is the weights and biases. Following the log-linear model formulation, we calculate a local scoring function $g_q = \log f_q(X; \theta)$ and a corresponding negative term $g_{\neg q} = \log(1 - f_q(X; \theta))$. We calculate the total scoring function for the output y by a linear model,

$$g(y, X; \theta) = \sum_{q \in \mathbf{Q}} g_q y_q + g_{\neg q} y_{\neg q}. \tag{2}$$

	Y_q	f_q	NLL	IML	
				$y_q^* = 0$	$y_q^* = 1$
TP	1	$\uparrow$	$-\log f_q(X; \theta)$	$-\log f_q(X; \theta)$	0^-
TN	0	$\downarrow$	$-\log(1 - f_q(X; \theta))$	0^-	$-\log(1 - f_q(X; \theta))$
FP	0	$\uparrow$	$-\log(1 - f_q(X; \theta))$	0^+	$-\log(1 - f_q(X; \theta))$
FN	1	$\downarrow$	$-\log f_q(X; \theta)$	$-\log f_q(X; \theta)$	0^+

Notations:
$\uparrow$: a probability close to 1; $\downarrow$: a probability close to 0.
0^-: a voided penalty in IML when the local prediction is correct;
0^+: a voided penalty in IML when the false prediction is corrected by the inference.

Table 1: Penalty term of NLL and IML regarding one predicate q.

The inference problem in (1) can be solved efficiently by off-the-shelf solvers[2].

3.2 The Loss Function

The commonly-used negative log-likelihood (NLL) loss function is given by

$$\mathcal{L}_{\text{NLL}} = \sum_{q \in \mathbf{Q}} -Y_q \log f_q(X; \theta) - (1 - Y_q) \log(1 - f_q(X; \theta)). \tag{3}$$

In fact, the ground-truth Y_q and the term $(1 - Y_q)$ serves as a selective mask to select penalizing the parameters of negative log-likelihood $-\log f_q(X; \theta)$ or $-\log(1 - f_q(X; \theta))$.

Inspired by the idea mentioned above, we introduce the IML. For each predicate q, if the associated component in the global inference y_q^* is correct according to the ground-truth Y_q, IML will nullify the corresponding terms by the mask. The resulting IML is as follows,

$$\begin{aligned}\mathcal{L}_{\text{IML}} = \sum_{q \in \mathbf{Q}} & -\left(1 - y_q^*\right) Y_q \log f_q(X; \theta) \\ & - y_q^* (1 - Y_q) \log(1 - f_q(X; \theta)).\end{aligned} \tag{4}$$

It can be observed that both masks $(1 - y_q^*)Y_q$ and $y_q^*(1 - Y_q)$ will be zero if $y_q^* = Y_q$. When $y_q^* \neq Y_q$, the masks will become Y_q and $(1 - Y_q)$ as in the NLL that, consequently, selects either the $-\log f_q(X; \theta)$ or $-\log(1 - f_q(X; \theta))$ to be penalized according to the ground-truth.

Table 1 shows the term that is added to the total loss regarding a predicate q based on NLL and IML, respectively. The first column denotes the True/False Positive/Negative notations regarding the local prediction. The $\uparrow$ in the third column indicates a probability close to 1 while a $\downarrow$ indicates that close to 0.

In the NLL, the term is selected by the ground-truth Y_q. Moreover, the magnitude is determined by how correct (or incorrect) the local prediction $f_q(X; \theta)$ is. As we introduced the inference to IML, the conditions become different. 0^+ and 0^- are the cases that the global inference gives the correct result ($H_q^* = Y_q$), while the others are for the cases when the inference is wrong. (FN, $y_q^* = 0$) and (FP, $y_q^* = 1$) are the criminals that we want to penalize. The 0^+ are the false local predictions that can be corrected by inference. Therefore, we do not need to penalize them. The (TP, $y_q^* = 0$) and (TN, $y_q^* = 1$) are innocent cases because they were correct locally and inference changed them to be wrong. However,

[2]We solve the ILP problems by Gurobi https://www.gurobi.com

we argue that if they get a higher local score, they should be able to help in correcting other false predictions. The 0^- are not problematic because they are correct, and they support the correct inference. No update needs to be applied in this case.

3.3 Combination

Training solely with IML in (4) could lead to "lazy" update and is more likely to be trapped in a local minimum compared to traditional loss because IML would ignore many of the local errors. The effect is two-fold. On one hand, with IML, the model receives fewer updates, which leads to a slower convergence. On the other hand, using IML would miss the opportunity to update a weak local model, which may cause trouble at testing time. We combine NLL and IML to enforce accurate local prediction.

$$\begin{aligned}
&\mathcal{L}_{\text{IML}(\lambda)} \\
&= \lambda \mathcal{L}_{\text{NLL}} + (1-\lambda)\mathcal{L}_{\text{IML}} \\
&= \sum_{q \in \mathbf{Q}} -\left(1-(1-\lambda)\, y_q^*\right) Y_q \log f_q\left(X;\theta\right) \\
&\qquad -\left(\lambda + (1-\lambda)\, y_q^*\right)\left(1-Y_q\right) \log\left(1-f_q\left(X;\theta\right)\right),
\end{aligned} \tag{5}$$

where λ is weighting the trade-off between NLL and IML, which is a hyper-parameter for the proposed loss. λ is also interpreted as the ratio of penalty to apply when the inference can resolve the errors on the local prediction and correct them in the global inference. λ times the corresponding NLL loss term will be penalized. Otherwise, if there are still errors according to the ground-truth, the penalty term is the same as NLL. This leads to having a better local prediction even if the inference can resolve the incorrectness. It is shown to be helpful in the Experiments Section and Analysis Section. However, λ is a new (and the only) hyper-parameter in this work. Experimental results with different λ will be reported and analyzed in Section 4.

3.4 Structure and Domain Knowledge

To represent the structure of the output and the domain knowledge, we follow the line of study on declarative knowledge representation for inference. We consider three types of generic relations between the parts: *is-a* indicates sub-typing of categorical predicates; *disjoint-with* indicates mutual exclusiveness of categorical predicates; and *has-a* indicates the composition of parts. These relations can be expressed by logical expressions. More specifically, *is-a* and *has-a* are mapped to implication $q_u \Rightarrow q_v$. *disjoint-with* is implemented by alternative denial $\neg\left(q_u \wedge q_v\right)$. Then we derive the linear constraints from the logical expressions by the rules in [Rizzolo and Roth, 2007].

4 Experiments

We evaluate the proposed approach with several structured learning tasks: Two different entity relation extraction (ER) tasks and spatial role labeling (SpRL) task. We investigate the entity and relation recognition corpora (CoNLL04) [Roth and Yih, 2004] and ACE 2005 Corpus (ACE2005) [Li and Ji, 2014] for ER task. The two datasets contain different types of entities and relationships. For SpRL task, CLEF 2017 mSpRL dataset (SpRL2017) [Kordjamshidi *et al.*, 2017a; Kordjamshidi *et al.*, 2017b] is investigated.

CoNLL04. CoNLL04 [Roth and Yih, 2004] is a publicly available corpus[3] for ER. The task is to recognize four types of entities among tokens in a sentence and classify five types of relations between entities. This corpus contains $5,516$ sentences, $11,182$ entities, and $2,048$ relations. The applied hard constraints are between the types of relations and the types of their two entities [Roth and Yih, 2004].

ACE2005. The ACE dataset contains documents with annotations defined for several tasks, including Named Entity Recognition, Relation Extraction, and Event Detection and Recognition. The dataset contains seven types of entities and 45 sub-entity types. We use the same data split used in [Li and Ji, 2014]. The training set contains $10,360$ sentences each of which includes at least one entity. The test set contains $2,637$ sentences some of which may not contain any entities. The total number of entities within the sentences of the training set is $47,406$, while the testing set contains $10,675$ of them.

CLEF 2017 mSpRL. The SpRL task is to identify and classify the spatial arguments of the spatial expression in a sentence [Kordjamshidi *et al.*, 2017a]. To be specific, we identify spatial roles, including "Trajector", "Spatial_indicator", and "Landmark", and detect their spatial triplet relation. We evaluated with CLEF 2017 mSpRL dataset [Kordjamshidi *et al.*, 2017a], which has 600 sentences in the training set and 613 sentences in the testing set. The dataset is more challenging because of the complicated triplet relations and fewer examples compared to other tasks.

4.1 Experimental Setup and Results

In this section, we will show the experiment settings for each task and report the results. We implemented all the experiments using Pytorch[4]. In all the tables IML(λ) denotes the training with the combined loss and a specified λ, and the prediction without inference. IML(λ)+ denotes training with the combined loss and a specified λ and the prediction with inference. Similarly, NLL denotes the training with NLL and prediction without inference while NLL+ means training with NLL and prediction with inference.

CoNLL04

We employ 768-dimensional pre-trained BERT [Devlin *et al.*, 2019], followed by an fully-connected layer of 768-dimensional hidden neurons as the token representation. For pairs based on which the relations are classified, we concatenate the representations of its two tokens and map it to a representation of the pairs by an additional fully-connected layer with 768-dimensional hidden neurons. Leaky ReLU, with a negative slope of 0.01, is used in the fully-connected layers. Independent Softmax of two classes based on the representation of the token or the pair are applied for local predictions. We handcrafted a simple ontology based on the constraints

[3] https://cogcomp.seas.upenn.edu/page/resource_view/43

[4] Our code is available at https://github.com/HLR/Inference-Masked-Loss

Task	Constraints
CoNLL04	1. Entity Disjoint; 2. Relation Composition.
ACE05	1. Entity Disjoint; 2. Entity Sub-typing.
CLEF 2017 mSpRL	1. Triplet Composition.

Table 2: Constraints used in different experiments.

	B1 (L+I)	B2 (IBT)	IML (0.6)	IML $(0.6)+$
Location	0.8369	0.8160	0.8534	**0.8707**
Organization	0.7246	0.7115	0.7801	**0.8035**
People	0.7857	0.7888	0.9329	**0.9381**
Other	-	-	0.6885	**0.6893**
Average Entities	-	-	0.8137	**0.8254**
Kill	-	-	**0.9871**	0.9855
Live-in	0.6075	0.6948	0.9511	**0.9614**
Located-in	-	-	0.9416	**0.9457**
Orgbase-on	-	-	0.9487	**0.9514**
Work-for	0.5797	0.6900	0.9607	**0.9610**
Average Relations	-	-	0.9578	**0.9610**
Average All	-	-	0.8938	**0.9007**

B1, B2: Baseline [Kordjamshidi *et al.*, 2015]

Table 3: F1 score compared with baseline, on CoNLL04.

in [Roth and Yih, 2004], adding disjoint constraints among all entity types. As shown in Table 2, we consider two types of constraints for CoNLL04. Entity disjoint constraint means each token can have at most one entity type. Relation composition constraint indicates that the arguments of a relation must have entity types consistent with the relation type as defined in [Roth and Yih, 2004]. The constraints are used for global inference in both training and testing. IML (0.6) (i.e. with $\lambda = 0.6$) is used and optimized by Adam optimizer [Kingma and Ba, 2014] in batches of 8 examples. We train for 100 epochs with the learning rate of $1e-4$, which is decayed 10 times every ten epochs. We use weight decay $1e-5$ and a dropout rate of 0.35 to avoid over-fitting. Focal loss $\gamma = 2$ and label smoothing 0.01 are used for imbalanced class labels. We conducted five-fold cross-validation over all samples for the final evaluation.

The baseline [Kordjamshidi *et al.*, 2015] used sophisticated linguistic features and structured features with a linear model and constraints [Roth and Yih, 2004]. Different training and testing settings are reported: local models (LO), Local learning and global prediction (L+I), joint training (IBT). Table 3 shows our F1-score compared with the baseline using L+I and IBT setting. Our model shows significant improvement, however, mainly because of BERT, which is capable of encoding each token with its context. We also achieve significant additional improvement by making inference after local prediction for most of the cases. The only case that inference decreases the performance is for the "Kill" relation, where local prediction performance is very high. It is connected to two "People", whose local prediction performance is rather weak, which may cause trouble for inference of "Kill". However, "Kill" gives "People" a boost. This is a trade-off between strong and weak local models.

ACE2005

We have designed a model as a baseline to experiment with both NNL and IML loss functions. The model is using BERT [Devlin *et al.*, 2019]+FLAIR [Akbik *et al.*, 2018]+GLoVe [Pennington *et al.*, 2014] for the representation of tokens. The first layer of the model is a Bi-directional LSTM layer converting the $5,236$-dimensional input of each token to a 2×240 representation vector. It is followed by a shared fully connected layer to a 480-dimensional representation and then classified by logistic regression for each class of entity and sub-entities. We exploit two different types of constraints for ACE05 as shown in Table 2. First, the disjoint property of different entity types. Second, the hierarchical constraints between entity types and sub-types in the dataset. The results of this experiment are listed in Table 4. Training with NLL, the model exceeds the published results [Straková *et al.*, 2019] after inference. With IML (0.5), we gain better results even without inference and get additional improvement after inference.

	NER	NLL	NLL+	IML (0.5)	IML $(0.5)+$
Entities	84.3	84.19	84.95	84.5	**85.05**

NER: [Straková *et al.*, 2019]

Table 4: F1-score of The ACE2005 entity recognition.

	NLL	NLL+	IML (0)	IML $(0)+$	IML (0.6)	IML $(0.6)+$
Entities	76.98	77.32	10.8	68.41	77.43	**77.65**

Table 5: F1-score of the ACE2005 entity recognition with hierarchical constraint on reduced training set

The IML (0.5) (with $\lambda = 0.5$) is optimized by Adam optimizer. We train for 100 epochs, with the learning rate of 0.04. We use focal loss $\gamma = 2$ and label smoothing 0.01 to deal with imbalanced class labels.

We have also compared the results of training on a smaller dataset (1000 sentences) to show whether IML improves the predictions in the low data regime. Table 5 shows the result of this experiment. In this experiment, we increased our representation space from 480 to 1000. While overall constraints helped in improving the results in the previous experiment we noticed the hierarchical constraints did not help significantly when working on the whole dataset. However, both types of constraints were helpful to achieve some improvements when training on the smaller number of examples in Table 5.

CLEF 2017 mSpRL

In this task, we firstly encode the input phrases with different linguistic features generated by SpaCy[5], such as POS-tag, lemma, dependency path. For spatial relation extraction, we concatenate the phrase encoding of the three spatial roles. Then, we use Recurrent Neural Networks and Logistic Regression to predict spatial role labels of phrases and extract spatial triplet relations among them.

We consider only one type of constraint according to the Table 2. Spatial triplets must include the three spatial roles, which are "Trajector", "Spatial Indicator", and "Landmark". We jointly predict the spatial roles and triplets. We trained IML (0.6) ($\lambda = 0.6$) by Adam for 20 epochs with a learning rate of 0.005, weight decay of 0.001 and dropout rate of 0.5 to avoid over-fitting.

[5]https://spacy.io/

	B	B+	NLL	NLL+	IML (0.6)	IML $(0.6)+$
Trajector	0.5787	0.6255	0.7598	0.7604	0.7604	0.7649
Landmark	0.7638	0.8065	0.8674	0.8674	0.8904	0.8904
Spatial Indicator	0.9520	0.9496	0.9482	0.9482	0.9509	0.9509
Spatial Triplet	0.6282	0.6825	0.5070	0.5138	0.5279	0.5346

B: Baseline [Manzoor and Kordjamshidi, 2018]

Table 6: CLEF 2017 mSpRL results compared with baseline.

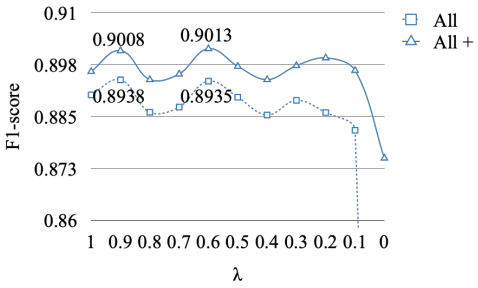

Figure 1: Different λ.

Table 6 shows IML improves the performance of spatial role labeling. Specifically, the inference improves the prediction results in most cases, and IML strengthens the utilization of inference compared to NLL. We use totally different parsing tools from the published result [Manzoor and Kordjamshidi, 2018], which leads to different phrases and relation candidates. Therefore, the input features are different and our results of triplet extractions which needs a lot of feature engineering are not competitive. However, we improve the spatial role extractions and IML+ consistently provides the best results for the triplet extraction compared to NLL and other variations in the scope of our own feature representations.

5 Analysis

To investigate the impact of IML and to find the optimal configuration, we conducted a series of variant experiments on EMR with the CoNLL04. For the sake of experimental time efficiency, we only evaluated using the first data split of the five-folds. All the experimental setting remains the same except those specified for each experiment.

Variation on λ. In IML (λ), λ is the parameter that controls the penalty that we apply on a false local prediction even if it can be corrected by inference. IML (0) is identical to IML while IML (1) is equivalent to NLL. The performance of the model with different λ values is shown in Figure 1. The dashed line with boxes shows the performance of local predictions, and the solid line with triangles shows the performance of global inference. Inference improves local predictions most of the time. $\lambda = 1$ (NLL) achieved a strong baseline performance. However, it is not the best. $\lambda = 0.6$ achieves the best inference result in this experiment. The performance with $\lambda = 0$ (IML) is too low to be shown properly in the chart. However, the corresponding inference performance is still competitive. λ is the only hyper-parameter proposed in this work.

Data Subsampling. We evaluate the training approach by using different variants of IML (λ) with less training data by subsampling the training set (4,413 sentences). The result is shown in Figure 2. The dashed lines with boxes are the performance of local prediction, and the solid lines with triangles are the performance of the global inference. Blue, green, and red curves show the result of different λs. In terms of inference performance, $\lambda = 0$ is rather weak compared with $\lambda = 1$ and $\lambda = 0.6$ when training with full data. For $\lambda = 1, 0.6$, the performance keeps decreasing when the data scale is reduced, while $\lambda = 0$ performs quite stable. It becomes competitive when 1/2 of the data is used and among the best is when only 1/8 of the data is available. When only 1/16 of the data is used, $\lambda = 0$ surpasses the other two settings by 9.6%. The prediction performance with $\lambda = 0$ is clearly very stable all over the time. With IML ($\lambda = 0$), the model can be trained to use local neural networks to learn minimal facts from the data and count on the constraints to get a robust global result. This is because IML decouples the structured learning task and leads to the need for fewer data.

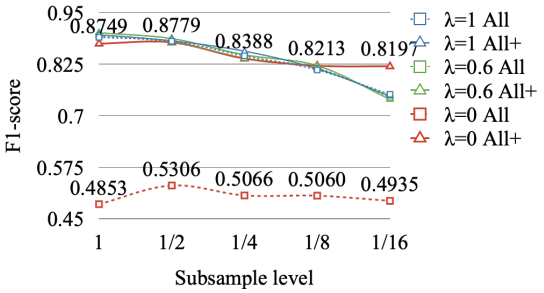

Figure 2: Different portion training set.

6 Conclusion

We investigated the combination of deep neural networks with structured output by introducing the inference-masked loss. The proposed structured output learning model imposes the structure of data and domain knowledge in the form of logical constraints that describe the correlations between output variables. Inference-masked loss takes the inference into account based on the domain knowledge compiled from logical expressions. It allows local deep neural networks to make false local predictions that can be corrected by the inference. The loss helps to decompose the learning task and let the neural networks focus on local representations and make local predictions. The inference collects the local predictions based on the structure of output and domain knowledge. One advantage of IML approach is being agnostic to the inference model and treating it as a black box. This helps to plug in any inference mechanism in the loop of deep learning iterations. The proposed approach improves the generalizability of the model and robustness of training, leading to state-of-the-art results on entity relation extraction and spatial role labeling tasks, with CoNLL04, ACE2005, and CLEF 2017 mSpRL datasets, respectively. In particular, it shows improvements when there is only a small set of annotated data available.

Acknowledgments

This work is (partially) supported by the Office of Naval Research grant N00014-19-1-2308.

References

[Akbik *et al.*, 2018] Alan Akbik, Duncan Blythe, and Roland Vollgraf. Contextual string embeddings for sequence labeling. In *COLING 2018, 27th International Conference on Computational Linguistics*, pages 1638–1649, 2018.

[Chang *et al.*, 2012] Ming-Wei Chang, Lev Ratinov, and Dan Roth. Structured learning with constrained conditional models. *Machine learning*, 88(3):399–431, 2012.

[Chen *et al.*, 2015] Liang-Chieh Chen, Alexander Schwing, Alan Yuille, and Raquel Urtasun. Learning deep structured models. In Francis Bach and David Blei, editors, *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pages 1785–1794, Lille, France, 07–09 Jul 2015. PMLR.

[Cho *et al.*, 2014] Kyunghyun Cho, Bart van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October 2014. Association for Computational Linguistics.

[Collins, 2002] Michael Collins. Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP 2002)*, pages 1–8. Association for Computational Linguistics, July 2002.

[Devlin *et al.*, 2019] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

[Edunov *et al.*, 2018] Sergey Edunov, Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. Classical structured prediction losses for sequence to sequence learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 355–364. Association for Computational Linguistics, June 2018.

[Gimpel and Smith, 2010] Kevin Gimpel and Noah A. Smith. Softmax-margin CRFs: Training log-linear models with cost functions. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 733–736, Los Angeles, California, June 2010. Association for Computational Linguistics.

[Guo *et al.*, 2019] Quan Guo, Andrzej Uszok, and Parisa Kordjamshidi. From ontologies to learning-based programs. In *IJCAI 2019 Workshop on DeLBP*, 2019.

[Ilharco *et al.*, 2019] Gabriel Ilharco, Yuan Zhang, and Jason Baldridge. Large-scale representation learning from visually grounded untranscribed speech. In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 55–65. Association for Computational Linguistics, November 2019.

[Kingma and Ba, 2014] Diederik P Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2014.

[Kordjamshidi *et al.*, 2015] Parisa Kordjamshidi, Hao Wu, and Dan Roth. Saul: Towards declarative learning based programming. In *Proc. of the International Joint Conference on Artificial Intelligence (IJCAI)*, 7 2015.

[Kordjamshidi *et al.*, 2017a] Parisa Kordjamshidi, Taher Rahgooy, Marie-Francine Moens, James Pustejovsky, Umar Manzoor, and Kirk Roberts. Clef 2017: Multimodal spatial role labeling (msprl) task overview. In *International Conference of the Cross-Language Evaluation Forum for European Languages*, pages 367–376. Springer, 2017.

[Kordjamshidi *et al.*, 2017b] Parisa Kordjamshidi, Martijn van Otterlo, and Marie-Francine Moens. *Spatial Role Labeling Annotation Scheme*, pages 1025–1052. Springer Netherlands, Dordrecht, 2017.

[Li and Ji, 2014] Qi Li and Heng Ji. Incremental joint extraction of entity mentions and relations. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 402–412, 2014.

[Li and Srikumar, 2019] Tao Li and Vivek Srikumar. Augmenting neural networks with first-order logic. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 292–302, Florence, Italy, July 2019. Association for Computational Linguistics.

[Liang *et al.*, 2019] Chen Liang, Ni Lao, Wang Ling, Zita Marinho, Yuandong Tian, Lu Wang, Jason D Williams, Audrey Durand, and Andre Martins. Deep Reinforcement Learning Meets Structured Prediction, 2019.

[Ma and Hovy, 2016] Xuezhe Ma and Eduard Hovy. 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)*, pages 1064–1074, Berlin, Germany, August 2016. Association for Computational Linguistics.

[Maes *et al.*, 2009] Francis Maes, Ludovic Denoyer, and Patrick Gallinari. Structured prediction with reinforcement learning. *Machine learning*, 77(2-3):271, 2009.

[Manzoor and Kordjamshidi, 2018] Umar Manzoor and Parisa Kordjamshidi. Anaphora resolution for improving spatial relation extraction from text. In *Proceedings of the First International Workshop on Spatial Language Understanding*, pages 53–62, 2018.

[Marquez Neila *et al.*, 2017] Pablo Marquez Neila, Mathieu Salzmann, and Pascal Fua. Imposing hard constraints on deep networks: Promises and limitations. In *CVPR Workshop on Negative Results in Computer Vision*, 2017.

[Muralidhar *et al.*, 2019] Nikhil Muralidhar, Mohammad Raihanul Islam, Manish Marwah, Anuj Karpatne, and Naren Ramakrishnan. Incorporating Prior Domain Knowledge into Deep Neural Networks. *Proceedings - 2018 IEEE International Conference on Big Data, Big Data 2018*, pages 36–45, 2019.

[Nandwani *et al.*, 2019] Yatin Nandwani, Abhishek Pathak, Mausam, and Parag Singla. A primal dual formulation for deep learning with constraints. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 12157–12168. Curran Associates, Inc., 2019.

[Pathak *et al.*, 2015] Deepak Pathak, Philipp Krahenbuhl, and Trevor Darrell. Constrained convolutional neural networks for weakly supervised segmentation. In *Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV)*, ICCV '15, pages 1796–1804, Washington, DC, USA, 2015. IEEE Computer Society.

[Pennington *et al.*, 2014] Jeffrey Pennington, Richard Socher, and Christopher D. Manning. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, 2014.

[Rizzolo and Roth, 2007] N. Rizzolo and D. Roth. Modeling discriminative global inference. In *Proc. of the IEEE International Conference on Semantic Computing (ICSC)*, pages 597–604, Irvine, California, 9 2007. IEEE.

[Roth and Yih, 2004] D. Roth and W. Yih. A linear programming formulation for global inference in natural language tasks. In Hwee Tou Ng and Ellen Riloff, editors, *Proc. of the Conference on Computational Natural Language Learning (CoNLL)*, pages 1–8. Association for Computational Linguistics, 2004.

[Roth and Yih, 2005] D. Roth and W. Yih. Integer linear programming inference for conditional random fields. In *Proc. of the International Conference on Machine Learning (ICML)*, pages 737–744, 2005.

[Shen *et al.*, 2019] Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. Ordered neurons: Integrating tree structures into recurrent neural networks. In *International Conference on Learning Representations*, 2019.

[Straková *et al.*, 2019] Jana Straková, Milan Straka, and Jan Hajič. Neural architectures for nested ner through linearization. *arXiv preprint arXiv:1908.06926*, 2019.

[Sutton and McCallum, 2006] Charles Sutton and Andrew McCallum. An introduction to conditional random fields for relational learning. In Lise Getoor and Ben Taskar, editors, *Introduction to Statistical Relational Learning*. MIT Press, 2006.

[Weiss *et al.*, 2015] David Weiss, Chris Alberti, Michael Collins, and Slav Petrov. 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 (Volume 1: Long Papers)*, pages 323–333. Association for Computational Linguistics, July 2015.

[Wu *et al.*, 2019] Zonghan Wu, Shirui Pan, Fengwen Chen, Guodong Long, Chengqi Zhang, and Philip S Yu. A comprehensive survey on graph neural networks. *arXiv preprint arXiv:1901.00596*, 2019.

[Yang *et al.*, 2019] Yinfei Yang, Gustavo Hernandez Abrego, Steve Yuan, Mandy Guo, Qinlan Shen, Daniel Cer, Yun-hsuan Sung, Brian Strope, and Ray Kurzweil. Improving multilingual sentence embedding using bi-directional dual encoder with additive margin softmax. In *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence, IJCAI-19*, pages 5370–5378. International Joint Conferences on Artificial Intelligence Organization, 7 2019.