

---

# Sequential Learning of Classifiers for Structured Prediction Problems

---

**Dan Roth**

Dept. of Computer Science  
Univ. of Illinois at U-C  
danr@illinois.edu

**Kevin Small**

Dept. of Computer Science  
Univ. of Illinois at U-C  
ksmall@illinois.edu

**Ivan Titov**

Dept. of Computer Science  
Univ. of Illinois at U-C  
titov@illinois.edu

## Abstract

Many classification problems with structured outputs can be regarded as a set of inter-related sub-problems where constraints dictate valid variable assignments. The standard approaches to these problems include either independent learning of individual classifiers for each of the sub-problems or joint learning of the entire set of classifiers with the constraints enforced during learning. We propose an intermediate approach where we learn these classifiers in a sequence using previously learned classifiers to guide learning of the next classifier by enforcing constraints between their outputs. We provide a theoretical motivation to explain why this learning protocol is expected to outperform both alternatives when individual problems have different ‘complexity’. This analysis motivates an algorithm for choosing a preferred order of classifier learning. We evaluate our technique on artificial experiments and on the entity and relation identification problem where the proposed method outperforms both joint and independent learning.

## 1 INTRODUCTION

Classification problems with structured output spaces are becoming increasingly common in different disciplines including natural language processing, computational biology and computer vision. Solving many of these problems involves solving a large set of inter-related sub-problems. As an example consider

the semantic role labeling (SRL) task (Carreras and Màrquez, 2004a), where a model needs to predict positions of verbal and nominal predicates in a sentence, select their sense, and identify their arguments for each possible argument role. For many of these problems, it is easy to define a set of constraints which enforce coherence of the output structure. In semantic role labeling, we can observe that a subject and an object of a verbal predicate always appear on opposite sides of the verb or given a predicate some of its arguments are illegal, e.g., the verb *say* cannot have an object (Punyakanok et al., 2008).

The most common approaches to solving these problems are either joint learning where global inference is used to enforce constraints during training (Inference Based Training, IBT) or independent learning of individual classifiers with no global inference (Learning Only, LO). Consistency of the output of a model learned with the LO method can be improved by enforcing constraints at test time (Learning + Inference, L+I). On a number of problems it has been observed that when individual sub-problems are easy to learn in isolation, L+I outperforms IBT (Punyakanok et al., 2005; Carreras and Màrquez, 2004b). Punyakanok et al. (2005) presented a theoretical analysis which suggests that if individual problems are linearly separable then LO should outperform IBT, assuming that each individual classifier uses a feature set which is smaller in size than the combined feature set used by the entire set of classifiers. Artificial experiments (Punyakanok et al., 2005) also suggested that with limited amount of training data and low degree of local non-separability, LO achieves comparable performance with IBT, and L+I significantly outperforms IBT. However, this separability requirement is very strong and not realistic in most real applications; it is very unlikely that all the sub-problems are easy to learn in isolation. This limitation motivates the development of new techniques which both do not require local separability for every sub-problem and achieve better generalization properties with less training data than IBT.

---

Appearing in Proceedings of the 12<sup>th</sup> International Conference on Artificial Intelligence and Statistics (AISTATS) 2009, Clearwater Beach, Florida, USA. Volume 5 of JMLR: W&CP 5. Copyright 2009 by the authors.

In this paper, we relax the local separability assumption and hypothesize that it is possible to order sub-problems such that for every sub-problem a set of learned classifiers for the preceding sub-problems helps to learn a subsequent classifier which renders the considered sub-problem separable. We propose a training protocol (sequential learning, SL) which discovers such an ordering and trains classifiers in this sequence, using preceding classifiers to guide learning of the next classifier by enforcing constraints between their outputs. Easier sub-problems are solved using a simpler class of functions, thereby resulting in better generalization with a limited amount of training data. In section 3.1, we provide theoretical analysis to confirm this claim. Moreover, whereas enforcing global coherence at testing time with LO classifiers lacks a good probabilistic motivation, the SL protocol corresponds to estimating parameters of a joint model. At each step of the learning sequence, the SL model is refined to account for labels of the classifier considered on this step. Therefore, enforcing the constraints with SL corresponds to finding the most likely sequence according to this learned model, i.e., the Viterbi decoding. This argument motivates section 3.2, where we demonstrate that L+I is likely to make incorrect predictions on rather simple distributions, whereas joint decoding with SL predicts correct outputs.

The proposed technique is related to classifier pipelines, standard in statistical natural language processing (NLP) (Finkel et al., 2006; Bunescu, 2008), where the output of a classifier trained for one sub-problem is provided as an input to the subsequent classifier. As an example, consider the case where predicted part-of-speech tags of words in a sentence are given as input to a syntactic parser which, in turn, outputs hierarchical syntactical representation used as input to a SRL model (Carreras and Màrquez, 2004a). The primary motivation for pipelines is to reduce computational expense relative to joint learning and, therefore, there is relatively little work analyzing the generalization properties of pipelines. From this perspective, the contribution of this paper is construction of a novel pipeline architecture where subsequent stages are integrated only by means of constraints, development of a criterion for finding a preferred order of sub-problems in these pipelines, and theoretical analysis of the resulting method. Interestingly, our conclusions dispute a general belief in NLP that joint modeling is preferable to pipelines.<sup>1</sup> However, the most

successful joint learning techniques (Collobert and Weston, 2008; Henderson et al., 2008) exploit shared internal representation, such as vectors of latent variables in graphical models or hidden layers in neural networks, to relate underlying properties of jointly learned sub-problems. This aspect of joint learning lies outside of the scope of this paper.

We evaluate our approach with artificial experiments and a real problem, the entity and relation identification task (Roth and Yih, 2004). For artificial experiments, we consider sequence labeling with randomly generated constraints and variable complexity of the individual sub-problems. The model trained with the SL method achieves significantly higher accuracy than LO, L+I, and IBT when the amount of training data is limited. As expected, IBT achieves the best performance as the amount of training data is increased. On the entity and relation identification task we show that the SL method outperforms all alternatives.

The remainder of the paper is structured as follows. Section 2 begins with a formal definition of the problem and the standard approaches. We then introduce the SL method as a way to address the discovered limitation of IBT, LO, and L+I. Section 3 starts with consideration of learning bounds for LO, SL, and IBT, demonstrating that the SL method is expected to achieve better accuracy under the stated assumptions. The second part of section 3 considers the probabilistic argument behind SL and argues that joint inference with SL is appropriate. In section 4, we provide an empirical evaluation.

## 2 SEQUENTIAL LEARNING

As discussed in the preceding section, we consider structured problems where we need to learn a mapping from the input space  $\mathcal{X}$  to the structured output space  $\mathcal{Y}$ . We further assume that the task can be decomposed into  $T$  sub-problems. For each of these sub-problems,  $t$ , we associate a scoring function  $\phi(y_t, x_t, w_t)$ , where  $y_t$  is a vector of variables to be inferred,  $w_t$  is a vector of parameters and  $x_t$  is the features of  $x \in \mathcal{X}$  used by the classifier  $t$ . Vector  $x_t$  is normally a collection of local features of input  $x$  for each variable in the vector  $y_t$ . For simplicity of notation we will assume that exactly one variable is associated with every sub-problem for every input element  $x$  (i.e.,  $y_t$  is a scalar), though methods discussed in this section apply to the general case and in the experimental evaluation we also consider a more complex scenario. Global inference is used to maintain structural consistency of the outputs:

$$\hat{y} = \arg \max_y (c(y) + \sum_{t=1}^T \phi(y_t, x_t, w_t)),$$

<sup>1</sup>The CoNLL 2008 shared task examined joint learning of syntactic and semantic structures (Surdeanu et al., 2008), hoping to use joint learning to improve upon the standard pipeline approach. Results were mixed with the best systems not using joint learning, but other competitive systems getting improvement from using joint learning.

where  $c$  is a predefined function which equals  $-\infty$  if output  $y$  is illegal and 0 otherwise, possibly depending on  $x$ . In the LO and L+I scenarios, each classifier is trained independently, whereas with IBT all  $T$  classifiers are trained jointly to optimize a global objective. The structured Perceptron algorithm (Collins, 2002) is one example of an IBT training algorithm.

```

for  $t = 1 : T$  do
  for  $k \notin \{s_1, \dots, s_{t-1}\}$  do
    (1) Estimate  $w_k$  of the model
    
$$P(y_k | x_{<t}, w_{<t}, w_k) \propto e^{\phi(y_k, x_k, w_k)} \sum_{y_{<t}} e^{c(y_{<t}, y_k) + \sum_{j=1}^{t-1} \phi(y_{s_j}, x_{s_j}, w_{s_j})}$$

    on  $D^t = \{(x_{<t}^{(i)}, x_k^{(i)}, y_k^{(i)})_{i=1}^N\}$ 
    (2) Compute expected error estimate  $\epsilon_k$ 
  end for
   $s_t = \arg \min_k \epsilon_k$ 
end for
    
```

Figure 1: Sequential learning algorithm

As argued above, for many real tasks different sub-problems might have different properties; whereas some of them can be successfully learned in isolation, other probably require some form of joint learning to capture more complex interdependencies. With sequential learning, we are attempting to construct an ordering  $s_1, \dots, s_T$  of these sub-problems such as that every sub-problem  $s_t$  can be successfully learned by jointly learning sub-problems  $s_1, \dots, s_t$ . Moreover, when considering a problem  $s_t$  we do not attempt to retrain parameters for the preceding classifiers,  $w_{s_1}, \dots, w_{s_{t-1}}$ , focusing only on estimating parameter vector  $w_{s_t}$ . Therefore the previously learned classifiers guide learning by decreasing the score for local predictions  $y_{s_t}$  if  $y_{s_t}$  can only be composed with unlikely outputs  $y_{s_1}, \dots, y_{s_{t-1}}$  and increasing in the opposite case. In order to find a preferred ordering  $s_1, \dots, s_T$ , on every step  $t$  we train each of the remaining classifiers  $k \notin \{s_1, \dots, s_{t-1}\}$  jointly with all the classifiers  $s_1, \dots, s_{t-1}$  to predict  $y_k$  given  $x_{s_1}, \dots, x_{s_{t-1}}$  and  $x_k$ . Then, using the test error estimators appropriate for the learning method, we select the problem  $s_t$  which is expected to have the most reliable predictions. This algorithm is presented formally in Figure 1. Note that in Figure 1, we assume that we have a log-linear probability model and, therefore, can marginalize over predictions of the previous classifiers  $y_{s_1}, \dots, y_{s_{t-1}}$ . Also, we use  $x_{<t}$  and  $y_{<t}$  to denote sequences of variables  $(x_{s_1} \dots x_{s_{t-1}})$  and  $(y_{s_1} \dots y_{s_{t-1}})$  respectively. The constraint indicator function  $c$ , when applied to an assignment of a subset of variables, equals zero if there exists such an assignment in coordination with other output variables such that the composed output is legal. Otherwise,  $c$  equals negative infinity.

In practice, marginalization over the set of variables may not be feasible either for computational reasons or because the learner does not automatically induce a probability estimator. E.g., even though a linear classifier learned with Perceptron or SVM can be used to estimate the probability (Platt, 1999), the factor regulating sharpness of the distribution needs to be selected by hand and may even require adjustment during the course of learning. Therefore, in this case we propose replacing marginalization by selecting the score of the most likely sequence  $y_{s_1}, \dots, y_{s_{t-1}}$  which, when composed with the considered  $y_k$ , satisfies the constraints. The algorithm for the Perceptron training presented in Figure 2 is an example instantiation of the conceptual algorithm in Figure 1. For simplicity, we assume that only one pass over the training is done when training each vector  $w_k$  although more passes would normally be required. As motivated by the Novikoff Theorem (Novikoff, 1963), we use the number of Perceptron updates as a criterion when deciding which of the candidate classifiers is more reliable.<sup>2</sup>

```

for  $t = 1 : T$  do
  for  $k \notin \{s_1, \dots, s_{t-1}\}$  do
     $\epsilon_k = 0, w_k = 0$ 
    for  $i = 1 : N$  do
       $\hat{y}_k = \arg \max_{y_k} w_k^T f(x_k^{(i)}, y_k)$ 
       $+ \max_{y_{<t}} c(y_{<t}, y_k) + \sum_{j=1}^{t-1} w_{s_j}^T f(x_{s_j}^{(i)}, y_{s_j})$ 
      if  $\hat{y}_k \neq y_k^{(i)}$  then
         $w_k = w_k + f(x_k^{(i)}, y_k^{(i)}) - f(x_k^{(i)}, \hat{y}_k)$ 
         $\epsilon_k = \epsilon_k + 1$  { counting updates }
      end if
    end for
     $s_t = \arg \min_k \epsilon_k$ 
  end for
end for
    
```

Figure 2: Sequential learning with Perceptron

### 3 ANALYSIS

In this section we provide a theoretical motivation for the SL algorithm. First, we demonstrate the use of sequential learning results in better covering number bounds (Zhang, 2002) on the expected classification error than that of LO and IBT if sub-problems are not locally separable but can be made separable with the help of the preceding classifiers in the learning sequence. Secondly, we demonstrate that enforcing global coherence of the SL classifier is statistically motivated and preferable to L+I even when sub-problems are equally complex to learn.

<sup>2</sup>If the number of variable instances differs between sub-problems, the number of Perceptron updates should be scaled by the proportion of variable instances appearing in the training set for the considered classifier  $k$ .

### 3.1 ERROR BOUNDS

This section focuses on the per-variable error as it is the most commonly used method to measure accuracy of a classifier for structured predictions. Also, predicting a partially correct structure is generally preferable to predicting a completely incorrect structure.<sup>3</sup> Therefore, for SL and LO methods we bound the expected error of individual classifiers and for IBT we bound the per-variable error of the entire predicted sequence. The goal is to demonstrate that under the conditions stated above, the average of the upper bounds for SL over  $t$  is lower than the average of the bounds for LO and also lower than the bound for IBT. The key intuition is that the LO method uses a low capacity function class for every sub-problem, and might not be sufficiently powerful for some of these sub-problems. With IBT, we use a powerful function class for all the sub-problems although some of them are likely to be easier and not requiring this power. Conversely, in the SL approach we try to use simpler function classes for simpler problems and more powerful classes for more complex problems, therefore balancing their capacity and their ability to learn from the training data.

Without loss of generality, we assume that the discovered order of classification agrees with the original ordering of the sub-problems, i.e.,  $s_t = t$ . We also consider only linear models and the modification of the SL algorithm described above, where we add to the local score,  $w_t^T f(x_t^{(i)}, y_t)$ , the sum of the scores of the most likely sequence  $y_1, \dots, y_{t-1}$ , as in Figure 2. We limit our consideration to each sub-problem being a binary classification task,  $y_t \in \{0, 1\}$ , although the results trivially generalize to multi-class and multi-variable cases.

As the LO protocol involves independent binary classifiers for each sub-problem, we can simply restate the result described in (Zhang, 2002):

**Theorem 1.** ((Zhang, 2002), Th. 6) *If the data is bounded  $\|x_t\| \leq a_t$  then there is a constant  $C$  such that with probability  $1 - \eta$  over  $n > 1$  random samples, the classification error of  $w_t^{LO}$  is bounded*

$$\text{err}(w_t^{LO}) \leq \frac{L^{LO}(w_t^{LO})}{n} + \sqrt{\frac{C}{n} \left( \ln \frac{1}{\eta} + a_t^2 \|w_t^{LO}\|_2^2 \ln n \right)},$$

where  $L^{LO}(w_t^{LO}) = |\{i : (w_t^{LO})^T x_t^{(i)} (2y_t^{(i)} - 1) < 1\}|$  is the number of samples with the margin less than 1.

We cannot directly apply results of (Zhang, 2002) to structured prediction with constrained classifiers (IBT), but we can use the technique similar to that considered in (Collins, 2002; Taskar et al., 2004); the

<sup>3</sup>The average per-label loss is equal to the expected Hamming error divided by the length of the sequence.

key point is to demonstrate that the set of predefined constraints do not influence generalization performance.

**Theorem 2.** *If the data is bounded  $\|x_t\|_2 \leq a_t$  then there is a constant  $C$  such that with probability  $1 - \eta$  over  $n > 1$  random samples, the per-variable error of the constrained model parametrized by  $w^{IBT}$  is bounded*

$$\text{err}(w^{IBT}) \leq \frac{L^{IBT}(w^{IBT})}{n} + \sqrt{\frac{C}{n} \left( \ln \frac{1}{\eta} + \sum_{t=1}^T a_t^2 \|w_t^{IBT}\|_2^2 \ln n \right)},$$

where  $L^{IBT}(w^{IBT})$  equals to

$$\sum_{i=1}^n \sup_{v: |v(y) - \varphi(y, x^{(i)}, w^{IBT})| \leq d_H(y, y^{(i)})} \frac{d_H(\arg \max_y v(y), y^{(i)})}{T}$$

is the per-variable margin loss and the Hamming distance  $d_H(y, y')$  is the number of mismatching variables, the function  $\varphi(y, x, w^{IBT}) = c(y, x) + \sum_{t=1}^T (w_t^{IBT})^T f(x_t, y_t)$  is the global score for the output sequence  $y$ ,  $v: \mathcal{Y} \rightarrow \mathcal{R}$  is a distorted scoring function  $\varphi(y, x^{(i)}, w^{IBT})$ .

*Proof.* The space constraints do not allow us to present the proof in detail and we explain only the proof strategy. We bound the multi-error-level covering number  $\mathcal{N}_\infty^{mul}(\mathcal{F}_c, \epsilon, n)$  for the considered constrained function class  $\mathcal{F}_c = \{\varphi(y, x, w), \forall w, |w_t^{IBT}| < b_t\}$  (see (Taskar, 2004), def. A.1.9 for the multi-error-level covering number definition) by the product of the covering numbers for linear function with bounded norms of parameter vectors  $\prod_{t=1}^T \mathcal{N}_\infty(\mathcal{F}_L, \epsilon, n)$ . This is done using the technique similar to (Collins, 2002; Taskar, 2004). Using the bound on  $\mathcal{N}_\infty(\mathcal{F}_L, \epsilon, n)$  (Th. 4, (Zhang, 2002)), we can substitute the  $\mathcal{N}_\infty^{mul}(\mathcal{F}_c, \epsilon, n)$  in theorem A.1.12 (Taskar, 2004). Then the statement of this theorem follows using the same argument as outlined in (Taskar, 2004).  $\square$

For the SL method, we state the following result which is similar to Theorem 2, but individually bounds the expected error for every sub-problem and not the average error over them:

**Theorem 3.** *If the data is bounded  $\|x_t\|_2 \leq a_t$  then there is a constant  $C$  such that with probability  $1 - \eta$  over  $n > 1$  random samples, the classification error of the SL classifier for problem  $t$  is bounded*

$$\text{err}(w_{\leq t}^{SL}) \leq \frac{L^{SL}(w_{\leq t}^{SL})}{n} + \sqrt{\frac{C}{n} \left( \ln \frac{1}{\eta} + \sum_{t'=1}^t a_{t'}^2 \|w_{t'}^{SL}\|_2^2 \ln n \right)},$$

where the margin loss  $L^{SL}(w_{\leq t}^{SL})$

$$\sum_{i=1}^n \sup_{v: |v(y) - \varphi_{\leq t}(y, x^{(i)}, w^{IBT})| \leq d_H(y, y^{(i)})} I[(\arg \max v(y))_t = y_t^{(i)}]$$

is the number of training instances for which the distorted scoring function predicts the wrong label at position  $t$ ,  $\varphi_{\leq t}(y, x, w^{SL}) = c(y, x) + \sum_{t'=1}^t (w_{t'}^{SL})^T f(x_{t'}, y_{t'})$  is the local score  $(w_t^{SL})^T f(x_t, y_t)$  corrected by the scores given by the classifiers for the previous sub-problems.

*Proof.* The proof is similar to the proof of Theorem 2. The crucial difference is that instead of using the average per-variable loss we use the loss which penalizes errors only on the component  $y_t$ . Note, that the multi-error-level metric and the multi-error-level covering number are still defined in terms of the Hamming distance.<sup>4</sup>  $\square$

The theorem suggests instead of fixing the weights learned on the previous iteration, we may want to train the entire vector  $w_{\leq t}^{SL}$ . However, it would not only increase computational expense but also complicate joint inference (see section 3.2).

Comparing the bound for the SL method (Theorem 3) and the bound for the LO method (Theorem 1), we can observe that there exist tasks which are difficult to solve locally but easy to solve in a sequence, i.e., for every  $t > 1$  we can achieve the same margin loss term (first term) in LO only for which  $a_t^2 \|w_t^{LO}\|_2^2 > \sum_{t'=1}^t \|w_{t'}^{SL}\|_2^2$ .<sup>5</sup> This situation might happen, e.g., when there are no predictive features for decision  $y_t^{(i)}$  on many samples  $x^{(i)}$  whereas features for previous sub-problems  $x_{t'}^{(i)}$ ,  $t' < t$  are reliable predictors and output  $y_{t'}^{(i)}$  constrains assignments of  $y_t^{(i)}$ . Therefore, for such tasks we expect better accuracy with the SL method than with the use of the LO approach.

Similarly, if every sub-problem is as easy to learn by SL method as by the global scoring function used in IBT, i.e., formally  $L^{IBT}(w^{IBT}) \geq \frac{\sum_{t=1}^T L^{SL}(w_{\leq t}^{SL})}{T}$  for  $\|w_t^{SL}\|_2 \leq \|w_t^{IBT}\|_2$ , then the bound for the IBT method (Theorem 2) is greater than the average over  $T$  bounds for the SL method (Theorem 3). This is the case when features predictive of  $y_t$  in windows  $x_{t'}$

<sup>4</sup>This corresponds to the analysis in (McAllester, 2007), where arbitrary loss functions are bounded using PAC-Bayesian bounds based upon the Hamming distance.

<sup>5</sup>Here we assume that  $C$  is equal for both problems. Formally, we should explicitly define the  $C$  coefficient for each bound to perform comparison. However, the behavior of the coefficients  $C$  should be expected similar for every bound. Also, if we unwrap  $C$  presenting lower-order terms this would not give any additional insight.

$t' > t$  are noisy or the subsequent sub-problems are much harder than sub-problem  $t$  and do not help to increase the margin. This implies that the bounds will predict that the SL method is expected to outperform the IBT method under such conditions.

We demonstrated that for tasks which are decomposable into sub-problems easily solvable in a sequence but not solvable in isolation that the SL method is expected to perform better than both LO and IBT methods. The difference in accuracy with respect to IBT is likely to be especially large when only small amount of training data is available because in this case stronger regularization is required to achieve competitive accuracy and any difference in the regularization term will have a larger effect.

### 3.2 PROBABILISTIC ARGUMENT

If we ignore any prior distribution over the parameter vector  $w_t$  we can assume that the step (1) in the SL algorithm in Figure 1 consists of maximization of the following conditional log-likelihood

$$\sum_{x_{\leq t}, y_t} P_D(x_{\leq t}) P_D(y_t | x_{\leq t}) \ln \frac{e^{\phi(y_t, x_t, w_t)} \hat{P}(y_t | x_{\leq t}, w_{< t})}{\sum_{y'_t} e^{\phi(y'_t, x_t, w_t)} \hat{P}(y'_t | x_{\leq t}, w_{< t})},$$

where  $P_D$  is the empirical distribution and

$$\hat{P}(y_t | x_{\leq t}, w_{< t}) \propto \sum_{y_{< t}} e^{c(y_t, y_{< t}) + \sum_{t'=1}^{t-1} \phi(y_{t'}, x_{t'}, w_{t'})}$$

can be regarded as the probability estimate for  $y_t$  based on the previous  $w_{t'}$  and  $x_{< t}$ . This estimate is proportional to the total probability which the model associates with all the assignments to the sequence of variables  $y_{< t}$  compatible with the proposed assignment to  $y_t$ . Therefore during this learning step we try to find the parameter vector  $w_t$  which corrects this estimate  $\hat{P}(y_t | x_{\leq t}, w_{< t})$  to get the best possible estimate of  $P_D(y_t | x_{\leq t})$ . When we compute estimates  $\hat{P}$ , we sum over all the possible sequences and it follows that

$$\hat{P}(y|x, w) = \frac{e^{c(y) + \sum_{t=1}^T \phi(y_t, x_t, w_t)}}{\sum_{y'} e^{c(y') + \sum_{t=1}^T \phi(y'_t, x_t, w_t)}}$$

is the estimate of the conditional probability  $P(y|x)$ . Consequently, we can use this estimate to find the most likely structure according to the model trained with the SL method. Therefore, enforcing constraints at test time is appropriate for the SL method.

It is well known that the accuracy of the LO method can also be improved by using global inference at test time (L+I) (Punyakankok et al., 2005). This approach normally increases both the per-variable accuracy and leads to more coherent structures. However, it lacks a good probabilistic motivation. LO is

trained to be an estimator of the local probability  $P(y_t|x_t)$ :  $\hat{P}(y_t|x_t, w_t) \propto e^{\phi(y_t, x_t, w_t)}$ , and if we substitute these estimates in the global model, as done in L+I, the marginal distributions  $\hat{P}(y_t|x_t)$  of the resulting global model will differ from the empirical distribution  $P_D(y_t|x_t)$ .

This drawback is not only of theoretical interest, but results in wrong predictions on a rather simple task. As an example, consider the following illustrative problem. The goal is to predict a binary output sequence of 3 variables given 3 variable input sequence  $x = (x_1, x_2, x_3) \in \{0, 1\}^3$  such that the input distribution is uniform. The constraints dictate that only two output sequences are possible:  $(0, 0, 0)$  and  $(1, 1, 1)$ , the second sequence is generated if all the input elements are equal to 1, otherwise the former is selected. We consider prediction of  $y_t$  given the single binary variable  $x_t$  as each of 3 sub-problems. The LO method estimated on an infinite amount of training data will predict  $(0, 0, 0)$  on any input. Even enforcing constraints would not help because the L+I model will overestimate the probability  $P(y_2 = 1|x_1 = 1, x_2 = 1)$  and  $P(y_3 = 1|x = (1, 1, 1))$ . Of critical note is that this would not happen with the SL learning method. The reason is clear if we recollect that  $w_2$  is selected to correct the estimate  $\hat{P}(y_t|x_{<t}, w_{<t})$ . This correction eliminates the underestimation mentioned above and leads to the correct classification on all inputs. Even though this problem is artificial and in solving such a task, we would not normally want to decompose the problem according to the method described, this illustrates general problems with features predictive of  $y_t$  not appearing in  $x_t$ , but appearing in  $x_{t'}$ ,  $t' \neq t$ . Also, this demonstrates that sequential learning helps even when the sub-problems are equally difficult to learn.

## 4 EMPIRICAL EVALUATION

In this section, we present experiments on artificial and real data. For artificial experiments, we generate data such that complexity of individual sub-problems varies. For experiments on real data, we consider the entity and relation recognition problem.

### 4.1 ARTIFICIAL EXPERIMENTS

In these experiments, we generally follow the set-up of the synthetic experiments in (Punyakano et al., 2005). Each example is a set of  $T$  points in  $d$ -dimensional real space with its labels being a sequence of binary variables,  $y \in \{0, 1\}^T$ , labeled according to:

$$y = \arg \min_y c(y) + \sum_{t=1}^T (2y_t - 1)w_t^T x_t,$$

where  $c(y)$  is equal to 0 on all the outputs except to the random number of illegal vectors  $y$ , where it equals negative infinity. We sample components of each parameter vector  $w_t$  uniformly in the range  $[-2^{T-t}, 2^{T-t}]$  and the input distribution  $x_t$  is uniform over  $[-1, +1]^d$ . It follows that the components of  $x_t$  with smaller  $t$  are better features of  $y_t$  than components of  $x_t$  with greater  $t$ . Therefore, we would expect that sub-problems with smaller  $t$  can be successfully learned in isolation whereas sequential learning should help with other sub-problems. This corresponds to what we would expect in real applications; some of the problems are easier to learn than the others. Note, that we do not explicitly enforce existence of a sub-problem ordering such that the SL method is able to classify elements with a large margin.

Per-label accuracy curves for IBT, L+I and IBT are presented in Figures 3 and 4 respectively. We do not present curves for LO, but, as expected, the accuracy of LO was significantly lower than for L+I (Punyakano et al., 2005), e.g., even on smaller datasets of 100 and 200 instances LO was below L+I by 2% and 3% in average, respectively.

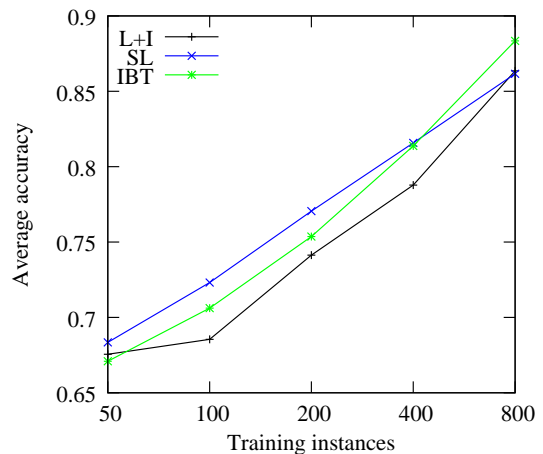


Figure 3: Comparison of learning strategies with different training dataset size

For the SL model, we used joint inference as discussed in section 3.2. To obtain both curves we considered sequences of 5 elements where the input dimensionality  $d$  for each problem was set to 100. The results are averaged over 10 different problem instances generated as explained above. The learning algorithm used for all experiments is the averaged Perceptron (Freund and Schapire, 1998). The SL protocol, except for preserving the averaged vector and running the training iteration more than once,<sup>6</sup> was identical to the one shown in Figure 2. The size of the testing set was equal

<sup>6</sup>The number of iterations was tuned independently for each learning strategy on an additional problem instance.

to 10,000 sequences. Figure 3 illustrates how accuracy changes with increase of the dataset size, the number of illegal output sequences was randomly chosen in each of 10 problem instances. For experiments in Figure 4, we fix the dataset size and vary the number of illegal output sequences.

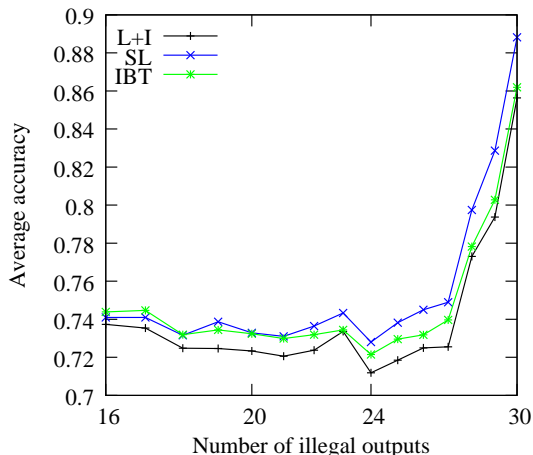


Figure 4: Comparison of learning strategies with different number of illegal outputs

As shown in Figure 3, the SL method was better than the alternative approaches when the number of training instances less or equal to 400. This improvement was consistent across different problem instances: the SL methods was statistically significantly better on 9, 8 and 9 problems out of 10 when using 50, 100 and 200 training instances, respectively. This agrees with theoretical analysis presented in section 3.1. From Figure 4, we can conclude that the SL method was better than the alternatives with any number of illegal outputs and the improvement increases with the number of illegal outputs. For the smaller number of illegal outputs, results of SL, IBT and L+I are virtually the same. This is not surprising, because the SL method (as well as IBT) can improve accuracy only if individual classifiers are indeed constrained. Even a single real constraint (e.g. the constraint for the SRL problem that an object and a subject of each verb predicate should lie on opposite sides of the verb) normally reduces the number of legal outputs by at least a factor of two, so this is not a strong requirement.

## 4.2 REAL-WORLD DATA

The second set of experiments deals with the problem of simultaneously recognizing entities and relations (Roth and Yih, 2004). We use the same datasets as in (Roth and Yih, 2004), which defines 3 types of entities (*person*, *location* and *organization*) and 5 types of binary relations (*located\_in*, *work\_for*, *org\_based\_in*,

*live\_in*, *kill*). The goal is to decide on the entity type and predict for each pair of entities in a sentence whether they participate in a relation and, if so, select its type. Detailed description of the dataset and the task can be found in (Roth and Yih, 2004). We split the dataset into a training set (815 sentences, 2632 entities, 1206 relations), a development set (333 sentences, 1070 entities and 426 relations) and a testing set (287 sentences, 943 entities and 432 relations). As features for entity classifiers we use: words inside the entity (unigrams), number of words in the entity, 2 preceding and 2 subsequent words, and also we test whether a word of the entity belongs to the list of common personal names and large cities as described in (Roth and Yih, 2004). Similarly, in a relation classifier we use the bag-of-words representation of the sentence, words in the potential arguments, and distance between them in the sentence. We define only a simple set of constraints which enforce for each relation consistency between its type and types of its arguments, e.g., for the relation *live\_in* the first argument should be *person* and the second one is *location*. The learning algorithm we use is the averaged Perceptron with thick separation. All the parameters were tuned on the development set and a single model trained with each strategy was evaluated on the final testing set.

We consider each multiclass classification problem as a set of binary classification tasks (one-vs-all) and the SL method attempts to select a preferred order for constrained learning of these 14 classification tasks (note that two classifiers corresponds to each ordering of arguments in a relation, one per each ordering of its arguments and there exists also an additional classifier which predicts that a pair of entities is not related).

Results for the SL, IBT and L+I strategies are shown in Table 1. We observe that the SL strategy outperforms both alternatives on relations. Results on entities are similar for all the strategies, which is not surprising as the entity predictions are more reliable and less affected by the constraints. Not surprisingly, the entity classifiers were learned before the relation classifiers in the preferred order were automatically discovered by the SL method. Importantly, additional experiments on the development set demonstrate that the discovered ordering of relation learning is indeed meaningful; we observe that random perturbations of this order seriously affect the resulting accuracy, by up to 8% in the  $F_1$  score for relations. Also, despite training in our experiments on a smaller training set and using a simpler feature set we achieve the accuracy competitive with the one reported in (Roth and Yih, 2004): around 86% on entities and around 55% on relations with their best approach.

Table 1: Scores on the joint named entity and relation recognition problem.

	RELATIONS			Entities Accuracy	Average F <sub>1</sub>
	P	R	F <sub>1</sub>		
SL	73.5	35.9	48.2	89.5	68.9
IBT	81.2	31.0	44.8	89.2	67.0
L+I	74.5	31.7	44.5	89.2	66.8

## 5 CONCLUSIONS

In this paper, we propose a novel strategy for learning classifiers for problems with structured output. The proposed method selects a preferred order of training of classifiers for individual sub-problems and trains them in this order using previously learned classifiers to guide learning of the subsequent ones by enforcing coherence constraints between their predictions. The proposed approach is theoretically motivated and demonstrated to outperform both joint and independent learning strategies on synthetic data and on a real natural language processing task. The approach has a potentially large number of applications in various areas where structured prediction problems are considered. The method can be easily generalized to use ‘weak’ constraints more common in such areas as computer vision where it is difficult to define constraints that hold for all the data instances.

### Acknowledgements

The authors thank Alex Klementiev and the reviewers for helpful comments regarding this work. This work is partly supported by NSF grant SoD-HCER-0613885, Swiss NSF scholarship PBGE22-119276, and DARPA funding under the Bootstrap Learning Program.

### References

- Bunescu, R. C. (2008). Learning with probabilistic features for improved pipeline models. In *Proc. of Conf. on Empirical Methods in Natural Language Processing*.
- Carreras, X. and Màrquez, L. (2004a). Introduction to the CoNLL-2004 shared tasks: Semantic role labeling. In *Proceedings of CoNLL-2004*, pages 89–97.
- Carreras, X. and Màrquez, L. (2004b). Online learning via global feedback for phrase recognition. In *Proceedings of NIPS-2003*.
- Collins, M. (2002). Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms. In *Proc. of Conf. on Empirical Methods in Natural Language Processing*.
- Collobert, R. and Weston, J. (2008). A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proc. ICML*.
- Finkel, J. R., Manning, C. D., and Ng, A. Y. (2006). Solving the problem of cascading errors: Approximate bayesian inference for linguistic annotation pipelines. In *In Proc. of Conference on Empirical Methods in Natural Language Processing*.
- Freund, Y. and Schapire, R. (1998). Large margin classification using the Perceptron algorithm. In *Proc. COLT*.
- Henderson, J., Merlo, P., Musillo, G., and Titov, I. (2008). A latent variable model of synchronous parsing for syntactic and semantic dependencies. In *Proc. CoNLL-2008 Shared Task*.
- McAllester, D. (2007). Generalization bounds and consistency for structured labeling. In Bakir, G., Hofmann, T., Scholkopf, B., Smola, A., Taskar, B., , and Vishwanathan, S. V. N., editors, *Predicting Structured Data*. MIT Press.
- Novikoff, A. (1963). On convergence proofs for perceptrons. In *Proceeding of the Symposium on the Mathematical Theory of Automata*, volume 12.
- Platt, J. C. (1999). Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In Smola, A., Bartlett, P., Scholkopf, B., and Schuurmans, D., editors, *Advances in Large Margin Classifiers*, pages 61–74. MIT Press.
- Punyakanok, V., Roth, D., and Yih, W. (2008). The importance of syntactic parsing and inference in semantic role labeling. *Computational Linguistics*, 34(2).
- Punyakanok, V., Roth, D., Yih, W., and Zimak, D. (2005). Learning and inference over constrained output. In *Proc. IJCAI*.
- Roth, D. and Yih, W. (2004). A linear programming formulation for global inference in natural language tasks. In *Proc. of the Annual Conference on Computational Natural Language Learning (CoNLL)*.
- Surdeanu, M., Johansson, R., Meyers, A., Màrquez, L., and Nivre, J. (2008). The CoNLL-2008 shared task on joint parsing of syntactic and semantic dependencies. In *Proceedings of the 12th Conference on Computational Natural Language Learning (CoNLL-2008)*.
- Taskar, B. (2004). *Learning Structured Prediction Models: A Large Margin Approach*. PhD thesis, Stanford University.
- Taskar, B., Guestrin, C., and Koller, D. (2004). Max-margin markov networks. In *Proc. NIPS 16*.
- Zhang, T. (2002). Covering number bounds of certain regularized linear function classes. *J. Mach. Learn. Res.*, 2:527–550.