Learning from Partially Annotated Sequences

Eraldo R. Fernandes[†] and Ulf Brefeld[‡]

[†] Pontifícia Universidade Católica do Rio de Janeiro, Brazil
efernandes@inf.puc-rio.br

[‡] Yahoo! Research, Barcelona, Spain
brefeld@yahoo-inc.com

Abstract. We study sequential prediction models in cases where only fragments of the sequences are annotated with the ground-truth. The task does not match the standard semi-supervised setting and is highly relevant in areas such as natural language processing, where completely labeled instances are expensive and require editorial data. We propose to generalize the semi-supervised setting and devise a simple transductive loss-augmented perceptron to learn from inexpensive partially annotated sequences that could for instance be provided by laymen, the wisdom of the crowd, or even automatically. Experiments on mono- and cross-lingual named entity recognition tasks with automatically generated partially annotated sentences from Wikipedia demonstrate the effectiveness of the proposed approach. Our results show that learning from partially labeled data is never worse than standard supervised and semi-supervised approaches trained on data with the same ratio of labeled and unlabeled tokens.

1 Introduction

The problem of labeling, annotating, and segmenting observation sequences arises in many applications across various areas such as natural language processing, information retrieval, and computational biology; exemplary applications include named entity recognition, information extraction, and protein secondary structure prediction.

Traditionally, sequence models such as hidden Markov models [26, 14] and variants thereof have been applied to label sequence learning [9] tasks. Learning procedures for generative models adjust the parameters such that the joint likelihood of training observations and label sequences is maximized. By contrast, from an application point of view, the true benefit of a label sequence predictor corresponds to its ability to find the correct label sequence given an observation sequence. Thus, many variants of discriminative sequence models have been explored, including maximum entropy Markov models [20], perceptron re-ranking [7, 8], conditional random fields [16, 17], structural support vector machines [2, 34], and max-margin Markov models [32].

Learning discriminative sequential prediction models requires ground-truth annotations and compiling a corpus that allows for state-of-the-art performance

Table 1. Different interpretations for "I saw her duck under the table" [15].

I saw [NP her] [VP duck under the table].	→ She ducked under the table.
I [VP saw [NP her duck] [PP under the table]].	→ The seeing is done under the table.
I saw [NP her duck [PP under the table]].	→ The duck is under the table.

on a novel task is not only financially expensive but also in terms of the time it takes to manually annotate the observations. Frequently, annotating data with ground-truth cannot be left to laymen due to the complexity of the domain. Instead trained editors need to deal with the pitfalls of the domain at hand such as morphological, grammatical, and word sense disambiguation when dealing with natural language. Table 1 shows an ambiguous sentence with three different interpretations that cannot be resolved without additional context.

Semi-supervised learning approaches [6] aim at reducing the need for large annotated corpora by incorporating unlabeled examples in the optimization; to deal with the unlabeled data one assumes that the data meets certain criteria. A common assumption exploits that similar examples are likely to have similar labelings. This so-called cluster assumption can be incorporated into semi-supervised structural prediction models by means of Laplacian priors [17, 1], entropy-based criterions [18], transduction [37], or SDP relaxations [35]. Although these methods have been shown to improve over the performance of purely supervised structured baselines, they do not reduce the amount of required labeled examples significantly as it is sometimes the case for univariate semi-supervised learning. One of the key reasons is the variety and number of possible annotations for the same observation sequence; there are $|\Sigma|^T$ different annotations for a sequence of length T with tag set Σ and many of them are similar in the sense that they differ only in a few labels.

In this paper, we extend the semi-supervised learning setting and study learning from partially annotated data. That is, in our setting only some of the observed tokens are annotated with the ground-truth while the rest of the sequence is unlabeled. The rational is as follows: If the target concept can be learned from partially labeled sequences, annotation costs can be significantly reduced. Large parts of an unlabeled corpus could for instance be labeled by laypeople using the wisdom of the crowd via platforms like MechanicalTurk[1] or CrowdFlower[2]. Prospective workers could be asked to only annotate those parts of a sequence they feel confident about and if two or more workers disagree on the labeling of a sentence, the mismatches are simply ignored in the model generation. In the example in Table 1, one could label the invariant token *her=NP* and leave the ambiguous parts of the sentence unlabeled.

We devise a straight-forward transductive extension of the structured loss-augmented perceptron that allows to include partially labeled sequences in the training process. This extension contains the supervised and the semi-supervised structured perceptron as special cases. To demonstrate that we can learn from

[1] https://www.mturk.com
[2] http://www.crowdflower.com

inexpensive data, we evaluate our method on named entity recognition tasks. We show in a controlled experiment that learning with partially labeled data is always on par or better than standard supervised and semi-supervised baselines (trained on data with the same ratio of labeled and unlabeled tokens). Moreover, we show that mono- and cross-lingual named entity recognition can significantly be improved by using additional corpora that are automatically extracted from Wikipedia[3] at factually no costs at all.

The remainder is structured as follows. Section 2 reviews related work and Section 3 introduces label sequence learning. We devise the transductive perceptron in Section 4. Section 5 reports on the empirical results and Section 6 concludes.

2 Related Work

Learning from partially annotated sequences has been studied by [30] who extend HMMs to explicitly exclude states for some observations in the estimation of the models. [22] propose to incorporate domain-specific ontologies into HMMs to provide labels for the unannotated parts of the sequences, [10] cast learning an HMM for partially labeled data into a large-margin framework and [33] present an extension of maximum entropy Markov models (MEMMs) and conditional random fields (CRFs). The latent-SVM [36] allows for the incorporation of latent variables in the underlying graphical structure; the additional variables implicitly act as indicator variables and conditioning on their actual value eases model adaptation because it serves as an internal clustering.

The generalized perceptron for structured output spaces is introduced by [7, 8]. Altun et al. [2] leverage this approach to support vector machines and explore label sequence learning tasks with implicit 0/1 loss. McAllester et al. [19] propose to incorporate loss functions into the learning process of perceptron-like algorithms. Transductive approaches for semi-supervised structured learning are for instance studied in [17, 1, 35, 18, 37], where the latter is the closest to our approach as the authors study transductive support vector machines with completely labeled and unlabeled examples.

Generating fully annotated corpora from Wikipedia has been studied by [24, 27, 21]. While [21] focus on English and exploit the semi-structured content of the info-boxes, [24] and [27] propose heuristics to assign tags to Wikipedia entries by manually defined patterns.

3 Preliminaries

The task in label sequence learning [9] is to find a mapping from a sequential input $\boldsymbol{x} = \langle x_1, \ldots, x_T \rangle$ to a sequential output $\boldsymbol{y} = \langle y_1, \ldots, y_T \rangle$, where $y_t \in \Sigma$; i.e., each element of $\boldsymbol{x}$ is annotated with an element of the output alphabet Σ which denotes the set of tags. We denote the set of all possible labelings of $\boldsymbol{x}$ by $\mathcal{Y}(\boldsymbol{x})$.

[3] http://www.wikipedia.com

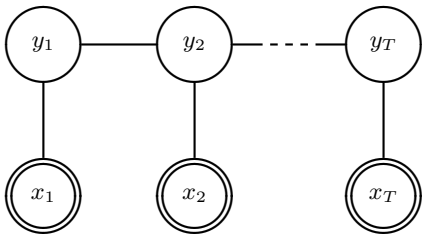

Fig. 1. A Markov random field for label sequence learning. The x_t denote observations and the y_i their corresponding hidden class variables.

The sequential learning task can be modeled in a natural way by a Markov random field where we have edges between neighboring labels and between label-observation pairs, see Figure 1. The conditional density $p(\boldsymbol{y}|\boldsymbol{x})$ factorizes across the cliques [12] and different feature maps can be assigned to the different types of cliques, ϕ_{trans} for transitions and ϕ_{obs} for emissions [2, 16]. Finally, interdependencies between $\boldsymbol{x}$ and $\boldsymbol{y}$ are captured by an aggregated joint feature map $\phi : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}^d$,

$$\phi(\boldsymbol{x}, \boldsymbol{y}) = \left(\sum_{t=2}^{T} \phi_{trans}(\boldsymbol{x}, y_{t-1}, y_t)^\top, \sum_{t=1}^{T} \phi_{obs}(\boldsymbol{x}, y_t)^\top \right)^\top$$

which gives rise to log-linear models of the form

$$g(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{w}) = \boldsymbol{w}^\top \phi(\boldsymbol{x}, \boldsymbol{y}).$$

The feature map exhibits a first-order Markov property and as a result, decoding can be performed by a Viterbi algorithm [11, 31] in $\mathcal{O}(T|\Sigma|^2)$ so that, once optimal parameters $\boldsymbol{w}^*$ have been found, these are used as plug-in estimates to compute the prediction for a new and unseen sequence $\boldsymbol{x}'$,

$$\hat{\boldsymbol{y}} = f(\boldsymbol{x}'; \boldsymbol{w}^*) = \operatorname*{argmax}_{\tilde{\boldsymbol{y}} \in \mathcal{Y}(\boldsymbol{x}')} g(\boldsymbol{x}', \tilde{\boldsymbol{y}}; \boldsymbol{w}^*). \tag{1}$$

The optimal function $f(\cdot; \boldsymbol{w}^*)$ minimizes the expected risk $\mathbb{E}\left[\ell(\boldsymbol{y}, f(\boldsymbol{x}; \boldsymbol{w}^*)\right]$ where ℓ is a task-dependent, structural loss function. In the remainder, we will focus on the 0/1- and the Hamming loss to compute the quality of predictions,

$$\ell_{0/1}(\boldsymbol{y}, \tilde{\boldsymbol{y}}) = 1_{[\boldsymbol{y} \neq \tilde{\boldsymbol{y}}]}; \qquad \ell_h(\boldsymbol{y}, \tilde{\boldsymbol{y}}) = \sum_{t=1}^{|\boldsymbol{y}|} 1_{[y_t \neq \tilde{y}_t]} \tag{2}$$

where the indicator function $1_{[u]} = 1$ if u is true and 0 otherwise.

4 Transductive Loss-Augmented Perceptrons

4.1 The Structured Perceptron

The structured perceptron [7, 2] is analogous to its univariate counterpart. Given an infinite sequence $(\boldsymbol{x}_1, \boldsymbol{y}_1), (\boldsymbol{x}_2, \boldsymbol{y}_2), \ldots$ drawn *i.i.d.* from $p(\boldsymbol{x}, \boldsymbol{y})$, the structured perceptron generates a sequence of models $\boldsymbol{w}_0 = \mathbf{0}, \boldsymbol{w}_1, \boldsymbol{w}_2, \ldots$. At time t, an update is performed if the the prediction $\hat{\boldsymbol{y}}_t = f(\boldsymbol{x}_t; \boldsymbol{w}_t)$ does not coincide with the true output $\boldsymbol{y}_t$; the update rule is given by

$$\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t + \phi(\boldsymbol{x}_t, \boldsymbol{y}_t) - \phi(\boldsymbol{x}_t, \hat{\boldsymbol{y}}_t).$$

Note that in case $\hat{\boldsymbol{y}}_t = \boldsymbol{y}_t$ the model is not changed, that is $\boldsymbol{w}_{t+1} \leftarrow \boldsymbol{w}_t$. After an update, the model favors $\boldsymbol{y}_t$ over $\hat{\boldsymbol{y}}_t$ for the input $\boldsymbol{x}_t$ and a simple extension of Novikoff's theorem [25] shows that the structured perceptron is guaranteed to converge to a zero loss solution (if one exists) in at most $t \leq (\frac{r}{\tilde{\gamma}})^2 \|\boldsymbol{w}^*\|^2$ steps, where r is the radius of the smallest hypersphere enclosing the data points and $\tilde{\gamma}$ is the functional margin of the data [8, 2].

4.2 Loss-augmented Perceptrons

The above update formula intrinsically minimizes the $0/1$-loss which is generally too coarse for differentiating the severity of erroneous annotations. To incorporate task-dependent loss functions into the perceptron, the structured hinge loss of a margin-rescaled SVM [34, 19] can be used. The respective decoding problem becomes

$$\begin{aligned}\hat{\boldsymbol{y}} &= \operatorname*{argmax}_{\tilde{\boldsymbol{y}} \in \mathcal{Y}(\boldsymbol{x}_t)} \left[\ell(\boldsymbol{y}_t, \tilde{\boldsymbol{y}}) - \boldsymbol{w}_t^\top \phi(\boldsymbol{x}_t, \boldsymbol{y}_t) + \boldsymbol{w}_t^\top \phi(\boldsymbol{x}_t, \tilde{\boldsymbol{y}})\right] \\ &= \operatorname*{argmax}_{\tilde{\boldsymbol{y}} \in \mathcal{Y}(\boldsymbol{x}_t)} \left[\ell(\boldsymbol{y}_t, \tilde{\boldsymbol{y}}) + \boldsymbol{w}_t^\top \phi(\boldsymbol{x}_t, \tilde{\boldsymbol{y}})\right].\end{aligned}$$

Margin-rescaling can be intuitively motivated by recalling that the size of the margin $\gamma = \tilde{\gamma}/\|\boldsymbol{w}\|$ quantifies the confidence in rejecting an erroneously decoded output $\tilde{\boldsymbol{y}}$. Re-weighting $\tilde{\gamma}$ with the current loss $\ell(\boldsymbol{y}, \tilde{\boldsymbol{y}})$ leads to a weaker rejection confidence when $\boldsymbol{y}$ and $\tilde{\boldsymbol{y}}$ are similar, while large deviations from the true annotation imply a large rejection threshold. Rescaling the margin by the loss implements the intuition that the confidence of rejecting a mistaken output is proportional to its error.

Margin-rescaling can always be integrated into the decoding algorithm when the loss function decomposes over the latent variables of the output structure as it is the case for the Hamming loss in Eq. (2). The final model $\boldsymbol{w}^*$ is a minimizer of a convex-relaxation of the theoretical loss (the generalization error) and given by

$$\boldsymbol{w}^* = \operatorname*{argmin}_{\tilde{\boldsymbol{w}}} \mathbb{E}\left[\max_{\tilde{\boldsymbol{y}} \in \mathcal{Y}(\boldsymbol{x}_t)} \ell(\boldsymbol{y}_t, \tilde{\boldsymbol{y}}) - w_t^\top \left(\phi(\boldsymbol{x}_t, \boldsymbol{y}_t) - \phi(\boldsymbol{x}_t, \tilde{\boldsymbol{y}})\right)\right].$$

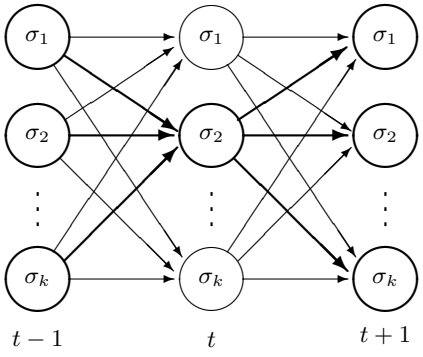

Fig. 2. The constrained Viterbi decoding (emissions are not shown). If time t is annotated with σ_2, the light edges are removed before decoding to guarantee that the optimal path passes through σ_2.

4.3 Transductive Perceptrons for Partially Labeled Data

We derive a straight-forward transductive extension of the loss-augmented perceptron that allows for dealing with partially annotated sequences. Instead of the ground-truth annotation $\boldsymbol{y}$ of an observed sequence $\boldsymbol{x}$, we are now given a set $\boldsymbol{z} = \{(t_j, \sigma_j)\}_{j=1,\ldots,m}$ with $1 \leq t_j \leq T$ and $\sigma_j \in \Sigma$ of token annotations such that the time slices x_{t_j} of $\boldsymbol{x}$ are labeled with $y_{t_j} = \sigma_j$ while the remaining parts of the label sequence are unlabeled.

To learn from the partially annotated input stream $(\boldsymbol{x}_1, \boldsymbol{z}_1), (\boldsymbol{x}_2, \boldsymbol{z}_2), \ldots$, we perform a transductive step to extrapolate the fragmentary annotations to the unlabeled tokens so that we obtain a reference labeling as a makeshift for the missing ground-truth. Following the transductive principle, we use a constrained Viterbi algorithm [5] to decode a pseudo ground-truth $\boldsymbol{y}_p$ for the tuple $(\boldsymbol{x}, \boldsymbol{z})$,

$$\boldsymbol{y}_p = \operatorname*{argmax}_{\tilde{\boldsymbol{y}} \in \mathcal{Y}(\boldsymbol{x})} \boldsymbol{w}^\top \phi(\boldsymbol{x}, \tilde{\boldsymbol{y}}) \quad \text{s.t.} \quad \forall (t, \sigma) \in \boldsymbol{z} : \quad \tilde{y}_t = \sigma.$$

The constrained Viterbi decoding guarantees that the optimal path passes through the already known labels by removing unwanted edges, see Figure 2. Assuming that a labeled token is at position $1 < t < T$, the number of removed edges is precisely $2(k-1)k$, where $k = |\Sigma|$. Algorithmically, the constrained decoding splits sequences at each labeled token in two halves which are then treated independently of each other in the decoding process.

Given the pseudo labeling $\boldsymbol{y}_p$ for an observation $\boldsymbol{x}$, the update rule of the loss-augmented perceptron can be used to complement the transductive perceptron. The inner loop of the resulting algorithm is shown in Table 2. Note that augmenting the loss function into the computation of the argmax (step 2) gives $\boldsymbol{y}_p = \hat{\boldsymbol{y}}$ if and only if the implicit loss-rescaled margin criterion is fulfilled for all alternative output sequences $\tilde{\boldsymbol{y}}$.

Table 2. The transductive perceptron algorithm

Input: Partially labeled example $(\boldsymbol{x}, \boldsymbol{z})$, model $\boldsymbol{w}$

1: $\boldsymbol{y}_p \leftarrow \operatorname{argmax}_{\tilde{\boldsymbol{y}} \in \mathcal{Y}(\boldsymbol{x})} \left[\boldsymbol{w}^\top \phi(\boldsymbol{x}, \tilde{\boldsymbol{y}})\right]$ s.t. $\forall (t, \sigma) \in \boldsymbol{z} : \ \tilde{y}_t = \sigma$.
2: $\hat{\boldsymbol{y}} \leftarrow \operatorname{argmax}_{\tilde{\boldsymbol{y}} \in \mathcal{Y}(\boldsymbol{x})} \left[\ell_h(\boldsymbol{y}_p, \tilde{\boldsymbol{y}}) + \boldsymbol{w}^\top \phi(\boldsymbol{x}, \tilde{\boldsymbol{y}})\right]$
3: $\boldsymbol{w}' \leftarrow \boldsymbol{w} + \phi(\boldsymbol{x}, \boldsymbol{y}_p) - \phi(\boldsymbol{x}, \hat{\boldsymbol{y}})$

Output: Updated model $\boldsymbol{w}'$

Kernelization Analogously to the regular perceptron algorithm, its transductive generalization can easily be kernelized. The weight vector at time t is given by

$$\boldsymbol{w}_t = \boldsymbol{0} + \sum_{j=1}^{t-1} \phi(\boldsymbol{x}_j, \boldsymbol{y}_j^p) - \phi(\boldsymbol{x}_j, \hat{\boldsymbol{y}}_j) \tag{3}$$

$$= \sum_{(\boldsymbol{x}, \boldsymbol{y}_p, \hat{\boldsymbol{y}})} \alpha_{\boldsymbol{x}}(\boldsymbol{y}_p, \hat{\boldsymbol{y}}) \left[\phi(\boldsymbol{x}, \boldsymbol{y}_p) - \phi(\boldsymbol{x}, \hat{\boldsymbol{y}})\right] \tag{4}$$

with appropriately chosen α's that act as virtual counters, detailing how many times the prediction $\hat{\boldsymbol{y}}$ has been decoded instead of the pseudo-output $\boldsymbol{y}_p$ for an observation $\boldsymbol{x}$. Thus, the dual perceptron has virtually exponentially many parameters, however, these are initialized with $\alpha_{\boldsymbol{x}}(\boldsymbol{y}, \boldsymbol{y}') = 0$ for all triplets $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{y}')$ so that the counters only need to be instantiated once the respective triplet is actually seen. Using Eq. (4), the decision function depends only on inner products of joint feature representations which can then be replaced by appropriate kernel functions $k(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{x}', \boldsymbol{y}') = \phi(\boldsymbol{x}, \boldsymbol{y})^\top \phi(\boldsymbol{x}', \boldsymbol{y}')$.

Parameterization Anecdotal evidence shows that unlabeled examples often harm the learning process when the model is weak as the unlabeled data outweigh the labeled part and hinder adaptation to the target concept. A remedy is to differently weight the influence of labeled and unlabeled data or to increase the influence of unlabeled examples during the learning process [13, 37]. In our experiments we parameterize the Hamming loss to account for labeled and unlabeled tokens,

$$\ell_h(\boldsymbol{y}_p, \hat{\boldsymbol{y}}) = \sum_{t=1}^{|\boldsymbol{y}_p|} \lambda(\boldsymbol{z}, t) 1_{[y_t^p \neq \hat{y}_t]}$$

where $\lambda(\boldsymbol{z}, t) = \lambda_L$ if t is a labeled time slice, that is $(t, \cdot) \in \boldsymbol{z}$, and $\lambda(\boldsymbol{z}, t) = \lambda_U$ otherwise. Appropriate values of λ_L and λ_U can be found using cross-validation or using holdout data.

Discussion Trivially, the traditional supervised and semi-supervised counterparts of the perceptron are obtained as special cases. That is, if for all examples $|\boldsymbol{z}| = |\boldsymbol{x}|$ holds, we recover the traditional supervised learning setting and in case either $|\boldsymbol{z}| = |\boldsymbol{x}|$ or $\boldsymbol{z} = \emptyset$ holds, we obtain the standard semi-supervised setting. This observation allows us to design the experiments in the next section simply by changing the data, that is the distribution of the annotations across tokens, while keeping the algorithm fixed. For supervised and semi-supervised scenarios, we only need to alter the label distribution so that it gives rise to either completely labeled or unlabeled sequences.

Using the results by Zinkevich et al. [38] the proposed transductive perceptron can easily be distributed on several machines. Note that the inner loop of the algorithm, displayed in Table 2 depends only on the input $(\boldsymbol{x}, \boldsymbol{z})$ and the actual model $\boldsymbol{w}$. Consequentially, several models can be trained in parallel on disjoint subsets of the data. A subsequent merging process aggregates the models where each model's impact is proportional to the amount of data it has been trained on.

5 Empirical Results

In this section, we will show that (i) one can effectively learn from partial annotations and that (ii) our approach is superior to standard semi-supervised setting. We thus compare the transductive loss-augmented perceptron to its supervised and semi-supervised counterparts. Experiments with CoNLL data use the original splits of the respective corpora into training, holdout, and test set, where parameters are adjusted on the holdout sets. We report on averages of training $3 \times 4 = 12$ repetitions, involving 3 perceptrons and 4 data sets to account for the random effects in the algorithm and data generation; error bars indicate standard error.

Due to the different nature of the algorithms, we need to provide different ground-truth annotations for the algorithms. While the transductive perceptron is simply trained on arbitrarily (e.g., partially) labeled sequences, the supervised baseline needs completely annotated sentences and the semi-supervised perceptron allows for the inclusion of additional unlabeled examples. In each setting, we use the same observation sequences for all methods and only change the distribution of the labels so that it meets the requirements of the respective methods; however note that the number of labeled tokens is identical for all methods. We describe the generation of the training sets in greater detail in the following subsections. All perceptrons are trained for 100 epochs.

5.1 English CoNLL 2003

The first study is based on the CoNLL 2003 shared task [29], an English corpus that includes annotations of four types of entities: person (PER), organization (ORG), location (LOC), and miscellaneous (MISC). This corpus is assembled

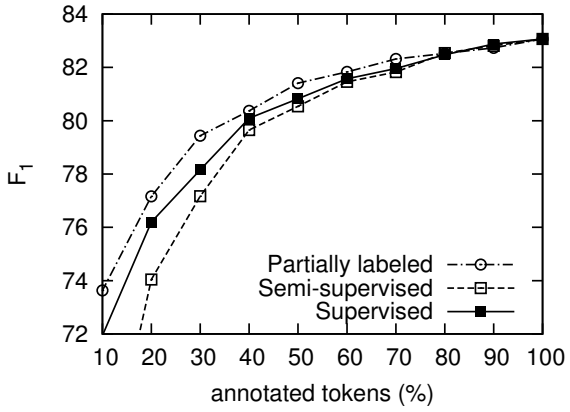

Fig. 3. Results for CoNLL.

from Reuters News stories and divided into three parts: 203,621 training, 51,362 development, and 46,435 test tokens.

We first study the impact of the ratio of labeled and unlabeled tokens in a controlled setting. To generate the respective training sets for supervised and semi-supervised settings, we proceed as follows. For each ratio, we draw sentences at random until the amount of tokens matches (approximately) the required number of labeled examples. These sentences are then completely labeled and form the training set for the supervised perceptron. The semi-supervised perceptron additionally gets the remaining sentences from the original training set as unlabeled examples. The partially labeled training data is generated by randomly removing token annotations from the original training split until the desired ratio of labeled/unlabeled tokens is obtained. Note that the underlying assumptions on the annotations are much stronger for the completely annotated data.

Figure 3 shows F1 scores for different ratios of labeled and unlabeled tokens. Although the baselines are more likely to capture transitions well because the labeled tokens form complete annotations, they are significantly outperformed by the transductive perceptron in case only 10-50% of the tokens are labeled. For 60-100% all three algorithms perform equally well which is still notable because the partial annotations are inexpensive and easier to obtain. By contrast, the semi-supervised perceptron performs worst and is not able to benefit from many unlabeled examples.

We now study the impact of the amount of additional labeled tokens. In Figure 4, we fix the amount of completely annotated sentences at 20% (left figure) and 50% (right), respectively, and vary the amount of additional partially annotated tokens. The supervised and semi-supervised baselines are constant as they cannot deal with the additional data where the semi-supervised perceptron treats the remaining 80% and 50% of the data as unlabeled sentences. Notice

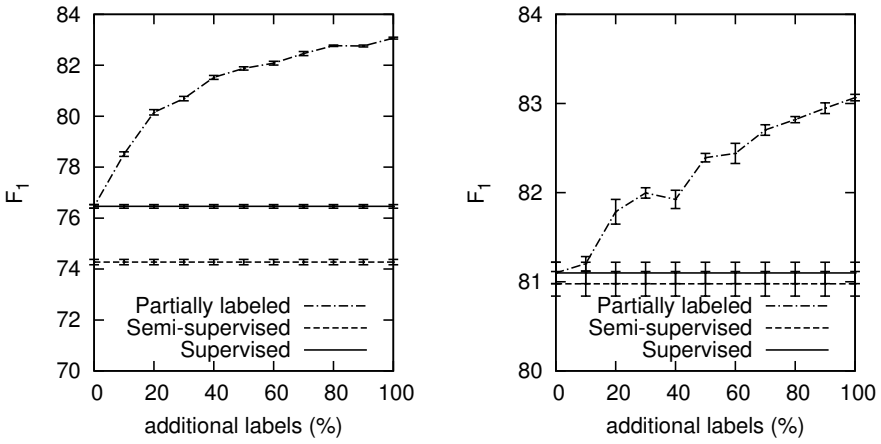

Fig. 4. Varying the amount of additional labeled tokens with 20% (left) and 50% (right) completely labeled examples.

that the semi-supervised baseline performs poorly; as in the previous experiment, the additional unlabeled data seemingly harm the training process. Similar observations have for instance been made by [4, 23] and particularly for structural semi-supervised learning by [37]. By contrast, the transductive perceptron shows in both figures an increasing performance for the partially labeled setting when the amount of labeled tokens increases. The gain in predictive accuracy is highest for settings with only a few completely labeled examples (Figure 4, left).

5.2 Wikipedia – Mono-Lingual Experiment

We now present an experiment using automatically annotated real-world data extracted from Wikipedia. To show that incorporating partially labeled examples improves performance, we proceed as follows: The training set consists of completely labeled sentences which are taken from the English CoNLL data and partially labeled data that is extracted automatically from Wikipedia. One of the major goals in the data generation is to render human interaction unnecessary or at least as low as possible. In the following we briefly describe a simple way to automatically annotate Wikipedia data using existing resources.

Atserias et al. [3] provide a tagged version of the English Wikipedia that preserves the link structure. We collect the tagged entities in the text that are linked to a Wikipedia article. In case the tagged entity does not perfectly match the hyperlinked text we treat it as untagged. This gives us a distribution of tags for each Wikipedia article as the tagging is noisy and depends highly on the context.[4] The linked entities referring to Wikipedia articles are now re-annotated

[4] For instance, a school could be either tagged as a location or an organization, depending on the context.

Table 3. An exemplary partially labeled sentence extracted from Wikipedia. The country *Hungary* is labeled as a location (LOC) due to the majority vote, while *Bukkszek* could not be linked to a tagged article and remains unlabeled.

$\boldsymbol{x}$ =	Bukkszek	is	a	small	village	in	the	north	of	Hungary				
$\boldsymbol{y}$ =	?	?	?	?	?	?	?	?	?	PER: 7	LOC: **10498**	ORG: 42	MISC: 2288	O: 374

with the most frequent tag of the referenced Wikipedia article. Table 3 shows an example of an automatically annotated sentence. Words that are not linked to a Wikipedia article (e.g., *small*) as well as words corresponding to Wikipedia articles which have not yet been tagged (e.g., *Bikkszek*) remain unlabeled.

Table 4. Characteristics of the English data sets.

	CoNLL	**Wikipedia**
tokens	203,621	1,205,137,774
examples	14,041	58,640,083
tokens per example	14.5	20.55
entities	23,499	22,632,261
entities per example	1.67	0.38
MISC	14.63%	18.17%
PER	28.08%	19.71%
ORG	26.89%	30.98%
LOC	30.38%	31.14%

Table 4 shows some descriptive statistics of the extracted data. Since the automatically generated data is only partially annotated, the average number of entities in sentences is much lower compared to that of CoNLL. That is, there are potentially many unidentified and missed entities in the data. By looking at the numbers one could assume that particularly persons (PER) are underrepresented in the Wikipedia data while organizations (ORG) and others (MISC) are slightly overrepresented. Locations (LOC) are seemingly well captured.

The experimental setup is as follows. We use all sentences contained in the CoNLL training set as completely labeled examples and add randomly drawn partially labeled sentences that are automatically extracted from Wikipedia. Figure 5 (left) shows F1 scores for varying numbers of additional data. The leftmost point coincides with the supervised perceptron that only processes the labeled CoNLL data. Adding partially labeled data shows a slight but significant improvement over the supervised baseline. Interestingly, the observed improvement increases with the number of partially labeled examples although these come from a different distribution as shown in Table 4.

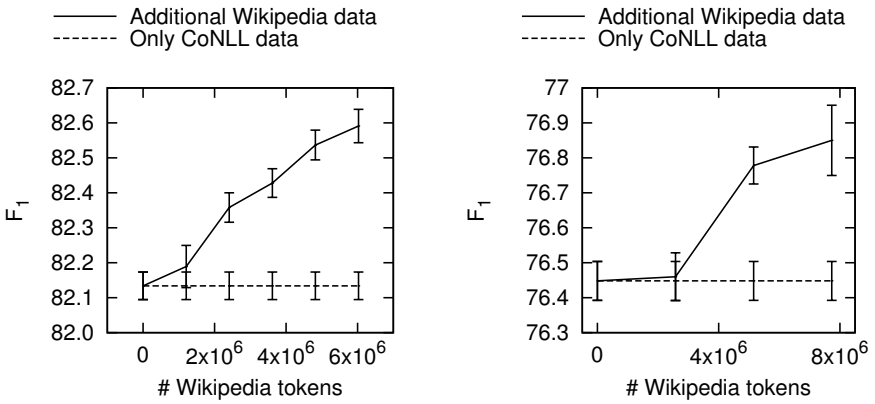

Fig. 5. Results for mono-lingual (left) and cross-lingual (right) Wikipedia experiments.

5.3 Wikipedia – Cross-Lingual Experiment

This experiment aims at studying whether we could enrich a small data set in the target language (here: Spanish) by exploiting resources in a source language (here: English). For the cross-language scenario we use the CoNLL'2002 corpus [28] for evaluation. The corpus consists of Spanish news wire articles from the *EFE*[5] news agency and is annotated with four types of entities: person (PER), organization (ORG), location (LOC), and miscellaneous (MISC). The additional Wikipedia resource is generated automatically as described in the previous section, however, we add an intermediate step to translate the English pages into Spanish by exploiting Wikipedia language links. The automatic data generation now consists of the following three steps: (1) Count the tagged entities that are linked to an English Wikipedia article. (2) Translate the article into Spanish by using the language links. In case such a link does not exist we ignore the article. (3) Annotate mentions of the Spanish entity in the Spanish Wikipedia with the most frequent tag of its English counterpart of step 1.

Table 5 shows some descriptive statistics of the extracted data from the Spanish Wikipedia. The number of contained entities is again much lower than in the CoNLL data. Compared to Table 4, the percentage of persons (PER) matches that of the Spanish CoNLL, however locations (LOC) and other entities (MISC) show large deviations. This is probably due to missing language links between the Wikipedias (the Spanish Wikipedia is much smaller than the one for English) and caused by differences in the respective languages.

Our experimental setup is identical to that of the previous section, except that we now use the training set of the Spanish CoNLL together with the automatically extracted data from the Spanish Wikipedia. Figure 5 (right) shows the results. A relatively small number of additional partially labeled examples

[5] http://efe.com/

Table 5. Characteristics of the Spanish data sets.

	CoNLL	**Wikipedia**
tokens	264,715	257,736,886
examples	8,323	9,500,804
tokens per example	31.81	27.12
entities	18,798	8,520,454
entities per example	2.26	0.89
MISC	11.56%	27.64%
PER	22.99%	23.71%
ORG	39.31%	32.63%
LOC	26.14%	16.02%

does not have an impact on the performance of the transductive perceptron. We credit this finding to noisy and probably weak annotations caused by the language transfer. However, when we add more than 6 million automatically labeled tokens, the generalized problem setting pays off and the performance increases, slightly, but significantly.

5.4 Execution Time

Figure 6 reports on execution times on an Intel(R) Core(TM)2 Duo CPU (E8400 model) with 3.00GHz and 6 MB cache memory. We use the same experimental setup as for Figure 4 (left). That is we use 20% of the English CoNLL sequences as completely labeled examples and vary the number of additional annotations on the remaining tokens. The figure shows that the execution time decreases for an increasing number of labels because the decoding is less expensive until it reaches the performance of the the standard loss-augmented perceptron which is trained on the completely labeled training set of the CoNLL data. The results also hold for the Wikipedia experiments. Using additional 0.1% of the English Wikipedia (which is about 5 times the size of the CoNLL training set) takes about 18 minutes. In sum, we observe a linear growing execution time in the size of the corpus given a fixed ratio of labeled/unlabeled tokens.

6 Conclusion

In this paper, we showed that surprisingly simple methods, such as the devised transductive perceptron, allow for learning from sparse and partial labelings. Our empirical findings show that a few, randomly distributed labels often lead to better models than the standard supervised and semi-supervised settings based on completely labeled ground-truth; the transductive perceptron was observed to be always better or on par as its counterparts trained on the same amount of labeled data. Immediate consequences arise for the data collection: while the standard semi-supervised approach requires completely labeled editorial data, we can effectively learn from partial annotations that have been generated automatically and without manual interaction; using additional, automatically labeled

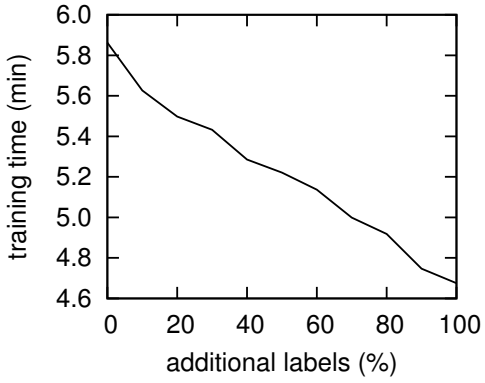

Fig. 6. Execution time.

data from Wikipedia lead to a significant increase in performance in mono- and cross-lingual named entity recognition tasks. We emphasize that these improvements come at factually no additional labeling costs at all.

Future work will extend our study towards larger-scales. It will certainly be of interest to extend the empirical evaluation to other sequential tasks, output structures. As the developed transductive perceptron is a relatively simple algorithm, more sophisticated ways for dealing with partially labeled data are also interesting research areas.

Acknowledgments

We thank Jordi Atserias for helping us to generate the Wikipedia data. This work was partially funded by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior (CAPES), Brazil.

References

1. Y. Altun, D. McAllester, and M. Belkin. Maximum margin semi–supervised learning for structured variables. In *Advances in Neural Information Processing Systems*, 2006.
2. Y. Altun, I. Tsochantaridis, and T. Hofmann. Hidden Markov support vector machines. In *Proceedings of the International Conference on Machine Learning*, 2003.
3. Jordi Atserias, Hugo Zaragoza, Massimiliano Ciaramita, and Giuseppe Attardi. Semantically annotated snapshot of the english wikipedia. In European Language Resources Association (ELRA), editor, *Proceedings of the Sixth International Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco, may 2008.
4. S. Baluja. Probabilistic modeling for face orientation discrimination: Learning from labeled and unlabeled data. In *Advances in Neural Information Processing Systems*, 1998.

5. L. Cao and C. W. Chen. A novel product coding and recurrent alternate decoding scheme for image transmission over noisy channels. *IEEE Transactions on Communications*, 51(9):1426 – 1431, 2003.
6. O. Chapelle, B. Schölkopf, and A. Zien. *Semi–supervised Learning*. MIT Press, 2006.
7. M. Collins. Discriminative reranking for natural language processing. In *Proceedings of the International Conference on Machine Learning*, 2000.
8. M. Collins. Ranking algorithms for named-entity extraction: Boosting and the voted perceptron. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, 2002.
9. T.G. Dietterich. Machine learning for sequential data: A review. In *Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*, 2002.
10. T.-M.-T. Do and T. Artieres. Large margin training for hidden Markov models with partially observed states. In *Proceedings of the International Conference on Machine Learning*, 2009.
11. G. D. Forney. The Viterbi algorithm. *Proceedings of IEEE*, 61(3):268–278, 1973.
12. J. M. Hammersley and P. E. Clifford. Markov random fields on finite graphs and lattices. Unpublished manuscript, 1971.
13. T. Joachims. Transductive inference for text classification using support vector machines. In *Proceedings of the International Conference on Machine Learning*, 1999.
14. B. Juang and L. Rabiner. Hidden Markov models for speech recognition. *Technometrics*, 33:251–272, 1991.
15. T. H. King, S. Dipper, A. Frank, J. Kuhn, and J. Maxwell. Ambiguity management in grammar writing. In *Proceedings of the ESSLLI 2000 Workshop on Linguistic Theory and Grammar Implementation*, 2000.
16. J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning*, 2001.
17. J. Lafferty, X. Zhu, and Y. Liu. Kernel conditional random fields: representation and clique selection. In *Proceedings of the International Conference on Machine Learning*, 2004.
18. C. Lee, S. Wang, F. Jiao, R. Greiner, and D. Schuurmans. Learning to model spatial dependency: Semi–supervised discriminative random fields. In *Advances in Neural Information Processing Systems*, 2007.
19. D. McAllester, T. Hazan, and J. Keshet. Direct loss minimization for structured perceptronsi. In *Advances in Neural Information Processing Systems*, 2011.
20. A. McCallum, D. Freitag, and F. Pereira. Maximum entropy Markov models for information extraction and segmentation. In *Proceedings of the International Conference on Machine Learning*, 2000.
21. Peter Mika, Massimiliano Ciaramita, Hugo Zaragoza, and Jordi Atserias. Learning to tag and tagging to learn: A case study on wikipedia. *IEEE Intelligent Systems*, 23:26–33, 2008.
22. S. Mukherjee and I. V. Ramakrishnan. Taming the unstructured: Creating structured content from partially labeled schematic text sequences. In *On the Move to Meaningful Internet Systems 2004: CoopIS, DOA, and ODBASE, OTM Confederated International Conferences*, 2004.
23. Kamal Nigam, Andrew K. McCallum, Sebastian Thrun, and Tom M. Mitchell. Text classification from labeled and unlabeled documents using EM. *Machine Learning*, 39(2/3):103–134, 2000.

24. Joel Nothman, Tara Murphy, and James R. Curran. Analysing wikipedia and gold-standard corpora for ner training. In *EACL '09: Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics*, pages 612–620, Morristown, NJ, USA, 2009. Association for Computational Linguistics.
25. A. B. Novikoff. On convergence proofs on perceptrons. In *Proceedings of the Symposium on the Mathematical Theory of Automata*, 1962.
26. L. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77(2):257–285, 1989.
27. Alexander E. Richman and Patrick Schone. Mining wiki resources for multilingual named entity recognition. In *Proceedings of ACL-08: HLT*, pages 1–9, Columbus, Ohio, June 2008. Association for Computational Linguistics.
28. Erik F. Tjong Kim Sang. Introduction to the CoNLL-2002 shared task: language-independent named entity recognition. In *COLING-2002: proceedings of the 6th conference on Natural language learning*, pages 1–4, Morristown, NJ, USA, 2002. Association for Computational Linguistics.
29. Erik F. Tjong Kim Sang and Fien De Meulder. Introduction to the CoNLL-2003 shared task: Language-independent named entity recognition. In *Proceedings of CoNLL-2003*, pages 142–147, 2003.
30. T. Scheffer and S. Wrobel. Active hidden Markov models for information extraction. In *Proceedings of the International Symposium on Intelligent Data Analysis*, 2001.
31. R. Schwarz and Y. L. Chow. The n-best algorithm: An efficient and exact procedure for finding the n most likely hypotheses. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 1990.
32. B. Taskar, C. Guestrin, and D. Koller. Max–margin Markov networks. In *Advances in Neural Information Processing Systems*, 2004.
33. T. T. Truyen, H. H. Bui, D. Q. Phung, and S. Venkatesh. Learning discriminative sequence models from partially labelled data for activity recognition. In *Proceedings of the Pacific Rim International Conference on Artificial Intelligence*, 2008.
34. I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484, 2005.
35. L. Xu, D. Wilkinson, F. Southey, and D. Schuurmans. Discriminative unsupervised learning of structured predictors. In *Proceedings of the International Conference on Machine Learning*, 2006.
36. C.-N. Yu and T. Joachims. Learning structural svms with latent variables. In *Proceedings of the International Conference on Machine Learning*, 2009.
37. A. Zien, U. Brefeld, and T. Scheffer. Transductive support vector machines for structured variables. In *Proceedings of the International Conference on Machine Learning*, 2007.
38. M. Zinkevich, M. Weimer, A. Smola, and L. Li. Parallelized stochastic gradient descent. In *Advances in Neural Information Processing Systems 23*, 2011.