Semi-Supervised Structured Output Learning based on a Hybrid Generative and Discriminative Approach

Jun Suzuki, Akinori Fujino and Hideki Isozaki
NTT Communication Science Laboratories, NTT Corp.
2-4 Hikaridai, Seika-cho, Soraku-gun, Kyoto, 619-0237 Japan
{jun, a.fujino, isozaki}@cslab.kecl.ntt.co.jp

Abstract

This paper proposes a framework for semi-supervised structured output learning (SOL), specifically for sequence labeling, based on a hybrid generative and discriminative approach. We define the objective function of our hybrid model, which is written in log-linear form, by discriminatively combining discriminative structured predictor(s) with generative model(s) that incorporate unlabeled data. Then, unlabeled data is used in a generative manner to increase the sum of the discriminant functions for all outputs during the parameter estimation. Experiments on named entity recognition (CoNLL-2003) and syntactic chunking (CoNLL-2000) data show that our hybrid model significantly outperforms the state-of-the-art performance obtained with supervised SOL methods, such as conditional random fields (CRFs).

1 Introduction

Structured output learning (SOL) methods, which attempt to optimize an interdependent output space globally, are important methodologies for certain natural language processing (NLP) tasks such as part-of-speech tagging, syntactic chunking (Chunking) and named entity recognition (NER), which are also referred to as sequence labeling tasks. When we consider the nature of these sequence labeling tasks, a semi-supervised approach appears to be more natural and appropriate. This is because the number of features and parameters typically become extremely large, and labeled examples can only sparsely cover the parameter space, even if thousands of labeled examples are available. In fact, many attempts have recently been made to develop semi-supervised SOL methods (Zhu et al., 2003; Li and McCallum, 2005; Altun et al., 2005; Jiao et al., 2006; Brefeld and Scheffer, 2006).

With the generative approach, we can easily incorporate unlabeled data into probabilistic models with the help of expectation-maximization (EM) algorithms (Dempster et al., 1977). For example, the Baum-Welch algorithm is a well-known algorithm for training a hidden Markov model (HMM) of sequence learning. Generally, with sequence learning tasks such as NER and Chunking, we cannot expect to obtain better performance than that obtained using discriminative approaches in supervised learning settings.

In contrast to the generative approach, with the discriminative approach, it is not obvious how unlabeled training data can be naturally incorporated into a discriminative training criterion. For example, the effect of unlabeled data will be eliminated from the objective function if the unlabeled data is directly used in traditional i.i.d. conditional-probability models. Nevertheless, several attempts have recently been made to incorporate unlabeled data in the discriminative approach. An approach based on pairwise similarities, which encourage nearby data points to have the same class label, has been proposed as a way of incorporating unlabeled data discriminatively (Zhu et al., 2003; Altun et al., 2005; Brefeld and Scheffer, 2006). However, this approach generally requires joint inference over the whole data set for prediction, which is not practical as regards the large data sets used for standard sequence labeling tasks in NLP. Another discriminative approach to semi-supervised SOL involves the incorporation of an entropy regularizer (Grand-

Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, pp. 791–800, Prague, June 2007.

valet and Bengio, 2004). Semi-supervised conditional random fields (CRFs) based on a minimum entropy regularizer (SS-CRF-MER) have been proposed in (Jiao et al., 2006). With this approach, the parameter is estimated to maximize the likelihood of labeled data and the negative conditional entropy of unlabeled data. Therefore, the structured predictor is trained to separate unlabeled data well under the entropy criterion by parameter estimation.

In contrast to these previous studies, this paper proposes a semi-supervised SOL framework based on a hybrid generative and discriminative approach. A hybrid approach was first proposed in a supervised learning setting (Raina et al., 2003) for text classification. (Fujino et al., 2005) have developed a semi-supervised approach by discriminatively combining a supervised classifier with generative models that incorporate unlabeled data. We extend this framework to the structured output domain, specifically for sequence labeling tasks. Moreover, we reformalize the objective function to allow the incorporation of discriminative models (structured predictors) trained from labeled data, since the original framework only considers the combination of generative classifiers. As a result, our hybrid model can significantly improve on the state-of-the-art performance obtained with supervised SOL methods, such as CRFs, even if a large amount of labeled data is available, as shown in our experiments on CoNLL-2003 NER and CoNLL-2000 Chunking data. In addition, compared with SS-CRF-MER, our hybrid model has several good characteristics including a low calculation cost and a robust optimization in terms of a sensitiveness of hyper-parameters. This is described in detail in Section 5.3.

2 Supervised SOL: CRFs

This paper focuses solely on sequence labeling tasks, such as named entity recognition (NER) and syntactic chunking (Chunking), as SOL problems. Thus, let $\boldsymbol{x}=(x_1,\dots,x_S)\in\mathcal{X}$ be an input sequence, and $\boldsymbol{y}=(y_0,\dots,y_{S+1})\in\mathcal{Y}$ be a particular output sequence, where y_0 and y_{S+1} are special fixed labels that represent the beginning and end of a sequence.

As regards supervised sequence learning, CRFs are recently introduced methods that constitute flexible and powerful models for structured predictors based on undirected graphical models that have been globally conditioned on a set of inputs (Lafferty et al., 2001). Let $\boldsymbol{\lambda}$ be a parameter vector and $\boldsymbol{f}(y_{s-1},y_s,\boldsymbol{x})$ be a (local) feature vector obtained from the corresponding position s given $\boldsymbol{x}$. CRFs define the conditional probability, $p(\boldsymbol{y}|\boldsymbol{x})$, as being proportional to a product of potential functions on the cliques. That is, $p(\boldsymbol{y}|\boldsymbol{x})$ on a (linear-chain) CRF can be defined as follows:

$$p(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\lambda}) = \frac{1}{Z(\boldsymbol{x})}\prod_{s=1}^{S+1}\exp(\boldsymbol{\lambda}\cdot\boldsymbol{f}(y_{s-1},y_s,\boldsymbol{x})).$$

$Z(\boldsymbol{x})=\sum_{\boldsymbol{y}}\prod_{s=1}^{S+1}\exp(\boldsymbol{\lambda}\cdot\boldsymbol{f}(y_{s-1},y_s,\boldsymbol{x}))$ is a normalization factor over all output values, $\mathcal{Y}$, and is also known as the partition function.

For parameter estimation (training), given labeled data $\mathcal{D}_l=\{(\boldsymbol{x}^k,\boldsymbol{y}^k)\}_{k=1}^K$, the Maximum a Posteriori (MAP) parameter estimation, namely maximizing $\log p(\boldsymbol{\lambda}|\mathcal{D}_l)$, is now the most widely used CRF training criterion. Thus, we maximize the following objective function to obtain optimal $\boldsymbol{\lambda}$:

$$\mathcal{L}^{\text{CRF}}(\boldsymbol{\lambda}) = \sum_k\Big[\boldsymbol{\lambda}\cdot\sum_s\boldsymbol{f}_s - \log Z(\boldsymbol{x}^k)\Big] + \log p(\boldsymbol{\lambda}), \quad (1)$$

where $\boldsymbol{f}_s$ is an abbreviation of $\boldsymbol{f}(y_{s-1},y_s,\boldsymbol{x})$ and $p(\boldsymbol{\lambda})$ is a prior probability distribution of $\boldsymbol{\lambda}$. A gradient-based optimization algorithm such as L-BFGS (Liu and Nocedal, 1989) is widely used for maximizing Equation (1). The gradient of Equation (1) can be written as follows:

$$\begin{aligned}\nabla\mathcal{L}^{\text{CRF}}(\boldsymbol{\lambda}) = &\sum_k E_{\tilde{p}(\boldsymbol{y}^k,\boldsymbol{x}^k;\boldsymbol{\lambda})}\big[\sum_s\boldsymbol{f}_s\big] \\ &-\sum_k E_{p(\mathcal{Y}|\boldsymbol{x}^k;\boldsymbol{\lambda})}\big[\sum_s\boldsymbol{f}_s\big] + \nabla\log p(\boldsymbol{\lambda}).\end{aligned}$$

Calculating $E_{p(\mathcal{Y}|\boldsymbol{x},\boldsymbol{\lambda})}$ as well as the partition function $Z(\boldsymbol{x})$ is not always tractable. However, for linear-chain CRFs, a dynamic programming algorithm similar in nature to the forward-backward algorithm in HMMs has already been developed for an efficient calculation (Lafferty et al., 2001).

For prediction, the most probable output, that is, $\hat{\boldsymbol{y}}=\arg\max_{\boldsymbol{y}\in\mathcal{Y}}p(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\lambda})$, can be efficiently obtained by using the Viterbi algorithm.

3 Hybrid Generative and Discriminative Approach to Semi-Supervised SOL

In this section, we describe our formulation of a hybrid approach to SOL and a parameter estimation method for sequence predictors. We assume that we have a set of labeled and unlabeled data, $\mathcal{D} = \{\mathcal{D}_l, \mathcal{D}_u\}$, where $\mathcal{D}_l = \{(\boldsymbol{x}^n, \boldsymbol{y}^n)\}_{n=1}^N$ and $\mathcal{D}_u = \{\boldsymbol{x}^m\}_{m=1}^M$.

Let us assume that we have I-units of discriminative models, p_i^D, and J-units of generative models, p_j^G. Our hybrid model for a structured predictor is designed by the discriminative combination of several joint probability densities of $\boldsymbol{x}$ and $\boldsymbol{y}$, $p(\boldsymbol{x}, \boldsymbol{y})$. That is, the posterior probability of our hybrid model is defined by providing the log-values of $p(\boldsymbol{x}, \boldsymbol{y})$ as the features of a log-linear model, such that:

$$\begin{aligned} & R(\boldsymbol{y}|\boldsymbol{x}; \boldsymbol{\Lambda}, \boldsymbol{\Theta}, \boldsymbol{\Gamma}) \\ &= \frac{\prod_i p_i^D(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\lambda}_i)^{\gamma_i} \prod_j p_j^G(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}_j)^{\gamma_j}}{\sum_{\boldsymbol{y}} \prod_i p_i^D(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\lambda}_i)^{\gamma_i} \prod_j p_j^G(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}_j)^{\gamma_j}} \\ &= \frac{\prod_i p_i^D(\boldsymbol{y}|\boldsymbol{x}; \boldsymbol{\lambda}_i)^{\gamma_i} \prod_j p_j^G(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}_j)^{\gamma_j}}{\sum_{\boldsymbol{y}} \prod_i p_i^D(\boldsymbol{y}|\boldsymbol{x}; \boldsymbol{\lambda}_i)^{\gamma_i} \prod_j p_j^G(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}_j)^{\gamma_j}}. \end{aligned} \tag{2}$$

Here, $\boldsymbol{\Gamma} = \{\{\gamma_i\}_{i=1}^I, \{\gamma_j\}_{j=I+1}^{I+J}\}$ represents the discriminative combination weight of each model where $\gamma_i, \gamma_j \in [0,1]$. Moreover, $\boldsymbol{\Lambda} = \{\boldsymbol{\lambda}_i\}_{i=1}^I$ and $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_j\}_{j=1}^J$ represent model parameters of individual models estimated from labeled and unlabeled data, respectively. Using $p^D(\boldsymbol{x}, \boldsymbol{y}) = p^D(\boldsymbol{y}|\boldsymbol{x})p^D(\boldsymbol{x})$, we can derive the third line from the second line, where $p_i^D(\boldsymbol{x}; \boldsymbol{\lambda}_i)^{\gamma_i}$ for all i are canceled out. Thus, our hybrid model is constructed by combining discriminative models, $p_i^D(\boldsymbol{y}|\boldsymbol{x}; \boldsymbol{\lambda}_i)$, with generative models, $p_j^G(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}_j)$.

Hereafter, let us assume that our hybrid model consists of CRFs for discriminative models, p_i^D, and HMMs for generative models, p_j^G, shown in Equation (2), since this paper focuses solely on sequence modeling. For HMMs, we consider a first order HMM defined in the following equation:

$$p(\boldsymbol{x}, \boldsymbol{y}|\boldsymbol{\theta}) = \prod_{s=1}^{S+1} \theta_{y_{s-1}, y_s} \theta_{y_s, x_s},$$

where θ_{y_{s-1}, y_s} and θ_{y_s, x_s} represent the transition probability between states y_{s-1} and y_s and the symbol emission probability of the s-th position of the corresponding input sequence, respectively, where $\theta_{y_{S+1}, x_{S+1}} = 1$.

It can be seen that the formalization in the log-linear combination of our hybrid model is very similar to that of LOP-CRFs (Smith et al., 2005). In fact, if we only use a combination of discriminative models (CRFs), which is equivalent to $\gamma_j = 0$ for all j, we obtain essentially the same objective function as that of the LOP-CRFs. Thus, our framework can also be seen as an extension of LOP-CRFs that enables us to incorporate unlabeled data.

3.1 Discriminative Combination

For estimating the parameter $\boldsymbol{\Gamma}$, let us assume that we already have discriminatively trained models on labeled data, $p_i^D(\boldsymbol{y}|\boldsymbol{x}; \boldsymbol{\lambda}_i)$. We maximize the following objective function for estimating parameter $\boldsymbol{\Gamma}$ under a fixed $\boldsymbol{\Theta}$:

$$\mathcal{L}^{\text{HySOL}}(\boldsymbol{\Gamma}|\boldsymbol{\Theta}) = \sum_n \log R(\boldsymbol{y}^n|\boldsymbol{x}^n; \boldsymbol{\Lambda}, \boldsymbol{\Theta}, \boldsymbol{\Gamma}) + \log p(\boldsymbol{\Gamma}). \tag{3}$$

where $p(\boldsymbol{\Gamma})$ is a prior probability distribution of $\boldsymbol{\Gamma}$.

The value of $\boldsymbol{\Gamma}$ providing a global maximum of $\mathcal{L}^{\text{HySOL}}(\boldsymbol{\Gamma}|\boldsymbol{\Theta})$ is guaranteed under an arbitrary fixed value in the $\boldsymbol{\Theta}$ domain, since $\mathcal{L}^{\text{HySOL}}(\boldsymbol{\Gamma}|\boldsymbol{\Theta})$ is a concave function of $\boldsymbol{\Gamma}$. Thus, we can easily maximize Equation (3) by using a gradient-based optimization algorithm such as (bound constrained) L-BFGS (Liu and Nocedal, 1989).

3.2 Incorporating Unlabeled Data

We cannot directly incorporate unlabeled data for discriminative training such as Equation (3) since the correct outputs $\boldsymbol{y}$ for unlabeled data are unknown. On the other hand, generative approaches can easily deal with unlabeled data as incomplete data (data with missing variable $\boldsymbol{y}$) by using a mixture model. A well-known way to achieve this incorporation is to maximize the log likelihood of unlabeled data with respect to the marginal distribution of generative models as

$$\mathcal{L}(\boldsymbol{\theta}) = \sum_m \log \sum_{\boldsymbol{y}} p(\boldsymbol{x}^m, \boldsymbol{y}; \boldsymbol{\theta}).$$

In fact, (Nigam et al., 2000) have reported that using unlabeled data with a mixture model can improve the text classification performance.

According to Bayes' rule, $p(\boldsymbol{y}|\boldsymbol{x}; \boldsymbol{\theta}) \propto p(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta})$, the discriminant functions of generative classifiers are provided by generative models $p(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta})$. Therefore, we can regard $\mathcal{L}(\boldsymbol{\theta})$ as the logarithm of the sum of discriminant functions for all missing variables $\boldsymbol{y}$ of unlabeled data. Following this view, we can directly incorporate unlabeled data into our hybrid model by maximizing the discriminant functions g of our hybrid model in the same way as for a mixture model as explained above. Thus, we maximize the following objective function for estimating the model parameters $\mathbf{\Theta}$ for generative models of unlabeled data:

$$\mathcal{G}(\mathbf{\Theta}|\mathbf{\Gamma}) = \sum_m \log \sum_{\boldsymbol{y}} g(\boldsymbol{x}^m, \boldsymbol{y}; \mathbf{\Theta}) + \log p(\mathbf{\Theta}). \quad (4)$$

where $p(\mathbf{\Theta})$ is a prior probability distribution of $\mathbf{\Theta}$. Here, the discriminant function g of output $\boldsymbol{y}$ given input $\boldsymbol{x}$ in our hybrid model can be obtained by the numerator on the third line of Equation (2), since the denominator does not affect the determination of $\boldsymbol{y}$, that is,

$$g(\boldsymbol{x}, \boldsymbol{y}; \mathbf{\Theta}) = \prod_i p_i^D(\boldsymbol{y}|\boldsymbol{x}; \boldsymbol{\lambda}_i)^{\gamma_i} \prod_j p_j^G(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta}_j)^{\gamma_j}.$$

Under a fixed $\mathbf{\Gamma}$, we can estimate the local maximum of $\mathcal{G}(\mathbf{\Theta}|\mathbf{\Gamma})$ around the initialized value of $\mathbf{\Theta}$ by an iterative computation such as the EM algorithm (Dempster et al., 1977). Let $\mathbf{\Theta}''$ and $\mathbf{\Theta}'$ be estimates of $\mathbf{\Theta}$ in the next and current steps, respectively. Using Jensen's inequality, $\log a \leq a - 1$, we obtain a Q-function that satisfies the inequality $\mathcal{G}(\mathbf{\Theta}''|\mathbf{\Gamma})$–$\mathcal{G}(\mathbf{\Theta}'|\mathbf{\Gamma}) \geq Q(\mathbf{\Theta}'', \mathbf{\Theta}'; \mathbf{\Gamma})$–$Q(\mathbf{\Theta}', \mathbf{\Theta}'; \mathbf{\Gamma})$, such that

$$\begin{aligned} &Q(\mathbf{\Theta}'', \mathbf{\Theta}'; \mathbf{\Gamma}) \\ &= \sum_j \gamma_j \sum_m \sum_{\boldsymbol{y}} R(\boldsymbol{y}|\boldsymbol{x}^m; \mathbf{\Lambda}, \mathbf{\Theta}', \mathbf{\Gamma}) \log p_j^G(\boldsymbol{x}^m, \boldsymbol{y}; \mathbf{\Theta}'') \\ &\quad + \log p(\mathbf{\Theta}''). \end{aligned} \quad (5)$$

Since $Q(\mathbf{\Theta}', \mathbf{\Theta}'; \mathbf{\Gamma})$ is independent of $\mathbf{\Theta}''$, we can improve the value of $\mathcal{G}(\mathbf{\Theta}|\mathbf{\Gamma})$ by computing $\mathbf{\Theta}''$ to maximize $Q(\mathbf{\Theta}'', \mathbf{\Theta}'; \mathbf{\Gamma})$. We can obtain a $\mathbf{\Theta}$ estimate by iteratively performing this update while $\mathcal{G}(\mathbf{\Theta}|\mathbf{\Gamma})$ is hill climbing.

As shown in Equation (5), R is used for estimating the parameter $\mathbf{\Theta}$. The intuitive effect of maximizing Equation (4) is similar to performing 'soft-clustering'. That is, unlabeled data is clustered with respect to the R distribution, which also includes information about labeled data, under the constraint of generative model structures.

3.3 Parameter Estimation Procedure

According to our definition, the $\mathbf{\Theta}$ and $\mathbf{\Gamma}$ estimations are mutually dependent. That is, the parameters of the hybrid model, $\mathbf{\Gamma}$, should be estimated using Equation (3) with a fixed $\mathbf{\Theta}$, while the parameters of the generative models, $\mathbf{\Theta}$, should be estimated using Equation (4) with a fixed $\mathbf{\Gamma}$. As a solution to our parameter estimation, we search for the $\mathbf{\Theta}$ and $\mathbf{\Gamma}$ that maximize $\mathcal{L}^{\text{HySOL}}(\mathbf{\Gamma}|\mathbf{\Theta})$ and $\mathcal{G}(\mathbf{\Theta}|\mathbf{\Gamma})$ simultaneously. For this search, we compute $\mathbf{\Theta}$ and $\mathbf{\Gamma}$ by maximizing the objective functions shown in Equations (4) and (3) iteratively and alternately. We summarize the algorithm for estimating these model parameters in Figure 1.

1.Given training set: $\mathcal{D}_u = \{\boldsymbol{x}^m\}_{m=1}^M$ and
$\quad\mathcal{D}_l = \{\mathcal{D}'_l = \{(\boldsymbol{x}^k, \boldsymbol{y}^k)\}_{k=1}^K, \mathcal{D}''_l = \{(\boldsymbol{x}^n, \boldsymbol{y}^n)\}_{n=1}^N\}$
2.Compute $\mathbf{\Lambda}$, using $\mathcal{D}'_l$.
3.Initialize $\mathbf{\Gamma}^{(0)}$, $\mathbf{\Theta}^{(0)}$ and $t \leftarrow 0$.
4.Perform the following until $\frac{|\mathbf{\Theta}^{(t+1)} - \mathbf{\Theta}^{(t)}|}{|\mathbf{\Theta}^{(t)}|} < \epsilon$.
  4.1. Compute $\mathbf{\Theta}^{(t+1)}$ to maximize Equation (4) under fixed $\mathbf{\Gamma}^{(t)}$ and $\mathbf{\Lambda}$ using $\mathcal{D}_u$.
  4.2. Compute $\mathbf{\Gamma}^{(t+1)}$ to maximize Equation (3) under fixed $\mathbf{\Theta}^{(t+1)}$ and $\mathbf{\Lambda}$ using $\mathcal{D}''_l$.
  4.3. $t \leftarrow t + 1$.
5.Output a structured predictor $R(\boldsymbol{y}|\boldsymbol{x}, \mathbf{\Lambda}, \mathbf{\Theta}^{(t)}, \mathbf{\Gamma}^{(t)})$.

Figure 1: Algorithm of learning model parameters used in our hybrid model.

Note that during the $\mathbf{\Gamma}$ estimation (procedure 4.2 in Figure 1), $\mathbf{\Gamma}$ can be over-fitted to the labeled training data if we use the same labeled training data as used for the $\mathbf{\Lambda}$ estimation. There are several possible ways to reduce this over-fit. In this paper, we select one of the simplest; we divide the labeled training data $\mathcal{D}_l$ into two distinct sets $\mathcal{D}'_l$ and $\mathcal{D}''_l$. Then, $\mathcal{D}'_l$ and $\mathcal{D}''_l$ are individually used for estimating $\mathbf{\Lambda}$ and $\mathbf{\Gamma}$, respectively. In our experiments, we divide the labeled training data $\mathcal{D}_l$ so that 4/5 is used for $\mathcal{D}'_l$ and the remaining 1/5 for $\mathcal{D}''_l$.

3.4 Efficient Parameter Estimation Algorithm

Let $\mathcal{N}_R(\boldsymbol{x})$ represent the denominator of Equation (2), that is the normalization factor of R. We can rearrange Equation (2) as follows:

$$R(\boldsymbol{y}|\boldsymbol{x}; \mathbf{\Lambda}, \mathbf{\Theta}, \mathbf{\Gamma}) = \frac{\prod_s \prod_i \left[V_{i,s}^D\right]^{\gamma_i} \prod_j \left[V_{j,s}^G\right]^{\gamma_j}}{\mathcal{N}_R(\boldsymbol{x}) \prod_i [Z_i(\boldsymbol{x})]^{\gamma_i}}, \quad (6)$$

where $V_{i,s}^D$ represents the potential function of the s-th position of the sequence in the i-th CRF and $V_{j,s}^G$ represents the probability of the s-th position in the j-th HMM, that is, $V_{i,s}^D = \exp(\boldsymbol{\lambda}_i \cdot \boldsymbol{f}_s)$ and $V_{j,s}^G = \theta_{y_{s-1},y_s}\theta_{y_s,x_s}$, respectively. See the Appendix for the derivation of Equation (6) from Equation (2).

To estimate $\mathbf{\Gamma}^{(t+1)}$, namely procedure 4.2 in Figure 1, we employ the derivatives with respect to γ_i and γ_j shown in Equation (6), which are the parameters of the discriminative and generative models, respectively. Thus, we obtain the following derivatives with respect to γ_i:

$$\frac{\partial \mathcal{L}^{\text{HySOL}}(\mathbf{\Gamma}|\mathbf{\Theta})}{\partial \gamma_i} = \sum_n \log p_i^D(\boldsymbol{y}^n|\boldsymbol{x}^n) + \sum_n \log Z_i^D(\boldsymbol{x}^n) - \sum_n E_{R(\mathcal{Y}|\boldsymbol{x}^n;\mathbf{\Lambda},\mathbf{\Theta},\mathbf{\Gamma})}\big[\sum_s \log V_{i,s}^D\big].$$

The first and second terms are constant during iterative procedure 4 in our optimization algorithm shown in Figure 1. Thus, we only need to calculate these values once at the beginning of procedure 4. Let $\boldsymbol{\alpha}_s(y)$ and $\boldsymbol{\beta}_s(y)$ represent the forward and backward state costs at position s with output y for corresponding input $\boldsymbol{x}$. Let $\mathcal{V}_s(y,y')$ represent the products of the total value of the transition cost between $s-1$ and s with labels y and y' in the corresponding input sequence, that is, $\mathcal{V}_s(y,y') = \prod_i [V_{i,s}^D(y,y')]^{\gamma_i} \prod_j [V_{j,s}^G(y,y')]^{\gamma_j}$. The third term, which indicates the expectation of potential functions, can be rewritten in the form of a forward-backward algorithm, that is,

$$E_{R(\mathcal{Y}|\boldsymbol{x};\mathbf{\Lambda},\mathbf{\Theta},\mathbf{\Gamma})}\big[\sum_s \log V_{i,s}^D\big] = \frac{1}{Z_R(\boldsymbol{x})} \sum_s \sum_{y,y'} \boldsymbol{\alpha}_{s-1}(y) \mathcal{V}_s(y,y') \boldsymbol{\beta}_s(y') \log V_{i,s}^D(y,y'), \tag{7}$$

where $Z_R(\boldsymbol{x})$ represents the partition function of our hybrid model, that is, $Z_R(\boldsymbol{x}) = \mathcal{N}_R(\boldsymbol{x}) \prod_i [Z_i(\boldsymbol{x})]^{\gamma_i}$. Hence, the calculation of derivatives with respect to γ_i is tractable since we can incorporate the same forward-backward algorithm as that used in a standard CRF.

Then, the derivatives with respect to γ_j, which are the parameters of generative models, can be written as follows:

$$\frac{\partial \mathcal{L}^{\text{HySOL}}(\mathbf{\Gamma}|\mathbf{\Theta})}{\partial \gamma_j} = \sum_n \log p_j^G(\boldsymbol{x}^n, \boldsymbol{y}^n) - \sum_n E_{R(\mathcal{Y}|\boldsymbol{x}^n;\mathbf{\Lambda},\mathbf{\Theta},\mathbf{\Gamma})}\big[\sum_s \log V_{j,s}^G\big].$$

Again, the second term, which indicates the expectation of transition probabilities and symbol emission probabilities, can be rewritten in the form of a forward-backward algorithm in the same manner as γ_i, where the only difference is that $V_{i,s}^D$ is substituted by $V_{j,s}^G$ in Equation (7).

To estimate $\mathbf{\Theta}^{(t+1)}$, which is procedure 4.1 in Figure 1, the same forward-backward algorithm as used in standard HMMs is available since the form of our Q-function shown in Equation (5) is the same as that of standard HMMs. The only difference is that our method uses marginal probabilities given by R instead of the $p(\boldsymbol{x}, \boldsymbol{y}; \boldsymbol{\theta})$ of standard HMMs.

Therefore, only a forward-backward algorithm is required for the efficient calculation of our parameter estimation process. Note that even though our hybrid model supports the use of a combination of several generative and discriminative models, we only need to calculate the forward-backward algorithm once for each sample during optimization procedures 4.1 and 4.2. This means that the required number of executions of the forward-backward algorithm for our parameter estimation is independent of the number of models used in the hybrid model.

In addition, after training, we can easily merge all the parameter values in a single parameter vector. This means that we can simply employ the Viterbi-algorithm for evaluating unseen samples, as well as that of standard CRFs, without any additional cost.

4 Experiments

We examined our hybrid model (HySOL) by applying it to two sequence labeling tasks, named entity recognition (NER) and syntactic chunking (Chunking). We used the same Chunking and 'English' NER data as those used for the shared tasks of CoNLL-2000 (Tjong Kim Sang and Buchholz, 2000) and CoNLL-2003 (Tjong Kim Sang and Meulder, 2003), respectively.

For the baseline method, we performed a conditional random field (CRF), which is exactly the same training procedure described in (Sha and Pereira, 2003) with L-BFGS. Moreover, LOP-CRF (Smith et al., 2005) is also compared with our hybrid model, since the formalism of our hybrid model can be seen as an extension of LOP-CRFs as described in Section 3. For CRF, we used the Gaussian prior as the second term on the RHS in Equation (1), where δ^2 represents the hyper-parameter in the Gaussian prior. In contrast, for LOP-CRF and HySOL, we used the Dirichlet priors as the second term on the

$\boldsymbol{\lambda}_1$	f(word$_s$), f(lword$_s$), f(pos$_s$), f(wtype$_s$), f(pos$_{s-1}$, pos$_s$), f(wtype$_{s-1}$, wtype$_s$), f(pos$_s$, pos$_{s+1}$), f(wtype$_s$, wtype$_{s+1}$), f(pref1$_s$), f(pref2$_s$), f(pref3$_s$), f(pref4$_s$), f(suf1$_s$), f(suf2$_s$), f(suf3$_s$), f(suf4$_s$)
$\boldsymbol{\lambda}_2$	f(word$_s$), f(lword$_s$), f(pos$_s$), f(wtype$_s$), f(word$_{s-1}$), f(lword$_{s-1}$), f(pos$_{s-1}$), f(wtype$_{s-1}$), f(word$_{s-2}$), f(lword$_{s-2}$), f(pos$_{s-2}$), f(wtype$_{s-2}$), f(pos$_{s-2}$, pos$_{s-1}$), f(wtype$_{s-2}$, wtype$_{s-1}$)
$\boldsymbol{\lambda}_3$	f(word$_s$), f(lword$_s$), f(pos$_s$), f(wtype$_s$), f(word$_{s+1}$), f(lword$_{s+1}$), f(pos$_{s+1}$), f(wtype$_{s+1}$), f(word$_{s+2}$), f(lword$_{s+2}$), f(pos$_{s+2}$), f(wtype$_{s+2}$), f(pos$_{s+1}$, pos$_{s+2}$), f(wtype$_{s+1}$, wtype$_{s+2}$)
$\boldsymbol{\lambda}_4$	all of the above

lword : lowercase of word, wtype : 'word type'
pref1-4: 1-4 character prefix of word
suf1-4 : 1-4 character suffix of word

Table 1: Features used in NER experiments

RHS in Equations (3), and (4), where ξ and η are the hyper-parameters in each Dirichlet prior.

4.1 Named Entity Recognition Experiments

The English NER data consists of 203,621, 51,362 and 46,435 words from 14,987, 3,466 and 3,684 sentences in training, development and test data, respectively, with four named entity tags, PERSON, LOCATION, ORGANIZATION and MISC, plus the 'O' tag. The unlabeled data consists of 17,003,926 words from 1,029,122 sentences. These data sets are exactly the same as those provided for the shared task of CoNLL-2003.

We slightly extended the feature set of the supplied data by adding feature types such as 'word type', and word prefix and suffix. Examples of 'word type' include whether the word is capitalized, contains digit or contains punctuation, which basically follows the baseline features of (Sutton et al., 2006) without regular expressions. Note that, unlike several previous studies, we did not employ additional information from external resources such as gazetteers. All our features can be automatically extracted from the supplied data.

For LOP-CRF and HySOL, we used four base discriminative models trained by CRFs with different feature sets. Table 1 shows the feature sets we used for training these models. The design of these feature sets was derived from a suggestion in (Smith et al., 2005), which exhibited the best performance in the several feature division. Note that the CRF for the comparison method was trained by using all feature types, namely the same as $\boldsymbol{\lambda}_4$.

As we explained in Section 3.3, for training HySOL, the parameters of four discriminative models, $\boldsymbol{\Lambda}$, were trained from 4/5 of the labeled training data, and $\boldsymbol{\Gamma}$ were trained from remaining 1/5. For the features of the generative models, we used all of the feature types shown in Figure 1. Note that one feature type corresponds to one HMM. Thus, each HMM maintains to consist of a non-overlapping feature set since each feature type only generates one symbol per state.

$\boldsymbol{\lambda}_1$	f(word$_s$), (pos$_s$), f(word$_{s-1}$, word$_s$), f(pos$_{s-1}$, pos$_s$), f(word$_s$, word$_{s+1}$), f(pos$_s$, pos$_{s+1}$)
$\boldsymbol{\lambda}_2$	f(word$_s$), (pos$_s$), f(word$_{s-1}$), f(pos$_{s-1}$), f(word$_{s-2}$), f(pos$_{s-2}$), f(word$_{s-2}$, word$_{s-1}$), f(pos$_{s-2}$, pos$_{s-1}$)
$\boldsymbol{\lambda}_3$	f(word$_s$), (pos$_s$), f(word$_{s+1}$), f(pos$_{s+1}$), f(word$_{s+2}$), f(pos$_{s+2}$), f(word$_{s+1}$, word$_{s+2}$), f(pos$_{s+1}$, pos$_{s+2}$)
$\boldsymbol{\lambda}_4$	all of the above

Table 2: Features used in Chunking experiments

4.2 Syntactic Chunking Experiments

CoNLL-2000 Chunking data was obtained from the Wall Street Journal (WSJ) corpus: sections 15-18 as training data (8,936 sentences and 211,727 words), and section 20 as test data (2,012 sentences and 47,377 words), with 11 different chunk-tags, such as NP and VP plus the 'O' tag, which represents the region outside any target chunk.

For LOP-CRF and HySOL, we also used four base discriminative models trained by CRFs with different feature sets. Table 2 shows the feature set we used in the Chunking experiments. We used the feature set of the supplied data without any extension of additional feature types.

To train HySOL, we used the same unlabeled data as used for our NER experiments (17,003,926 words from the Reuters corpus). Moreover, the division of the labeled training data and the feature set of the generative models were derived in the same manner as our NER experiments (see Section 4.1). That is, we divided the labeled training data into 4/5 for estimating $\boldsymbol{\Lambda}$ and 1/5 for estimating $\boldsymbol{\Gamma}$; one feature type shown in Table 2 is assigned in one generative model.

methods	(hyper-params)	$F_{\beta=1}$	(gain)	Sent	(gain)
CRF	(δ^2=100.0)	84.70	-	78.30	-
(4/5 labeled data, δ^2=100.0)		83.74	(-0.96)	77.06	(-1.24)
LOP-CRF	(ξ'=0.1)	84.90	(+0.20)	79.02	(+0.72)
HySOL	(ξ'=0.1,η'=0.0001)	**87.20**	(+2.50)	**81.19**	(+2.89)
	(w/o prior)	86.86	(+2.16)	80.75	(+2.45)
w/o $p_j^G \forall j$	(ξ'=1.0)	84.56	(-0.14)	78.23	(-0.07)

Table 3: NER performance (CoNLL-2003)

methods	(hyper-params)	$F_{\beta=1}$	(gain)	Sent	(gain)
CRF	(δ^2=10.0)	93.87	-	59.84	-
(4/5 labeled data, δ^2=10.0)		93.70	(-0.17)	58.85	(-0.99)
LOP-CRF	(ξ'=0.1)	93.91	(+0.04)	60.34	(+0.50)
HySOL	(ξ'=1.0,η'=0.0001)	**94.30**	(+0.43)	**61.73**	(+1.89)
	(w/o prior)	94.17	(+0.30)	61.23	(+1.39)
w/o $p_j^G \forall j$	(ξ'=1.0)	93.84	(-0.03)	59.74	(-0.10)

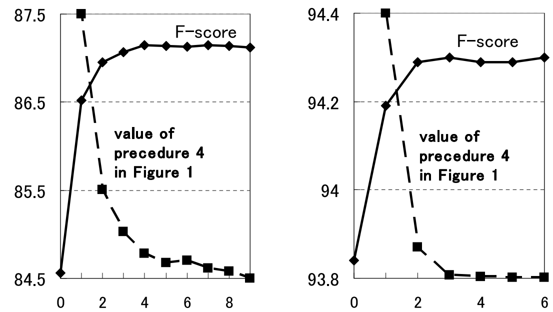
Table 4: Chunking performance (CoNLL-2000)

5 Results and Discussion

We evaluated the performance in terms of the $F_{\beta=1}$ score, which is the evaluation measure used in CoNLL-2000 and 2003, and sentence accuracy, since all the methods in our experiments optimize sequence loss. Tables 3 and 4 show the results of the NER and Chunking experiments, respectively. The $F_{\beta=1}$ and 'Sent' columns show the performance evaluated using the $F_{\beta=1}$ score and sentence accuracy, respectively. δ^2, ξ and η, which are the hyper-parameters in Gaussian or Dirichlet priors, are selected from a certain value set by using a development set[1], that is, $\delta^2 \in \{0.01, 0.1, 1, 10, 100, 1000\}$, $\xi - 1 = \xi' \in \{0.01, 0.1, 1, 10\}$ and $\eta - 1 = \eta' \in \{0.00001, 0.0001, 0.001, 0.01\}$. The second rows of CRF in Tables 3 and 4 represent the performance of base discriminative models used in HySOL with all the features, which are trained with 4/5 of the labeled training data. The third rows of HySOL show the performance obtained without using generative models (unlabeled data). The model itself is essentially the same as LOP-CRFs. However the performance in the third HySOL rows was consistently lower than that of LOP-CRF since the discriminative models in HySOL are trained with 4/5 labeled data.

As shown in Tables 3 and 4, HySOL significantly improved the performance of supervised setting, CRF and LOP-CRF, as regards both NER and Chunking experiments.

[1]Chunking (CoNLL-2000) data has no common development set. Thus, our preliminary examination employed by using 4/5 labeled training data with the remaining 1/5 as development data to determine the hyper-parameter values.



(a) NER (b) Chunking

Figure 2: Changes in the performance and the convergence condition value (procedure 4 in Figure 1) of HySOL.

5.1 Impact of Incorporating Unlabeled Data

The contributions provided by incorporating unlabeled data in our hybrid model can be seen by comparison with the performance of the first and third rows in HySOL, namely a 2.64 point F-score and a 2.96 point sentence accuracy gain in the NER experiments and a 0.46 point F-score and a 1.99 point sentence accuracy gain in the Chunking experiments.

We believe there are two key ideas that enable the unlabeled data in our approach to exhibit this improvement compared with the the state-of-the-art performance provided by discriminative models in supervised settings. First, unlabeled data is only used for optimizing Equation (4) to obtain a similar effect to 'soft-clustering', which can be calculated without information about the correct output. Second, by using a combination of generative models, we can enhance the flexibility of the feature design for unlabeled data. For example, we can handle arbitrary overlapping features, similar to those used in discriminative models, for unlabeled data by assigning one feature type for one generative model as in our experiments.

5.2 Impact of Iterative Parameter Estimation

Figure 2 shows the changes in the performance and the convergence condition value of HySOL during parameter estimation iteration in our NER and Chunking experiments, respectively. As shown in the figure, HySOL was able to reach the convergence condition in a small number of iterations in our experiments. Moreover, the change in the performance remains quite stable during the iteration. However, theoretically, our optimization procedure is not guaranteed to converge in the $\mathbf{\Gamma}$ and $\mathbf{\Theta}$ space, since the optimization of $\mathbf{\Theta}$ has local maxima. Even if we were unable to meet the convergence condition, we were easily able to obtain model parameters by performing a sufficient fixed number of iterations, and then select the parameters when Equation (4) obtained the maximum objective value.

5.3 Comparison with SS-CRF-MER

When we consider semi-supervised SOL methods, SS-CRF-MER (Jiao et al., 2006) is the most competitive with HySOL, since both methods are defined based on CRFs. We planned to compare the performance with that of SS-CRF-MER in our NER and Chunking experiments. Unfortunately, we failed to implement SS-CRF-MER since it requires the use of a slightly complicated algorithm, called the 'nested' forward-backward algorithm.

Although, we cannot compare the performance, our hybrid approach has several good characteristics compared with SS-CRF-MER. First, it requires a higher order algorithm, namely a 'nested' forward-backward algorithm, for the parameter estimation of unlabeled data whose time complexity is $O(L^3S^2)$ for each unlabeled data, where L and S represent the output label size and unlabeled sample length, respectively. Thus, our hybrid approach is more scalable for the size of unlabeled data, since HySOL only needs a standard forward-backward algorithm whose time complexity is $O(L^2S)$. In fact, we still have a question as to whether SS-CRF-MER is really scalable in practical time for such a large amount of unlabeled data as used in our experiments, which is about 680 times larger than that of (Jiao et al., 2006). Scalability for unlabeled data will become really important in the future, as it will be natural to use millions or billions of unlabeled data for further improvement. Second, SS-CRF-MER has a sensitive hyper-parameter in the objective function, which controls the influence of the unlabeled data. In contrast, our objective function only has a hyper-parameter of prior distribution, which is widely used for standard MAP estimation. Moreover, the experimental results shown in Tables 3 and

	$F_{\beta=1}$	additional resources
ASO-semi (Ando and Zhang, 2005)	89.31	unlabeled data (27M words)
(Florian et al., 2003)	88.76	their own large gazetteers, 2M-word labeled data
(Chieu and Ng, 2003)	88.31	their own large gazetteers, very elaborated features
HySOL	**88.14**	unlabeled data (17M words) supplied gazetters
HySOL	**87.20**	unlabeled data (17M words)

Table 5: Previous top systems in NER (CoNLL-2003) experiments

	$F_{\beta=1}$	additional resources
ASO-semi (Ando and Zhang, 2005)	94.39	unlabeled data (15M words: WSJ)
HySOL	**94.30**	unlabeled data (17M words: Reuters)
(Zhang et al., 2002)	94.17	full parser output
(Kudo and Matsumoto, 2001)	93.91	–

Table 6: Previous top systems in Chunking (CoNLL-2000) experiments

4 indicate that HySOL is rather robust with respect to the hyper-parameter since we can obtain fairly good performance without a prior distribution.

5.4 Comparison with Previous Top Systems

With respect to the performance of NER and Chunking tasks, the current best performance is reported in (Ando and Zhang, 2005), which we refer to as 'ASO-semi', as shown in Figures 5 and 6. ASO-semi also incorporates unlabeled data solely for the additional information in the same way as our method. Unfortunately, our results could not reach their level of performance, although the size and source of the unlabeled data are not the same for certain reasons. First, (Ando and Zhang, 2005) does not describe the unlabeled data used in their NER experiments in detail, and second, we are not licensed to use the TREC corpus including WSJ unlabeled data that they used for their Chunking experiments (training and test data for Chunking is derived from WSJ). Therefore, we simply used the supplied unlabeled data of the CoNLL-2003 shared task for both NER and Chunking. If we consider the advantage of our approach, our hybrid model incorporating generative models seems rather intuitive, since it is sometimes difficult to find out a design of effective auxiliary problems for the target problem.

Interestingly, the additional information obtained

	$F_{\beta=1}$	(gain)
HySOL (ξ'=0.1,η'=0.0001)	87.20	-
+ w/ F-score opt. (Suzuki et al., 2006)	88.02	(+0.82)
+ unlabeled data (17M → 27M words)	88.41	(+0.39)
+ supplied gazetters	88.90	(+0.49)
+ add dev. set for estimating $\mathbf{\Gamma}$	89.27	(+0.37)

Table 7: The HySOL performance with the F-score optimization technique and some additional resources in NER (CoNLL-2003) experiments

	$F_{\beta=1}$	(gain)
HySOL (ξ'=0.1,η'=0.0001)	94.30	-
+ w/ F-score opt. (Suzuki et al., 2006)	94.36	(+0.06)

Table 8: The HySOL performance with the F-score optimization technique on Chunking (CoNLL-2000) experiments

from unlabeled data appear different from each other. ASO-semi uses unlabeled data for constructing auxiliary problems to find the 'shared structures' of auxiliary problems that are expected to improve the performance of the main problem. Moreover, it is possible to combine both methods, for example, by incorporating the features obtained with their method in our base discriminative models, and then construct a hybrid model using our method. Therefore, there may be a possibility of further improving the performance by this simple combination.

In NER, most of the top systems other than ASO-semi boost performance by employing external hand-crafted resources such as large gazetteers. This is why their results are superior to those obtained with HySOL. In fact, if we simply add the gazetteers included in CoNLL-2003 supplied data as features, HySOL achieves 88.14.

5.5 Applying F-score Optimization Technique

In addition, we can simply apply the F-score optimization technique for the sequence labeling tasks proposed in (Suzuki et al., 2006) to boost the HySOL performance since the base discriminative models $p^D(\boldsymbol{y}|\boldsymbol{x})$ and discriminative combination, namely Equation (3), in our hybrid model basically uses the same optimization procedure as CRFs. Tables 7 and 8 show the F-score gain when we apply the F-score optimization technique. As shown in the Tables, the F-score optimization technique can easily improve the (F-score) performance without any additional resources or feature engineering.

In NER, we also examined HySOL with additional resources to observe the performance gain. The third row represents the performance when we add approximately 10M words of unlabeled data (total 27M words)[2] that are derived from 1996/11/15-30 articles in Reuters corpus. Then, the fourth and fifth rows represent the performance when we add the supplied gazetters in the CoNLL-2003 data as features, and adding development data as training data of $\mathbf{\Gamma}$. In this case, HySOL achieved a comparable performance to that of the current best system, ASO-semi, in both NER and Chunking experiments even though the NER experiment is not a fair comparison since we added additional resources (gazetters and dev. set) that ASO-semi does not use in training.

6 Conclusion and Future Work

We proposed a framework for semi-supervised SOL based on a hybrid generative and discriminative approach. Experimental results showed that incorporating unlabeled data in a generative manner has the power to further improve on the state-of-the-art performance provided by supervised SOL methods such as CRFs, with the help of our hybrid approach, which discriminatively combines with discriminative models. In future we intend to investigate more appropriate model and feature design for unlabeled data, which may further improve the performance achieved in our experiments.

Appendix

Let $V^D_{i,s} = \exp(\boldsymbol{\lambda} \cdot \boldsymbol{f}_s)$ and $V^G_{j,s} = \theta_{y_{s-1},y_s}\theta_{y_s,x_s}$. Equation (6) can be obtained by the following rearrangement of Equation (2) :

$$
\begin{aligned}
&R(\boldsymbol{y}|\boldsymbol{x};\boldsymbol{\Lambda},\boldsymbol{\Theta},\boldsymbol{\Gamma})\\
&= \frac{\prod_i p^D_i(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\lambda}_i)^{\gamma_i}\prod_j p^G_j(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\theta}_j)^{\gamma_j}}{\sum_{\boldsymbol{y}}\prod_i p^D_i(\boldsymbol{y}|\boldsymbol{x},\boldsymbol{\lambda}_i)^{\gamma_i}\prod_j p^G_j(\boldsymbol{x},\boldsymbol{y},\boldsymbol{\theta}_j)^{\gamma_j}}\\
&= \frac{1}{\mathcal{N}_R(\boldsymbol{x})}\prod_i\Big[\frac{\prod_s V^D_{i,s}}{Z_i(\boldsymbol{x})}\Big]^{\gamma_i}\prod_j\Big[\prod_s V^G_{j,s}\Big]^{\gamma_j}\\
&= \frac{1}{\mathcal{N}_R(\boldsymbol{x})\prod_i[Z_i(\boldsymbol{x})]^{\gamma_i}}\prod_i\Big[\prod_s V^D_{i,s}\Big]^{\gamma_i}\prod_j\Big[\prod_s V^G_{j,s}\Big]^{\gamma_j}\\
&= \frac{1}{\mathcal{N}_R(\boldsymbol{x})\prod_i[Z_i(\boldsymbol{x})]^{\gamma_i}}\prod_s\prod_i[V^D_{i,s}]^{\gamma_i}\prod_j[V^G_{j,s}]^{\gamma_j}.
\end{aligned}
$$

[2] In order to keep the consistency of POS tags, we re-attached POS tags of the supplied data set and new 10M words of unlabeled data using a POS tagger trained from WSJ corpus.

References

Y. Altun, D. McAllester, and M. Belkin. 2005. Maximum Margin Semi-Supervised Learning for Structured Variables. In *Proc. of NIPS*2005*.

R. Ando and T. Zhang. 2005. A High-Performance Semi-Supervised Learning Method for Text Chunking. In *Proc. of ACL-2005*, pages 1–9.

U. Brefeld and T. Scheffer. 2006. Semi-Supervised Learning for Structured Output Variables. In *Proc. of ICML-2006*.

H. L. Chieu and Hwee T. Ng. 2003. Named Entity Recognition with a Maximum Entropy Approach. In *Proc. of CoNLL-2003*, pages 160–163.

A. P. Dempster, N. M. Laird, and D. B. Rubin. 1977. Maximum Likelihood from Incomplete Data via the EM Algorithm. *Journal of the Royal Statistical Society, Series B*, 39:1–38.

R. Florian, A. Ittycheriah, H. Jing, and T. Zhang. 2003. Named Entity Recognition through Classifier Combination. In *Proc. of CoNLL-2003*, pages 168–171.

A. Fujino, N. Ueda, and K. Saito. 2005. A Hybrid Generative/Discriminative Approach to Semi-Supervised Classifier Design. In *Proc. of AAAI-05*, pages 764–769.

Y. Grandvalet and Y. Bengio. 2004. Semi-Supervised Learning by Entropy Minimization. In *Proc. of NIPS*2004*, pages 529–536.

F. Jiao, S. Wang, C.-H. Lee, R. Greiner, and D. Schuurmans. 2006. Semi-Supervised Conditional Random Fields for Improved Sequence Segmentation and Labeling. In *Proc. of COLING/ACL-2006*, pages 209–216.

T. Kudo and Y. Matsumoto. 2001. Chunking with Support Vector Machines. In *Proc. of NAACL 2001*, pages 192–199.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data. In *Proc. of ICML-2001*, pages 282–289.

W. Li and A. McCallum. 2005. Semi-Supervised Sequence Modeling with Syntactic Topic Models. In *Proc. of AAAI-2005*, pages 813–818.

D. C. Liu and J. Nocedal. 1989. On the Limited Memory BFGS Method for Large Scale Optimization. *Math. Programming, Ser. B*, 45(3):503–528.

K. Nigam, A. McCallum, S. Thrun, and T. Mitchell. 2000. Text Classification from Labeled and Unlabeled Documents using EM. *Machine Learning*, 39:103–134.

R. Raina, Y. Shen, A. Y. Ng, and A. McCallum. 2003. Classification with Hybrid Generative/Discriminative Models. In *Proc. of NIPS*2003*.

F. Sha and F. Pereira. 2003. Shallow Parsing with Conditional Random Fields. In *Proc. of HLT/NAACL-2003*, pages 213–220.

A. Smith, T. Cohn, and M. Osborne. 2005. Logarithmic Opinion Pools for Conditional Random Fields. In *Proc. of ACL-2005*, pages 10–17.

C. Sutton, M. Sindelar, and A. McCallum. 2006. Reducing Weight Undertraining in Structured Discriminative Learning. In *Proc. of HTL-NAACL 2006*, pages 89–95.

J. Suzuki, E. McDermott, and H. Isozoki. 2006. Training Conditional Random Fields with Multivariate Evaluation Measure. In *Proc. of COLING/ACL-2006*, pages 217–224.

E. F. Tjong Kim Sang and S. Buchholz. 2000. Introduction to the CoNLL-2000 Shared Task: Chunking. In *Proc. of CoNLL-2000 and LLL-2000*, pages 127–132.

E. T. Tjong Kim Sang and F. De Meulder. 2003. Introduction to the CoNLL-2003 Shared Task: Language-Independent Named Entity Recognition. In *Proc. of CoNLL-2003*, pages 142–147.

T. Zhang, F. Damerau, and D. Johnson. 2002. Text Chunking based on a Generalization of Winnow. *Machine Learning Research*, 2:615–637.

X. Zhu, Z. Ghahramani, and J. Lafferty. 2003. Semi-Supervised Learning using Gaussian Fields and Harmonic Functions. In *Proc.of ICML-2003*, pages 912–919.