Co-Training of Conditional Random Fields for Segmenting Sequence Data

Xuan-Hieu Phan, Le-Minh Nguyen, and Yasushi Inoguchi
Graduate School of Information Science
Japan Advanced Institute of Science and Technology
1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan
{hieuxuan, nguyenml, inoguchi}@jaist.ac.jp

ABSTRACT

This paper presents a semi-supervised co-training approach for discriminative sequential learning models, such as conditional random fields (CRFs). In this framework, different CRF models are trained on an initial set of sequence data according different views. The bootstrapping process is performed by iteratively adding new reliably inferred data sequences to the training data sets of CRF models retraining them. Reliable data sequences are inferred from a huge set of unlabeled data by estimating entropy values of predicted labels at time positions in data sequences. The inference and re-train operations are repeated a number of times in order that each CRF model should gain as much useful evidence from unlabeled data and the other CRF models as possible. The proposed method was tested on noun phrase chunking and achieved significant results.

Keywords: semi-supervised learning, co-training, conditional random fields, text labeling and segmentation.

1. INTRODUCTION

Learning from both labeled and unlabeled data, also known as semi-supervised learning, has received much attention of machine learning and data mining communities during the past few years. There have been many existing semi-supervised learning approaches to the traditional classification such as co-training [1] [2], Gaussian mixture models with EM [3], minimizing separation (transductive SVMs, Gaussian processes information regularization) [4], and graph-based methods [5].

Recently, there is a subdirection of semi-supervised learning that focuses on sequential modeling models, such as HMMs [6] and CRFs [7]. To gain additional benefit from unlabeled data for POS tagging and word segmentation, Li and McCallum [8] presented a clustering method to partition words into different syntactic and semantic topics based on word's content and their surrounding context. Those clusters were then used as input features for training CRFs from a huge set of unlabeled words. Although this method showed a significant improvement in accuracy, the approach tends to be task and data-dependent. Lafferty et al. [9] introduced kernel conditional random fields for semi-supervised learning. This model can learn from unlabeled data by relying on the similarities between labeled and unlabeled observations using kernel functions. Brefeld et al. [10] presented a multi-view discriminative sequential learning method that is based on the principle of maximizing the consensus among multiple independent hypotheses. Other semi-supervised learning methods focus on sequential labeling for text data, such as unsupervised models for named entity recognition [11], semi-supervised learning from thousands of auxiliary classification problems [12], and contrastive estimation for log-linear models [13]. Those models are more or less domain and task-dependent, and thus have some difficulties when being applied to other sequential learning applications.

In this paper, we present a semi-supervised learning method for CRFs that is based on co-training philosophy [1], i.e., try to gain extra useful information/evidence from unlabeled data by relying on the agreement among different hypotheses. Technically, we have different CRFs models trained according to different views on the small initial set of labeled data. Those models are bootstrapped by being iteratively re-trained on additional confident labeled data sequences inferred from a huge set of unlabeled data. The selection of confident data sequences is performed by estimating entropy values of predicted labels at time positions in every sequence. Sequences with small entropy values for one CRF models should be confident and can be used to train the others in the next step. In addition, some confident sequences can be re-corrected from unconfident ones, and very useful for the bootstrapping process. The re-correction operation is based not oly on the entropy values but also on the consensus of independent CRFs.

The main advantages of the proposed semi-supervised learning method are threefold. First, this method dedicated to discriminative models rather than generative ones. Second, it is easy for implementation because it is only based on simple entropy estimation. Finally, the method is task and domain independent, i.e., one can apply this method with CRFs for any sequential learning applica-

tion and for any kind of data provided that the learning task can be separated into different views.

The remaining of the paper is organized as follows. Section 2 briefly introduces sequential learning with CRFs. Section 3 presents the proposed co-training method for CRFs. Section 4 presents empirical evaluation and some discussion. Finally, conclusions are given in Section 6.

2. SEGMENTING FOR SEQUENCE DATA WITH CONDITIONAL RANDOM FIELDS

The goal of labeling/segmenting for sequence data is to learn to map observation sequences to their corresponding label sequences, e.g., a POS tag sequence for words in a sentence. Discriminative sequential modeling models, such as CRFs [7] and Discriminative HMMs [14], were particularly designed for such sequential learning applications. In this paper, CRFs are referred to as conditionally-trained finite state machines and will be used to demonstrate our co-training method.

2. 1. Conditional Random Fields

Let $o = \{o_1, o_2, \ldots, o_T\}$ be some observation sequence. Let **S** be a set of states, each of which is associated with a label, $l \in \mathbf{L}$. Let $s = \{s_1, s_2, \ldots, s_T\}$ be some state sequence, Lafferty et al. [7] define CRF as the conditional probability of a state sequence *s* given data observation sequence *o* as follows,

$$p_\theta(s \mid o) = \frac{1}{Z(o)} \exp\left(\sum_{t=1}^{T} \sum_{k} \lambda_k f_k(s_{t-1}, s_t, o, t) \right) \quad (1)$$

where $Z(o)$ is the normalization summing over all label sequences. f_k denotes a feature function in the language of maximum entropy modeling and λ_k is a learned weight associated with feature f_k. Each f_k is either a transition or a per-state feature:

$$f_k^{(per-state)}(s_t, o, t) = \delta(s_t, l) x_k(o, t) \quad (2)$$

$$f_k^{(transition)}(s_{t-1}, s_t, t) = \delta(s_{t-1}, l')\delta(s_t, l) \quad (3)$$

where δ denotes the Kronecker-δ function. A per-state feature (2) combines label *l* of the current state s_t and a characteristic (sometimes called "context predicate") of of the observation sequence *o* at time position *t*. For example, the label of the current state is JJ (adjective) and the current word is "sequential". A transition feature (3) represents a sequential dependency by combining the label *l'* of the previous state s_{t-1} and the label *l* of the current state s_t, such as the previous label *l'* = JJ (adjective) and the current label *l* = NN (noun).

2. 2. Inference in Conditional Random Fields

Inference in CRFs is to find the most likely state/label sequence *s** given an observation *o*:

$$s^* = \arg\max_s p_\theta(s \mid o)$$
$$= \arg\max_s \left\{ \exp\left(\sum_{t=1}^{T} \sum_{k} \lambda_k f_k(s_{t-1}, s_t, o, t) \right) \right\} \quad (4)$$

In order to find *s**, one can apply a dynamic programming technique with a slightly modified version of the original Viterbi algorithm for HMMs [6]. To avoid an exponential-time search over all possible settings of *s*, Viterbi stores the probability of the most likely path up to time *t* which accounts for the first *t* observations and ends in state s_i. We denote this probability to be $\varphi_t(s_i)$ ($0 \le t \le T\text{-}1$). The recursion is given by:

$$\varphi_{t+1}(s_i) = \max_{s_j} \left\{ \varphi_t(s_j) \exp\left(\sum_{k} \lambda_k f_k(s_j, s_i, o, t) \right) \right\} \quad (5)$$

The recursion terminates when *t* = T-1 and the biggest unnormalized probability is $p^* = \text{argmax}_i\{\varphi_T(s_i)\}$. At this time, we can backtrack through the stored information to find the most likely label sequence *s**.

2. 3. Training Conditional Random Fields

CRFs are trained by setting the weight set $\theta = \{\lambda_1, \ldots\}$ to maximize the log-likelihood function, ***L***, of a given training dataset $\mathbf{D} = \{(o^j, l^j)\}_{j=1..N}$:

$$L = \sum_{j=1}^{N} \log\left(p_\theta(l^j \mid o^j) \right) - \sum_{k} \frac{\lambda_k^2}{2\sigma^2} \quad (6)$$

where the second sum is a Gaussian prior over weight set with variance σ^2, which provides smoothing to deal with sparsity in data [15]. It has been proved that the above log-likelihood function is convex, thus searching the global optimum is guaranteed [16]. However, the optimum cannot be found analytically. The parameter estimation requires an iterative procedure. It has been shown that quasi-Newton methods, such as L-BFGS [17], are more efficient than the others [18] [19]. This method can avoid the explicit estimation of the Hessian matrix of the log-likelihood by building up an approximation of it using successive evaluations of the gradient.

3. CO-TRAINING OF CONDITIONAL RANDOM FIELDS

3. 1. Co-training Framework for CRFs

The co-training framework for CRFs is similar to the general co-training framework for classification problem [1]. Initially, *k* CRF models (CRF_1, …, CRF_k) are trained according to different and independent views on the same small set of labeled data $\mathbf{D}_L$. The selection of independent views will be discussed latter. The CRF models are then bootstrapped by the co-training procedure as follows. First, all CRF models are used to predict labels for unlabeled data set $\mathbf{D}_U$. Then, we choose a subset of confidently predicted data sequences from $\mathbf{D}_U$ to add to the training sets of those CRF models and retrain them. This procedure will be performed repeatedly several times so that useful information from unlabeled data is utilized. The final CRF models are expected to predict labels for sequence data better than those trained on the original set of labeled data. The key step in our method is how to identify reliably predicted data sequences from unlabeled data to enrich the training set of the next co-training iteration. This key problem will be thoroughly discussed in the next subsection.

3. 2. Entropy-Based Estimation of Reliably Inferred Sequence Data in CRFs

Table 1. An example of a reliably inferred observation sequence based on entropy estimation

True Label	Predicted Label	Entropy $H(o_t)$	$Path_1$ 0.978	$Path_2$ 0.012	$Path_3$ 0.006	$Path_4$ 0.002	...	Observation sequence	
								Word	POS-tag
B-NP	B-NP	0.0	B-NP	B-NP	B-NP	B-NP	...	Other	JJ
I-NP	I-NP	0.0	I-NP	I-NP	I-NP	I-NP	...	changes	NNS
O	O	0.0	O	O	O	O	...	,	,
O	O	0.0	O	O	O	O	...	including	VBG
O	O	0.0061	O	O	O	B-NP	...	Easing	VBG
B-NP	B-NP	0.0052	B-NP	B-NP	B-NP	I-NP	...	restrictions	NNS
O	O	0.0	O	O	O	O	...	on	IN
B-NP	B-NP	0.0	B-NP	B-NP	B-NP	B-NP	...	travel	NN
O	O	0.0	O	O	O	O	...	for	IN
B-NP	B-NP	0.0	B-NP	B-NP	B-NP	B-NP	...	East	NNP
I-NP	I-NP	0.0137	I-NP	I-NP	O	I-NP	...	Germans	NNPS
O	O	0.0227	O	I-NP	O	O	...	,	,
O	O	0.0002	O	O	O	O	...	are	VBP
O	O	0.0	O	O	O	O	...	expected	VBN
O	O	0.0	O	O	O	O	...	.	.

Table 2. An example of an unreliably inferred observation sequence based on entropy estimation

True Label	Predicted Label	Entropy $H(o_t)$	$Path_1$ 0.774	$Path_2$ 0.101	$Path_3$ 0.057	$Path_4$ 0.048	...	Observation sequence	
								Word	POS-tag
O	O	0.0	O	O	O	O	...	However	RB
O	O	0.0	O	O	O	O	...	,	,
B-NP	B-NP	0.0	B-NP	B-NP	B-NP	B-NP	...	dealers	NNS
O	O	0.0	O	O	O	O	...	caution	VBP
O	O	0.0	O	O	O	O	...	that	IN
B-NP	B-NP	0.0	B-NP	B-NP	B-NP	B-NP	...	any	DT
I-NP	I-NP	0.0	I-NP	I-NP	I-NP	I-NP	...	increase	NN
O	O	0.0	O	O	O	O	...	would	MD
O	O	0.0	O	O	O	O	...	be	VB
B-NP	B-NP	0.0	B-NP	B-NP	B-NP	B-NP	...	$	$
I-NP	I-NP	0.0	I-NP	I-NP	I-NP	I-NP	...	1	CD
I-NP	O	0.1219	O	I-NP	O	O	...	to	TO
I-NP	B-NP	0.1196	B-NP	I-NP	B-NP	B-NP	...	$	$
I-NP	I-NP	0.0002	I-NP	I-NP	I-NP	I-NP	...	2	CD
O	O	0.0	O	O	O	O	...	at	IN
O	B-NP	0.0807	B-NP	B-NP	O	B-NP	...	most	RBS
O	O	0.0711	O	O	O	I-NP	...	.	.

This section discusses the selection of confidently inffered data sequences based on the entropy estimation of predicted labels at different time positions of unlabeled

data sequences. Confident sequences are those having small entropy values of predicted labels.

Let **L** = {l_1, l_2, …, l_Q} be the set of all possible class labels. Let $o = \{o_1, o_2, ..., o_T\}$ be some data observation sequence. Let $l^1 = \{l_1^1, l_2^1, ..., l_T^1\}$, $l^1 = \{l_1^2, l_2^2, ..., l_T^2\}$, ..., and $l^n = \{l_1^n, l_2^n, ..., l_T^n\}$ the be the n-best predicted label sequences (commonly known as n-best label paths with path values: p^1, p^2, …, p^n) for the observation sequence *o*. Table 1 shows an example of n-best label sequences in which the observation *o* consists of an English words (a sentence) and their POS tags. The problem is to predict a phrase chunk label (B-NP indicates the begin of a noun phrase, I-NP indicates inside of a noun phrase, and O indicates outside of a noun phrase) for each word in a sentence. We can see the best label path l^1 = {B-NP, I-NP, O, O, O, B-NP, O, B-NP, O, B-NP, I-NP, O, O, O, O} with the path value p^1 = 0.978. Similarly, the second path value p^2 = 0.012, the third path value p^3 = 0.006, etc. If n equals to N possible label paths of observation sequence *o*, then {p^1, p^2, …, p^N} will be a distribution, i.e., sum(p^1, p^2, …, p^N) = 1. However, in CRFs, n-best path values are much larger than the remaining ones and we can normalize so that {p^1, p^2, …, p^n} is a probabilistic distribution.

For each time position *t* (1 ≤ t ≤ T) in the observation sequence *o*, the portion of the label $l_i \in$ **L** that are assigned for the data observation o_t in the n-best paths is $P(l_i)$ and can be calculated as follows.

$$P(l_i, o_t) = \sum p^j (\forall l^j : l_t^j = l_i), \qquad (7)$$

Then, the entropy of predicted labels of the observation sequence *o* at the position *t* is defined as H(o_t):

$$\mathrm{H}(o_t) = -\sum_{i=1}^{Q} P(l_i, o_t) \log P(l_i, o_t), \qquad (8)$$

For the sake of simplicity, we normalize the entropy value H(o_t) (i.e., scaling it to [0, 1]) by dividing by log(Q) (the maximum entropy value).

For example, the observation o_5 (word = "Easing", POS-tag = "VBG") in Table 1 has the entropy value H(o_5) = 0.0061. In this example, we use 10-best (n = 10) label sequences and the number of label is Q = 3 (i.e., **L** = {B-NP, I-NP, O}). Similarly, H(o_6) = 0.0052 (there is at least a little bit change of the predicted label at the $Path_4$), H(o_7) = 0.0 (there is no change of the predicted label of the 10-best paths at the time position *t* = 7). In general, all observations of the observation sequence in Table 1 have small entropy values (the largest value is H(o_{12}) = 0.0227.

Table 2 shows another example in which entropy values are much larger than those in Table 1. For example, the observation o_{12} (word = "to", POS-tag = "TO") has the entropy value H(o_{12}) = 0.1219. This is because the best label path value ($Path_1$ = 0.774) is not confident enough and there is a major change in the predicted label at this position (l^1_{12} = O, l^2_{12} = I-NP, l^3_{12} = O, …).

Intuitively, H(o_t) measures the uncertainty of the predicted label of observation o_t. In other words, in general the uncertainty of the predicted label of o_t is high if H(o_t) is large. Let $l(o_t)$ be the predicted label in the best label path l^1 of the observation o_t. We have the following definition of "reliably predicted label":

Definition 1: label $l(o_t)$ is a "reliably predicted label" of the observation o_t if the corresponding entropy value H(o_t) is smaller than or equal to an entropy threshold H_{th}, i.e., H(o_t) ≤ H_{th}.

Based on definition (1), we have definition of "reliably inferred label sequence" below.

Definition 2: Let $l(o)$ be a predicted label sequence of observation sequence *o*. Then, $l(o)$ is called the "reliably inferred label sequence" if every label $l(o_t)$ of $l(o)$ is "reliable predicted label", i.e., H(o_t) ≤ H_{th} (1 ≤ t ≤ T).

For example, setting the threshold H_{th} = 0.06, the best label path ($Path_1$) in Table 1 is a reliably inferred label sequence because every H(o_t) ≤ 0.06. On the other hand, the best label path in Table 2 does not satisfy definition (2) because there are some time positions whose entropy values are larger than 0.06 (e.g., o_{12}, o_{13}, o_{16}, o_{17}). We also compare the best label sequence and the true label sequence (humans annotated labels) in both Table 1 and Table 2 in order to demonstrate the reasonableness of our assumption about the relationship between the entropy values and the confidence of predicted labels. We can see that the best label path in Table 1 is the same as the true label sequence while in the Table 2 the predicted labels with high entropy values (at o_{12}, o_{13}, o_{16}) are different from the true labels (I-NP | O; I-NP | B-NP; O | B-NP). In general, label sequences with small entropy values tend to be confident enough for retraining CRFs.

3. 3. Co-training Algorithm for CRFs

This section presents the co-training algorithm for CRF models. Let **CRFs** = {CRF_1, CRF_2, …, CRF_k} be *k* CRF models according to different and independent views. The next section will discuss how to select different

views for co-training CRFs. Let $\mathbf{D}_L = \{(o^i, l^i)\}_{i=1..L}$ be the initial training set of labeled sequence data. Let $\mathbf{D}_U = \{(o^j)\}_{j=L+1..U}$ be the huge set of unlabeled sequence data. The co-training algorithm for CRFs is presented in Table 3.

Table 3. Co-training algorithm for CRFs

In	**CRFs** = $\{CRF_1, CRF_2, \ldots, CRF_k\}$, $\mathbf{D}_U$, $\mathbf{D}_L$
Out	**CRFs** trained on both $\mathbf{D}_U$ and $\mathbf{D}_L$
0.	$\mathbf{D}_{Li} = \mathbf{D}_L$ (i = 1..k)
1.	Train CRF_i (i = 1..k) on $\mathbf{D}_{Li}$ independently
2.	Use trained CRF_i (i = 1..k) to predict n-best label sequences for all observation sequence in $\mathbf{D}_U$ to obtain $\mathbf{D}_{Ui.}$
3.	$\mathbf{D}_{Li} = \mathbf{D}_{Li} \cup \mathbf{ConfSeq_1}(\mathbf{D}_{Uj})$ (j = 1..k, j ≠ i)
4.	$\mathbf{D}_{Li} = \mathbf{D}_{Li} \cup \mathbf{ConfSeq_2}(\mathbf{D}_{U1}, \mathbf{D}_{U2}, \ldots, \mathbf{D}_{Uk})$
5.	If #iterations ≥ **I** Then stop Else go to step 1.

The algorithm first trains CRF models ($CRF_1, \ldots, CRF_k$) on the initial set of labeled sequence data $\mathbf{D}_{Li} = \mathbf{D}_L$ (i.e., step 1). In step 2, it uses the trained CRF models to predict n-best label paths for all observation sequences in $\mathbf{D}_U$ to obtain $\mathbf{D}_{Ui}$ (corresponding to CRF_i). Steps 3 and 4 try to gain confident (labeled) sequences from $\mathbf{D}_{Ui}$ to add to the labeled training set of each CRF_i. The first operation (step 3) is $\mathbf{ConfSeq_1}(\mathbf{D}_{Uj})$ (j = 1..k, j ≠ i). This means that it collects all reliably inferred sequences predicted by the other CRF models (CRF_j, j = 1..k, j ≠ i) and add to the labled training data set of the current model (i.e., CRF_i). After collecting all confident data sequences, the algorithm focuses on unreliable sequences: the second operation $\mathbf{ConfSeq_2}(\mathbf{D}_{U1}, \mathbf{D}_{U2}, \ldots, \mathbf{D}_{Uk})$. In this operation, we look entropy values generated by *k* CRF models for each "unreliable sequence" in order to utilize the significant difference in entropy values that derives from the independent views of those models. In other words, a label sequence may not confident when we examine the its entropy values generated by each CRF_i separately. However, we can re-correct its label sequences if looking concurrently to *k* entropy paths generated by *k* **CRFs** in order to obtain more "confident sequences" from unlabeled data $\mathbf{D}_U$. The second operation is very important because those confident sequences returned by this operation help the models to improve themselves very much. After gaining confident sequences from $\mathbf{D}_U$ and add to labeled data set $\mathbf{D}_{Li}$ for CRF_i, the algorithm check the stopping condition to stop, otherwise it goes to step 1 to re-train the CRF models on their new labeled data sets.

3. 4. Multi-View Representation for Co-training

The original work on co-training [1] proposed that one can use independent set of features for different and independent views. However, the feature set independence assumption is usually too restricted to obey. Thus, one can relax this assumption to a lower level: features are divided into subsets that are as much independent as possible.

We present another choice of multi-view representation for co-training. That is label representation. For many segmenting sequence data applications, we have the different choice for representing label sequence. For example, in NP chunking we have at least five choices.

	IOB1	**IOB2**	**IOE1**	**IOE2**	**Start/End**
In	O	O	O	O	O
early	I-NP	B-NP	I-NP	I-NP	B-NP
trading	I-NP	I-NP	I-NP	E-NP	E-NP
in	O	O	O	O	O
Busy	I-NP	B-NP	I-NP	I-NP	B-NP
Hong	I-NP	I-NP	I-NP	I-NP	I-NP
Kong	I-NP	I-NP	E-NP	E-NP	E-NP
Monday	B-NP	B-NP	I-NP	E-NP	S-NP
,	O	O	O	O	O
gold	I-NP	B-NP	I-NP	E-NP	S-NP
Was	O	O	O	O	O

IOB1 representation was first introduced in [20]. The others (IOB2, IOE1, IOE2) were introduced by Tjong Kim Sang [21]. The last style was introduced in [22]. These representation styles have been used for phrase chunking application. However, they can be applied for any kind of data and any kind of sequence segmentation applications.

IOB1: I (the current token is inside of a segment), O (the current token is outside of any segment), and B (current token is the beginning of a segment which immediately follows another segment). IOB2: a B tag is given for every token which exists at the beginning of a segment. Other tokens are the same as IOB1. IOE1: an E tag is used to mark the last token of a segment immediately preceding another segment. IOE2: an E tag is given for every token which exists at the end of a segment. Start/End: B (current token is the start of a segment consisting of more than one token), E (current token is the end of a segment consisting of more than one token), I (current token is a middle of a segment consisting of more than two tokens), S (current token is a segment consisting of only one token), and O (current token is outside of any segment).

Although these representation styles have been mainly used for phrase chunking, they should be useful and suitable for co-training because we believe that they should provide different views into training data set and

thus making a significant difference among CRFs. We used these representation styles for multi-view co-training of CRFs for noun phrase chunking problem.

4. EMPIRICAL EVALUATION

We evaluate our co-training method on noun phrase chunking problem. Noun phrase chunking, an intermediate step toward full parsing of natural language, identifies noun phrase (NP) in text sentences. Here is an example of a sentence with noun phrase marking: "[NP He] reckons [NP the current account deficit] will narrow to [NP only # 1.8 billion] in [NP September]".

4.1. Data

The training and testing data for this task is available at the shared task for CoNLL-2000. The data consist of the same sections of the WSJ corpus: section 15-18 as training data (8936 sentences, 211727 tokens) and section 20 as testing data (2012 sentences, 47377 tokens). Each line in the annotated data is for a token and consists of three columns: the token (a word or a punctuation mark), the POS tag of the token, and noun phrase label (label for short) of the token. The representation for label can be one of IOB1, IOB2, IOE1, IOE2, Start/End mentioned above. Two consecutive sequences (sentences) are separated by a blank line.

For co-training of CRFs, we divided the training set into 30 parts. Each part (297 sequences) can be used as the small original set of labeled data (i.e., $\mathbf{D}_L$). Another part was used as the development set to tune the entropy threshold (i.e., H_{th}). We removed the noun phrase labels of the remaining 28 parts and used these parts as unlabeled data set (i.e., $\mathbf{D}_U$). We keep the same testing set of CoNLL-2000 (i.e., the section 20 of WSJ) as the testing set of our CRF models.

4.2. Multi-view Representation for Co-training

We used four label representation styles IOB1, IOB2, IOE1, IOE2 for different CRFs (CRF_1, CRF_2, CRF_3, and CRF_4). The training data set of CRF_1, CRF_2, CRF_3, CRF_4 are $\mathbf{D}_{L1}$, $\mathbf{D}_{L2}$, $\mathbf{D}_{L3}$, $\mathbf{D}_{L4}$ and their label representation styles are IOB1, IOB2, IOE1, IOE2, respectively. All our CRF models obey the first-order Markov property, i.e., the current state only depends on the previous label.

4.3. Feature Selection for CRFs

We used the same feature selection for four CRFs. The transition features obey the first-order Markov property. Per-state features are the combinations of the label of the current state and one context predicate within a sliding window of size 5 (i.e., -2, -1, 0, 1, 2). Context predicates can be a token or POS tag within the sliding window, the combination of the current token and the previous token, the combination of the current token and the next token, the combination of two or three consecutive POS tags within the sliding window.

4.4. Results

Table 4 shows the results of the four CRF models using the proposed co-training algorithm. The first column is the number of co-training iterations. The next four large double-columns are corresponding to four CRF models. At each co-training iteration, the labeled training data set ($\mathbf{D}_{Li}$) of those models were added by selecting reliably inferred data sequences from unlabeled data set ($\mathbf{D}_U$).

We used a development set to tune the entropy threshold ($H_{th} = 0.06$). We can see that after three co-training iterations, the error rate decrease significantly (16.5%, 13.2%, 16.4%, and 19.0%). The phrase-based error rate reductions are around 15.0%. Four CRF models used around 7000 sequences from un- labeled data set in order to improve the learning performance

Table 4. Error rate reduction of four CRF models using co-training

Iteration	**CRF_1** (IOB1)		**CRF2** (IOB2)		**CRF_3** (IOE1)		**CRF_4** (IOE2)	
	$\mathbf{D}_{L1}$ #seq.	F_1 (%)	$\mathbf{D}_{L2}$ #seq.	F_1 (%)	$\mathbf{D}_{L3}$ #seq.	F_1 (%)	$\mathbf{D}_{L4}$ #seq.	F_1 (%)
0	297	96.43	297	95.21	297	96.35	297	95.32
1	3267	96.79	3362	95.69	3329	96.86	3117	95.90
2	5701	96.93	5569	95.74	5660	**96.99**	5745	96.00
3	6730	**97.02**	6999	**95.84**	6769	96.95	7260	**96.21**
Total		16.5% error rate reduction		13.2% error rate reduction		16.4% error rate reduction		19.0% error rate reduction

5. CONCLUSIONS

In this paper, we presented a semi-supervised learning framework for conditional random fields based on the co-training technique and the entropy estimation to determine confident sequences inferred from a huge set of unlabeled data. The proposed method has some advantages comparing to the other semi-supervised learning methods for sequence data. First, this method is domain and data independent. This means that we can apply this method to any sequential learning problems to improve the prediction accuracy. Second, it is easy to implement because it is only based on a simple and fast entropy estimation. Finally, one can freely choose a multi-view representation and apply this framework to build a CRF co-training application.

The future work will focus on the complex analysis of entropy values and how to select reliably inferred data sequences from unlabeled data more accurately and efficiently. We will also try with other multi-view representation ways to see that whether our method can be adaptive to different kinds of sequence data and sequential learning applications.

REFERENCES

[1] Blum, A. and Mitchell, T. Combining labeled and unlabeled data with co-training. In *COLT Workshop on Computational Learning Theory*.

[2] Nigram, K., McCallum, A., Thrun, S., and Mitchell, T. Learning to classify text from labeled and unlabeled documents. In *AAAI-98*.

[3] Cozman, F.G., Cohen, I., and Cirelo, M.C. Semi-supervised learning of mixture models. In *ICML-2003*.

[4] Szummer, M. and Jaakkola, T. Partially labeled classification with markov random walks. In *NIPS-2001*.

[5] Zhu, X., Ghahramani, Z., and Lafferty, J. Semi-supervised learning using gaussian fields and harmonic functions. In *ICML-2003*.

[6] Rabiner, L.R. A tutorial on hidden markov models and selected applications in speech recognition. In *IEEE-1989*.

[7] Lafferty, J., McCallum, A., and Pereira, F. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *ICML-2001*.

[8] Li, W. and McCallum, A. A semi-supervised sequence modeling with syntatic topic models. In *AAAI-2005*.

[9] Lafferty, J., Zhu, X., and Liu, Y. Kernel conditional random fields: representation and clique selection. In *ICML-2003*.

[10] Brefeld, U., Buscher, C., and Scheffer, T. Multi-view discriminative sequential learning. In *ECML-2005*.

[11] Collins, M. and Singer, Y. Unsupervised models for named entity classification. In *EMNLP-1999*.

[12] Ando, R.K. and Zhang, T. A high-performance semi-supervised learning method for text chunking. In *ACL-2005*.

[13] Smith, N.A. and Eisner, J. Contrastive estimation: training log-linear models on unbalanced data. In *ACL-2005*.

[14] Collins, M. Discriminative training methods for hidden markov models: theory and experiment with perceptron algorithms. In *EMNLP-2002*.

[15] Chen, S.F. and Rosenfeld, R. A gaussian prior for smoothing maximum entropy models. Technical Report CMU-CS-99-108, CMU, 1999.

[16] McCallum, A. Efficently inducing features of conditional random fields. In *UAI-2003*.

[17] Liu, D. and Nocedal, J. On the limited memory BFGS method for large-scale optimization. *Mathematical Programming*, 45:503--528, 1989.

[18] Malouf, R. A comparison of algorithms for maximum entropy parameter estimation. In *CoNLL-2002*.

[19] Sha, F. and Pereira, F. Shallow parsing with conditional random fields. In HLT/NAACL 2003.

[20] Ramshaw, L.A. and Marcus, P. Text chunking using transformation-based learning. In *Workshop on Very Large Corpora*, 1995.

[21] Tjong Kim Sang, E.F. and Veenstra, J. Representing text chunks. In *EACL-1999*.

[22] Uchimoto, K., Ma, Q., Murata, M., Ozaku, H., and Isahara, H. Named entity recognition based on a maximum entropy model and transformation rules. In *ACL-2000*.

[23] Abney, S. Bootstrapping. In *ACL-2002*.

[24] Clark, S., Curran, J.R., and Osborne, M. Bootstrapping POS taggers using unlabeled data. In *CoNLL-2003*.

[26] Berger, A., Pietra, A.D., and Pietra, J.D. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39-71, 1996.

[26] Kudo, T. and Matsumoto, Y. Chunking with support vector machines. In ACL/ NAACL-2001.