Structural Classification Methods Based on Weighted Finite-State Transducers for Automatic Speech Recognition

Yotaro Kubo, *Member, IEEE*, Shinji Watanabe, *Senior Member, IEEE*, Takaaki Hori, *Member, IEEE*, and Atsushi Nakamura, *Senior Member, IEEE*

***Abstract*—The potential of structural classification methods for automatic speech recognition (ASR) has been attracting the speech community since they can realize the unified modeling of acoustic and linguistic aspects of recognizers. However, the structural classification approaches involve well-known tradeoffs between the richness of features and the computational efficiency of decoders. If we are to employ, for example, a frame-synchronous one-pass decoding technique, features considered to calculate the likelihood of each hypothesis must be restricted to the same form as the conventional acoustic and language models. This paper tackles this limitation directly by exploiting the structure of the weighted finite-state transducers (WFSTs) used for decoding. Although WFST arcs provide rich contextual information, close integration with a computationally efficient decoding technique is still possible since most decoding techniques only require that their likelihood functions are factorizable for each decoder arc and time frame. In this paper, we compare two methods for structural classification with the WFST-based features; the structured perceptron and conditional random field (CRF) techniques. To analyze the advantages of these two classifiers, we present experimental results for the TIMIT continuous phoneme recognition task, the WSJ transcription task, and the MIT lecture transcription task. We confirmed that the proposed approach improved the ASR performance without sacrificing the computational efficiency of the decoders, even though the baseline systems are already trained with discriminative training techniques (e.g., MPE).**

***Index Terms*—Automatic speech recognition (ASR), structural classification, weighted finite-state transducers (WFST).**

I. Introduction

One-pass decoding is an important technique for ensuring the real-time property of systems involving automatic speech recognition (ASR) technologies such as systems for the DARPA TRANSTAC project, meeting recognition systems [1], [2] and closed-captioning systems [3]. To enable such applications, the recent development of ASR decoders has led to fast and accurate speech recognition based on hidden Markov model (HMM)-based acoustic models and N-gram language models. However, due to the generative formulations of these acoustic and language models, their performance is not directly maximized in terms of error rates. With the aim of achieving the direct minimization of the error rates, several discriminative training methods have been proposed that optimize the parameters of the generative models with respect to the discriminative criteria [4]–[11]. Because discriminative training methods preserve the form of the original generative models, the decoding techniques developed for the conventional ASR models are available even though the parameters are trained with these methods.

In practice, conventional discriminative training methods train the acoustic and language models separately; however, it is well-known that in theory the discriminative objective function should mutually involve both acoustic and language model parameters. Therefore, several joint training methods have recently been proposed. For example, hidden conditional random fields (HCRFs) [12] have been introduced to achieve the joint optimization of parameter vectors that can be translated into HMM and N-gram parameters. Chien [13] proposed a joint training method that directly optimizes these generative model parameters based on the maximum entropy principle. However, these studies still use a model structure that is identical to that of conventional HMMs and N-gram models. More specifically, these methods still involve independent acoustic and language models in their decoding processes even though these are estimated so that the joint performance is maximized.

To overcome these restrictions on model structure, several unified modeling techniques based on structural classification methods have been discussed. Unlike the joint training methods, unified modeling techniques attempt to use expanded model structures to represent and leverage the interdependency of the acoustic and language aspects of ASR. To leverage these interdependencies, most methods based on the structural classification approach attempt to estimate the structural labels using features extracted from both the input and output of the classifiers, where the input represents an acoustic observation sequence and the output represents a linguistic symbol sequence in terms of ASR. One of the earliest instances of structural classification is conditional random fields (CRFs) [14] where the interdependency of sequential inputs and outputs is directly handled. Subsequently, structured support vector machines (structured SVMs) are introduced by combining the advantage of the max-margin property of the SVMs and the essence of structural

Manuscript received January 17, 2012; revised April 19, 2012; accepted April 24, 2012. Date of publication May 11, 2012; date of current version August 09, 2012. This work was supported in part by the Japan Society for the Promotion of Science under Grant-in-Aid Scientific Research No. 22300064. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Brian Kingsbury.

Y. Kubo, T. Hori, and A. Nakamura are with the NTT Communication Science Laboratories, NTT Corporation, Kyoto 619-0237, Japan (e-mail: kubo.yotaro@lab.ntt.co.jp; hori.t@lab.ntt.co.jp; nakamura.atsushi@lab.ntt.co.jp).

S. Watanabe was with the NTT Communication Science Laboratories, NTT Corporation, Kyoto 619-0237, Japan. He is now with the Mitsubishi Electric Research Laboratories, Cambridge, MA 02139 USA (e-mail: shinjiw@ieee.org).

Digital Object Identifier 10.1109/TASL.2012.2199112

classification [15]. Structured perceptrons have also been proposed to enable efficient online learning [16]. Since perceptron training is typically performed by an online algorithm, structured perceptrons are used to overcome large-scale datasets. An advantage of the use of the perceptron approach is that several large-margin training methods have been proposed for perceptrons [16], [17]. The efficiency of these structural classification approaches has been very well accepted by the natural language processing (NLP) and machine learning (ML) research communities, and verified by many NLP tasks [18], [19]. Considering the close relationship between the ASR, NLP, and ML research fields, the structural classification approaches would also be very promising in relation to ASR problems.

In ASR problems, especially in large-vocabulary continuous speech recognition (LVCSR) problems, computationally efficient approaches are adopted since large-scale datasets are typically used. The structural classification approaches that have been used for ASR can be divided into two categories. The first category involves classical CRFs that attempt to model the posterior distribution of word sequences. For example, [20] enabled the integration of various detectors to achieve a flexible rescoring framework based on an extension of CRFs. Even though sophisticated learning algorithms based on SVMs and perceptrons have been proposed, the CRF-based framework is still competitive since it enables easy and efficient parallelization. The second category involves a structured perceptron. [21] introduces a rescoring framework based on weighted finite-state transducers (WFSTs) and achieves the efficient use of duration information. Even these methods are potentially capable of handling various features to leverage interdependency between acoustic and linguistic aspects, features must be restricted to the same form with the conventional acoustic and language models if we are to apply one-pass decoding techniques to these methods. In other words, since structural classification is a general approach for realizing unified modeling, and enables the use of even computationally inefficient features, the realization of computationally efficient decoding would be difficult if the features used are not carefully parametrized.

This paper deals directly with this parametrization problem, and proposes computationally efficient WFST-based parametrization. The proposed methods involve parameter vectors and features for each WFST arc in the decoding network. Since WFST arcs represent multiple lexical contexts, such as word contexts, phoneme contexts, and HMM state contexts, WFST-based parametrization enables the utilization of such contextual information. Furthermore, thanks to these dependencies on the WFST arcs, we can realize the tight and straightforward integration of the structural classifiers and the WFST-based computationally efficient decoders, such as [22]–[25]. The proposed approach can be assumed to be a straightforward extension of the conventional transition weight optimization approaches [26], [27] that does not optimize the acoustic aspect of ASR. The proposed approach can be incorporated with both perceptron approaches and CRF approaches of structural classifiers. This paper is the extended version of our previous papers where the averaged perceptron-based technique [28] and CRF-based technique [29] are separately discussed. In this paper, we investigate their properties, e.g., relationships to vocabulary size, acoustic model complexity, and error measurement, by comparing these methods.

The rest of this paper is organized as follows. Section II describes a general formulation of the proposed approach for structural classification. Section III describes a method for optimizing the structural classifier based on the perceptron techniques. Section IV describes an alternative optimization method based on conditional random fields. Section V discusses and compares the experimental results. Section VI summarizes the achievements and findings, and suggests future work.

II. Structural Classification Approach

This section reduces the conventional WFST-based ASR decoding process to a linear classification process called structural classification. ASR is performed by obtaining a label sequence $\hat{\boldsymbol{\ell}}$ that maximizes the posterior distribution $P(\boldsymbol{\ell}|\mathbf{X})$, given an observation vector sequence $\mathbf{X}$, as follows:

$$\hat{\boldsymbol{\ell}} = \arg\max_{\boldsymbol{\ell}} \ \log P(\boldsymbol{\ell}|\mathbf{X}) \tag{1}$$

where, in general, an observation vector sequence $\mathbf{X}$ is represented by a T-element sequence of D-dimensional vectors, i.e., $\mathbf{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_t, \ldots, \boldsymbol{x}_T \mid \boldsymbol{x}_t \in \mathbb{R}^D\}$, and a label sequence $\boldsymbol{\ell}$ is represented by a sequence of symbols in a label set $\mathbb{L}$, i.e., $\boldsymbol{\ell} = \{\ell_1, \ell_2, \ldots, \ell_m, \ldots \mid \ell_m \in \mathbb{L}\}$. Conventionally, this posterior distribution is modeled by using HMM-based acoustic models, and N-gram language models, and approximated by applying the Viterbi approximation (hard decision of an HMM state sequence), as follows:

$$\begin{aligned}\hat{\boldsymbol{\ell}} &= \arg\max_{\boldsymbol{\ell}} \ \log \overbrace{\sum_{\mathbf{q}} P(\mathbf{X}|\mathbf{q})P(\mathbf{q}|\boldsymbol{\ell})}^{\text{HMM}} \quad \overbrace{P(\boldsymbol{\ell})}^{\text{N-gram model}} \\ &\approx \arg\max_{\boldsymbol{\ell}} \ \log \underbrace{\max_{\mathbf{q}}}_{\text{Viterbi}} \ P(\mathbf{X}|\mathbf{q})P(\mathbf{q}|\boldsymbol{\ell})P(\boldsymbol{\ell})\end{aligned} \tag{2}$$

where $\mathbf{q}$ is an HMM state sequence. The probability distribution $P(\mathbf{q}|\boldsymbol{\ell})$ is conventionally decomposed into three elements, i.e., a pronunciation lexicon model, a context-dependency model, and an HMM state transition model, as follows:

$$P(\mathbf{q}|\boldsymbol{\ell}) \overset{\text{def}}{=} \sum_{\tilde{\mathbf{p}}} \sum_{\mathbf{p}} \quad \overbrace{P(\mathbf{q}|\tilde{\mathbf{p}})}^{\text{State transition}} \quad \overbrace{P(\tilde{\mathbf{p}}|\mathbf{p})}^{\text{Context dependency}} \quad \overbrace{P(\mathbf{p}|\boldsymbol{\ell})}^{\text{Lexicon}} \tag{3}$$

or, optionally, we can use the following approximated form for computational efficiency:

$$P(\mathbf{q}|\boldsymbol{\ell}) \approx \max_{\mathbf{p},\tilde{\mathbf{p}}} P(\mathbf{q}|\tilde{\mathbf{p}})P(\tilde{\mathbf{p}}|\mathbf{p})P(\mathbf{p}|\boldsymbol{\ell}) \tag{4}$$

where $\mathbf{p}$, and $\tilde{\mathbf{p}}$ are variables denoting sequences of phonemes and context-dependent phonemes, respectively, $P(\mathbf{p}|\boldsymbol{\ell})$ is a pronunciation lexicon model that defines the probabilities of a phoneme sequence, given the word sequences, that is usually defined as a rule-based model, $P(\tilde{\mathbf{p}}|\mathbf{p})$ is the probability of context-dependent phonemes, given the phonemes, and $P(\mathbf{q}|\tilde{\mathbf{p}})$

is a HMM state probability distribution defined as a Markov chain.

By introducing the WFST-based decoders, these nested optimizations and marginalizations in (2) are reduced to the equivalent flat optimization, as follows:

$$\begin{aligned}\hat{\boldsymbol{\ell}} &= \boldsymbol{O}[\hat{\mathbf{a}}]\\ \hat{\mathbf{a}} &= \arg\max_{\mathbf{a}\in\boldsymbol{D}} f(\mathbf{a};\mathbf{X})\end{aligned} \tag{5}$$

where $\boldsymbol{O}[\hat{\mathbf{a}}]$ is an operator that extract output word sequences from the given WFST arc sequence $\hat{\mathbf{a}}$, $\boldsymbol{D}$ is a decoding WFST represented as a set of arc sequences, and $f(\mathbf{a};\mathbf{X})$ is a score function of the hypothesis arc sequence $\mathbf{a}$, given the observation sequence $\mathbf{X}$. In this paper, we consider a WFST to be a set of arc sequences that start from an initial state, and reach a final state. Each arc a_m in the arc sequence contains information about an input symbol $I[a_m] \in \{\epsilon\}\cup\{1,\ldots,S\}$, an output symbol $O[a_m] \in \{\epsilon\}\cup\mathbb{L}$, a start time $T[a_m] \in \{1,\ldots,T\}$, a stop time $\tilde{T}[a_m] \in \{1,\ldots,T\}$, and a real valued weight $H[a_m] \in \mathbb{R}^+$ where ϵ denotes the epsilon symbol that represents null. Although time alignment information of the WFST arcs $T[a_m], \tilde{T}[a_m]$ is only observable after decoding, we introduced these variables for simplicity of notation. We should note that the use of these time variables in decoding processes can be avoided in the proposed method.

In general, the decoding WFST $\boldsymbol{D}$ is created by composing an HMM-state sequence transducer, a context-dependency transducer, a pronunciation lexicon transducer, and a language model acceptor [22]. By using the above-mentioned WFST notations, the score function $f(\mathbf{a};\mathbf{X})$ is conventionally defined as follows:

$$f(\mathbf{a};\mathbf{X}) = \sum_m -W(a_m;\mathbf{X}) \tag{6}$$

where $W(a_m;\mathbf{X})$ is an arc-wise score function defined by using the acoustic score function $g(a_m;\mathbf{X})$, as follows:

$$\begin{aligned}W(a_m;\mathbf{X}) &= g(a_m;\mathbf{X}) + \alpha H[a_m]\\ g(a_m;\mathbf{X}) &= -\sum_{t=T[a_m]}^{\tilde{T}[a_m]} \log P(\boldsymbol{x}_t|q_t = I[a_m]).\end{aligned} \tag{7}$$

Here, α is a tunable parameter, called a language model scale factor, that controls the importance of language model constraints. The above equation assumes that each non-epsilon input symbol $I[a_m]$ represents just one HMM state. However, the above equation can also be generalized to the WFSTs optimized with the factorization operation, which combines several input symbols to make a concatenated input symbol, by introducing the Viterbi acoustic score as $g(a_m;\mathbf{X})$.

In this paper, aiming at a decoder-friendly structural classification, we extend this arc-wise score function. Since most one-pass decoding techniques assume that the score function of the arc sequence $f(\mathbf{a};\mathbf{X})$ can be factorized into the arc-wise terms $W(a_m;\mathbf{X})$, the extension of the arc-wise function does not exclude the availability of one-pass decoding techniques if the arc-wise function can be computed on a frame-by-frame basis. In this paper, the following extended function is used instead of the original arc-wise score function:

$$W'(a_m;\mathbf{X},\boldsymbol{\Lambda}) = g(a_m;\mathbf{X}) + \alpha H[a_m] + \boldsymbol{\lambda}_{N[a_m]}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{X},a_m). \tag{8}$$

where $\boldsymbol{\Lambda} \overset{\text{def}}{=} \{\boldsymbol{\lambda}_1,\boldsymbol{\lambda}_2,\ldots\}$ is a set of parameter vectors, and $\boldsymbol{\phi}(\mathbf{X};a_m)$ is the arc feature extraction function described below. $N[a_m]$ is the arc identifier, which is used for tying the parameter vectors, defined depending on the task design. For example, unique identifiers given to each arc in the WFST were used in the following phoneme recognition experiments, and the unique numbers given to each arc in the unigram WFSTs were used as the arc identifiers of the composed WFST in the following LVCSR experiments. By substituting the original arc-wise score function with the above extended one, the extended decoding rule can be expressed as follows:

$$\begin{aligned}\hat{\mathbf{a}} &= \arg\max_{\mathbf{a}\in\boldsymbol{D}} f'(\mathbf{a};\mathbf{X},\boldsymbol{\Lambda})\\ f'(\mathbf{a};\mathbf{X},\boldsymbol{\Lambda}) &= \sum_m -W'(a_m;\mathbf{X},\boldsymbol{\Lambda})\end{aligned} \tag{9}$$

In this paper, to ensure the capability of the frame-by-frame computation of the arc-wise function, the arc feature extraction function is designed as follows:

$$\boldsymbol{\phi}(\mathbf{X},a_m) \overset{\text{def}}{=} \begin{cases}[\mathbf{0},1]^{\mathsf{T}}, & I[a_m]=\epsilon\\ \left[\sum_{t=T[a_m]}^{\tilde{T}[a_m]}\boldsymbol{\psi}(\boldsymbol{x}_t),1\right]^{\mathsf{T}}, & \text{otherwise.}\end{cases} \tag{10}$$

Here, we consider two kinds of features; the frame-level acoustic feature $\boldsymbol{\psi}(\boldsymbol{x}_t)$ and the arc occupancy feature, which is fixed to 1 for all arcs. By introducing this feature function, the dot product in (8) can be computed, as follows:

$$\boldsymbol{\lambda}_{N[a_m]}^{\mathsf{T}}\boldsymbol{\phi}(\mathbf{X},a_m) = \sum_{t=T[a_m]}^{\tilde{T}[a_m]}\boldsymbol{\psi}(\boldsymbol{x}_t)^{\mathsf{T}}\boldsymbol{\lambda}'_{N[a_m]} + \lambda_{N[a_m],D} \tag{11}$$

where $\boldsymbol{\lambda}'_{N[a_m]}$ is the reduced parameter vector constructed by discarding the last component of $\boldsymbol{\lambda}_{N[a_m]}$ corresponding to the arc occupancy feature, and $\lambda_{N[a_m],D}$ is the last component of $\boldsymbol{\lambda}_{N[a_m]}$ corresponding to the arc occupancy feature. By computing this dot product on the frame-by-frame basis, we can straightforwardly integrate the extended decoding rule into the conventional one-pass decoding techniques.

This extension is similar to that of discriminative language models [30], [31], which also involve the linear corrective terms optimized by using training datasets. In contrast to the discriminative language models, the proposed approach can handle the acoustic information extracted by using $\boldsymbol{\psi}$; however, the use of flexible features is discarded to ensure the availability of one-pass decoding techniques. Actually, most implementations of the discriminative language models are based on a rescoring framework.

The following sections describe methods for optimizing parameter $\boldsymbol{\Lambda}$. Section III describes the parameter optimization method based on the perceptron approach. Section IV describes the parameter optimization methods based on the

CRF approach. It should be noted that this paper achieves discriminative unified modeling by optimizing the additional term introduced into the arc-wise score function; therefore, we do not mention the training of acoustic and language models hereafter. Combination with sophisticated training methods used for acoustic and language models is possible and promising.

III. Perceptron-Based Training

The perceptron approach is a very well-known way of optimizing the linear discriminant function. Since our score function is based on a linear function, as in (9), the perceptron approach is applicable straightforwardly. Hereafter, we denote the training dataset as $\mathcal{Z} = \{(\mathbf{X}^{(n)}, \boldsymbol{\ell}^{(n)}) \mid \forall n \in \{1, \ldots, N\}\}$ where n is the utterance index, and $\mathbf{X}^{(n)}$ and $\boldsymbol{\ell}^{(n)}$ are the nth observation vector sequence and label sequence, respectively.

The strategy used in perceptron learning is based on the stochastic gradient descent optimization of a hinge loss function [32]. By considering the score of the correct word sequence $\hat{f}$ and the recognized word sequence $\bar{f}$, we can design the hinge loss function, as follows:

$$L(\mathcal{Z}) = \sum_n \max\{0, \bar{f}(\mathbf{X}^{(n)}) - \hat{f}(\mathbf{X}^{(n)})\} \tag{12}$$

where the score functions, $\hat{f}(\mathbf{X}^{(n)})$ and $\bar{f}(\mathbf{X}^{(n)})$, are defined as follows:

$$\begin{aligned} \hat{f}(\mathbf{X}^{(n)}) &\stackrel{\text{def}}{=} \max_{\mathbf{a} \in (\boldsymbol{D} \circ \boldsymbol{\ell}^{(n)})} f'(\mathbf{a}; \mathbf{X}^{(n)}, \boldsymbol{\Lambda}) \\ \bar{f}(\mathbf{X}^{(n)}) &\stackrel{\text{def}}{=} \max_{\mathbf{a} \in \boldsymbol{D}} f'(\mathbf{a}; \mathbf{X}^{(n)}, \boldsymbol{\Lambda}). \end{aligned} \tag{13}$$

Here, $(\boldsymbol{D} \circ \boldsymbol{\ell}^{(n)})$ denotes the composition of the decoding network WFST $\boldsymbol{D}$ and the finite-state acceptor that accepts the nth word sequence $\boldsymbol{\ell}^{(n)}$. The gradient vector (or a subgradient vector) with respect to the ith parameter vector $\boldsymbol{\lambda}_i$ can be expressed as a sum of the (sub)gradient vectors of sample-wise loss functions, as follows:

$$\nabla_{\boldsymbol{\lambda}_i} L(\mathcal{Z}) = \sum_n \nabla_{\boldsymbol{\lambda}_i} \max\{0, \bar{f}(\mathbf{X}^{(n)}) - \hat{f}(\mathbf{X}^{(n)})\}. \tag{14}$$

The stochastic gradient method [33] approximates the gradient vector by omitting summation, and only using the gradient vector of the sample-wise loss function that corresponds to a sample $\mathbf{X}^{(r)}$ randomly sampled from the training data, as follows:

$$\nabla_{\boldsymbol{\lambda}_i} L(\mathcal{Z}) \approx \nabla_{\boldsymbol{\lambda}_i} \max\{0, \bar{f}(\mathbf{X}^{(r)}) - \hat{f}(\mathbf{X}^{(r)})\} \stackrel{\text{def}}{=} \tilde{\nabla}^{(r)}_{\boldsymbol{\lambda}_i}. \tag{15}$$

Here, the sample-wise gradient vector can be expressed as follows:

$$\tilde{\nabla}^{(r)}_{\boldsymbol{\lambda}_i} = \begin{cases} \mathbf{0}, & \bar{f}(\mathbf{X}^{(r)}) - \hat{f}(\mathbf{X}^{(r)}) > 0 \\ \sum_m \delta(N[\bar{a}_m]; i) \boldsymbol{\phi}(\mathbf{X}^{(r)}, \bar{a}_m) & \\ \quad -\delta(N[\hat{a}_m]; i) \boldsymbol{\phi}(\mathbf{X}^{(r)}, \hat{a}_m), & \\ & \text{otherwise.} \end{cases} \tag{16}$$

where $\delta(x; y)$ is Kronecker's delta, i.e., $\delta(x; y) = 1$ if $x = y$ otherwise $\delta(x; y) = 0$.

By using a normalized sample-wise gradient vector instead of the true gradient vector, the update rule of the normalized perceptron learning can be defined as follows:

$$\boldsymbol{\lambda}_i^{[e+1]} = \boldsymbol{\lambda}_i^{[e]} - \frac{\gamma^{[e]}}{\left\|\tilde{\nabla}^{(r)}_{\boldsymbol{\lambda}_i}\right\|} \tilde{\nabla}^{(r)}_{\boldsymbol{\lambda}_i} \Big|_{\boldsymbol{\lambda}_i = \boldsymbol{\lambda}_i^{[e]}} \tag{17}$$

where the superscripts $[e]$ and $[e+1]$ denote the numbers of optimization iterations, and $\gamma^{[e]}$ is the learning rate of the eth update. By increasing $e \rightarrow \infty$, it is shown that the optimization converges at an optimum point if the learning rate sequence $\gamma^{[e]}$ is designed appropriately [33].

A. Averaged Perceptron

The averaged perceptron (AP) technique is introduced to obtain robustness as regards the choice of learning rate and the randomness of the selected learning sample r[16]. The AP is trained by averaging all parameter vectors $\boldsymbol{\lambda}_i^{[e']}$ obtained as optimization iterates. The solution for AP $\hat{\boldsymbol{\lambda}}_i$ can be expressed as follows:

$$\hat{\boldsymbol{\lambda}}_i = \frac{1}{E} \sum_{e'} \boldsymbol{\lambda}_i^{[e']}. \tag{18}$$

Here, E is the total number of iterations. Thanks to the robustness of this algorithm, a constant learning rate $\gamma^{[e]} = \gamma$ can be used in practice. Algorithm 1 is a learning algorithm for the averaged perceptron adapted to the proposed structural classifier. In the following sections, we denote this training method as "AP."

Algorithm 1 Averaged Perceptron Training

```
Input: training data $\mathcal{Z}$, decoding network $\boldsymbol{D}$, learning rate $\gamma$
$\boldsymbol{\lambda}_i \leftarrow \mathbf{0}$ $(\forall i)$
$E \leftarrow 0$
loop
    $r \leftarrow$ random sample from $\{1, \ldots, N\}$
    $\hat{\mathbf{a}} \leftarrow \arg\max_{\mathbf{a} \in (\boldsymbol{D} \circ \boldsymbol{\ell}^{(r)})} f'(\mathbf{a}; \mathbf{X}^{(r)}, \boldsymbol{\Lambda})$
    $\bar{\mathbf{a}} \leftarrow \arg\max_{\mathbf{a} \in \boldsymbol{D}} f'(\mathbf{a}; \mathbf{X}^{(r)}, \boldsymbol{\Lambda})$
    if $\hat{\mathbf{a}} \neq \bar{\mathbf{a}}$ then
        for all $\hat{a}_m \in \hat{\mathbf{a}}$ do
            $\boldsymbol{\lambda}_{N[\hat{a}_m]} = \boldsymbol{\lambda}_{N[\hat{a}_m]} + (\gamma / \|\boldsymbol{\phi}(\mathbf{X}^{(r)}, \hat{a}_m)\|) \boldsymbol{\phi}(\mathbf{X}^{(r)}, \hat{a}_m)$
        end for
        for all $\bar{a}_m \in \bar{\mathbf{a}}$ do
            $\boldsymbol{\lambda}_{N[\bar{a}_m]} = \boldsymbol{\lambda}_{N[\bar{a}_m]} - (\gamma / \|\boldsymbol{\phi}(\mathbf{X}^{(r)}, \bar{a}_m)\|) \boldsymbol{\phi}(\mathbf{X}^{(r)}, \bar{a}_m)$
        end for
    end if
    $\boldsymbol{\lambda}_i^{(e)} \leftarrow \boldsymbol{\lambda}_i$ $(\forall i)$
    $E \leftarrow E + 1$
end loop
Result: $1/E \sum_{e=0}^{E} \boldsymbol{\lambda}_i^{(e)}$ $(\forall i)$
```

B. Distributed Perceptron

Parallelization is important as regards adapting the algorithm to large-scale tasks. This paper employs a parallelized variant of a perceptron proposed by McDonald *et al.* [34]. In [34], it is shown that the distributed perceptron algorithm performed in multiple threads achieved performance comparable to that of conventional perceptron algorithms performed in a single thread. Since the theoretical convergence property of the distributed perceptron has been well discussed, we adopted the distributed perceptron algorithm for our structural classification problem.

The strategy of the distributed perceptron training is quite simple. Since our objective function is convex, the linear interpolation of solutions always achieves a better objective function than, at least, the worst solution; i.e., $L(\mathcal{Z})\mid_{\boldsymbol{\Lambda}=\bar{\boldsymbol{\Lambda}}}\leq \max_s\{L(\mathcal{Z})\mid_{\boldsymbol{\Lambda}=\boldsymbol{\Lambda}_s}\}$ where $\bar{\boldsymbol{\Lambda}}$ is an average of all hypothesis parameters $\boldsymbol{\Lambda}_s$. A hypothesis parameter $\boldsymbol{\Lambda}_s$ can be obtained by using a subset $\mathcal{Z}_s$ of the given training dataset and the perceptron training method. An example of distributed perceptron training is summarized in Algorithm 2. In this algorithm, a perceptron algorithm, denoted as $\mathrm{Perceptron}(\mathcal{Z}_s,\boldsymbol{\Lambda})$, is used as a subroutine that performs perceptron training with a given training dataset $\mathcal{Z}_s$ and a given initial value $\boldsymbol{\Lambda}$. The convergence property of the algorithm is shown in [34]. Since each perceptron subroutine can be executed in parallel, most of the algorithm is parallelizable. In this paper, we used the conventional normalized perceptron approach [(17)] as a perceptron subroutine. In the following sections, we denote this training method as "DP."

Algorithm 2 Distributed Perceptron Training

```
Input: training data $\mathcal{Z}$, # of threads $S$
Divide data $\mathcal{Z}$ into $S$ subsets $\mathcal{Z}_s$ $(s \in \{1,\ldots,S\})$
loop
    for all $s \in \{1,\ldots,S\}$ do
        $\boldsymbol{\Lambda}_s \leftarrow \mathrm{Perceptron}(\mathcal{Z}_s,\boldsymbol{\Lambda})$
    end for
    $\boldsymbol{\Lambda} \leftarrow (1/S)\sum_s \boldsymbol{\Lambda}_s$
end loop
```

IV. CRF-Based Training

Since the perceptron approaches omit probabilistic representations by directly optimizing the score function f', these approaches are not compatible with sophisticated training criteria developed for acoustic model discriminative training, such as the *maximum-mutual-information* (MMI), boosted MMI, and differenced MMI criteria. Alternatively, since the CRF approach retains probabilistic representations, these training methods are applicable straightforwardly. This section discusses training criteria adapted to the CRF framework from the acoustic model training methods.

In the following training methods, instead of handling the score function $f'(\mathbf{a};\mathbf{X},\boldsymbol{\Lambda})$ directly, we deal with the following CRF-based posterior distribution of the arc sequences:

$$P(\mathbf{a}|\mathbf{X},\boldsymbol{\Lambda}) \overset{\mathrm{def}}{=} \frac{1}{Z(\mathbf{X},\boldsymbol{\Lambda})}\exp f'(\mathbf{a};\mathbf{X},\boldsymbol{\Lambda}) \tag{19}$$

where Z is a partition function that is defined as follows:

$$Z(\mathbf{X},\boldsymbol{\Lambda}) = \sum_{\mathbf{a}'\in\mathcal{D}}\exp f'(\mathbf{a}';\mathbf{X},\boldsymbol{\Lambda}). \tag{20}$$

The decoding criterion is the same as the perceptron case (5) since the partition function $Z(\mathbf{X},\boldsymbol{\Lambda})$ can be assumed to be a constant during the decoding processes. In contrast to the conventional CRF approaches, since the proposed approach employs a computationally efficient score function [f' in (9)], compatibility with the one-pass decoding methods is ensured.

The following sections describe three training methods derived from the MMI-based criteria.

A. Maximum Mutual Information

The MMI criterion is regarded as a natural training criterion for posterior distribution models including CRFs since the MMI training attempts to fit the posterior distribution to that of the training label set, as follows:

$$\begin{aligned}\hat{\boldsymbol{\Lambda}} &= \arg\max_{\boldsymbol{\Lambda}} \sum_n \log P(\boldsymbol{\ell}^{(n)}|\mathbf{X}^{(n)},\boldsymbol{\Lambda})\\ &= \arg\max_{\boldsymbol{\Lambda}} \underbrace{\sum_n \log\frac{\sum_{\mathbf{a}\in(\boldsymbol{D}\circ\boldsymbol{\ell}^{(n)})}\exp\left\{f'(\mathbf{a};\mathbf{X}^{(n)},\boldsymbol{\Lambda})\right\}}{\sum_{\mathbf{a}'\in\boldsymbol{D}}\exp\left\{f'(\mathbf{a}';\mathbf{X}^{(n)},\boldsymbol{\Lambda})\right\}}}_{F^{\mathrm{MMI}}(\boldsymbol{\Lambda})}.\end{aligned} \tag{21}$$

In this paper, for the CRF-based training methods, we introduce a lattice approximation. With this approximation, the summation over all possible arc sequences ($\mathbf{a}'\in\boldsymbol{D}$ and $\mathbf{a}\in(\boldsymbol{D}\circ\boldsymbol{\ell}^{(n)})$) is approximated by using a set of arc sequences in the competitor lattice and the reference lattice ($\mathbf{a}'\in\bar{\boldsymbol{L}}^{(n)}$ and $\mathbf{a}\in\hat{\boldsymbol{L}}^{(n)}$). In practice, the competitor and reference lattices can be obtained by running the decoder once, where the perceptron-based training methods proposed in this paper perform the decoding for each parameter update. By introducing the lattice approximation, the gradient vector of this objective function is obtained as follows:

$$\begin{aligned}\nabla_{\boldsymbol{\lambda}_i}F^{\mathrm{MMI}}(\boldsymbol{\Lambda}) &= \sum_n \boldsymbol{\xi}^{\mathrm{num}}_{n,i}(\boldsymbol{\Lambda}) - \boldsymbol{\xi}^{\mathrm{den}}_{n,i}(\boldsymbol{\Lambda})\\ \boldsymbol{\xi}^{\mathrm{num}}_{n,i}(\boldsymbol{\Lambda}) &= \sum_{\mathbf{a}\in\hat{\boldsymbol{L}}^{(n)}} \hat{q}(\mathbf{a};\mathbf{X}^{(n)},\boldsymbol{\Lambda})\\ &\quad\times\sum_m \delta(i;N[a_m])\boldsymbol{\phi}(\mathbf{X}^{(n)},a_m)\\ \boldsymbol{\xi}^{\mathrm{den}}_{n,i}(\boldsymbol{\Lambda}) &= \sum_{a'\in\bar{\boldsymbol{L}}^{(n)}} \bar{q}(\mathbf{a}';\mathbf{X}^{(n)},\boldsymbol{\Lambda})\\ &\quad\times\sum_m \delta(i;N[a'_m])\boldsymbol{\phi}(\mathbf{X}^{(n)},a'_m)\end{aligned} \tag{22}$$

where $\hat{q}(\mathbf{a};\mathbf{X}^{(n)},\boldsymbol{\Lambda})$ and $\bar{q}(\mathbf{a};\mathbf{X}^{(n)},\boldsymbol{\Lambda})$ are the posterior distribution of the arc sequence over all possible sequences in the reference and competitor lattices, respectively, defined as follows:

$$\begin{aligned}\hat{q}(\mathbf{a};\mathbf{X}^{(n)},\boldsymbol{\Lambda}') &\overset{\mathrm{def}}{=} \frac{\exp\left\{\kappa f'(\mathbf{a};\mathbf{X}^{(n)},\boldsymbol{\Lambda}')\right\}}{\sum_{\mathbf{a}'\in\hat{\boldsymbol{L}}^{(n)}}\exp\left\{\kappa f'(\mathbf{a}';\mathbf{X}^{(n)},\boldsymbol{\Lambda}')\right\}}\\ \bar{q}(\mathbf{a}';\mathbf{X}^{(n)},\boldsymbol{\Lambda}') &\overset{\mathrm{def}}{=} \frac{\exp\left\{\kappa f'(\mathbf{a}';\mathbf{X}^{(n)},\boldsymbol{\Lambda}')\right\}}{\sum_{\mathbf{a}''\in\bar{\boldsymbol{L}}^{(n)}}\exp\left\{\kappa f'(\mathbf{a}'';\mathbf{X}^{(n)},\boldsymbol{\Lambda}')\right\}}.\end{aligned} \tag{23}$$

Here, a lattice smoothing factor κ is introduced for smoothing the lattice probability. This gradient vector can easily be computed by employing the lattice-based forward–backward algorithm. In contrast to the AP approach, the MMI objective function and its gradient vector consider all competing hypotheses.

In the following sections, we denote this training criterion as "MMI-CRF."

B. Transition Error Count and Boosted MMI

The recent development of MMI-based training techniques for acoustic models suggests that we can devise more sophisticated training criteria than the original MMI criterion. Since MMI estimates parameters to maximize the posterior distribution of the given label sequence, the training procedure only takes account of sequence-level errors. However, if we are to reduce word error rates, a sequence level error measurement is not sufficient. Therefore, to reduce word errors, it is important to use of fine-grained error measurements. Several conventional structural classification methods [35], [36] that are proposed for enhancing acoustic representation also showed the efficiency of the fine-grained error measurements, such as the *minimum word error* (MWE) criterion. This section proposes a finer measurement specialized for our WFST-based representation.

Several methods for measuring errors can be identified. Since the motivation is to reduce the word errors, the use of approximated word error counts looks promising, as employed in MWE training [6]. However, several studies have reported the advantages of finer measurements, e.g., phoneme errors and frame errors [6], [37]. In this paper, we focus on the *transition errors* . Conventional decoders, including WFST decoders, are modeled as finite-state machines (FSMs), and all computations during decoding processes can be described as a state transition of the FSMs. In that sense, the transitions in FSMs can be assumed to be a primitive decision in the ASR processes. Therefore, a transition error measurement can be considered a finest-grained error measurement.

To take account of arbitrary error measurements, we extended the boosted MMI technique [8] for our CRF training. The boosted MMI method is proposed for the discriminative training of acoustic models to reduce phoneme (or word) errors. This method achieves such error reduction by emphasizing the importance of erroneous sequences that include a lot of phoneme errors. In this paper, by adapting the boosted MMI to our model, we defined the following objective function:

$$
\begin{aligned}
&F_{\sigma}^{\mathrm{bMMI}}(\Lambda)\\
&\stackrel{\mathrm{def}}{=}\sum_{n}\log\frac{\sum_{\mathbf{a}\in\hat{\boldsymbol{L}}^{(n)}}\exp\left\{f'(\mathbf{a};\mathbf{X}^{(n)},\boldsymbol{\Lambda})\right\}}{\sum_{\mathbf{a}'\in\bar{\boldsymbol{L}}^{(n}}\exp\left\{f'(\mathbf{a}';\mathbf{X}^{(n)},\boldsymbol{\Lambda})+\sigma E^{(n)}(\mathbf{a}')\right\}}
\end{aligned}
\tag{24}
$$

where σ is a tunable parameter that adjusts the importance of fine-grained errors, and $E^{(n)}(\mathbf{a}')$ is a transition error count function. To define the error count function, the time-aligned arc-id $A(t;.)$ is defined as follows:

$$
A(t;\mathbf{a})\stackrel{\mathrm{def}}{=}N[a_m]\ \text{s.t.}\ T[a_m]\leq t<\tilde{T}[a_m], a_m\in\mathbf{a}. \tag{25}
$$

By using this time-aligned arc-id function, the error count function is defined as follows:

$$
E^{(n)}(\mathbf{a}')=\sum_{t}\left(1-\delta(A(t,\hat{\mathbf{a}}^{(n)});A(t,\mathbf{a}'))\right) \tag{26}
$$

where $\hat{\mathbf{a}}^{(n)}$ is the reference arc sequence obtained as $\hat{\mathbf{a}}^{(n)}=\arg\max_{\mathbf{a}\in\hat{\boldsymbol{L}}^{(n)}} f'(\mathbf{a};\mathbf{X}^{(n)},\boldsymbol{\Lambda})$. The gradient vector of the objective function [(24)] can also be computed by using the lattice-based forward–backward algorithm [8]. In the following sections, we denote this training criterion as "bMMI-CRF."

C. Direct Error Minimization via Differenced MMI

In the fields related to the discriminative training of acoustic models, several methods utilize the approximated error count as an objective function [5], [6]. For example, *minimum classification error* (MCE) [5] uses an objective function that approximates the number of sequence classification errors, and *minimum phone error* (MPE) [6] uses the model expectation of the phoneme error count function.

In this section, similar to MPE, we focus on the model expectation of the above-mentioned transition error function. We defined the objective function to be maximized as the model-based expectation of the negated error count function, as follows:

$$
\begin{aligned}
F^{\mathrm{MPE}}(\Lambda)&\stackrel{\mathrm{def}}{=}\sum_{n}\left\langle -E^{(n)}(\mathbf{a})\right\rangle_{P(\mathbf{a}|\mathbf{X}^{(n)},\boldsymbol{\Lambda})}\\
&=\sum_{n}\frac{-\sum_{\mathbf{a}\in\boldsymbol{D}}\exp\left\{f'(\mathbf{a};\mathbf{X}^{(n)},\boldsymbol{\Lambda})\right\}E^{(n)}(\mathbf{a})}{\sum_{\mathbf{a}'\in\boldsymbol{D}}\exp\left\{f'(\mathbf{a}';\mathbf{X}^{(n)},\boldsymbol{\Lambda})\right\}}.
\end{aligned}
\tag{27}
$$

In this paper, to ensure generality and ease of implementation, we adopt a variant of the MPE criterion, called differenced MMI (dMMI), originally proposed for acoustic model training [9]. The dMMI criterion is based on the following identity:

$$
F^{\mathrm{MPE}}(\Lambda)=\frac{\partial}{\partial\sigma}F_{\sigma}^{\mathrm{bMMI}}(\Lambda)\mid_{\sigma=0}. \tag{28}
$$

By utilizing this identity, the dMMI objective function is defined by performing the partial derivative in the above identity by using a numerical differentiation technique, as follows:

$$
F_{(\sigma_1,\sigma_2)}^{\mathrm{dMMI}}(\Lambda)=\frac{1}{\sigma_2-\sigma_1}\left(F_{\sigma=\sigma_2}^{\mathrm{bMMI}}(\Lambda)-F_{\sigma=\sigma_1}^{\mathrm{bMMI}}(\Lambda)\right). \tag{29}
$$

This objective function $F_{(\sigma_1,\sigma_2)}^{\mathrm{dMMI}}(\Lambda)$ converges to the MPE objective function in the limit of $\sigma_1\rightarrow -0$, $\sigma_2\rightarrow +0$. Furthermore, this objective function also converges to the bMMI objective function $F_{\sigma}^{\mathrm{bMMI}}(\Lambda)$ in the limit of $\sigma_1\rightarrow -\infty$, $\sigma_2\rightarrow\sigma$. Thus, we can define intermediates between MPE and bMMI by tuning the dMMI criteria. Furthermore, an optimization algorithm of this objective function can be performed simply by using the lattice-based forward–backward algorithm as with bMMI-CRF where the straightforward implementation of MPE requires the modified forward–backward algorithm [8]. In the following sections, we denote this training criterion as "dMMI-CRF."

V. Experiments

We conducted speech recognition experiments to evaluate the efficiency of our proposed approach. We applied the proposed method to the TIMIT continuous phoneme recognition task, the MIT-OCW/World lecture transcription task [38], and

TABLE I
TASK DESCRIPTIONS

	Training	Development	Evaluation
TIMIT			
# utterances	3,696	1,152	192
Duration	3h	59min	9.7min
# phonemes	48 → 39		
# running phonemes	140,099	43,539	7,215
MIT-OCW			
# utterances	60,392	813	6,989
Duration	101h	0.9h	7.8h
# lexicon words	44,485		
# running words	1,076,647	11,082	74,827
WSJ20k	(SI284)	(Nov'92)	(Nov'93)
# utterances	37,513	333	215
Duration	73.6h	0.7h	0.4h
# lexicon words	19,982		
# running words	1,281,540	5,646	3,852

TABLE II
ACOUSTIC ANALYSIS CONFIGURATION

A/D conversion	16 kHz / 16 bit
Window	Hamming (size: 25ms, shift: 10ms)
Observation vector	12 order MFCC with log-energy + Δ + $\Delta\Delta$
Front-end processing	Cepstral mean normalization

the WSJ transcription task. The task descriptions are summarized in Table I. The acoustic analysis configuration we used for all the experiments is summarized in Table II.

A. Continuous Phoneme Recognition

As the first experiment, we performed the TIMIT continuous phoneme recognition task. Since this task is strictly standardized, it is informative to compare the error rates with other results those are reported in other papers.

We evaluated both the context-independent (CI) HMM case and the context-dependent (CD) HMM case. In the CI case, the decoding network was constructed as $\boldsymbol{D} = \mathrm{opt}(\boldsymbol{H} \circ \boldsymbol{G})$ where $\boldsymbol{H}$ was a WFST that output HMM state sequences, $\boldsymbol{G}$ was a WFST based on a phoneme bigram model trained by using the maximum-likelihood method, and $\mathrm{opt}(.)$ was a WFST optimization operator that performed determinization, pushing, and minimization of WFSTs in this order. In the CD case, the decoding network was constructed as $\boldsymbol{D} = \mathrm{opt}(\mathrm{opt}(\boldsymbol{H} \circ \boldsymbol{C}) \circ \boldsymbol{G})$ where $\boldsymbol{C}$ was a context-dependency WFST obtained by using an acoustic model selection method based on a variational Bayesian estimation and clustering (VBEC) approach [39]. The composed decoding networks were further optimized by using factorization, epsilon-removal, and minimization. All the WFST operations used for constructing these WFSTs are detailed in [22].

The arc identifier $N[a_m]$ was uniquely numbered for all the arcs in the decoding networks. The numbers of arcs (i.e., the numbers of unique arc identifiers) in the CI and CD WFSTs were 1725 and 3362, respectively. In these experiments, we used a simple frame-level acoustic feature function, defined as $\boldsymbol{\psi}(\boldsymbol{x}) \overset{\mathrm{def}}{=} [\boldsymbol{x}^{\mathsf{T}}, 1]^{\mathsf{T}}$, i.e., we used the raw 39-dimensional observation vectors augmented by a bias term. We trained HMMs that involved three left-to-right hidden states as an acoustic model. The language model scale factor α was set at 5 and 10 for the CI and CD cases, respectively, determined by using the development dataset. The lattice smoothing factor κ was set at 1.0 determined by using the development dataset. The optimization loops of the structural classifiers were stopped when the performance on the development dataset was saturated. The Rprop method [40] was adopted for optimization of the CRF-based objective functions since this method was confirmed to be effective in other ASR tasks.

Tables III and IV show the experimental results of the decoders based on CI HMMs with diagonal covariance and full covariance acoustic models, respectively. The phoneme error rates presented with † marks are confirmed to achieve statistically significant improvements from the "ML-HMM" baseline system at significance level of $p = 0.05$ (computed using the Z-test). The results in the right-most columns of the tables are excerpts from the papers that present the results obtained in experiments using very similar configurations to those employed in this paper.

At first, we observed that the structural classification approach outperformed the conventional HMM and N-gram based approaches. Although we did not consider the variances and correlations of the observation vectors, the proposed classifier trained with bMMI or dMMI criteria outperformed the discriminatively trained acoustic models, where the second-order statistics and latent variables are involved. We consider that this advantage results from rich information about the contextual features represented in the WFST arcs. The AP approach and the MMI-CRF (bMMI-CRF with $\sigma = 0$) are theoretically similar in terms of their loss functions since both methods only take account of sequence level errors. However, the performances of these two methods were different, and we found that the AP approach was superior to MMI training in the TIMIT experiments. The main reason for this superiority might be due to the fact that the AP method can be considered as an approximation of the voted perceptron [17] that strictly ensures the large-margin property. The introduction of transition error minimization worked efficiently in all settings. This might be attributed to the robustness acquired by employing transition error minimization. Specifically, since transition error minimization attempts to discard WFST transitions that lead to a large expectation number of consequent transition errors, the decoder behavior is optimized with respect to the most primitive operations. We also confirmed that the dMMI methods worked slightly better than the bMMI methods. Our analysis shows that this advantage arises from the fact that the dMMI objective function directly represents the number of incorrect decisions. Although the large margin HMMs trained with the non-convex optimization method [41] outperformed the proposed method when the number of mixture components was less than 4, the proposed method outperformed this sophisticated discriminative training method in the 8 mixture setting. This suggests a limitation of the Gaussian mixture models without considering contextual information, since the proposed method enables the utilization of rich contextual information, the word error rates continued to decrease as the model complexity increased. We should note that a combination with large margin HMMs is also possible and promising. We demonstrate the efficiency of a combination consisting of the proposed method and the discriminatively trained acoustic models in the following LVCSR experiments.

In the TIMIT dataset, since we used the continuous phoneme decoding network, contextual information utilized by the proposed method is a kind of phoneme context. Therefore, we can

TABLE III
PHONEME ERROR RATES OF THE TIMIT EXPERIMENTS (CONTEXT INDEPENDENT, DIAGONAL COVARIANCE)

	ML-HMM	+AP	+MMI-CRF	+bMMI-CRF				+dMMI-CRF					(MCE-HMM)
σ (σ_2)	–	–	0.0	1.0	2.0	4.0	8.0	0.0625	1.0	2.0	4.0	8.0	[5]
σ_1								-0.0625	-1.0	-2.0	-4.0	-8.0	
1 mix.	42.1	**34.4**†	38.4†	36.7†	35.2†	35.0†	36.4†	38.1†	35.6†	34.8†	34.6†	36.7†	37.4
2 mix.	39.2	33.3†	36.8†	34.5†	34.1†	33.2†	34.3†	35.8†	33.6†	**32.7**†	33.6†	34.5†	NA
4 mix.	37.4	32.2†	35.0†	33.4†	32.5†	32.9†	34.6†	34.6†	32.5†	**32.0**†	33.6†	34.1†	32.5
8 mix.	34.2	30.5†	32.9†	31.4†	30.9†	**30.3**†	31.8†	32.0†	31.6†	30.4†	30.5†	31.6†	32.1
16 mix.	32.9	30.2†	32.1	31.2†	29.6†	**29.3**†	31.1†	31.0†	30.1†	**29.3**†	29.5†	31.4†	30.5
32 mix.	32.1	29.9†	31.9	30.8†	29.5†	29.1†	30.5†	30.5†	29.9†	**28.8**†	28.9†	30.5†	NA

TABLE IV
PHONEME ERROR RATES OF THE TIMIT EXPERIMENTS (CONTEXT INDEPENDENT, FULL COVARIANCE)

	ML-HMM	+AP	+MMI-CRF	+bMMI-CRF				+dMMI-CRF					(LM-HMM)
σ (σ_2)	–	–	0.0	1.0	2.0	4.0	8.0	0.0625	1.0	2.0	4.0	8.0	[39]
(σ_1)								-0.0625	-1.0	-2.0	-4.0	-8.0	
1 mix.	36.3	31.4†	33.8†	32.7†	31.9†	31.3†	32.7†	33.4†	32.1†	31.7†	**31.2**†	32.8†	29.6
2 mix.	33.1	29.3†	31.4†	30.3†	29.7†	**28.5**†	29.5†	30.3†	29.6†	28.6†	28.7†	29.8†	28.0
4 mix.	30.5	29.1†	30.0	29.1†	28.6†	28.5†	29.8	29.2†	28.1†	**27.9**†	28.5†	30.0	27.7
8 mix.	29.3	28.0†	28.7	28.2	27.6†	**27.1**†	27.5†	28.1	27.8†	27.2†	27.2†	28.0†	27.6
16 mix.	28.8	28.4	28.8	28.4	28.1	**27.1**†	27.5†	28.1	28.0	27.4†	**27.1**†	27.5†	NA
32 mix.	28.8	27.8	28.5	28.5	28.0	**26.9**†	27.0†	28.3	28.0	27.8	27.1†	27.1†	NA

TABLE V
PHONEME ERROR RATES OF THE TIMIT EXPERIMENTS (CONTEXT DEPENDENT, DIAGONAL COVARIANCE)

	ML-HMM	+AP	+MMI-CRF	+bMMI-CRF				+dMMI-CRF				
σ (σ_2)	–	–	0.0	1.0	2.0	4.0	8.0	0.0625	1.0	2.0	4.0	8.0
(σ_1)								-0.0625	-1.0	-2.0	-4.0	-8.0
1 mix.	34.7	30.9†	33.2†	31.8†	30.8†	29.8†	30.3†	32.8†	30.8†	29.7†	**29.6**†	30.4†
2 mix.	32.0	29.5†	31.4	30.4†	29.5†	**28.6**†	29.0†	30.9	29.5†	29.0†	28.9†	28.7†
4 mix.	29.3	27.9†	28.9	28.6	28.1	**27.3**†	28.0†	28.8	28.5	27.9†	27.4†	27.8†
8 mix.	29.1	28.4	28.2	28.4	27.4†	**26.9**†	27.7†	28.2	28.1	28.0	**26.9**†	27.2†
16 mix.	30.1	29.0	29.2	28.7†	27.9†	**26.7**†	27.7†	29.2	28.0†	27.4†	27.0†	27.8†
32 mix.	30.1	28.8†	29.5	29.2	28.1†	27.7†	28.0†	29.3	28.6†	28.0†	**27.5**†	28.0†

anticipate that the proposed method will be similar to context dependent HMMs. Tables V and VI are the phoneme error rates of systems with CD models. We still observed the advantages of the proposed method in these experimental results. The CD models uses clustered triphones that can be assumed to be a structure that shares the same GMM parameter among several WFST arcs. Contrastingly, the proposed method involves independent parameters that represent the linear classifier for each WFST arc. We consider that these advantages originate from the unified modeling of acoustic and language models. Although the contextual information considered in the proposed method is similar to that of CD-HMMs, the advantage attributed to the joint optimization of the acoustic and language models is still in effect. We also observed that the relative advantages of the AP, compared with MMI-CRF, were weaker in the CD cases. This might be due to the sparseness in the AP approach. Since AP only considers one competitor hypothesis each update, the parameters tend to be sparse, especially with large WFSTs. Even though sparseness is important for computational efficiency, it makes the intermediate solutions unreliable. Although the convergence of the AP training is ensured, since well-converged parameters often lead to overfitting, the efficiency of the intermediate solutions is important. The relative advantage of CRF-based methods might be attributed to the lattice-based optimization method that is well-suited to early-stopping of optimization. We observed that the best hyper-parameters σ, σ_1, σ_2 minimizing the error rates of the development dataset also achieved the best performance for the evaluation dataset. Specifically, we observed a 24.7 phoneme error rate for the development dataset when the dMMI-CRF method was used with $\sigma_1 = -8$, $\sigma_2 = 8$ and 4-mixture HMMs that also achieved the best result (26.1) for the evaluation dataset.

By discarding the terms related to MFCC features in (8) from the CI systems, we can obtain a variant of the discriminative language model technique [30] that models phoneme sequences by using phoneme bigram features. Table VII shows the phoneme error rates of the proposed system with and without MFCC features. The number of mixture components, training method, and hyper-parameters used in the experiments are selected by using the development dataset. The table reveals that the proposed method achieved slight improvements from the baseline even without MFCC features; however, we also observed the advantage of introducing acoustic features. Further enhancement of the features is possible and would be promising.

B. Large Vocabulary Continuous Speech Recognition

This section describes the results of the LVCSR experiments. We primary used the MIT-OCW corpus, and to ensure performance stability, we also used the WSJ corpus (see Table I).

As with the previous experiments, we trained our structural classifiers by using the perceptron and the CRF approaches. As we described above, perceptron training algorithms perform

TABLE VI
PHONEME ERROR RATES OF THE TIMIT EXPERIMENTS (CONTEXT DEPENDENT, FULL COVARIANCE)

	ML-HMM	+AP	+MMI-CRF	+bMMI-CRF				+dMMI-CRF				
σ (σ_2)	–	–	0.0	1.0	2.0	4.0	8.0	0.0625	1.0	2.0	4.0	8.0
(σ_1)								-0.0625	-1.0	-2.0	-4.0	-8.0
1 mix.	28.4	27.6	28.8	28.4	27.7	27.8	26.7†	28.0	27.8	**26.3**†	26.6†	26.9†
2 mix.	28.1	26.7†	27.5	27.4	27.1	27.0	26.6†	27.5	27.1	26.8†	**26.2**†	26.3†
4 mix.	27.1	27.1	27.3	27.0	27.4	26.7	26.2	27.3	27.0	27.0	26.7	**26.1**
8 mix.	28.4	28.4	27.6	27.5	27.5	27.2	26.8†	27.2	27.6	27.5	27.3	**26.6**†
16 mix.	29.0	29.0	29.2	28.9	29.0	28.4	**28.1**	28.6	28.3	28.7	28.5	**28.1**
32 mix.	31.5	30.6	30.1†	30.0†	30.0†	28.8†	28.7†	30.4	30.2†	30.2†	**28.6**†	**28.6**†

TABLE VII
PHONEME ERROR RATES (PERS) OF THE PROPOSED METHOD (WITH CONTEXT-INDEPENDENT HMMS) OBTAINED BY VARYING USED FEATURES

features	baseline	without MFCCs	with MFCCs
PERs [%]	28.8	28.5	26.9

TABLE VIII
WORD ERROR RATES OF WFST-BASED CRF AND PERCEPTRON (THE VALUES IN PARENTHESES ARE THE WORD ERROR RATES OF THE DEVELOPMENT SETS)

Method	σ (bMMI) (σ_1, σ_2) (dMMI)	**MIT-OCW** **WER [%]**	**WSJ20k** **WER [%]**
ML-HMM	–	32.8 (40.8)	7.8 (11.5)
MPE-HMM	–	28.3 (39.0)	7.8 (11.5)
bMMI-HMM [1]	–	28.3 (37.4)	7.7 (11.5)
dMMI-HMM [2]	–	28.2 (38.2)	7.7 (11.5)
+ DP	–	27.8 (37.8)	7.9 (10.9)
+ MMI-CRF	0.0	27.7 (37.3)	7.6 (11.1)
+ bMMI-CRF	1.0	27.4 (37.3)	7.6 (11.0)
	2.0	**27.1** (36.2)	7.4 (11.0)
	4.0	27.4 (36.4)	**7.3** (11.2)
+ dMMI-CRF	(-1.0, 1.0)	27.8 (37.5)	7.4 (10.7)
	(-2.0, 2.0)	27.3 (36.4)	7.4 (11.0)
	(-4.0, 4.0)	27.6 (36.5)	**7.3** (11.1)
+ dMMI-CRF	(-0.25, 0.25)	27.9 (37.6)	7.9 (11.1)
(→ MPE)	(-0.0625, 0.0625)	28.0 (37.6)	7.7 (11.3)
+ dMMI-CRF	(-10.0, 2.0)	27.3 (36.0)	7.8 (12.0)
(→ bMMI)	(-50.0, 2.0)	**27.1** (36.2)	**7.3** (11.0)

[1] $(\sigma_1, \sigma_2) = (-3, 3)$ for MIT-OCW and $(\sigma_1, \sigma_2) = (-0.01, 0.01)$ for WSJ20k Were Tuned by Using the Development Datasets.
[2] $\sigma = 3.0$ for MIT-OCW and $\sigma = 1.0$ for WSJ20k Were Tuned by Using the Development Datasets

hypothesis generation in each iteration, which is corresponding to full ASR decoding in our case. However, ASR decoding processes generally require much computational costs. In such cases, the AP approach is not suitable since it cannot be parallelized effectively. Therefore, we used the DP approach described in Section III-B instead. The baseline acoustic models were obtained by using maximum likelihood training, MPE training, and dMMI training. All the structural classifiers were combined with the dMMI acoustic models. As in the previous experiments, triphone clustering was performed by using the VBEC method [39]. 2565 and 2466 clustered HMM states were obtained for the MIT-OCW and the WSJ20k tasks, respectively, and the number of mixture components was set at 32 for both tasks. The trigram language models were estimated by using the maximum-likelihood procedure and smoothed by using the Kneser–Ney smoothing method for both tasks. The language model scale factor α was set at 11 and 16 for the MIT-OCW and WSJ tasks, respectively, and the lattice smoothing factor κ was set at 1.0 for both tasks. These tuning factors were determined by using the development datasets. The definition of the frame-level acoustic feature function was the same as in the previous experiments; i.e., $\boldsymbol{\psi}(\boldsymbol{x}) \stackrel{\text{def}}{=} [\boldsymbol{x}^{\mathsf{T}}, 1]^{\mathsf{T}}$. The optimization loops of the structural classifiers (DP and CRF) were stopped when the performance on the development datasets was saturated.

The decoding networks $\boldsymbol{D}$ for both tasks were constructed as $\boldsymbol{D} = \boldsymbol{D}^{(1)} \diamond \boldsymbol{G}^{(2,3)}$ where $\diamond$ denotes a fast on-the-fly composition [24], and $\boldsymbol{G}^{(2,3)}$ denotes a weighted finite-state acceptor that represents bigram and trigram probability. $\boldsymbol{D}^{(1)}$ is a unigram WFST constructed as $\boldsymbol{D}^{(1)} = \mathrm{opt}(\mathrm{opt}(\boldsymbol{H} \circ \boldsymbol{C}) \circ \mathrm{opt}(\boldsymbol{L} \circ \boldsymbol{G}^{(1)}))$. To obtain the arc identifier $N[a_m]$, we numbered the arcs in the unigram decoding WFST $\boldsymbol{D}^{(1)}$. The arc identifier $N[a_m]$ was taken from the arc number annotated to the corresponding arcs in the first WFST $\boldsymbol{D}^{(1)}$. There were 108 485 and 47 826 unique WFST arc identifiers in the MIT-OCW and WSJ20k tasks, respectively.

Table VIII summarizes the word error rates of the proposed method and the compared methods. First, we confirmed that the structural classification approaches were efficient even in the LVCSR tasks. Even though the number of parameters were increased greatly, overfitting was not so crucial in these experiments. We confirmed that the structural classification methods outperformed techniques based on the state-of-the-art discriminative training methods (4.2% relative error reduction from MPE and 3.9% relative error reduction from dMMI). Even though statistical significance is not observed in the WSJ task, all the performance improvements in the MIT-OCW task are confirmed to be statistically significant compared with the "dMMI-HMM" baseline system (significance level $p = 0.05$; computed using the Z-test). In contrast to the TIMIT experiments, the advantage of the CRF approach is emphasized. As we observed in the previous experiments, we confirmed that the CRF-based optimization is advantageous when the decoding WFST is large. Furthermore, we confirmed that the bMMI criterion, which was inferior to the dMMI method in acoustic model training, worked efficiently when training the proposed classifier. This advantage might be due to the non-convexity of the dMMI criterion. It is well-known that high-dimensional parameters often exacerbate local optimum problems. Since the proposed classifier uses a high-dimensional representation of speech segments, the convexity in the training criteria is important. As in the dMMI theory [9], we also confirmed that the dMMI objective function properly converged with the bMMI objective function (see the last two rows of the table). We also examined the effectiveness of the MPE criterion realized by using the above convergence theory. In the experiments, we observed that the MPE criterion was not so effective for

TABLE IX
REAL-TIME FACTORS OF THE DECODERS IN THE MIT-OCW TASK

	Standard	+ Annotation	+ bMMI-CRF
RTFs	1.29x	2.10x	2.27x

training of the proposed classifiers. This also might be due to the non-convexity of the MPE criterion. Although we observed a small discrepancy between the best values of the hyper-parameters $(\sigma,\ \sigma_1,\ \sigma_2)$ for the development set and those for the evaluation dataset in the LVCSR experiments, the tendency was the same. Therefore, we can choose these values by using the development dataset as in the TIMIT experiments.

The training time required for the MIT-OCW task performed with 80 computational threads was approximately 5 hours with the distributed perceptron approach, and 5 minutes with the CRF approach (bMMI-CRF) for each iteration. We required around 15 iterations in the CRF experiments and three iterations in the DP experiments to obtain the best performance with the development dataset. This difference arises from the difference in the styles of implementation. Since, in this paper, we adopted full decoding in the perceptron-based training method and the lattice-based forward–backward algorithm in the CRF-based training method, the perceptron learning appears to be more computationally painful. However, the perceptron approach is still promising because the perceptron training would be faster than the CRF-based training if the lattice approximation was not used.

Table IX shows the real-time factors of the decoders used in the experiments. The RTFs are computed by using a 700 utterance subset of the MIT-OCW test dataset, and a single-thread program. The decoder tuning parameters (beam width and maximum number of hypotheses) are the same as in the previous experiments to minimize search errors in the development datasets. The compared decoders are the standard decoder, the annotation decoder, which preserves the rich information used for lattice outputs during the decoding process, and the structural classification. We confirmed that the introduction of structural classification led to slower decoding than in the conventional decoding processes. This additional computational cost mainly results from the computation of the inner products in (8). However, since the proposed structural classification is still performed in a one-pass manner, we could say that the proposed approach is computationally efficient. Thanks to the one-pass decoding, we can realize low-latency speech recognition that is well-suited for certain applications, such as [1]–[3]. In the MIT-OCW experiments, since the decoding network had 85 639 arcs with output symbols and 22 846 epsilon arcs, the total number of parameters is $40 \cdot 85, 639 + 22, 846 = 3\,448\,406$. We observed that 1 752 696 parameters (50.7%) were nonzero after the DP optimization suggesting that further speeding up of decoding with DPs would be possible by exploiting parameter sparseness. In contrast bMMI optimization does not produce sparse solutions: 3 330 862 (96.6%) parameters were nonzero after the bMMI optimization. The proposed structural classification would be faster than the conventional rescoring-based structural classification because these methods require lattices before the actual process starts. Thus, we confirmed that a one-pass structural classification method is efficiently realized by employing the proposed method.

VI. CONCLUSION

This paper proposed a novel modeling technique for automatic speech recognition based on a structural classification method. Unlike other structural classification approaches, the proposed classifier is capable of handling interdependency of acoustic and linguistic aspects without sacrificing the availability of the computationally efficient one-pass decoding techniques. The key to the work described in this paper is that it considers a WFST decoding process to be a structural classification process. Specifically, the proposed method introduces linear corrective terms, whose coefficients are trained discriminatively, for each WFST arc. Thanks to the WFST-based structures, the proposed approach can devise the rich information contained by the WFSTs without increasing computational cost of decoding drastically. We introduced two approaches for training the classifiers. The first approach involves the perceptron learning of the linear discriminant function. With the first approach, we proposed two training methods based on the averaged perceptron technique and the distributed perceptron technique. The second approach involves the CRF as *a posterior* distribution of WFST arc sequences. With the second approach, we proposed three training methods based on the MMI criterion, the boosted MMI (bMMI) criterion, and the dMMI criterion. The bMMI and dMMI methods are considered to be particularly important since these methods can reduce the number of error transitions to the incorrect decoding state. We evaluated the performance in terms of the speech recognition accuracy of the proposed methods by undertaking the TIMIT phoneme recognition task, the MIT-OCW lecture transcription task, and the WSJ broadcast transcription task. The experimental results revealed that the proposed method realized accurate speech recognition without the need for substantial computational resources. Furthermore, we confirmed that our proposed methods were scalable for large-vocabulary continuous speech recognition tasks. We finally confirmed that, by using WFST-based structural classification, the recognition accuracy can be improved compared with the original systems based on the generative formulation of speech recognition even if the classifiers involved are simply linear classifiers.

Future work will include comparative studies and nonlinear feature augmentation. Since the proposed method can be treated as a variant of discriminative language models, the comparison with the conventional discriminative language models in large-vocabulary continuous speech recognition experiments would be important. Furthermore, because the proposed method only introduces linear terms for each WFST arc, the nonlinear relationship between observation vectors and WFST arcs cannot be expressed. A promising approach for tackling this problem involves using nonlinearly warped features. Recently, several methods have been developed to identify the efficient warping of the feature spaces [42]–[44]. Thus, incorporating these methods in the proposed WFST-based structural classification approach would be promising.

Acknowledgment

The authors would like to thank the MIT Spoken Language Systems Group for helping them to perform experiments based on the MIT-OCW dataset. The authors would also like to thank Y. Fujii at Toyohashi University of Technology for helping them to set up the WSJ evaluation environment.

References

[1] S. Renals, T. Hain, and H. Bourlard, "Recognition and understanding of meetings the AMI and AMIDA projects," in *Proc. IEEE Workshop on ASRU*, Sep. 2007, pp. 238–247.

[2] T. Hori, S. Araki, T. Yoshioka, M. Fujimoto, S. Watanabe, T. Oba, A. Ogawa, K. Otsuka, D. Mikami, K. Kinoshita, T. Nakatani, A. Nakamura, and J. Yamato, "Low-latency real-time meeting recognition and understanding using distant microphones and omni-directional camera," *IEEE Trans. Audio, Speech, Lang. Process.*, vol. 20, no. 2, pp. 499–513, Feb. 2012.

[3] M. Saraclar, M. Riley, E. Bocchieri, and V. Goffin, "Towards automatic closed captioning: Low latency real time broadcast news transcription," in *Proc. ICSLP*, Sep. 2002, pp. 1741–1744.

[4] V. Valtchev, J. J. Odell, P. C. Woodland, and S. J. Young, "Lattice-based discriminative training for large vocabulary speech recognition," *Proc. ICASSP*, pp. 605–608, May 1996.

[5] E. McDermott and S. Katagiri, "String-level MCE for continuous phoneme recognition," *Proc. Eurospeech*, pp. 123–126, Sep. 1997.

[6] D. Povey and P. C. Woodland, "Minimum phone error and I-smoothing for improved discriminative training," *Proc. ICASSP*, vol. 1, pp. 105–108, May 2002.

[7] G. Heigold, T. Deselaers, and R. Schluter, "Modified MMI/MPE: A direct evaluation of the margin in speech recognition," *Proc. ICML*, pp. 384–391, Jul. 2008.

[8] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted MMI for model and feature-space discriminative training," *Proc. ICASSP*, pp. 4057–4060, Mar. 2008.

[9] E. McDermott, S. Watanabe, and N. Atsushi, "Discriminative training based on an integrated view of MPE and MMI in margin and error space," *Proc. ICASSP*, pp. 4894–4897, Mar. 2010.

[10] H.-K. J. Kuo, E. Fosle-Lussier, H. Jiang, and C. H. Lee, "Discriminative training of language models for speech recognition," *Proc. ICASSP*, vol. 1, pp. 325–328, May 2002.

[11] J.-W. Kuo and B. Chen, "Minimum word error based discriminative training language models," in *Proc. Interspeech*, Sep. 2005, pp. 1277–1280.

[12] A. Gunawardana, M. Mahajan, A. Acero, and J. C. Platt, "Hidden conditional random fields for phone classification," *Proc. Interspeech*, pp. 1117–1120, Sep. 2005.

[13] J.-T. Chien and C.-H. Chueh, "Joint acoustic and language modeling for speech recognition," *Speech Commun.*, vol. 52, no. 3, pp. 223–235, Mar. 2010.

[14] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data," *Proc. ICML*, pp. 282–289, Jun. 2001.

[15] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large margin methods for structured and interdependent output variables," *J. Mach. Learn. Res.*, vol. 6, pp. 1453–1484, Sep. 2005.

[16] M. Collins, "Discriminative training methods for hidden Markov models: Theory and experiments with perceptron algorithms," *Proc. EMNLP*, pp. 1–8, Jul. 2002.

[17] Y. Freund and R. E. Schapire, "Large margin classification using the perceptron algorithm," *Mach. Learn.*, vol. 37, no. 3, pp. 277–296, Dec. 1999.

[18] F. Sha and F. Pereira, "Shallow parsing with conditional random fields," *Proc. HLT-NAACL*, pp. 134–141, May 2003.

[19] T. Watanabe, J. Suzuki, H. Tsukada, and H. Isozaki, "Online large-margin training for statistical machine translation," *Proc. EMNLP-CoNLL*, pp. 764–773, Jun. 2007.

[20] G. Zweig and P. Nguyen, "A segmental CRF approach to large vocabulary continuous speech recognition," in *Proc. IEEE Workshop on ASRU*, Dec. 2009, pp. 152–157.

[21] M. Lehr and I. Shafran, "Learning a discriminative weighted finite state transducer for automatic speech recognition," *IEEE Trans. Acoust., Speech, Lang. Process.*, vol. 19, no. 5, pp. 1360–1367, Jul. 2011.

[22] M. Mohri, F. Pereira, and M. Riley, "Weighted finite-state transducers in speech recognition," *Comput. Speech Lang.*, vol. 16, no. 1, pp. 69–88, Jan. 2002.

[23] D. Moore, J. Dines, M. Doss, J. Vepa, O. Cheng, and T. Hain, "Juicer: A weighted finite-state transducer speech decoder," *Proc. Mach. Learn. for Multimodal Interact.*, pp. 285–296, May 2006.

[24] T. Hori, C. Hori, Y. Minami, and A. Nakamura, "Efficient WFST-based one-pass decoding with on-the-fly hypothesis rescoring in extremely large vocabulary continuous speech recognition," *IEEE Trans. Acoust., Speech, Lang. Process.*, vol. 15, no. 4, pp. 1352–1365, May 2007.

[25] J. R. Novak, P. R. Dixon, and S. Furui, "An empirical comparison of the T^3, juicer, Hdecode and Sphinx3 decoders," in *Proc. Interspeech*, Aug. 2010, pp. 1890–1893.

[26] S.-S. Lin and F. Yvon, "Discriminative training of finite state decoding graphs," *Proc. Interspeech*, pp. 733–736, Sep. 2005.

[27] H.-K. J. Kuo, B. Kingsbury, and G. Zweig, "Discriminative training of decoding graphs for large vocabulary continuous speech recognition," *Proc. ICASSP*, vol. 4, pp. 45–48, Apr. 2007.

[28] S. Watanabe, T. Hori, and A. Nakamura, "Large vocabulary continuous speech recognition using WFST-based linear classifier for structured data," *Proc. Interspeech*, pp. 346–349, Aug. 2010.

[29] Y. Kubo, S. Watanabe, and A. Nakamura, "Decoding network optimization using minimum transition error training," *Proc. ICASSP*, pp. 4197–4200, Mar. 2012.

[30] B. Roark, M. Saraclar, M. Collins, and M. Johnson, "Discriminative language modeling with conditional random fields and the perceptron algorithm," *Proc. ACL*, pp. 47–54, Jul. 2004.

[31] T. Oba, T. Hori, A. Ito, and A. Nakamura, "Round-robin duel discriminative language models in one-pass decoding with on-the-fly error correction," *Proc. ICASSP*, pp. 5588–5591, May 2011.

[32] R. Collobert and S. Bengio, "Links between perceptrons, MLPs and SVMs," *Proc. ICML*, pp. 23–29, Jul. 2004.

[33] H. J. Kushner and G. Yin*, Stochastic Approximation and Recursive Algorithms and Applications*. New York: Springer-Verlag, 2003, vol. 35.

[34] R. McDonald, K. Hall, and G. Mann, "Distributed training strategies for the structured perceptron," *Proc. HLT-NAACL*, pp. 456–464, 2010.

[35] A. Ragni and M. J. F. Gales, "Structured discriminative models for noise robust continuous speech recognition," *Proc. ICASSP*, pp. 4781–4791, Mar. 2011.

[36] S. X. Zhang and M. J. F. Gales, "Structured support vector machines for noise robust continuous speech recognition," *Proc. Interspeech*, pp. 989–992, Aug. 2011.

[37] J. Zheng and A. Stolcke, "Improved discriminative training using phone lattices," *Proc. Interspeech*, pp. 2125–2128, Sep. 2005.

[38] J. Glass, T. J. Hazen, S. Cyphers, I. Malioutov, D. Huynh, and R. Barzilay, "Recent progress in the MIT spoken lecture processing project," *Proc. Interspeech*, pp. 2553–2556, Aug. 2007.

[39] T.-M.-T. Do and T. Aritieres, "Large margin training for hidden Markov Models with partially observed states," *Proc. ICML*, pp. 265–272, Jun. 2009.

[40] S. Watanabe, Y. Minami, A. Nakamura, and N. Ueda, "Variational Bayesian estimation and clustering for speech recognition," *IEEE Trans. Speech Audio Process.*, vol. 12, no. 4, pp. 365–381, Jul. 2004.

[41] M. Riedmiller and H. Braun, "A direct adaptive method for faster backpropagation learning: The Rprop algorithm," in *Proc. ICNN*, Mar. 1993, pp. 586–591.

[42] Y. Kubo, S. Wiesler, R. Schlueter, H. Ney, S. Watanabe, A. Nakamura, and T. Kobayashi, "Subspace pursuit method for kernel-log-linear models," in *Proc. ICASSP*, Mar. 2011, pp. 4500–4503.

[43] S. Wiesler, M. Nussbaum-Thom, G. Heigold, R. Schlueter, and H. Ney, "Investigations on features for log-linear acoustic models in continuous speech recognition," in *Proc. IEEE Workshop on ASRU*, Dec. 2009, pp. 52–57.

[44] G. E. Dahl, M. Ranzato, A. Mohamed, and G. Hinton, "Phone recognition with the mean-covariance restricted Boltzmann machine," *Adv. in NIPS*, vol. 23, pp. 469–477, 2010.

Yotaro Kubo (M'10) received the B.E., M.E., and Dr. Eng. degrees from Waseda University, Tokyo, Japan, in 2007, 2008, and 2010, respectively.

He was a Visiting Scientist at RWTH Aachen University from April to October 2010. In 2010, he joined Nippon Telegraph and Telephone Corporation (NTT) and has been with NTT Communication Science Laboratories, Kyoto, Japan. His research interests include machine learning and signal processing.

Dr. Kubo is a member of the International Speech Communication Association (ISCA), the Acoustical Society of Japan (ASJ), the Institute of Electronics, Information and Communication Engineers (IEICE), and the Information Processing Society of Japan (IPSJ). He received the Awaya

Award from the ASJ in 2010, the Yamashita SIG Research Award from the IPSJ in 2011, and the IEEE Signal Processing Society Japan Chapter Student Paper Award in 2011.

Shinji Watanabe (SM'12) received the B.S., M.S., and Dr. Eng. degrees from Waseda University, Tokyo, Japan, in 1999, 2001, and 2006, respectively.

From 2001 to 2011, he was a Research Scientist at NTT Communication Science Laboratories, Kyoto, Japan. From January to March in 2009, he was a Visiting Scholar at the Georgia Institute of Technology, Atlanta. From 2011, he has been working at Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA. His research interests include Bayesian learning, pattern recognition, and speech and spoken language processing.

Dr. Watanabe is a member of the Acoustical Society of Japan (ASJ) and the Institute of Electronics, Information and Communications Engineers (IEICE). He received the Awaya Award from the ASJ in 2003, the Paper Award from the IEICE in 2004, the Itakura Award from ASJ in 2006, and the TELECOM System Technology Award from the Telecommunications Advancement Foundation in 2006.

Takaaki Hori (M'04) received the B.E. and M.E. degrees in electrical and information engineering and the Ph.D. degree in system and information engineering from Yamagata University, Yonezawa, Japan, in 1994, 1996, and 1999, respectively.

Since 1999, he has been engaged in research on spoken language processing at the Cyber Space Laboratories, Nippon Telegraph and Telephone (NTT) Corporation, Kyoto, Japan. He was a Visiting Scientist at the Massachusetts Institute of Technology, Cambridge, from 2006 to 2007. He is currently a senior Research Scientist in the NTT Communication Science Laboratories, NTT Corporation.

Dr. Hori received the 22nd Awaya Prize Young Researcher Award from the Acoustical Society of Japan (ASJ) in 2005 and the 24th TELECOM System Technology Award from the Telecommunications Advancement Foundation in 2009. He is a member of the Institute of Electronics, Information, and Communication Engineers (IEICE) and the ASJ.

Atsushi Nakamura (SM'07) received the B.E., M.E., and Dr.Eng. degrees from Kyushu University, Fukuoka, Japan, in 1985, 1987 and 2001, respectively.

In 1987, he joined Nippon Telegraph and Telephone Corporation (NTT), where he engaged in the research and development of network service platforms, including studies on application of speech processing technologies into network services, at Musashino Electrical Communication Laboratories, Tokyo, Japan. From 1994 to 2000, he was with Advanced Telecommunications Research (ATR) Institute, Kyoto, Japan, as a Senior Researcher, working on the research of spontaneous speech recognition, construction of spoken language database and development of speech translation systems. Since April, 2000, he has been with NTT Communication Science Laboratories, Kyoto, Japan. His research interests include acoustic modeling of speech, speech recognition and synthesis, spoken language processing systems, speech production and perception, computational phonetics and phonology, and application of learning theories to signal analysis and modeling.

Dr. Nakamura is a member of the Machine Learning for Signal Processing (MLSP) Technical Committee, as well as serving as a Vice Chair of the Signal Processing Society Kansai Chapter. He is also a member of the Institute of Electronics, Information and Communication Engineering (IEICE) and the Acoustical Society of Japan (ASJ). He received the IEICE Paper Award in 2004, and received twice the TELECOM System Technology Award of The Telecommunications Advancement Foundation, in 2006 and 2009.