LARGE VOCABULARY CONTINUOUS SPEECH RECOGNITION BASED ON WFST STRUCTURED CLASSIFIERS AND DEEP BOTTLENECK FEATURES

Yotaro Kubo, Takaaki Hori, Atsushi Nakamura

NTT Communication Science Laboratories, NTT Corporation, Kyoto 619–0237, Japan

{kubo.yotaro, hori.t, nakamura.atsushi}@lab.ntt.co.jp

ABSTRACT

Recently, structured classification approaches have been considered important with a view to achieving unified modeling of the acoustic and linguistic aspects of speech recognizers. With these approaches, unified representation is achieved by directly optimizing a score function that measures the correspondence between the input and output of the system. Since structured classifiers typically employ a linear function as a score function, extracting expressive features from the input and output of the system is very important. On the other hand, the effectiveness of deep neural networks has been verified by several experiments, and it has been suggested that the outputs of hidden layers in deep neural networks (DNNs) are essential speech features that purely express phonetic information. In this paper, we propose a method for structured classification with DNN features. The proposed method expands conventional DNN-based acoustic models so that they optimizes the weight terms of the arcs in a decoding WFST, which is constructed with the on-the-fly composition method. Since DNN-based features can be considered enhancements in the input representation, the enhancements in the output representation based on the WFST arcs are expected to complement the DNN-based features. The proposed method achieved an 8 % relative error reduction even compared with a strong acoustic model based on DNNs.

Index Terms— Speech recognition, weighed finite-state transducers, structured classification, deep neural networks

1. INTRODUCTION

Structured classification approaches have recently been considered successful ways of achieving the joint optimization of the acoustic and linguistic aspects in automatic speech recognition (ASR) [1–3]. These approaches involve the extraction of features that describe both the input and output of the system to be optimized, and the optimization of a function that represents the correspondence score of a given input and output pair. By using these features and score functions, the unified model for ASR can be consistently optimized by using a discriminative criterion. Even though there is a direct approach that jointly optimizes the parameters of conventional acoustic models and language models (e.g. [4]), structured classification has further advantages deriving from the flexibility of feature design.

On the other hand, recent developments of acoustic modeling based on deep neural networks (DNNs) has been producing very attractive results in several speech recognition tasks, including large vocabulary continuous speech recognition (LVCSR) tasks [5, 6]. DNNs are typically used to substitute Gaussian mixture models (GMMs) in continuous density hidden Markov model (CD-HMM) acoustic models. To date, DNNs have mainly been used to enhance the input representation of ASR.

In structured classifiers, a linear function is typically used as a function that represents the correspondence score to ensure convexity in the optimization. Therefore, several nonlinear features are used that map input samples to a linearly separable space so that the following linear function provides accurate classification. For example, the log-likelihoods of GMMs in HMM-based acoustic models are used in [7], coefficients of Fisher information matrices are used in [3], and higher order polynomial features are used in [8]. Recently, methods have been proposed that introduce the outputs of hidden layers in DNNs as features for structured classification [9, 10]. Since DNN-based acoustic models for ASR outperform conventional speech recognizers, the application of DNN-based features to structured classifiers might also be promising as regards enabling both the accurate acoustic representation and unified optimization of ASR. Even though these conventional structured classifiers are potentially capable of leveraging rich contextual features to represent the interdependence of the input and output of recognizers, features and models must be restricted to the same form as the conventional HMM-based acoustic model and N-gram based language models if we are to apply one-pass decoding techniques to these classifiers.

In this paper, instead of using the conventional HMM and N-gram model structure, we employed a structure determined by a WFST representing a decoding network. Since WFSTs are constructed by expanding all possible state transitions combinations in HMM, triphone model, pronunciation lexicon model, and so on, the state transitions in WFSTs contain rich lexical contexts. To leverage such lexical information, we replace the fixed cost parameters of the first WFST in an on-the-fly composition chain with optimizable functions that score DNN-based acoustic features by using linear functions with a parameter vector defined for each WFST arc. By optimizing the parameter vectors with sequential discriminative training criteria, whole aspects of the speech recognizers are optimized in terms of word error rates (WERs). Thanks to the WFST-based structure, most one-pass decoding techniques are still straightforwardly applicable to structural classifiers that have the same structure as conventional decoding networks.

This paper is an extension of [11]. The method proposed in [11] is not combined with large N-gram language models, and the preliminary results for a continuous phoneme recognition task are presented. To enable an effective combination with realistic N-gram language models, we exploit a structure of weight terms of WFSTs constructed with an on-the-fly composition algorithm. The strategy adapted in this work is based on the strategy used to scale-up the GMM-based structured classifiers to LVCSR problems [7]. We verify that this scaling up strategy is also applicable to DNN-based structured classification even though the features used are high dimensional and very expressive.

2. WFST-DNN STRUCTURED CLASSIFIERS

Speech recognizers based on WFSTs output the most plausible word sequence ($\hat{\boldsymbol{\ell}}$) corresponding to the input observation sequence $\mathbf{X}$ by

extracting the output labels $\boldsymbol{O}[\hat{\mathbf{a}}]$ assigned to each WFST arc in the most plausible arc sequence $\hat{\mathbf{a}}$, as follows:

$$\hat{\boldsymbol{\ell}} = \boldsymbol{O}[\hat{\mathbf{a}}] \text{ where } \hat{\mathbf{a}} = \underset{\mathbf{a}\in\boldsymbol{\mathcal{D}}}{\operatorname{argmax}} P(\mathbf{a}|\mathbf{X}), \tag{1}$$

where $\boldsymbol{\mathcal{D}}$ is a WFST represented as a set of possible arc sequences. In general, the probability of an arc sequence $\mathbf{a} = \{a_1, a_2, \cdots a_n \cdots\}$ is defined by using transition cost function $\omega(a_n; \mathbf{X})$, as follows:

$$P(\mathbf{a}|\mathbf{X}) \propto \exp\left\{\sum_n -\omega(a_n; \mathbf{X})\right\}. \tag{2}$$

Even though the proposed method is applicable to one-pass decoding, we consider that each arc a_j in the arc sequences has annotations providing alignment information; therefore, each arc has an output label $O[a_j]$, an input HMM-state label $I[a_j]$, a weight term $W[a_j]$, start time $T[a_j]$, and stop time $T'[a_j]$.

Since the number of arcs required to represent all possible state changes in a recognition system increases exponentially in large vocabulary systems, the on-the-fly composition method is introduced in such cases. With this method, the overall decoding network $\boldsymbol{\mathcal{D}}$ is decomposed into several networks $\boldsymbol{\mathcal{D}} = \boldsymbol{\mathcal{D}}^{(1)} \circ \boldsymbol{\mathcal{D}}^{(2)} \circ \cdots$, where $\circ$ is the composition operator. Moreover, each arc a_j in $\boldsymbol{\mathcal{D}}$ is represented as a tuple of arcs in the decomposed networks as $a_j = (a_{j_1}, a_{j_2}, \cdots)$.

With this notation the transition cost function can be denoted as follows:

$$\omega(a_j; \mathbf{X}) = g(a_{j_1}; \mathbf{X}) + c\sum_k W[a_{j_k}], \tag{3}$$

where c is a tuning parameter called a language model scale factor. Here, we introduce an acoustic cost function $g(a_{j_1}; \mathbf{X})$ that represents a negative logarithm of emission probability in HMMs. With DNN-based acoustic models, g is denoted as follows:

$$g(a_{j_1}; \mathbf{X}) = \sum_{\tau=T[a_{j_1}]}^{T'[a_{j_1}]} \left(\left(\boldsymbol{w}_{I[a_{j_1}]}^{(L+1)}\right)^{\mathsf{T}} \boldsymbol{h}^{(L)}(\boldsymbol{x}_\tau) + b_{I[a_{j_1}]}^{(L+1)}\right), \tag{4}$$

where L is the number of hidden layers in the DNN, $\boldsymbol{w}_{I[a_{j_1}]}^{(L+1)}$ is a vector containing the $I[a_{j_1}]^{\text{th}}$ row of the $(L+1)^{\text{th}}$ weight matrix (the last weight matrix), $b_{I[a_{j_1}]}^{(L+1)}$ is the $I[a_{j_1}]^{\text{th}}$ element of the $(L+1)^{\text{th}}$ bias vector, and $\boldsymbol{h}^{(L)}(\boldsymbol{x}_\tau)$ is the output vector of the L^{th} hidden layer (the last hidden layer) as a function of the input vector $\boldsymbol{x}_\tau$.

This paper enhances the acoustic cost function $g(a_{j_1}; \mathbf{X})$ to capture not only the acoustic costs but also the linguistic costs and interdependence of these costs by modifying the cost function directly depending on the arc variable a_{j_1}. We modified this cost function so that the parameters depend on the arc, not the HMM-states annotated to the arc, and introduced a corrective term for the transition cost. By introducing these modifications, the unified cost function can be defined as follows:

$$g(a_{j_1}; \mathbf{X}) = \gamma_{a_{j_1}} + \sum_{\tau=T[a_j]}^{T'[a_j]} \left(\left(\boldsymbol{\alpha}_{a_{j_1}}\right)^{\mathsf{T}} \boldsymbol{h}^{(L)}(\boldsymbol{x}_\tau) + \beta_{a_{j_1}}\right), \tag{5}$$

where $\gamma_{a_{j_1}}$ is a corrective term for the arc weight $W[a_{j_1}]$. It should be noted that the unified cost function introduces an untied parameter representation $(\boldsymbol{\alpha}_{a_{j_1}}, \beta_{a_{j_1}})$ that is tied with the HMM-state variable $I[a_{j_1}]$ as $\boldsymbol{w}_{I[a_{j_1}]}^{(L+1)}, b_{I[a_{j_1}]}^{(L+1)}$ in the original acoustic cost function (Eq. (4)). By optimizing the parameters $\Lambda \overset{\text{def}}{=} \{\boldsymbol{\alpha}_{a_{j_1}}, \beta_{a_{j_1}}, \gamma_{a_{j_1}} | \forall a_{j_1}\}$ discriminatively, we can perform the overall discriminative optimization of the speech recognizers. Even though the unified cost function depends only on the arc variable of the first WFST $\boldsymbol{\mathcal{D}}^{(1)}$ in the on-the-fly composition chain $\boldsymbol{\mathcal{D}}^{(1)} \circ \boldsymbol{\mathcal{D}}^{(2)} \circ \cdots$, the proposed method is capable of optimizing the linguistic aspects of the recognizer by designing the chain so that the first WFST $\boldsymbol{\mathcal{D}}^{(1)}$ expresses a sufficiently rich structure. In the following experiments, we used a WFST composed of hidden Markov models, triphone context models, pronunciation lexicon models, and unigram language models as $\boldsymbol{\mathcal{D}}^{(1)}$, and trigram language models as $\boldsymbol{\mathcal{D}}^{(2)}$.

Since the unified cost function is a sum of the frame synchronous term $(\boldsymbol{\alpha}_{a_{j_1}})^{\mathsf{T}} \boldsymbol{h}^{(L)}(\boldsymbol{x}_\tau)$ and the arc synchronous term $\gamma_{a_{j_1}}$, the actual decoding can be performed without fixing the alignment $T[a_{j_1}], T'[a_{j_1}]$. Therefore, the outputs of the proposed structured classifier can be computed by using conventional WFST-based decoding techniques. We call this structured classifier as "WFST-DNN" in the following sections.

3. TRAINING WFST-DNN MODELS

Several sequential training criteria can be used with the proposed method. In this paper, we adopt and describe two training criteria based on the maximum mutual information (MMI) criterion; boosted MMI (bMMI) and differenced MMI (dMMI). Hereafter, we refer to the i^{th} observation vector sequence in the training dataset as $\mathbf{X}^{(i)}$, the i^{th} transcription represented in WFST as $\boldsymbol{\mathcal{T}}^{(i)}$, and the i^{th} oracle arc sequence as $\mathbf{a}^{(i)}$. The oracle arc sequences are computed as follows:

$$\mathbf{a}^{(i)} = \underset{\mathbf{a}\in(\boldsymbol{\mathcal{D}}\circ\boldsymbol{\mathcal{T}}^{(i)})}{\operatorname{argmax}} P(\mathbf{a}|\mathbf{X}, \Lambda') \tag{6}$$

where Λ' is the initial value of the optimization. The initial value $\Lambda' = \{\boldsymbol{\alpha}'_{a_{j_1}}, \beta'_{a_{j_1}}, \gamma'_{a_{j_1}} | \forall a_{j_1}\}$ is taken from an optimized DNN system as follows:

$$\boldsymbol{\alpha}'_{a_{j_1}} = \boldsymbol{w}_{I[a_{j_1}]}^{(L+1)},\ \beta'_{a_{j_1}} = b_{I[a_{j_1}]}^{(L+1)},\ \gamma'_{a_{j_1}} = 0, \tag{7}$$

where $\boldsymbol{w}_{I[a_{j_1}]}^{(L)}$ and $b_{I[a_{j_1}]}^{(L)}$ are the corresponding parameters in the DNN system, and the corrective term for arc weights $\gamma'_{a_{j_1}}$ is initialized by 0. Furthermore, we obtained lattices $\mathcal{L}^{(i)}$ corresponding to each observation sequence $\mathbf{X}^{(i)}$ by using the DNN system. Taking initial values from the optimized DNNs is important to ensure the validity of competing hypotheses in the lattices. The DNNs used in this study are optimized by using generative pretraining based on the contrastive divergence method, and frame-wise discriminative training based on stochastic gradient descent.

The first criterion we adopted for training the WFST-DNN is the **boosted MMI (bMMI)** criterion, which is commonly used to devise a fine error measure in MMI [12]. The objective function of bMMI is defined with a hyperparameter σ, as follows:

$$F_\sigma^{\texttt{bMMI}}(\Lambda) = \sum_n \log \frac{\exp\left\{-\Omega\left(\mathbf{X}^{(i)}, \mathbf{a}^{(i)}\right)\right\}}{\sum_{\mathbf{a}'\in\mathcal{L}^{(i)}} \exp\left\{-\Omega\left(\mathbf{X}^{(i)}, \mathbf{a}'\right) + \sigma E(\mathbf{a}^{(i)}, \mathbf{a}')\right\}}, \tag{8}$$

where the total cost function $\Omega(\mathbf{X}^{(i)}, \mathbf{a})$ is defined as follows:

$$\Omega\left(\mathbf{X}^{(i)}, \mathbf{a}\right) = \sum_n -\omega(a_n; \mathbf{X}^{(i)}). \tag{9}$$

It should be noted that Ω also depends on the parameter Λ since we use the unified cost function as Eq. (5). Here, $E(\mathbf{a}^{(i)}, \mathbf{a}')$ is a measure of the errors included in the hypothesis $\mathbf{a}'$. In this paper, we use the transition error count, which is computed by counting the number of frames that produce an erroneous WFST transition [7], as the measure of the errors. The hyperparameter σ is used to adjust the impact of the error measure, which is usually tuned by performing validation on a development dataset. By setting $\sigma = 0$, the bMMI objective function is equivalent to that of MMI. One of the advantages of using bMMI is that the objective function is convex.

The **differenced MMI (dMMI)** criterion is a discriminative criterion that generalizes the bMMI, minimum phone error (MPE) [13], and other discriminative criteria [14]. The objective function of dMMI is defined with hyperparameters σ_1, σ_2, as follows:

$$F^{\texttt{dMMI}}_{\sigma_1,\sigma_2}(\Lambda) = \left(F^{\texttt{bMMI}}_{\sigma_2}(\Lambda) - F^{\texttt{bMMI}}_{\sigma_1}(\Lambda)\right) / \left(\sigma_2 - \sigma_1\right). \quad (10)$$

This objective function converges to the MPE objective function in the limit of $\sigma_1 \to -0, \sigma_2 \to +0$. Furthermore, this objective function also converges to the bMMI objective function with the hyperparameter σ in the limit of $\sigma_1 \to -\infty, \sigma_2 \to \sigma$.

In the following experiments, we introduced L2-regularization terms to the above basic objective functions, as follows:

$$\tilde{F}(\Lambda) = F(\Lambda) - p\sum_{a_{j_1}} ||\boldsymbol{\alpha}_{a_{j_1}}||_2^2 - q\sum_{a_{j_1}} ||\beta_{a_{j_1}}||_2^2 - r\sum_{a_{j_1}} ||\gamma_{a_{j_1}}||_2^2 \quad (11)$$

where $F(\Lambda)$ is one of the above-mentioned objective functions (dMMI or bMMI).

The parameter Λ with respect to these objective functions can be optimized by adopting an arbitrary gradient-based optimization method. The gradient vector of the parameters with respect to the bMMI objective function can be computed by using the lattice-based forward-backward algorithm, and that of the dMMI objective function can be obtained by computing the difference between two bMMI gradient vectors.

4. EXPERIMENTS

We conducted continuous speech recognition experiments to evaluate the efficiency of our proposed approach in LVCSR tasks. We applied the proposed method to the MIT-OCW/World lecture recognition task [15], and evaluated the WERs. The details of the corpus we used are provided in Table 1.

In the experiments, 12 Mel-frequency cepstral coefficients (MFCCs) and the logarithmic energy were extracted and augmented by their derivatives and accelerations. Furthermore, 11 consecutive frames of extracted MFCC feature vectors were concatenated to form input vectors for DNNs. The number of clustered HMM states and the number of mixture components per state were 2,565 and 32, respectively, as determined using variational Bayesian model clustering [16]. A speech recognizer with GMMs was optimized by using the dMMI training procedure [14] to compute the initial state alignment for DNN training.

Table 1. MIT-OCW/World lecture recognition task

	Training	Development	Evaluation
# utterances	55,763	726	6,989
Duration	101h	0.9h	7.8h
# lexicon words		44,485	
# running words	1,128,169	9,514	72,159

We constructed a basic DNN system with 8 hidden layers each of which has 2048 hidden units. The language model scale factor (c in Eq. (3)) was fixed at 8.0 for all DNN-based systems (including WFST-DNN), which was determined in order to minimize the WERs on the development dataset with this basic DNN acoustic model. Moreover, to improve the computational efficiency of the WFST-DNN system, we also prepared a DNN with a "bottleneck layer" (DNN-BN). The DNN-BN system had 8 hidden layers with 2048 hidden units and 1 hidden layer, called a "bottleneck layer", with 512 hidden units located adjacent to the output layer of the DNNs. Although the number of total parameters was reduced by introducing this bottleneck layer, we observed that the WERs on the development dataset were reduced from 28.7 to 28.3.

The decoding network was constructed as $\boldsymbol{\mathcal{D}} = \boldsymbol{\mathcal{D}}^{(1)} \circ \boldsymbol{\mathcal{D}}^{(2)}$ by using an efficient on-the-fly composition [17]. The first WFST $\boldsymbol{\mathcal{D}}^{(1)} = \mathrm{Opt}(\mathrm{Opt}(\boldsymbol{\mathcal{H}} \circ \boldsymbol{\mathcal{C}}) \circ \mathrm{Opt}(\boldsymbol{\mathcal{L}} \circ \boldsymbol{\mathcal{G}}^{(1)}))$, where $\mathrm{Opt}(.)$ was the WFST optimization operator, was constructed by statically composing hidden Markov models $\boldsymbol{\mathcal{H}}$, triphone context models $\boldsymbol{\mathcal{C}}$, pronunciation lexicon models $\boldsymbol{\mathcal{L}}$, and unigram language models $\boldsymbol{\mathcal{G}}^{(1)}$. The second WFST $\boldsymbol{\mathcal{D}}^{(2)} = \boldsymbol{\mathcal{G}}^{(2,3)}$ was a trigram language model normalized by unigram probabilities that was estimated by using a maximum likelihood procedure and the Kneser-Ney smoothing technique [18]. The numbers of arcs in $\boldsymbol{\mathcal{D}}^{(1)}$ and $\boldsymbol{\mathcal{D}}^{(2)}$ were 236,228 and 4,946,612, respectively. Since the first WFST $\boldsymbol{\mathcal{D}}^{(1)}$ had 234,838 arcs that had a non-epsilon input symbol, the total number of optimizable parameters in the WFST-DNNs was $234838 \times (512+1) + 236228 = 120708122$.

We used the Rprop method [19] to optimize parameter Λ since this method can be effectively performed with grid computers. With 100 parallel computation threads, the computation time required to compute a single iteration was around 10 minutes, and optimizations typically converged within 15 steps. We fixed the regularization terms in Eq. (11) as $p = 0.0002, q = 0, r = 0$ since the setting $p = 0.0002, q = 0$ is commonly used to train DNNs, and $r = 0$ would be appropriate for avoiding underfitting.

Figs. 1 and 2 show the WERs on the development dataset as functions of the numbers of iterations in the optimization of DNN-WFST. We observed that all settings successfully reduced the WERs from the initial parameters taken from the DNN-BN systems if the optimization was stopped with appropriate timing. Thus, it is suggested that the expanded representation is effective for performing improved classification. The optimal numbers of iterations of WFST-DNNs used in the following experiments were selected to minimize the WERs in Figs 1 and 2.

Table 2 shows a comparison of the training objective functions and the hyper-parameters. From the table, we confirmed that the hyperparameters $\sigma, \sigma_1, \sigma_2$ that minimized the WER on the development dataset also produced a smaller WER on the evaluation dataset even if the development dataset used in the experiments was not very large. In addition, we also confirmed that the choice of the training criteria was not very sensitive, which is similar to the previous experiments with GMM-based WFST structured classifiers [7], even though the choice of hyperparameters is crucial. Although we could not find any significant difference between bMMI and dMMI in the development dataset experiments, we picked bMMI with $\sigma = 2$ as the best configuration since bMMI was slightly better than dMMI (the numbers of word errors differed by just 2).

Table 3 summarizes the final results obtained in the experiments. WFST-GMM in the table is a WFST-based structured classifier that uses the raw MFCCs and GMM log-likelihoods as features where

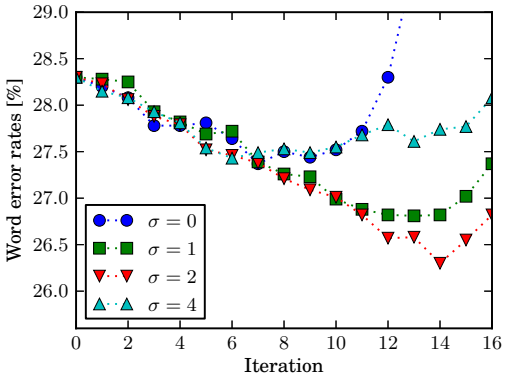

Fig. 1. WERs on the development dataset as functions of the numbers of iterations with bMMI.

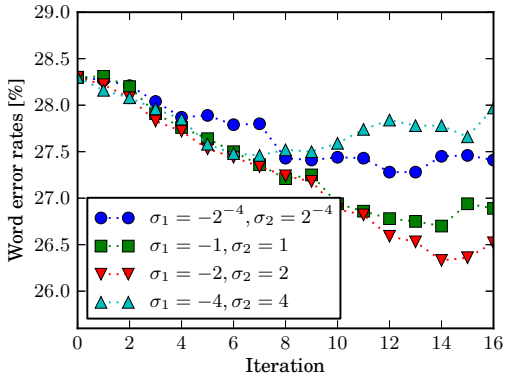

Fig. 2. WERs on the development dataset as functions of the numbers of iterations with dMMI.

WFST-DNNs use DNN-based features. These results shows that introducing WFST-based structured classification successfully reduced the WERs with both the WFST-GMMs and WFST-DNNs. However, the relative gain of WFST-DNN was bigger than that of the WFST-GMM. This might be attributed to the feature dimensionality. It seemed that the proposed method successfully leveraged the high-dimensional DNN-based features without suffering from overfitting. Finally, we reduced the WERs by 27% by introducing WFST-DNN structured classifiers into the GMM (dMMI) speech recognizers. We confirmed that the WFST-DNNs reduced the WER even compared with that of the strong DNN-BN acoustic models; an 8.0 % WER reduction was achieved by introducing structured classifiers into the DNN-BN systems.

We also measured real-time factors (RTFs) to verify the computational complexity. With the software we used, the acoustic score computation in the DNN-BN systems, and bottleneck feature $\boldsymbol{h}(.)$ computation in the WFST-DNN systems were accelerated by using graphics processing units (GPUs). The RTFs observed with this software were 0.48x in the DNN-BN systems, and 1.12x in the WFST-DNN systems. Thanks to the GPU-based acceleration, the computational time was still acceptable even if it was degraded when compared with a DNN-BN. It might be possible to accelerate WFST-DNN computation further by leveraging GPUs to compute the cost function directly, not simply the output of the hidden layers.

Table 2. Comparison of WERs obtained by varying training objective function and hyperparameters

Obj. Func.	σ	Dev. [%]	Eval. [%]
bMMI	0.0	27.4	21.8
bMMI	1.0	26.8	20.7
bMMI	2.0	**26.3**	**20.6**
bMMI	4.0	27.4	21.9
dMMI	$(-2^{-4}, 2^{-4})$	27.3	21.6
dMMI	$(-1.0, 1.0)$	26.7	20.7
dMMI	$(-2.0, 2.0)$	**26.3**	20.7
dMMI	$(-4.0, 4.0)$	27.5	21.9

Table 3. WERs on the evaluation dataset and relative error reduction rates (Rel.) from dMMI

Methods	WER [%]	Rel. [%]
GMM (ML)	32.6	-16.4
GMM (dMMI) [14]	28.2	–
WFST-GMM [7]	27.1	3.2
DNN	22.7	19.5
DNN-BN	22.4	20.6
WFST-DNN	**20.6**	27.0

5. CONCLUSION

In this paper, we proposed a method for large vocabulary continuous speech recognition (LVCSR) with structured classifier based on weighted finite-state transducer (WFST) and deep neural network (DNN) features. With the proposed method, features extracted from the bottleneck layers of DNNs are classified by parameter vectors that are independently optimized for each WFST arc. Thanks to the lexical context information represented in the WFST arcs, the proposed method yields a similar effect to that of whole-word acoustic models by only requiring few computational resources. The proposed classifier was confirmed to be effective even in large vocabulary continuous speech recognition experiments. Furthermore, the actual time required to process the inputs was not prohibitive by leveraging graphic processing units.

Future work will include an optimization of all DNN parameters. Even though it had been considered computationally prohibitive, it was recently shown that sequential DNN training can be efficiently parallelized by using Hessian-free optimization methods [20, 21]. Moreover, using more expressive speech features is also important. For example, several studies have suggested that the use of the logarithmic outputs of Mel filterbanks with a longer context window is an efficient way to find a precise DNN. Since the proposed method can also be viewed as a computationally efficient variant of whole-word HMMs, the use of longer context windows might be effective for capturing the long-term temporal dependency of acoustic observations. Even though the proposed method was developed for automatic speech recognition, its application to other domains is also promising since a WFST constitutes a common framework for several application fields such as speech summarization, speech translation, and dialogue systems.

6. REFERENCES

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

[2] M. Lehr and I. Shafran, "Learning a discriminative weighted finite state transducer for automatic speech recognition," *IEEE Trans. ASLP*, vol. 19, pp. 1360–1367, 2011.

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

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

[5] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *Proc. INTERSPEECH*, 2011.

[6] A. Mohamed, G. Dahl, and G. Hinton, "Acoustic modeling using deep belief networks," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 1, pp. 14–22, 2012.

[7] Y. Kubo, S. Watanabe, T. Hori, and A. Nakamura, "Structural classification methods based on weighted finite-state transducers for automatic speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 8, pp. 2240–2251, Oct 2012.

[8] A. Gunawardana, M. Mahajan, A. Acero, and J. C. Platt, "Hidden conditional random fields for phone classification," in *Proc. INTERSPEECH*, 2005.

[9] Y. Fujii, K. Yamamoto, and S. Nakagawa, "Deep-hidden conditional neural fields for continuous phoneme speech recognition," in *Proc. International Workshop of Statistical Machine Learning for Speech Processing (IWSML)*, 2012.

[10] Y. Wang and D. Wang, "Boosting classification based speech separation using temporal dynamics," in *Proc. INTERSPEECH*, Sep 2012.

[11] Y. Kubo, T. Hori, and A. Nakamura, "Integrating deep neural networks into structured classification approach based on weighted finite-state transducers," in *Proc. INTERSPEECH*, Sep 2012.

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

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

[14] E. McDermott, S. Watanabe, and A. Nakamura, "Discriminative training based on an integrated view of MPE and MMI in margin and error space," in *Proc. ICASSP*, 2010.

[15] J. Glass, T. J. Hazen, S. Cyphers, I. Malioutov, D. Huynh, and R. Barzilay, "Recent progress in the MIT Spoken Lecture Processing Project," in *Proc. INTERSPEECH*, 2007, pp. 2553–2556.

[16] S. Watanabe, Y. Minami, A. Nakamura, and N. Ueda, "Variational Bayesian estimation and clustering for speech recognition," *IEEE Transactions on Acoustics, Speech, and Language Processing*, vol. 12, pp. 365–381, July 2004.

[17] 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. ASLP*, vol. 15, pp. 1352–1365, 2007.

[18] R. Kneser and H. Ney, "Improved backing-off for m-gram language modeling," in *Proc. ICASSP*, 1995, vol. 1, pp. 181–184.

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

[20] O. Vinyals and D. Povey, "Krylov subspace descent for deep learning," in *Proc. AISTATS*, 2011.

[21] B. Kingsbury, T. N. Sainath, and H. Soltau, "Scalable minimum Bayes risk training of deep neural network acoustic models using distributed Hessian-free optimization," in *Proc. INTERSPEECH*, Sep 2012.