STANDALONE TRAINING OF CONTEXT-DEPENDENT DEEP NEURAL NETWORK ACOUSTIC MODELS

C. Zhang & P. C. Woodland

Cambridge University Engineering Dept, Trumpington St., Cambridge, CB2 1PZ U.K.
Email: {cz277,pcw}@eng.cam.ac.uk

ABSTRACT

Recently, context-dependent (CD) deep neural network (DNN) hidden Markov models (HMMs) have been widely used as acoustic models for speech recognition. However, the standard method to build such models requires target training labels from a system using HMMs with Gaussian mixture model output distributions (GMM-HMMs). In this paper, we introduce a method for training state-of-the-art CD-DNN-HMMs without relying on such a pre-existing system. We achieve this in two steps: build a context-independent (CI) DNN iteratively with word transcriptions, and then cluster the equivalent output distributions of the untied CD-DNN HMM states using the decision tree based state tying approach. Experiments have been performed on the Wall Street Journal corpus and the resulting system gave comparable word error rates (WER) to CD-DNNs built based on GMM-HMM alignments and state-clustering.

1. INTRODUCTION

There has been a long-term interest in recognising speech with a hybrid system that estimates HMM state emission probabilities with multi-layer perceptrons (MLPs) [1, 2]. However, it was only found recently that an MLP with a large set of context-dependent targets and many hidden layers, i.e., a context-dependent deep neural network (CD-DNN), could significantly improve recognition performance [3–5]. Although CD-DNNs have demonstrated favourable performance in various speech recognition tasks [4–9], an existing well-trained traditional GMM-HMM has to be used for two main aspects of training: state-to-frame alignments and defining a set of tied context-dependent states [3, 4].

The state-to-frame alignments serve as the training labels for CD-DNNs. Previous studies showed that shallow (single hidden layer) MLPs could be trained with iteratively refined Viterbi alignments and state occupancies generated by hybrid system itself [2, 10]. However, since high quality labels are crucial in DNN training, CD-DNNs are usually trained based on the alignments generated by a well-trained GMM-HMM system [4, 5]. Meanwhile, DNN targets are also derived from a decision-tree based tied-state GMM-HMM system [11, 12] built on the same data [3, 4]. In the decision tree approach, Gaussian distributed CD states are clustered based on the maximum likelihood (ML) criterion [11].

In this paper, we propose a method to train CD-DNNs that is independent of any existing system. The proposed method could be divided into two parts: discriminative pre-training with integrated re-alignment to first train context independent DNNs without relying on previously generated alignments; and CD-DNN decision tree target clustering, which is a modification of the standard decision tree state tying [11] based on explicitly estimating approximately equivalent terms to CD-DNN output distributions. Experiments show the proposed techniques yield comparable WER performance to CD-DNNs that rely on GMM-HMMs.

Section 2 briefly reviews CD-DNNs and GMM-HMM based decision tree state tying. The key components of the proposed method, standalone training of CI-DNNs and DNN based decision tree target clustering are described in Section 3 and Section 4, respectively. The experimental setup and results are presented in Section 5 and Section 6 which is followed by conclusions.

2. A REVIEW OF CD-DNN-HMMS

2.1. DNN-HMM Hybrid Acoustic Models

A DNN is an MLP with many hidden layers. The DNN input vector $\mathbf{x}_t$ is formed from a stacked set of adjacent frames of the acoustic feature vector, $\mathbf{o}_t$, for each frame.

In each hidden layer, the input to each unit is a weighted sum of the outputs from the previous layer [13]. A unit transforms its input with a hidden activation function, e.g., the *sigmoid function*. In the output layer, let $\mathbf{z}_t$ be the output vector of the last hidden layer, called *sigmoidal activations*, and a_k be the input to an output unit k, we have

$$a_k = \mathbf{w}_k^{\mathrm{T}}\mathbf{z}_t + b_k, \tag{1}$$

where $\mathbf{w}_k$ and b_k are the weights and bias associated with unit k. The a_k are termed *output activations*, and normalised to be the posterior probability for class $\mathcal{C}_k$ corresponding to the unit by using the *softmax* output activation function, i.e.,

$$p(\mathcal{C}_k|\mathbf{z}_t) = \frac{e^{a_k}}{\sum_{k'} e^{a_{k'}}}. \tag{2}$$

To interface a DNN with HMMs, the $p(\mathcal{C}_k|\mathbf{x}_t)$ are converted to the log-likelihood of $\mathbf{x}_t$ generated by an HMM state s_k by [4]

$$\ln p(\mathbf{x}_t|s_k) = \ln p(s_k|\mathbf{x}_t) + \ln p(\mathbf{x}_t) - \ln P(s_k), \tag{3}$$

where $P(s_k) = T_k/\sum_{k'} T_{k'}$, T_k is the number of frames associated with output s_k. $p(\mathbf{x}_t)$ is independent of the recognition result [5].

2.2. GMM-HMM based Decision Tree State Tying

It is well known that, when using CD models, the training data are usually spread unevenly for each model [11]. For a CD-DNN, despite the fact that the parameters in the hidden layers are shared,

Chao Zhang is supported by Cambridge International Scholarship from the Cambridge Commonwealth, European & International Trust and by the EPSRC Programme Grant EP/I031022/1 (Natural Speech Technology)

978-1-4799-2893-4/14/$31.00 ©2014 IEEE

those associated with some targets in the output layer may still suffer from data insufficiency. Furthermore, the softmax function needs to sum over all targets to normalise the output activations, which can be slow when very many targets are involved. As a result, for DNNs, the CD states need to be tied to form an adequate number of targets.

Decision-tree-based state tying clusters the CD states efficiently by dealing with their distributions instead of true data samples, and results in each tied-state being robustly estimated [11]. The decision tree is a binary tree built upon a set of pre-defined binary phonetic questions. At each non-leaf node, the states are classified into the node's children according to the answer to a question which is chosen to maximise the log-likelihood increase from splitting the context associated with the node. By assuming that the samples are (single) Gaussian distributed, the log-likelihood of the frames $\mathcal{O}$ generated by a node $\mathcal{S}_n$ can be approximated as [11],

$$\mathcal{L}_{\mathcal{S}_n}(\mathcal{O}) = -\frac{1}{2}\left(D\ln(2\pi) + D + \ln|\boldsymbol{\Sigma}_{\mathcal{S}_n}|\right)\sum_{s\in\mathcal{S}_n}\sum_{t}\gamma_s(t), \quad (4)$$

where D is the dimension of the data, $\gamma_s(t) = p(q_t = s|\mathcal{O}, \boldsymbol{\Lambda})$ is the ML state occupancy of being in state s at frame t. The covariance matrix $\boldsymbol{\Sigma}_{\mathcal{S}_n}$ is computed efficiently by

$$(\sigma_{ij}^{\mathcal{S}_n})^2 = \frac{\sum_{s\in\mathcal{S}_n}\theta_{ij}^{s}(\mathcal{O}^2)}{\sum_{s\in\mathcal{S}_n}\sum_t\gamma_s(t)} - \mu_i^{\mathcal{S}_n}\mu_j^{\mathcal{S}_n} \quad (5)$$

$$\mu_i^{\mathcal{S}_n} = \frac{\sum_{s\in\mathcal{S}_n}\theta_i^s(\mathcal{O})}{\sum_{s\in\mathcal{S}_n}\sum_t\gamma_s(t)}, \quad (6)$$

where $(\sigma_{ij}^{\mathcal{S}_n})^2$ are the elements of $\boldsymbol{\Sigma}_{\mathcal{S}_n}$; μ_i^s and $(\sigma_{ij}^s)^2$ are the elements of the means and the covariances of s; $\theta_i^s(\mathcal{O}) = \mu_i^s\sum_t\gamma_s(t)$ and $\theta_i^s(\mathcal{O}^2) = \left[(\sigma_{ij}^s)^2 + \mu_i^s\mu_j^s\right]\sum_t\gamma_s(t)$ are the first- and second-order statistics. During the clustering, the alignments are assumed to be fixed, which makes $\sum_t\gamma_s(t)$ constant.

3. PROPOSED TRAINING PROCEDURE FOR CI-DNNS

CD-DNN training depends on the availability of tied-state labels for all frames, which are often acquired by forced alignment with a high performance GMM-HMM system [4, 5]. To eliminate such a reliance, DNN-HMMs should be able to align the reference transcriptions themselves, and the CD states, either seen or unseen in the training set, should be tied based on DNN-HMMs rather than GMM-HMMs. In the following sections, approaches are discussed to address these issues.

3.1. Initial Alignment Refinement

To train a CI-DNN-HMM, the CI state-level transcriptions are generated from the word transcriptions. This is done by expanding every word to CI phones according to its first pronunciation in the dictionary [14], and then replacing every CI phone with its HMM states.

In order to align the CI state transcriptions without relying on an existing GMM-HMM system, an idea analogous to the *flat start* initialisation strategy used in GMM-HMM training [14] is employed, i.e., every state in an utterance is assigned an equal duration in the initial alignments. In this paper, the data are repeatedly realigned based on the word transcriptions. We call these initial uniformly segmented transcriptions *flat initial alignments*.

Since the states and frames are usually poorly aligned in the flat initial alignments, the alignments are refined by the following steps:

1. train a 3-layer (1 hidden layer) MLP with flat initial alignments for 1 epoch using error back-propagation (EBP) [1];
2. use the current MLP to realign the training set;
3. use the realignments to train a new 3-layer MLP from scratch for 1 epoch using EBP;
4. repeat step 2-4 for a number of iterations.

The above steps are similar to those used to obtain iterative Viterbi alignments in [2]. A major difference is 3-layer MLPs are trained from scratch in order to avoid the problem caused by bad initial alignments [15]. After being refined, the alignments are used for discriminative pre-training.

3.2. Discriminative Pre-training with Realignment

Instead of conventional layer-by-layer discriminative pre-training [5, 16], *discriminative pre-training with realignment* is proposed to build CI-DNNs. With this method, the data is realigned each time a new hidden layer is trained with EBP, to refine the training labels and to increase their match with the specific hidden layers. The steps are:

1. train a 3-layer MLP with the initial alignments for 1 epoch, and use the MLP to realign the data;
2. replace the current output layer with a hidden layer along with a new output layer;
3. train the modified MLP with the latest alignments for 1 epoch;
4. use the MLP to realign the reference transcriptions;
5. repeat step 2-5 until the planned DNN structure is realised.

After pre-training, all DNN layers are jointly trained by EBP to fine-tune the model parameters, which is called fine-tuning [5]. We found that realigning the data and retraining new CI-DNN-HMMs from scratch with conventional discriminative pre-training and fine-tuning could further improve the performance. After these steps, the required CI-DNN is trained.

4. CD-DNN TARGET CLUSTERING

4.1. Class-Conditional Distribution Interpretation

Since decision tree tying clusters the output probability density functions of the states, to modify the algorithm for DNN-HMMs, the equivalent class-conditional distributions from the DNN are needed.

The DNN output layer can be viewed as a single layer perceptron (SLP) that estimates the posterior probabilities with a softmax function, based on the input sigmoidal activation vector $\mathbf{z}_t$. Like the original GMM-HMM based algorithm [11], we assume the class-conditional distributions $p(\mathbf{z}_t|\mathcal{C}_k)$ to be Gaussian. If all untied Gaussian distributions have the same covariance matrix, i.e., $p(\mathbf{z}_t|\mathcal{C}_k) = \mathcal{N}(\mathbf{z}_t; \boldsymbol{\mu}_k, \boldsymbol{\Sigma})$, from Bayes' theorem [13], we have

$$p(\mathcal{C}_k|\mathbf{z}_t) = \frac{p(\mathbf{z}_t|\mathcal{C}_k)P(\mathcal{C}_k)}{\sum_{k'}p(\mathbf{z}_t|\mathcal{C}_{k'})P(\mathcal{C}_{k'})} \quad (7)$$

$$= \frac{\exp\{\boldsymbol{\mu}_k^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{z}_t - \frac{1}{2}\boldsymbol{\mu}_k^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \ln P(\mathcal{C}_k)\}}{\sum_{k'}\exp\{\boldsymbol{\mu}_{k'}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\mathbf{z}_t - \frac{1}{2}\boldsymbol{\mu}_{k'}^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_{k'} + \ln P(\mathcal{C}_{k'})\}}. \quad (8)$$

Eq. (7) has the same form as the posteriors generated by the SLP, obtained by substituting Eq. (1) into Eq. (2). Consequently, the relationship between the means and variances of the Gaussian distributions and the parameters of the SLP can be obtained as

$$\eta\,\mathbf{w}_k^{\mathrm{T}} = \boldsymbol{\mu}_k^{\mathrm{T}}\boldsymbol{\Sigma}^{-1} \quad (9)$$

$$\eta\,b_k = -\frac{1}{2}\boldsymbol{\mu}_k^{\mathrm{T}}\boldsymbol{\Sigma}^{-1}\boldsymbol{\mu}_k + \ln P(\mathcal{C}_k), \quad (10)$$

where η is any non-zero valued real number. Eq. (9) and Eq. (10) can be used to generate an SLP from known distributions. Actually, the output distributions could be more generally assumed to be any member of a particular form of the exponential family [13].

Since the output densities are estimated based on $\mathbf{z}$, the DNN-HMM based method clusters in the space of $\mathbf{z}$, $\mathbf{\Omega_z}$, while the GMM-HMM based decision tree state tying clusters in the space of the original observations $\mathbf{o}$, $\mathbf{\Omega_o}$.

4.2. DNN-HMM based Decision Tree Target Clustering

With the CI-DNN-HMM system obtained in Section 3, the CD states are clustered in the space of the sigmoidal activations $\mathbf{z}_t$ generated by the last hidden layer of the CI-DNN, $\mathbf{\Omega}_\mathbf{z}^{\text{CI}}$. The major steps of the modified method are illustrated in Fig. 1, and other parts follow standard GMM-HMM based state tying, as introduced in Section 2.2.

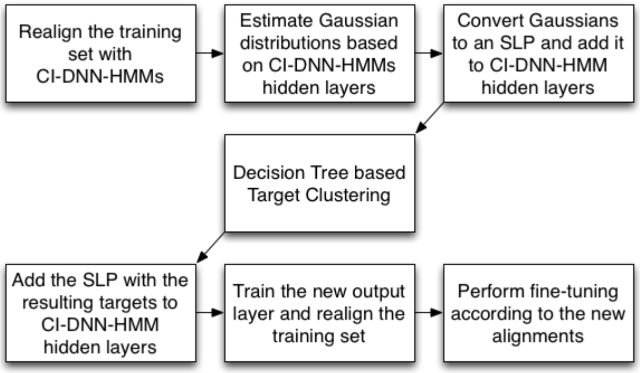
Realign the training set with CI-DNN-HMMs
Estimate Gaussian distributions based on CI-DNN-HMMs hidden layers
Convert Gaussians to an SLP and add it to CI-DNN-HMM hidden layers
Decision Tree based Target Clustering
Add the SLP with the resulting targets to CI-DNN-HMM hidden layers
Train the new output layer and realign the training set
Perform fine-tuning according to the new alignments

Fig. 1. Steps of DNN-HMM based decision tree target clustering.

4.2.1. Distribution Estimation based on Hidden Activations

To obtain the input to the decision tree clustering, the output densities for the untied states are required. The untied states together with their training labels are obtained by expanding the CI states with their surrounding phones, using the alignments generated by the CI-DNN-HMMs. Then the parameters of the distributions, i.e., the mean vectors and the common covariance matrix, are estimated based on the ML criterion.

$$\mu_{kd} = \frac{\sum_{\mathbf{z}_t \in \mathcal{Z}_k} z_{td}}{T_k} \tag{11}$$

$$(\sigma_{ij})^2 = \frac{\sum_k \sum_{\mathbf{z}_t \in \mathcal{Z}_k} (z_{ti} - \mu_{ki})(z_{tj} - \mu_{kj})}{\sum_k T_k}, \tag{12}$$

where z_{td}, μ_{kd}, and $(\sigma_{ij})^2$ are the elements of $\mathbf{z}_t$, $\boldsymbol{\mu}_k$, and $\mathbf{\Sigma}$; $\mathcal{Z}_k$ is the collection of $\mathbf{z}_t$ whose labels are $\mathcal{C}_k$.

Since $\mathbf{\Sigma}$ is usually a large full matrix, its determinant, which is used to get the log-likelihood by Eq. (4), is hard to compute. Therefore, we transform $\mathbf{\Sigma}$ to a diagonal matrix using a rotation, i.e., the orthonormal matrix $\mathbf{A}$ whose columns are the eigenvectors of $\mathbf{\Sigma}$ transforms the untied Gaussians to have a common diagonal covariance matrix by $p(\mathbf{z}|\mathcal{C}_k) = \mathcal{N}(\mathbf{A}^\text{T}\mathbf{z}; \mathbf{A}^\text{T}\boldsymbol{\mu}_k, \mathbf{A}^\text{T}\mathbf{\Sigma}\mathbf{A})$. Furthermore, to reduce the dimension of $\mathbf{A}^\text{T}\mathbf{\Sigma}\mathbf{A}$ and speed up the computation, we can discard some columns of $\mathbf{A}$ that are associated with very small eigenvalues.

4.2.2. Statistics Collection and Building CD-DNNs

After the parameters of the class-conditional distributions have been determined, they are converted into an SLP with untied state targets using Eq. (9) and Eq. (10). This SLP is then added in place of the original output layers of the CI-DNN-HMM, and used to collect the statistics $\sum_t p(\mathcal{C}_k|\mathbf{x}_t)$, which serves as the term of $\sum_t \gamma_k(t)$ used in Eq. (4)-(6). If the SLP is converted from diagonal Gaussians transformed by $\mathbf{A}$, then $\mathbf{A}^\text{T}$ with a zero bias vector should be treated as an extra layer with linear hidden activation function and interposed between the SLP and the existing hidden layers, to make the SLP take de-correlated inputs.

After clustering, the output layer with the newly clustered targets is added to the hidden layers of the CI-DNN-HMMs. The hidden layer weights are fixed and only the new output layer is trained. After this step, the resulting CD-DNN-HMMs are used to realign the training set, and fine-tuning applied according to the realignments. The resulting CD-DNN-HMMs are the required models.

If we denote $\mathbf{\Omega}_\mathbf{z}^{\text{CD}}$ as the space of the sigmoidal activations of the final CD-DNN-HMMs, the DNN-HMM based target clustering make predictions about the best targets of $\mathbf{\Omega}_\mathbf{z}^{\text{CD}}$ in $\mathbf{\Omega}_\mathbf{z}^{\text{CI}}$. In contrast, the GMM-HMM based state tying predicts CD state targets in $\mathbf{\Omega_o}$.

5. EXPERIMENTAL SETUP

The proposed techniques were evaluated by training systems on the Wall Street Journal (WSJ) training set (SI-284) and testing on the 1994 H1-dev (Dev) as well as Nov'94 H1-eval (Eval) sets, The LIMSI dictionary was used. A dictionary with 65k words, along with a backoff trigram language model, was used for all experiments. Detailed information about these can be found in [17]. The acoustic feature vector for ML trained GMM-HMM systems and all DNN-HMMs consist of 13d PLP coefficients with their Δ and $\Delta\Delta$, processed by utterance level cepstral mean normalisation (CMN) and global cepstral variance normalisation (CVN). Every HMM had 3 emitting states including the short pause model, whose states were tied to those of the silence model.

The GMM-HMM systems were trained and decoded using HTK [14, 17]. A triphone system with 1 Gaussian component per state was used for decision tree state tying, which was re-estimated to an ML trained GMM-HMM system with 5981 tied-states and 12 Gaussian components per state except for the 3 silence states that had 24 Gaussian components per state. This system was further extended to include $\Delta\Delta\Delta$ features with 39d using heteroscedastic linear discriminant analysis (HLDA) [18, 19], and discriminatively trained based on the minimum phone error (MPE) criterion [20].

The DNNs and MLPs were trained with an extension of ICSI's QuickNet software [21]. A sigmoid hidden activation function, a softmax output activation function, and the cross-entropy criterion were used. The input vector had 351 dimensions, which was produced by concatenating the current frame with 4 frames in its left and right contexts. Ten percent of the training set was selected as the held-out set for cross-validation. Parameter updates were averaged over a mini-batch with 800 frames and smoothed by adding a "momentum" term of 0.5 times the previous updates. A learning rate of 0.001 was used for pre-training. For fine-tuning, a learning rate of 0.002 was used for the first 6 epochs and 0.001 for the last 6 epochs. All DNNs had 5 hidden layers with 1000 nodes per layer.

6. EXPERIMENTAL RESULTS

6.1. Baseline System Performance

CI-DNN-HMM (I1) and CD-DNN-HMM (D1) baseline systems were built with conventional discriminative pre-training using the labels derived from the alignments generated by an HLDA MPE GMM-HMM system (G2). The triphone tied-state targets were generated by GMM-HMM based decision tree tying approach. I1 was used to realign the data and another CI-DNN-HMM baseline (I2) with the same configuration was trained from scratch based on the realignments. The baseline performance is listed in Table 1. GMM-HMMs results are also included as a comparison to previous work [17, 22].

ID	Type	DNN Alignments	WER%	
			Dev	Eval
G1	ML GMM-HMMs	—	9.1	9.5
G2	HLDA MPE GMM-HMMs		8.0	8.7
I1	CI-DNN-HMMs	G2	10.5	12.0
I2		I1	10.7	13.7
D1	CD-DNN-HMMs	G2	6.7	8.0

Table 1. Baseline system performance. The CI-DNN and CD-DNN structures are $351 \times 1000^5 \times 138$ and $351 \times 1000^5 \times 5981$.

6.2. CI-DNN-HMM Standalone Training

The flat initial alignments were first refined for 20 iterations. Afterwards, several CI-DNN-HMM systems were trained with different pre-training and conventional fine-tuning based on the alignments generated. One system (I3) was built using discriminative pre-training with realignment, and another (I4) was later built with conventional discriminative pre-training based on the alignments generated by I3. For comparison, one CI-DNN-HMM system (I5) was trained with conventional discriminative pre-training, whose realignments were used to build another set of CI-DNN-HMMs (I6) from scratch. The performance is presented in Table 2.

Comparing I3 with I5 and I4 with I6, we can see the systems with discriminative pre-training with realignment gave on average a 2.3% and 4.0% relative reduction in WER (on Dev and Eval combined). Retraining the systems from scratch for more passes caused the performance to fluctuate. As for I4 and I1, although I4 performed more poorly than I1, its results were achieved without the information from $\Delta\Delta\Delta$ features, HLDA transforms, and CD modelling embedded in the alignments. This conclusion is also be supported by system I2. I2 suffered from an 8% averaged relative WER increase compared to I1, since it excluded the above information.

ID	Training Route	WER%	
		Dev	Eval
I3	Realigned	12.2	14.3
I4	Realigned+Conventional	11.7	13.8
I5	Conventional	12.2	15.0
I6	Conventional+Conventional	12.0	14.6

Table 2. Results of CI-DNN-HMMs with different pre-trainings. "Training Route" shows the different pre-training methods for building the CI-DNN-HMM systems. "Realigned" means discriminative pre-training with realignments.

6.3. DNN-HMM based Target Clustering

The difference between GMM-HMM and DNN-HMM based decision tree state tying is now investigated. The experiment started from the best standalone CI-DNN-HMMs, I4. A total of 68,172 untied triphone states occur in the alignments generated by I4, in contrast to 68,034 untied states involved in GMM-HMM based state tying. The SLP estimated using the I4 alignments gave a 35.3% context-dependent frame classification accuracy on the combination of the training and held out sets. We de-correlated the common covariance matrix with a rotation and kept 300 dimensions (accounting for 96% of the variance) with the largest eigenvalues in the transformed diagonal covariance matrix. 5,996 tied states were generated by DNN-HMM based decision tree clustering. These clustered targets were added to the I4 hidden layers. The effect of different clusterings were examined by using EBP either through all layers or through the output layer only, based on the training labels derived from I4 alignments. The results are given in Table 3.

From the results, D2 slightly outperformed G3 (1% average relative WER reduction), which indicates that the tied-states clustered in $\boldsymbol{\Omega}_{\mathbf{z}}^{\text{CI}}$ (of I4) match the existing I4 hidden layers better than those clustered in $\boldsymbol{\Omega}_{\mathbf{o}}$. Meanwhile, if all layers were trained by EBP, D3 only outperformed G4 by 0.6% averaged relative WER. The performance difference for G4 and D3 is reduced compared to that between G3 and D2 due to the power of fine-tuning, which not only changes the output layers weights but changes their input features $\mathbf{z}_t$ as well.

Compared to the baseline CD-DNN-HMMs D1, the CD-DNN-HMMs trained in a standalone fashion, D3, performed 1.5% poorer on Dev but 2.5% better on Eval in terms of relative WER. As a result, we accomplished the task of training a state-of-the-art CD-DNN-HMM system without relying on any GMM-HMMs. In addition, the proposed training procedure is quite efficient since training and aligning the data with CI-DNN-HMMs can be much faster than with CD-DNN-HMMs.

ID	Clustering	EBP Layers	WER%	
			Dev	Eval
G3	GMM-HMM	Final Layer	7.6	9.0
G4		All Layers	6.8	7.9
D2	DNN-HMM	Final Layer	7.7	8.7
D3		All Layers	6.8	7.8

Table 3. Comparisons between GMM-HMM and CD-DNN-HMM based state tying. The hidden layers and alignments were from I4. The structures of the GMM-HMM and DNN-HMM clustered CD-DNNs are $351 \times 1000^5 \times 5981$ and $351 \times 1000^5 \times 5996$, separately.

7. CONCLUSIONS

A new CD-DNN training procedure has been presented, which unlike the standard approach to DNN training does not rely on a pre-existing speech recognition system. A CI-DNN is trained in an interleaved fashion by updating the model parameters with the reference labels and updating the labels by realigning the training set. Afterwards, a Gaussian distribution with a common covariance matrix is estimated for every untied CD state based on the hidden activation vectors generated by the last hidden layer of the CI-DNN, which are clustered by decision tree state tying. These are the converted to the output layer of a CD-DNN. Experiments on the standard SI-284 training setup for the Wall Street Journal corpus have shown that the proposed training procedure gives state-of-the-art performance.

8. REFERENCES

[1] D. E. Rumelhart, J. L. McClelland, and the PDP Research Group, *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations*, MIT Press, Cambridge, MA, USA, Jul. 1986.

[2] H. A. Bourlard and N. Morgan, *Connectionist Speech Recognition: A Hybrid Approach*, Kluwer Academic Publishers, Norwell, MA, USA, 1993.

[3] F. Seide, G. Li, and D. Yu, "Conversational speech transcription using context-dependent deep neural networks," in *Proc. Interspeech'11*, Florence, Italy, Sep. 2011, pp. 437–440.

[4] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, Jan. 2012.

[5] G. E. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kinsbury, "Deep neural networks for acoustic modeling in speech recognition," *IEEE Signal Processing Magazine*, pp. 2–17, Nov. 2012.

[6] M. L. Seltzer, D. Yu, and Y.-Q. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Proc. ICASSP'13*, Vancouver, Canada, 2013, pp. 7398–7402.

[7] S. Thomas, M. L. Seltzer, K. Church, and H. Hermansky, "Deep neural network features and semi-supervised training for low resource speech recognition," in *Proc. ICASSP'13*, Vancouver, Canada, 2013, pp. 6704–6708.

[8] P. Bell, P. Swietojanski, and S. Renals, "Multi-level adaptive networks in tandem and hybrid ASR systems," in *Proc. ICASSP'13*, Vancouver, Canada, 2013, pp. 7947–7951.

[9] K. M. Knill, M. J. F. Gales, S. P. Rath, P. C. Woodland, C. Zhang, and S.-X. Zhang, "Investigation of multilingual deep neural networks for spoken term detection," in *Proc. ASRU'13*, Olomouc, Czech Republic, 2013, pp. 138–143.

[10] Y.-H. Yan, M. Fanty, and R. Cole, "Speech recognition using neural networks with forward-backward probability generated targets," in *Proc. ICASSP'97*, Munich, Germany, 1997, pp. 3241–3244.

[11] S. J. Young, J. J. Odell, and P. C. Woodland, "Tree-based state tying for high accuracy acoustic modelling," in *Proc. Human Language Technology Workshop*, Plainsboro, NJ, USA, 1994, pp. 307–312.

[12] P. C. Woodland, J. J. Odell, V. Valtchev, and S. J. Young, "Large vocabulary continuous speech recognition using HTK," in *Proc. ICASSP'94*, Adelaide, Australia, 1994, vol. 2, pp. 125–128.

[13] C. M. Bishop, *Neural Networks for Pattern Recognition*, Oxford University Press, Oxford, UK, Nov. 1995.

[14] S. J. Young, G. Evermann, M. J. F. Gales, T. Hain., D. Kershaw, X.-Y. Liu, G. Moore, J. J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. C. Woodland, *The HTK book (for HTK version 3.4)*, Cambridge University Engineering Department, Cambridge, UK, 2006.

[15] E. Trentin and M. Gori, "Robust combination of neural networks and hidden Markov models for speech recognition," *IEEE Transactions on Neural Networks*, vol. 14, no. 6, pp. 1519–1531, Nov. 2003.

[16] F. Seide, G. Li, X. Chen, and Y. Dong, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *Proc. ASRU'11*, Waikoloa, HI, USA, 2011, pp. 24–29.

[17] P. C. Woodland, C. J. Leggetter, J. J. Odell, V. Valtchev, and S. J. Young, "The 1994 HTK large vocabulary speech recognition system," in *Proc. ICASSP'95*, Detroit, MI, USA, 1995, vol. 1, pp. 73–76.

[18] N. Kumar, *Investigation of silicon-auditory models and generalization of linear discriminant analysis for improved speech recognition*, Ph.D. thesis, John Hopkins University, Baltimore, MD, USA, 1997.

[19] X.-Y. Liu, M. J. F. Gales, and P. C. Woodland, "Automatic complexity control for HLDA systems," in *Proc. ICASSP'03*, Hong Kong, Hong Kong, Apr. 2003, vol. 1, pp. 132–135.

[20] D. Povey and P. C. Woodland, "Minimum phone error and I-smoothing for improved discriminative training," in *Proc. ICASSP'02*, Orlando, FL, USA, 2002, vol. 1, pp. 105–108.

[21] D. Johnson, "QuickNet," `www1.icsi.berkeley.edu/speech/qn.html`.

[22] D. Povey, *Discriminative Training for Large Vocabulary Speech Recognition*, Ph.D. thesis, Cambridge University Engineering Department, Cambridge, UK, 2003.