A STUDY OF AN IRRELEVANT VARIABILITY NORMALIZATION BASED DISCRIMINATIVE TRAINING APPROACH FOR LVCSR

Yu Zhang[1,2], *Jian Xu*[1,3], *Zhi-Jie Yan*[1], *Qiang Huo*[1]

[1]Microsoft Research Asia, Beijing, China
[2]MOE-Microsoft Key Laboratory for Intelligent Computing and Intelligent Systems
[2]Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China
[3] University of Science and Technology of China, Hefei, China
`sjtuzy@gmail.com, v-jiaxu@microsoft.com, zhijiey@microsoft.com, qianghuo@microsoft.com`

ABSTRACT

This paper presents a discriminative training (DT) approach to irrelevant variability normalization (IVN) based training of feature transforms and hidden Markov models for large vocabulary continuous speech recognition. A speaker-clustering based method is used for acoustic sniffing and maximum mutual information (MMI) is used as a training criterion. Combined with unsupervised adaptation of feature transforms, the IVN-based DT approach achieves a 14.5% relative word error rate reduction over an MMI-trained baseline system on a Switchboard-1 conversational telephone speech transcription task.

Index Terms— irrelevant variability normalization, discriminative training, unsupervised adaptation, LVCSR, acoustic modeling

1. INTRODUCTION

In [7], a maximum likelihood (ML) version of a so-called irrelevant variability normalization (IVN) based approach to large vocabulary continuous speech recognition (LVCSR) was studied and promising results were reported on a Switchboard-1 conversational telephone speech transcription task [2]. In this paper, we present a follow-up study of a discriminative training (DT) version of the IVN-based approach to LVCSR, where maximum mutual information (MMI) criterion (e.g. [6]) is used for DT. Fig. 1 illustrates how an IVN-based framework works for acoustic modeling, training and adaptation. In off-line training stage (upper part of the figure), one can train from a large amount of diversified training data, by using an IVN-based training procedure (ML or DT), a set of generic hidden Markov models (HMMs) good at discriminating different phonetic classes and a set of auxiliary transforms used to "absorb" factors irrelevant to phonetic classification. In recognition stage (lower part of the figure), given the sequence of feature vectors extracted from an unknown speech segment, an "*acoustic sniffing*" module will decide which transform to use for each feature vector to remove irrelevant information. The sequence of transformed feature vectors is then decoded by using a traditional LVCSR decoder with three knowledge sources, namely generic HMMs, a pronunciation lexicon, and a language model. After the first-pass recognition, the set of feature transforms is adapted under an ML criterion by using the previous recognition result and unknown speech segment itself, which is recognized again to achieve better accuracy by using the adapted feature transforms and the pre-trained generic HMMs. Additional adaptation and recognition passes may be performed subsequently until a predetermined criterion is met, such as a prescribed number of passes. While a so-called moving-window based frame labeling method was used for acoustic sniffing in [7], better results are achieved by using a speaker-clustering based method as reported in this paper.

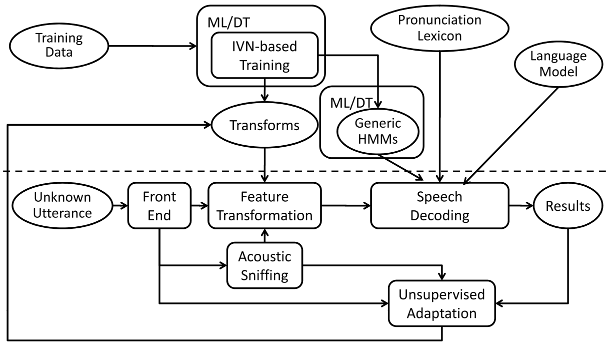

Fig. 1. *An illustration of IVN-based framework for acoustic modeling, training and adaptation.*

The rest of the paper is organized as follows. In Section 2, we present the IVN-based discriminative training approach used in our experiments. In Section 3, we report experimental results. Finally, we conclude the paper in Section 4.

2. APPROACH

2.1. Feature Transformation Function

As in [7], the following feature transformation (FT) function is used:

$$\boldsymbol{x}_t = \mathcal{F}(\boldsymbol{y}_t; \boldsymbol{\Theta}) = \boldsymbol{A}^{(e_t)}\boldsymbol{y}_t + \boldsymbol{b}^{(l_t)} \qquad (1)$$

where $\boldsymbol{y}_t$ is the t-th D-dimensional feature vector of the input feature vector sequence extracted by the "Front-End" module; $\boldsymbol{x}_t$ is the transformed feature vector; e_t and l_t are the labels (transform indices) informed by the "Acoustic Sniffing" module for the $D \times D$ nonsingular transformation matrix $\boldsymbol{A}^{(e_t)}$ and D-dimensional bias vector $\boldsymbol{b}^{(l_t)}$, respectively; and $\boldsymbol{\Theta} = \{\boldsymbol{A}^{(e)}, \boldsymbol{b}^{(l)} | e = 1, 2, \cdots, E; l = 1, 2, \cdots, L\}$ denotes the set of feature transformation parameters with E and L being the total number of tied transformation matrices and bias vectors, respectively. For the convenience of notation, we

This work was done while Yu Zhang and Jian Xu were interns in Microsoft Research Asia, Beijing, China.

also use hereinafter $\mathcal{F}(\boldsymbol{Y};\boldsymbol{\Theta})$ to denote the transformed version of a speech segment $\boldsymbol{Y}$ by transforming individual feature vector $\boldsymbol{y}_t$ of $\boldsymbol{Y}$ as defined in Eq. (1).

2.2. Speaker-Clustering based Approach to Acoustic Sniffing

In this study, a speaker-clustering based approach is used for acoustic sniffing. In training stage, given the feature vectors from each training speaker, the following procedure is used for speaker clustering:

Step 1: *Initialization*

Two Gaussian mixture models (GMMs) are trained first by using training data from male and female speakers, respectively. Each GMM represents a speaker cluster and has 1,024 Gaussian components in our experiments.

Step 2: *Speaker classification and GMM re-estimation*

Given the current set of GMMs, classify each speaker into the speaker cluster, which gives the highest likelihood of the training data from the speaker against the corresponding GMM. Given the new speaker clustering result, re-estimate GMM for each speaker cluster. Repeat the above two actions for several times.

Step 3: *Splitting of speaker clusters*

If a pre-determined number of speaker clusters is reached, stop; Otherwise, split each speaker cluster into two new clusters by perturbations of the mean vectors of the corresponding GMM, and go back to **Step 2**.

Given the above speaker clustering result, a simple acoustic sniffing scheme can work as follows:

- In IVN training, the labels for e_t and l_t are assigned as the speaker cluster label. By doing so, all the feature vectors from the same speaker cluster will share a single feature transform. The total number of feature transforms equals the number of speaker clusters.
- In recognition stage, given the chunk of data from an unknown speaker, speaker classification is performed first. The pre-trained feature transform from the corresponding speaker cluster is then used for feature transformation.

Although the above simple acoustic sniffing scheme was used in experiments reported here, other more flexible schemes are apparently possible. That explains why we give the most general formulations in both feature transformation function and the IVN-based training procedure to be described in the next subsection, where e_t and l_t can be assigned flexibly by an appropriate acoustic sniffing method.

2.3. IVN-based Discriminative Training

Let's assume that each basic speech unit in our speech recognizer is modeled by a Gaussian mixture continuous density HMM (CDHMM), whose parameters are denoted as $\lambda = \{\pi_s, a_{ss'}, c_{sm}, \boldsymbol{\mu}_{sm}, \boldsymbol{\Sigma}_{sm}; s, s' = 1, \cdots, S; m = 1, \cdots, M\}$, where S is the number of states, M is the number of Gaussian components for each state, $\{\pi_s\}$ is the initial state distribution, $a_{ss'}$'s are state transition probabilities, c_{sm}'s are Gaussian mixture weights, $\boldsymbol{\mu}_{sm} = [\mu_{sm1}, \cdots, \mu_{smD}]^T$ is a D-dimensional mean vector, and $\boldsymbol{\Sigma}_{sm} = \mathrm{diag}\{\sigma^2_{sm1}, \cdots, \sigma^2_{smD}\}$ is a $D \times D$ diagonal covariance matrix. Let $\boldsymbol{\Lambda} = \{\lambda\}$ denote the set of CDHMM parameters and $\mathcal{Y} = \{\boldsymbol{Y}_i | i = 1, 2, \cdots I\}$ the set of training data, where $\boldsymbol{Y}_i = (\boldsymbol{y}_1^{(i)}, \boldsymbol{y}_2^{(i)}, \cdots, \boldsymbol{y}_{T_i}^{(i)})$ is a sequence of D-dimensional feature vectors extracted from the i-th utterance. By using "acoustic sniffing", two sets of frame labels for transformation matrix and bias vector, $\mathcal{E}$ and $\mathcal{L}$, can be derived from $\mathcal{Y}$, respectively. So IVN-based training is to optimize, by adjusting feature transformation parameters $\boldsymbol{\Theta}$ and HMM parameters $\boldsymbol{\Lambda}$, a given discriminative training criterion. When MMI criterion is used, it is to maximize the following objective function:

$$\begin{aligned}\mathcal{F}_{\mathrm{MMI}}(\boldsymbol{\Theta}, \boldsymbol{\Lambda}) &= \sum_{i=1}^{I} \mathcal{F}_{\mathrm{MMI}}(\boldsymbol{\Theta}, \boldsymbol{\Lambda}; \boldsymbol{Y}_i, \mathcal{M}_i, \mathcal{E}, \mathcal{L}) \\ &= \sum_{i=1}^{I} \log \frac{p(\boldsymbol{Y}_i | \boldsymbol{\Theta}, \boldsymbol{\Lambda}; \mathcal{M}_i^+, \mathcal{E}, \mathcal{L})}{p(\boldsymbol{Y}_i | \boldsymbol{\Theta}, \boldsymbol{\Lambda}; \mathcal{M}_i^-, \mathcal{E}, \mathcal{L})} \qquad (2)\end{aligned}$$

where $\mathcal{M}_i^+$ and $\mathcal{M}_i^-$ stand for the reference model space and competing model space of $\boldsymbol{Y}_i$, respectively. Similar to the ML version of IVN-based training, the following *method of alternating variables* is used to maximize MMI objective function:

Step 1: *Initialization*

Feature transform and HMM parameters are initialized as the ones trained by using IVN-based ML approach in [7].

Step 2: *Estimate feature transformation parameters $\boldsymbol{\Theta}$ by fixing HMM parameters $\boldsymbol{\Lambda}$*

Given the fixed HMM parameters $\overline{\boldsymbol{\Lambda}}$, the MMI objective function $\mathcal{F}_{\mathrm{MMI}}(\boldsymbol{\Theta}, \overline{\boldsymbol{\Lambda}})$ can be optimized by increasing an auxiliary function iteratively as described in [3]. In IVN-based discriminative training, such an auxiliary function is as follows:

$$\mathcal{Q}(\boldsymbol{\Theta}, \overline{\boldsymbol{\Theta}}) = \mathcal{A}(\boldsymbol{\Theta}, \overline{\boldsymbol{\Theta}}) + \mathcal{A}^{\mathrm{sm}}(\boldsymbol{\Theta}, \overline{\boldsymbol{\Theta}}) \qquad (3)$$

where

$$\mathcal{A}(\boldsymbol{\Theta}, \overline{\boldsymbol{\Theta}}) = \sum_{\substack{s,m,l,e \\ \boldsymbol{y}_t \in \mathcal{L}_l \cap \mathcal{E}_e}} \left(\gamma_{sm}^+(t) - \gamma_{sm}^-(t)\right) \log p_{sm}(\boldsymbol{y}_t | \boldsymbol{\Theta}, \overline{\boldsymbol{\Lambda}})$$

$$p_{sm}(\boldsymbol{y}_t | \boldsymbol{\Theta}, \overline{\boldsymbol{\Lambda}}) = \mathcal{N}(\mathcal{F}(\boldsymbol{y}_t; \boldsymbol{\Theta}); \overline{\boldsymbol{\mu}}_{sm}, \overline{\boldsymbol{\Sigma}}_{sm}) |\det(\boldsymbol{A}^{(e_t)})| \;, \qquad (4)$$

$\mathcal{E}_e, \mathcal{L}_l$ is the set of training feature vectors with a transformation label e and a bias label l respectively, $\gamma_{sm}^+(t)$ and $\gamma_{sm}^-(t)$ denote occupancy statistics of Gaussian component m in state s of the observed feature vector $\boldsymbol{y}_t$, and

$$\mathcal{A}^{\mathrm{sm}}(\boldsymbol{\Theta}, \overline{\boldsymbol{\Theta}}) = \sum_{s,m,l,e} D_{sm}^{e,l} \int_{\boldsymbol{y}} p_{sm}(\boldsymbol{y} | \overline{\boldsymbol{\Theta}}, \overline{\boldsymbol{\Lambda}}) \log p_{sm}(\boldsymbol{y} | \boldsymbol{\Theta}, \boldsymbol{\Lambda}) \mathrm{d}\boldsymbol{y}$$

is a smoothing function to ensure that the $\mathcal{Q}$-function is concave. It is easy to verify that the $\mathcal{Q}$-function in Eq. (3) is a "week-sense" auxiliary function [6] for the MMI objective function, which can be maximized again using the *method of alternating variables*. The overall training procedure in **Step 2** is outlined as follows:

Step 2.1: Calculate $\gamma_{sm}^{(+/-)}(t)$ and accumulate relevant sufficient statistics.

Step 2.2: Increase $\mathcal{Q}$-function by the *method of alternating variables*:

Step 2.2.1: Estimate $\{\boldsymbol{A}^{(e)}\}$ by fixing $\{\boldsymbol{b}^{(l)}\}$

The derivation of the updating formula for $\boldsymbol{A}^{(e)}$ is similar to that in CMLLR [1]. By differentiating $\mathcal{Q}$-function w.r.t. the d-th row of $\boldsymbol{A}^{(e)}$ (hereinafter denoted as $\boldsymbol{A}_d^{(e)}$) and equating to zero, the following updating formula can be derived:

$$\boldsymbol{A}_d^{(e)} = \alpha_d^{(e)} \boldsymbol{c}_d^{(e)} \boldsymbol{F}_d^{(e)-1} + \boldsymbol{j}_d^{(e)} \boldsymbol{F}_d^{(e)-1} \qquad (5)$$

where $\boldsymbol{c}_d^{(e)}$ is the cofactor row vector $[c_{d1}^{(e)} \dots c_{dD}^{(e)}]$ with $c_{dj}^{(e)} = \mathrm{cof}(\boldsymbol{A}_{dj}^{(e)})$, and

$$\begin{aligned}
\boldsymbol{F}_d^{(e)-1} &= \sum_{s,m=1} \frac{1}{\sigma_{smd}^2} [\boldsymbol{G}_{sme} + \sum_l D_{sm}^{e,l} \boldsymbol{C}_{sml}] \\
\boldsymbol{j}_d^{(e)} &= \sum_{s,m=1} \Big[\sum_{\boldsymbol{y}_t \in \mathcal{E}_e} (\gamma_{sm}^{+}(t) - \gamma_{sm}^{-}(t)) \frac{(\boldsymbol{\mu}_{smd} - \boldsymbol{b}_d^{(l_t)})}{\sigma_{smd}^2} \boldsymbol{y}_t^\top \\
&\quad + \sum_l D_{sm}^{e,l} \frac{(\boldsymbol{\mu}_{smd} - \overline{\boldsymbol{b}}_d^{(l)})(\boldsymbol{\mu}_{sm} - \overline{\boldsymbol{b}}^{(l)})^\top}{\sigma_{smd}^2} \overline{\boldsymbol{A}}^{(e)-1\top} \Big] \\
\boldsymbol{G}_{sme} &= \sum_{\boldsymbol{y}_t \in \mathcal{E}_e} (\gamma_{sm}^{+}(t) - \gamma_{sm}^{-}(t)) \boldsymbol{y}_t \boldsymbol{y}_t^\top \\
\boldsymbol{C}_{sml} &= \overline{\boldsymbol{A}}^{(e)-1} [\boldsymbol{\Sigma}_{sm} + (\boldsymbol{\mu}_{sm} - \overline{\boldsymbol{b}}^{(l)})(\boldsymbol{\mu}_{sm} - \overline{\boldsymbol{b}}^{(l)})^\top] \overline{\boldsymbol{A}}^{(e)-1\top} \\
\alpha_d^{(e)} &= \frac{-\epsilon_2^{(e)} \pm \sqrt{(\epsilon_2^{(e)})^2 + 4\epsilon_1^{(e)} \beta^{(e)}}}{2\epsilon_1^{(e)}} \\
\epsilon_1^{(e)} &= \boldsymbol{c}_d^{(e)} \boldsymbol{F}_d^{(e)-1\top} \boldsymbol{c}_d^{(e)\top} \\
\epsilon_2^{(e)} &= \boldsymbol{c}_d^{(e)} \boldsymbol{F}_d^{(e)-1\top} \boldsymbol{j}_d^{(e)\top} \\
\beta^{(e)} &= \sum_{s,m} \sum_{\boldsymbol{y}_t \in \mathcal{E}_e} (\gamma_{sm}^{+}(t) - \gamma_{sm}^{-}(t)) + \sum_{s,m} \sum_l D_{sm}^{e,l} \,. \qquad (6)
\end{aligned}$$

The value of $\alpha_d^{(e)}$ is selected as what maximizes

$$\mathcal{Q}_e = \beta^{(e)} \log |\alpha_d^{(e)} \epsilon_1^{(e)} + \epsilon_2^{(e)}| - \frac{1}{2} \alpha_d^{(e)2} \epsilon_1^{(e)} \,. \qquad (7)$$

It is easy to verify that $\mathcal{Q}(\boldsymbol{\Theta}, \overline{\boldsymbol{\Theta}})$ is concave when $\beta^{(e)} > 0$ and $\boldsymbol{F}_d^{(e)}$ is positive definite. However, the lower bound of $D_{sm}^{e,l}$ from [6] could not be borrowed directly because $\boldsymbol{F}_d^{(e)}$ is a full matrix unlike the diagonal case in extended Baum-Welch (EBW) algorithm for updating HMM parameters. In our case, it can be proven that the $\mathcal{Q}$-function is concave when $D_{sm}^{e,l}$ satisfies the following constraint:

$$D_{sm}^{e,l} = \mathrm{EConst} * \max\{D_{\min}^{e}, \sum_{\boldsymbol{y}_t \in \mathcal{E}_e \cap \mathcal{L}_l} |\gamma_{sm}^{+}(t) - \gamma_{sm}^{-}(t)| + \frac{1}{\beta}\}$$

where $\mathrm{EConst} > 1$, $\frac{1}{\beta} > 0$, and

$$D_{\min}^{e} = \max_i \left| \frac{\det(\boldsymbol{G}_{sme}^{(ii)})}{\det([\sum_l \boldsymbol{C}_{sml}]^{(ii)})} \right| , \qquad (8)$$

$\boldsymbol{G}_{sme}^{(ii)}$, $[\sum_l \boldsymbol{C}_{sml}]^{(ii)}$ is the ith leading principal minors of $\boldsymbol{G}_{sme}$ and $\sum_l \boldsymbol{C}_{sml}$. In our experiments, we set $\mathrm{EConst} = 2$ and $\beta = 0.2$.

As in [1], $\boldsymbol{A}^{(e)}$ can be updated by using the above row-by-row updating formula. Let's use N_a to denote the number of iterations performed.

Step 2.2.2: Estimate $\{\boldsymbol{b}^{(l)}\}$ by fixing $\{\boldsymbol{A}^{(e)}\}$

By differentiating the $\mathcal{Q}$-function w.r.t. $\boldsymbol{b}^{(l)}$ and equating to zero, each $\boldsymbol{b}^{(l)}$ can be updated as follows:

$$\boldsymbol{b}_d^{(l)} = \frac{\Big[\sum\limits_{\substack{\boldsymbol{y}_t \in \mathcal{L}_l \\ s,m}} \frac{\gamma_{sm}^{+}(t) - \gamma_{sm}^{-}(t)}{\sigma_{smd}^2} (\boldsymbol{\mu}_{sm} - \boldsymbol{A}_d^{(e_t)} \boldsymbol{y}_t) + \sum\limits_{s,m,e} \frac{D_{m}^{e,l}}{\sigma_{smd}^2} \overline{\boldsymbol{b}}_d^{(l)} \Big]}{\Big[\sum_{s,m} \frac{\sum_e D_{sm}^{e,l} + \sum_{\boldsymbol{y}_t \in \mathcal{L}_l} (\gamma_{sm}^{+}(t) - \gamma_{sm}^{-}(t))}{\sigma_{smd}^2} \Big]}$$

where $\boldsymbol{b}_d^{(l)}$ is the d-th element of the bias vector $\boldsymbol{b}^{(l)}$, $\boldsymbol{A}_d^{(e_t)}$ is the d-th row of the *updated* matrix $\boldsymbol{A}^{(e_t)}$ obtained in **Step 2.2.1**.

Step 2.2.3: Repeat **Step 2.2.1** to **Step 2.2.2** N_{ab} times, and update $\boldsymbol{\Theta}$.

Step 2.3: Repeat **Step 2.1** to **Step 2.2** N_T times.

Step 3: *Estimate HMM parameters* $\boldsymbol{\Lambda}$ *by fixing feature transformation parameters* $\boldsymbol{\Theta}$

Given the *updated* transform parameters $\overline{\boldsymbol{\Theta}}$ obtained in **Step 2**, we first transform each training feature vector y_t by using the feature transformation $\mathcal{F}(\boldsymbol{y}_t; \overline{\boldsymbol{\Theta}})$. Then, N_h EBW iterations (e.g., [6]) are performed to re-estimate HMM parameters $\boldsymbol{\Lambda}$, which optimizes the MMI objective function $\mathcal{F}_{\mathrm{MMI}}(\overline{\boldsymbol{\Theta}}, \boldsymbol{\Lambda})$.

Step 4: *Repeat* ***Step 2*** *and* ***Step 3*** *N_c times*

This concludes the description of our IVN-based DT procedure.

2.4. Unsupervised Adaptation

To improve recognition accuracy, for each unknown speech segment, unsupervised adaptation can be performed as follows:

Step 1: Given an unknown speech segment $\boldsymbol{Y}$, do acoustic sniffing to identify the labels of feature transform for each feature vector. Transform $\boldsymbol{Y}$ as $\mathcal{F}(\boldsymbol{Y}; \boldsymbol{\Theta})$ with *pre-trained* transform parameters $\boldsymbol{\Theta}$. Do first-pass recognition by using generic HMMs.

Step 2: Given the recognized transcription, the transform parameters $\boldsymbol{\Theta}$ are re-estimated by using the IVN-based ML training procedure described in [7].

Step 3: Transform Y with the *updated* parameters $\boldsymbol{\Theta}$. Do recognition by using generic HMMs.

Step 4: Repeat **Step 2** and **Step 3** until a pre-specified criterion is satisfied (e.g., a fixed number of cycles).

3. EXPERIMENTS AND RESULTS

3.1. Experimental Setup

Switchboard-1 corpus [2] was used for our experiments. We used 4,870 sides of conversations (about 300 hours speech) from 520 speakers as training data, and 40 sides of Switchboard-1 conversations (about 2 hours speech) from the 2000 Hub5 evaluation as testing data. The minimum, maximum and average lengths of conversation sides are 4.84s, 547.16s, and 229.61s in the training set, and 73.12s, 279.77s, and 184.47s in the testing set, respectively.

For feature extraction in front-end, we used 39 PLP_E_D_A (in HTK's terminology [8]) features. Conversation-side based mean and variance normalization was applied in both training and recognition stages. For acoustic modeling, we used phonetic decision-tree based tied-state triphone CDHMMs with 9,302 states and 40 Gaussian components per state. Our recognition vocabulary contains 22,641 unique words. The pronunciation lexicon contains multiple pronunciations per word with a total of 28,649 unique pronunciations. A trigram language model trained on the transcription data of the Switchboard-1 acoustic training data and broadcast news data was used for recognition. All of the recognition experiments were performed with a Microsoft in-house decoder instead of using the HDecode engine of HTK3.4 toolkit [8] as in [7] because the former achieves a slightly better recognition accuracy. All the recognition results were calculated by using a NIST Scoring Toolkit SCTK [5].

For IVN-based MMI training, the feature transforms were estimated as described in Section 2, while the HMM parameters were optimized by using the conventional EBW algorithm (e.g., [6]). The relevant control parameters were set to the typical values as suggested by HTK [8], e.g., the learning constant $EConst = 2$, $\tau = 100$ for i-smoothing, acoustic scaling factor $\kappa = 1/11.25$. In speaker-clustering based "acoustic sniffing", 8 speaker clusters were trained, therefore $E = L = 8$. In all IVN-based discriminative training and adaptation experiments, the relevant control parameters are set as follows: $N_a = 10$, $N_{ab} = N_T = N_h = 1$.

Without IVN-based training, our ML-trained and MMI-trained baseline systems achieve a word error rate (WER) of 30.0% and 26.2% respectively.

3.2. Effects of IVN-based Discriminative Training

Starting from the ML-trained baseline system, we perform IVN-based ML training using speaker-clustering based acoustic sniffing. The WER is reduced from 30.0% for baseline system to 27.8% after 20 main cyles of IVN-based training, which is better than the WER, 28.5%, of the IVN-based ML-trained system if the moving-window based acoustic sniffing method as described in [7] is used. Therefore, the speaker-clustering based acoustic sniffing method is used in the remaining IVN experiments.

Starting from the above IVN-based ML-trained system, the following three sets of experiments are performed:

- perform 10 cycles of MMI training for feature transforms only;
- perform 5 EBW iterations of MMI training for HMMs only;
- perform 5 main cycles of MMI training for both feature transforms and HMMs.

The WERs of the above three systems are 27.0%, 25.0%, and 24.6% respectively. Compared with the respective baseline systems, the power of IVN-based discriminative training is clearly demonstrated.

3.3. Effects of Unsupervised Adaptation

Starting from the above four IVN-trained systems, we conduct conversation-side based unsupervised adaption as described in Section 2.4. After two cycles of recognition and adaptation, the WERs are reduced to 25.5%, 25.1%, 22.7%, and 22.4%, respectively.

For comparison, starting from the ML- and MMI-trained baseline systems, we perform conversation-side based unsupervised HMM adaptation using MLLR approach [4]. Eight regression classes are used and 3 EM iterations are performed to estimate the linear transforms. After two cycles of recognition and adaptation, the WERs are reduced to 28.4% and 24.8% respectively.

All the above results are summarized in Table 1 for easy comparison. Apparently IVN-based DT approach achieves the best performance. Compared with the "HMM-MMI" baseline system, the "FT-MMI+HMM-MMI" approach achieves relative WER reductions of 6.1% and 14.5% for without and with unsupervised adaptation respectively. Compared with the "HMM-MMI+MLLR adaptation" system, the "FT-MMI+HMM-MMI" approach with unsupervised adaptation achieves a relative WER reduction of 9.7%. Given the simplicity of the "FT-ML+HMM-MMI" approach and its promising results, it can be a good choice in practice.

4. CONCLUSION AND DISCUSSIONS

In this paper, we have investigated and confirmed the effectiveness of an IVN-based framework for acoustic modeling by using discriminative training to estimate feature transforms and/or generic HMMs for LVCSR. A new acoustic sniffing technique based on speaker clustering is studied and confirmed to perform better than a previous approach. Promising results are achieved on the difficult Switchboard-1 conversational telephone speech transcription task. Ongoing and future works for IVN-based framework include:

- to explore different acoustic sniffing techniques and identify the most effective ways for different LVCSR application scenarios and deployment requirements;
- to investigate other DT criteria and more effective optimization methods for IVN-based discriminative training;
- to investigate appropriate adaptation methods for application scenarios where only a short speech utterance is available for adaptation;
- to verify the effectiveness of the IVN-based framework for even larger scale LVCSR applications.

We will report those results elsewhere once they become available.

Table 1. *Comparison of different approaches (FT: feature transform, UA: unsupervised adaptation)*

Method			w/o UA		UA	
#	FT	HMM	WER(%)	Rel.(%)	WER(%)	Rel.(%)
1	-	ML	30.0	N/A	28.4	N/A
2	-	MMI	26.2	12.7	24.8	12.7
3	ML	ML	27.8	7.3	25.5	10.2
4	MMI	ML	27.0	10.0	25.1	11.6
5	ML	MMI	25.0	16.7	22.7	20.1
6	MMI	MMI	24.6	18.0	22.4	21.1

5. REFERENCES

[1] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, Vol. 12, pp.75-98, 1998.

[2] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "SWITCHBOARD: Telephone speech corpus for research and development," *Proc. ICASSP-1992*, pp.517-520. See also LDC website: http://www.ldc.upenn.edu for more details.

[3] A. Gunawardana, "Maximum Mutual Information Estimation of Acoustic HMM Emission Densities," *CLSP Research Note*, Note No. 40, CLSP, Johns Hopkins University, 2000.

[4] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models," *Computer Speech and Language*, Vol. 9, pp.171-185, 1995.

[5] NIST Scoring Toolkit SCTK, see the following site for details: http://itl.nist.gov/iad/mig/tests/rt/2002/software.htm.

[6] D. Povey, "Discriminative Training for Large Vocabulary Speech Recognition," *Ph.D. thesis*, Cambridge University, 2004.

[7] G.-C. Shi, Y. Shi, and Q. Huo, "A study of irrelevant variability normalization based training and unsupervised online adaptation for LVCSR," *Proc. Interspeech-2010*, pp.1357-1360.

[8] S. Young, *et al.*, The HTK Book (for HTK version 3.4), 2006.