A STUDY OF DISCRIMINATIVE FEATURE EXTRACTION FOR I-VECTOR BASED ACOUSTIC SNIFFING IN IVN ACOUSTIC MODEL TRAINING

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

[1]Microsoft Research Asia, Beijing, China
[2]Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China
[3]Department of Automation, University of Science and Technology of China, Hefei, China
sjtuzy@gmail.com, v-jiaxu@microsoft.com, zhijiey@microsoft.com, qianghuo@microsoft.com

ABSTRACT

Recently, we proposed an i-vector approach to acoustic sniffing for irrelevant variability normalization based acoustic model training in large vocabulary continuous speech recognition (LVCSR). Its effectiveness has been confirmed by experimental results on Switchboard-1 conversational telephone speech transcription task. In this paper, we study several discriminative feature extraction approaches in i-vector space to improve both recognition accuracy and run-time efficiency. New experimental results are reported on a much larger scale LVCSR task with about 2000 hours training data.

Index Terms— i-vector, irrelevant variability normalization, discriminative feature extraction

1. INTRODUCTION

Recently, a so-called i-vector approach [2] was proposed to extract a low-dimensional feature vector from a speech segment to represent speaker information, which has been successfully applied to speaker verification and become popular in speaker recognition community. In [8], an i-vector based approach was used for acoustic sniffing in irrelevant variability normalization (IVN) based acoustic model training for large vocabulary continuous speech recognition (LVCSR) (e.g., [6, 11]). In [12], a new i-vector approach was proposed by using a full factor analysis model with a residual term. Compared with the traditional i-vector approach, unfortunately only minor improvement in recognition accuracy was achieved when it was applied to acoustic sniffing for IVN-based acoustic model training. In [12], we also studied the effectiveness of using LDA (linear discriminant analysis) for i-vector transformation and dimension reduction, and promising results were achieved on Switchboard-1 conversational telephone speech transcription task. In this paper, we continue the above study by investigating two new discriminative feature extraction approaches based on minimum classification error (MCE) training and comparing their effectiveness with the LDA approach for i-vector transformation and dimension reduction. New experimental results of IVN-based acoustic model training are also reported on a much larger scale LVCSR task with about 2000 hours training data.

2. I-VECTOR APPROACH TO ACOUSTIC SNIFFING FOR IVN-BASED TRAINING

2.1. Raw i-Vector Extraction

In [8, 12], two approaches were proposed to extract an F-dimensional i-vector from a speech segment to represent acoustic information irrelevant to phonetic classification. In this study, we use the traditional i-vector approach as described in [8] because it has a lower computational complexity yet performs only slightly worse in recognition accuracy than the new i-vector approach in [12]. Readers are referred to [8] for technical details.

2.2. Discriminative Feature Extraction in i-Vector Space

If metadata (e.g., speaker ID in our experiments) for each speech segment is available, this information can be used (e.g., each speaker ID can be used as a class label in our experiments) to train an $F_1 \times F$ linear transform matrix $\boldsymbol{W}$, which can be used to transform each raw i-vector into a lower dimensional (i.e., $F_1 \leq F$) yet more discriminative feature space.

2.2.1. LDA based Feature Extraction

There are many ways to estimate the linear transformation $\boldsymbol{W}$ for discriminative feature extraction (DFE) and/or dimension reduction. One traditional way is to use LDA. As we demonstrated in [12], LDA-based i-vector transformation and dimension reduction indeed brings recognition accuracy improvement. However, when a cosine measure is used to measure the similarity between two transformed i-vectors, it is inconsistent with the underlying Euclidean metric used in LDA approach. Naturally, we want to know whether a more powerful DFE method could further improve the final speech recognition results. This is actually the main motivation of this study.

2.2.2. MCE based Discriminative Feature Extraction

In literature, many DFE methods based on MCE training have been proposed and studied. One example is the discriminative metric design (DMD) approach proposed in [7], which we used to develop our specific MCE-based DFE methods for nearest prototype classifiers using Euclidean distance based dissimilarity measure and cosine similarity measure, respectively. To the best of our knowledge, no study has been reported on the MCE-based DFE for the prototype-based classifier with cosine similarity measure.

Let's consider a speaker classification problem using an utterance-based i-vector. Suppose there are in total S training speakers denoted

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

as a speaker set $C = \{1, \ldots, S\}$, and for each training utterance i, a speaker label $c_i \in C$ is available. An F-dimensional raw i-vector $\boldsymbol{w}_i$ can be extracted for i, and then transformed into a new F_1-dimensional feature vector by using an $F_1 \times F$ linear transformation matrix W. A discriminant function can be defined for the s-th speaker as $g_s(\boldsymbol{w}_i, \boldsymbol{W})$, which enables the following speaker classification rule for an unknown i-vector $\boldsymbol{w}_i$:

$$C(\boldsymbol{w}_i) = \arg\max_s g_s(\boldsymbol{w}_i, \boldsymbol{W}). \tag{1}$$

A misclassification measure can be defined for each training i-vector from the s-th speaker as

$$d_s(\boldsymbol{w}_i, \boldsymbol{W}) = -g_s(\boldsymbol{w}_i, \boldsymbol{W}) + G_s(\boldsymbol{w}_i, \boldsymbol{W}), \tag{2}$$

where

$$G_s(\boldsymbol{w}_i, \boldsymbol{W}) = \max_{k, k \neq s} g_k(\boldsymbol{w}_i, \boldsymbol{W}). \tag{3}$$

The loss function is then defined as:

$$l_s(\boldsymbol{w}_i, \boldsymbol{W}) = \frac{1}{1 + \exp(-\gamma d_s + \theta)}, \tag{4}$$

where γ and θ are two control parameters. An empirical average loss can then be defined on the training set as

$$L_0(\boldsymbol{W}) = \frac{1}{N} \sum_s \sum_{i, c_i = s} l_s(\boldsymbol{w}_i, \boldsymbol{W}), \tag{5}$$

where N is the total number of training utterances.

In this study, two types of discriminant function are studied. The first one is based on Euclidean distance and defined as follows [7]:

$$g_s^{\text{euc}}(\boldsymbol{w}_i, \boldsymbol{W}) = -\frac{1}{2}(\boldsymbol{w}_i - \boldsymbol{\mu}_s)^\top \boldsymbol{W}\boldsymbol{W}^\top (\boldsymbol{w}_i - \boldsymbol{\mu}_s), \tag{6}$$

where $\boldsymbol{\mu}_s$ is the prototype parameter for the s-th speaker. In this study, we used a fixed $\boldsymbol{\mu}_s$, which is the mean of the training raw i-vectors for the s-th speaker, i.e.,

$$\boldsymbol{\mu}_s = \frac{1}{N_s} \sum_{i, c_i = s} \boldsymbol{w}_i, \tag{7}$$

in which N_s is the total number of utterances of speaker s.

Another discriminant function is based on cosine similarity and defined as follows:

$$g_s^{\cos}(\boldsymbol{w}_i, \boldsymbol{W}) = \frac{\boldsymbol{\mu}_s^\top}{||\boldsymbol{\mu}_s||} \cdot \frac{\boldsymbol{W}^\top \boldsymbol{w}_i}{\sqrt{\boldsymbol{w}_i^\top \boldsymbol{W}\boldsymbol{W}^\top \boldsymbol{w}_i}}, \tag{8}$$

where $\boldsymbol{\mu}_s$ is the prototype parameter for the s-th speaker calculated as follows:

$$\boldsymbol{\mu}_s = \frac{1}{N_s} \sum_{i, c_i = s} \frac{\boldsymbol{W}^\top \boldsymbol{w}_i}{\sqrt{\boldsymbol{w}_i^\top \boldsymbol{W}\boldsymbol{W}^\top \boldsymbol{w}_i}} \ . \tag{9}$$

Given the set of training i-vectors, the i-vector transform $\boldsymbol{W}$ can be estimated by minimizing $L_0(\boldsymbol{W})$ with the following optimization procedure:

Step 1: Initialize $\boldsymbol{W}$ as LDA transform. Set $t = 0$.

Step 2: Update $\boldsymbol{W}$ by fixing $\boldsymbol{\mu}_s$'s as follows:

$$\boldsymbol{W}^{t+1} = \boldsymbol{W}^t + \alpha \frac{\partial L_0(\boldsymbol{W})}{\partial \boldsymbol{W}}, \tag{10}$$

where α is a learning rate. The derivative $\frac{\partial L_0(\boldsymbol{W})}{\partial \boldsymbol{W}}$ is calculated for each type of discriminant function as follows:

- for Euclidean distance

$$\frac{\partial L_0(\boldsymbol{W})}{\partial \boldsymbol{W}} = \frac{1}{N} \sum_s \sum_{i, c_i = s} \gamma l_s(\boldsymbol{w}_i, \boldsymbol{W})(1 - l_s(\boldsymbol{w}_i, \boldsymbol{W}))$$
$$\Big\{ - (\boldsymbol{w}_i - \boldsymbol{\mu}_s)(\boldsymbol{w}_i - \boldsymbol{\mu}_s)^\top + (\boldsymbol{w}_i - \boldsymbol{\mu}_k)(\boldsymbol{w}_i - \boldsymbol{\mu}_k)^\top \Big\} \boldsymbol{W}$$

- for cosine similarity

$$\frac{\partial L_0(\boldsymbol{W})}{\partial \boldsymbol{W}} = \frac{1}{N} \sum_s \sum_{i, c_i = s} \gamma l_s(\boldsymbol{w}_i, \boldsymbol{W})(1 - l_s(\boldsymbol{w}_i, \boldsymbol{W}))$$
$$\Big\{ \frac{\boldsymbol{w}_i}{\sqrt{\boldsymbol{w}_i^\top \boldsymbol{W}\boldsymbol{W}^\top \boldsymbol{w}_i}} \cdot (\frac{\boldsymbol{\mu}_s^\top}{||\boldsymbol{\mu}_s||} - \frac{\boldsymbol{\mu}_k^\top}{||\boldsymbol{\mu}_k||})$$
$$- \frac{\boldsymbol{w}_i^\top \boldsymbol{W}}{\sqrt{\boldsymbol{w}_i^\top \boldsymbol{W}\boldsymbol{W}^\top \boldsymbol{w}_i}} \cdot (\frac{\boldsymbol{\mu}_s}{||\boldsymbol{\mu}_s||} - \frac{\boldsymbol{\mu}_k}{||\boldsymbol{\mu}_k||}) \frac{\boldsymbol{w}_i \boldsymbol{w}_i^\top}{\boldsymbol{w}_i^\top \boldsymbol{W}\boldsymbol{W}^\top \boldsymbol{w}_i} \boldsymbol{W} \Big\}$$

where

$$k = \arg\max_{k, k \neq c} g_k(\boldsymbol{w}_i, \boldsymbol{W}^t) \ .$$

Step 3: When cosine similarity is used, update the prototype for each speaker class by using Eq. (9).

Step 4: Repeat **Step 2** and **Step 3** until the decrease of $L_0(\boldsymbol{W})$ is smaller than a pre-specified threshold.

2.3. Acoustic Condition Clustering using i-Vectors

Given the set of raw or transformed (via LDA or MCE-DFE) training i-vectors, we use a hierarchical divisive clustering algorithm, namely LBG algorithm [4], to cluster them into multiple clusters. Either a Euclidean distance is used to measure the dissimilarity between two i-vectors, $\boldsymbol{w}_i$ and $\boldsymbol{w}_j$, or a cosine measure is used to measure the similarity between two i-vectors. In the latter case, we normalize each i-vector to have a unit norm so that the cosine similarity can be calculated simply as $\boldsymbol{w}_i^\top \boldsymbol{w}_j$.

After the convergence of the LBG clustering algorithm, we obtain E clusters of i-vectors with their centroids denoted as $\boldsymbol{c}_1^{(w)}, \boldsymbol{c}_2^{(w)}, \ldots, \boldsymbol{c}_E^{(w)}$, respectively. Then the speech segments in training set can be distributed to different clusters according to the one-to-one relationship with the corresponding i-vectors. By doing so, all the feature vectors from the same cluster will share a single linear feature transform in IVN-based acoustic model training (to be explained in the next subsection) and the total number of feature transforms equals the number of acoustic conditions.

2.4. i-Vector Approach to Acoustic Sniffing for IVN-based Training

As described in e.g., [6, 11, 8, 12], in IVN-based training, a set of linear feature transforms along with a set of generic hidden Markov models (HMMs) are trained using a maximum likelihood (ML) (e.g., [6]) and/or discriminative training (DT) (e.g., [11]) criterion. The feature transforms are used to normalize the irrelevant variabilities of different acoustic conditions. As Fig. 1 shows, given a speech segment (e.g., several frames of speech, an utterance, or several utterances), a so-called "acoustic sniffing" module is responsible for detecting the corresponding acoustic condition and choosing the most appropriate transform(s) accordingly. In the recognition stage, given an unknown speech segment, the "acoustic sniffing" module is

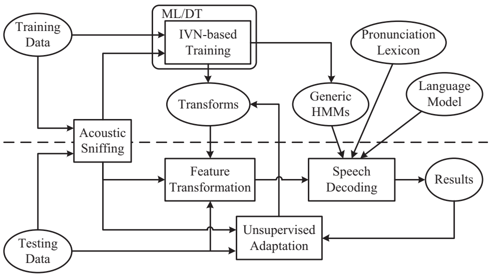

Fig. 1. *An illustration of IVN-based framework for acoustic modeling, training and adaptation.*

used again for choosing the pre-trained IVN transform(s). The transformed feature vector sequence is then decoded using a conventional LVCSR decoder. After the first-pass recognition, unsupervised adaptation can be performed to adapt the selected feature transform(s). Therefore, an improved recognition accuracy can be achieved in the second-pass decoding.

In this study, the following feature transformation (FT) function is used:

$$\boldsymbol{x}_t = \mathcal{F}(\boldsymbol{y}_t; \boldsymbol{\Theta}) = \boldsymbol{A}^{(e)}\boldsymbol{y}_t + \boldsymbol{b}^{(e)} \tag{11}$$

where $\boldsymbol{y}_t$ is the t-th D-dimensional feature vector of the input feature vector sequence $\boldsymbol{Y}$; $\boldsymbol{x}_t$ is the transformed feature vector; e is a label (transform index) informed by the "acoustic sniffing" module for the $D \times D$ nonsingular transformation matrix $\boldsymbol{A}^{(e)}$ and D-dimensional bias vector $\boldsymbol{b}^{(e)}$; and $\boldsymbol{\Theta} = \{\boldsymbol{A}^{(e)}, \boldsymbol{b}^{(e)} | e = 1, 2, \cdots, E\}$ denotes the set of feature transformation parameters with E being the total number of acoustic conditions as described in the previous subsection.

In IVN-based framework, the acoustic sniffing module is essential for both training and recognition. As mentioned previously, in [8, 12], the i-vector based approach was used for acoustic sniffing and promising results were achieved. In this study, we compare the effectiveness of the newly proposed MCE-based DFE methods with the traditional LDA transformation in this context. Given a speech segment $\boldsymbol{Y}$, i-vector based acoustic sniffing can be done as follows:

Step 1: Extract an i-vector $\boldsymbol{w}_i$ from $\boldsymbol{Y}$ as described in [8].

Step 2: Apply $\boldsymbol{W}$ for feature transformation (via LDA or MCE-DFE) if applicable. Further normalize the i-vector to have a unit norm if cosine similarity measure is used. Let's use $\hat{\boldsymbol{w}}$ to denote the final transformed i-vector.

Step 3: Classify the i-vector $\hat{\boldsymbol{w}}$ into an acoustic condition, e, as follows:

- If Euclidean distance is used as a dissimilarity measure,

$$e = \underset{l=1,2,\ldots,E}{\operatorname{argmin}} \; ||\hat{\boldsymbol{w}} - \boldsymbol{c}_l^{(w)}||;$$

- If cosine similarity measure is used,

$$e = \underset{l=1,2,\ldots,E}{\operatorname{argmax}} \; \hat{\boldsymbol{w}}^\top \boldsymbol{c}_l^{(w)} \; .$$

The pre-trained linear feature transform for IVN based training from the corresponding acoustic condition e will be used for feature transformation.

The same acoustic sniffing procedure is used in both training and recognition stages. It is noted that in the second case of **Step 3** of the above procedure, if i-vector $\hat{\boldsymbol{w}}$ and centroids $\boldsymbol{c}_l^{(w)}$'s have been normalized to unit norm, it can be proven that the above two decision rules will give the same result. Therefore the first decision rule can always be used in run-time for more efficient acoustic sniffing because a partial-distance based technique can be used to eliminate unnecessary computations.

3. EXPERIMENTS AND RESULTS

3.1. Experimental Setup

Two sets of experiments were performed in this study. The first one was done on a Switchboard-1 conversational telephone speech transcription task [3] with about 300 hours training data to study carefully the algorithmic issues of the proposed approach. The experimental setup is the same as in [11, 8].

The second experiment was performed on a "Jumbo" task with a training set consisting of about 300 hours training speech from Switchboard-1 corpus and about 1,700 hours training speech from Fisher English corpus (part 1 and 2) [1]. The Spring 2003 NIST rich transcription set (RT03S) with about 6.3 hours conversational telephone speech was used as the testing test. For front-end feature extraction, we used 52-dimensional PLP_E_D_A_T (in HTK's terminology [10]) features with mean and variance normalization. HLDA transformation was then applied to reduce the feature vector dimension to 39. For acoustic modeling, we used phonetic decision tree based tied-state triphone GMM-HMMs with 18,000 states and 72 Gaussian components per state. Our recognition vocabulary contained 47,633 unique words. The pronunciation lexicon contained multiple pronunciations per word with a total of 58,393 unique pronunciations. A trigram language model, which was trained on the 2000-hour Jumbo-corpus transcripts and interpolated with a written-text trigram, was used in decoding. All of the recognition experiments were performed with a Microsoft in-house decoder as in [11, 8] and the results were evaluated by using the NIST Scoring Toolkit SCTK [5].

The settings of relevant control parameters are as follows: The number of UBM-GMM components for i-vector extraction $K = 1,024$ [8]; The dimension of the raw i-vectors $F = 600$ for Switchboard-1 and $F = 400$ for Jumbo task; The i-vector dimension after LDA or MCE-DFE $F_1 = 100$; For IVN-based MMI training [11], the learning constant $EConst = 2$, i-smoothing $\tau = 100$, and acoustic scaling factor $\kappa = 1/12$; For acoustic sniffing, 128 acoustic conditions were clustered, therefore $E = 128$; For MCE training, the control parameters of the sigmoid function are $\gamma = 32$ and $\theta = 0$.

To handle large-scale training data, the tools for hyperparameter estimation in i-vector extraction, LBG clustering and GMM training have been implemented based upon MSR Asia's MPI-based machine learning platform [9]. This platform was developed on top of Microsoft Windows HPC Server, and optimized for various machine learning algorithms including speech training. With this high-performance parallel computing platform, experiments can be run very efficiently for large-scale tasks.

3.2. Experimental Results

Table 1 gives a comparison of speaker classification errors by using different DFE methods on Switchboard-I training set. It is quite clear that using cosine similarity based discriminant function for speaker classification and the corresponding MCE-DFE method (labeled as "DFE (COS)") achieves much lower error rate than that of using

Table 1. Comparison of speaker classification errors by using different DFE methods on Switchboard-I training set.

Measure	LDA	DFE (EUC)	DFE (COS)
Euclidean	13.9%	12.7%	N/A
Cosine	12.3%	N/A	6.0%

Table 2. Comparison of different acoustic sniffing approaches for IVN-based ML training by using recognition word error rate (WER in %) on Switchboard-I task as performance metric. Our ML-trained baseline system achieves a WER of 30.0%.

	eval2000			
Measure	w/o trans.	LDA	DFE (EUC)	DFE (COS)
Euclidean	27.3	26.5	26.7	N/A
Cosine	27.2	26.3	N/A	26.5

Euclidean distance based discriminant function and the corresponding MCE-DFE method (labeled as "DFE (EUC)"). Both MCE-DFE methods perform better than the LDA based DFE when the Euclidean distance based discriminant function is used. Interestingly, LDA based DFE with cosine similarity based discriminant function performs slightly better than the "DFE (EUC)" case with the Euclidean distance based discriminant function.

Table 2 gives a comparison of different acoustic sniffing approaches with different DFE methods for IVN-based ML training by using recognition word error rate (WER in %) on Switchboard-I task as performance metric. Unfortunately, MCE-DFE methods failed to outperform the LDA method, yet all the DFE methods for i-vector transformation and dimension reduction achieves better WER than the cases without i-vector transformation (labeled as "w/o trans."). It is noted that reducing the dimension of i-vector from 600 to 100 via LDA or MCE-DFE methods does not degrade recognition accuracy.

Since using LDA transformation with cosine similarity measure in acoustic sniffing gives us the best recognition accuracy, we used this setup for the set of experiments on "Jumbo" task. The experimental results are summarized in Table 3. Our ML- and MMI-trained baseline systems without IVN training achieved a WER of 30.2% and 26.6% respectively. After ML- and MMI-based IVN training but without using LDA for i-vector transformation, the WERs are reduced to 28.8% and 25.8% respectively. This demonstrates clearly the power of IVN training. After using LDA for i-vector transformation and dimension reduction (from 400 to 100), the WERs of IVN-based ML training and MMI training are further reduced to 28.2% and 25.4% respectively.

4. SUMMARY

In this paper, we have studied several discriminative feature extraction approaches in i-vector space and compared their effectiveness for acoustic sniffing in IVN-based acoustic model training. New experimental results are reported on "Jumbo" task with about 2000 hours training data. LDA-based i-vector transformation and dimension reduction plus using cosine similarity measure in acoustic sniffing has improved both recognition accuracy and run-time efficiency, therefore is the solution we recommended for others to use in practice.

Table 3. The effectiveness of using LDA for i-vector transformation and dimension reduction (from 400 to 100) on "Jumbo" task, and the relative error rate reduction against the ML baseline (FT: feature transform in IVN training).

		rt03			
Method		w/o LDA		LDA	
FT	HMM	WER(%)	Rel.(%)	WER(%)	Rel.(%)
-	ML	30.2	N/A	N/A	N/A
-	MMI	26.6	11.9	N/A	N/A
ML	ML	28.8	4.6	28.2	6.6
ML	MMI	25.8	14.6	25.4	15.9

5. ACKNOWLEDGEMENT

We thank our colleagues, Frank Seide and Gang Li for providing us the ML-trained baseline system on "Jumbo" task.

6. REFERENCES

[1] C. Cieri, D. Miller, and K. Walker, "The Fisher corpus: a resource for the next generations of speech-to-text," *Proc. 4th International Conference on Language Resources and Evaluation*, 2004, pp.69-71.

[2] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-end factor analysis for speaker verification," *IEEE Trans. on Audio, Speech and Language Processing*, Vol. 19, No. 4, pp.788-798, 2011.

[3] 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.

[4] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Trans. on Communication*, Vol. COM-28, pp.84-95, 1980.

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

[6] 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.

[7] H. Watanabe, T. Yamaguchi, and S. Katagiri, "Discriminative metric design for robust pattern recognition," *IEEE Trans. on Signal Processing*, Vol. 45, No. 11, pp.2655-2662, 1997.

[8] J. Xu, Y. Zhang, Z.-J. Yan, and Q. Huo, "An i-vector based approach to acoustic sniffing for irrelevant variability normalization based acoustic model training and speech recognition," *Proc. Interspeech-2011*, pp.1701-1704.

[9] Z.-J. Yan, T. Gao, and Q. Huo, "Designing an MPI-based parallel and distributed machine learning platform on large-scale HPC clusters," submitted to *IWSML-2012*.

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

[11] Y. Zhang, J. Xu, Z.-J. Yan, and Q. Huo, "A study of irrelevant variability normalization based discriminative training approach for LVCSR," *Proc. ICASSP-2011*, pp.5308-5311.

[12] Y. Zhang, Z.-J. Yan, and Q. Huo, "A new i-vector approach and its application to irrelevant variability normalization based acoustic model training," *MLSP-2011*, Beijing, China, 6 pages.