IRRELEVANT VARIABILITY NORMALIZATION BASED HMM TRAINING USING MAP ESTIMATION OF FEATURE TRANSFORMS FOR ROBUST SPEECH RECOGNITION

Donglai ZHU [1], *Qiang HUO*[2]

[1] Institute for Infocomm Research, Singapore
[2]Microsoft Research Asia, Beijing, China
(E-mails: dzhu@i2r.a-star.edu.sg, qianghuo@microsoft.com)

ABSTRACT

In the past several years, we've been studying feature transformation (FT) approaches to robust automatic speech recognition (ASR) which can compensate for possible "distortions" caused by factors irrelevant to phonetic classification in both training and recognition stages. Several FT functions with different degrees of flexibility have been studied and the corresponding maximum likelihood (ML) training techniques developed. In this paper, we study yet another new FT function which takes the most flexible form of frame-dependent linear transformation. Maximum *a posteriori* (MAP) estimation is used for estimating FT function parameters to deal with the possible problem of insufficient training data caused by the increased number of model parameters. The effectiveness of the proposed approach is confirmed by evaluation experiments on Finnish Aurora3 database.

Index Terms— robust speech recognition, feature transformation, MAP estimate, hidden Markov model.

1. INTRODUCTION

Using feature transformation (FT) in training and/or recognition stages to compensate for possible "distortions" caused by factors irrelevant for phonetic classification has been studied in robust automatic speech recognition (ASR) area for many years (see references in [7, 5, 11]). In the past several years, we've also been working on this research topic based on the concept of stochastic vector mapping (SVM) that performs a frame-dependent feature transformation to compensate for "environmental" variabilities in both training and recognition stages (e.g., [7, 8, 5, 11]). To avoid the possible confusion with a more popular jargon, namely support vector machine, we will use hereinafter FT instead of SVM to refer to our feature transformation approaches. In the following, our past attempts are summarized first.

Let's assume that a speech utterance corrupted by some "distortions" has been transformed into a sequence of feature vectors. Given a set of training data $\mathcal{Y} = \{Y_i\}_{i=1}^{I}$, where Y_i is a sequence of feature vectors of original speech, suppose that they can be partitioned into E "environment" classes, and the D-dimensional feature vector y under an environment class e follows the distribution of a mixture of Gaussians, $p(y|e) = \sum_{k=1}^{K} p(k|e)p(y|k,e) = \sum_{k=1}^{K} p(k|e)\mathcal{N}(y;\xi_k^{(e)}, R_k^{(e)})$, where $\mathcal{N}(\cdot;\xi,R)$ is a normal distribution with mean vector ξ and diagonal covariance matrix R. Readers are referred to [9] for the approach we used for the automatic clustering of environment conditions from training data $\mathcal{Y}$, the labeling of an utterance Y to a specific environment condition, and the estimation of the above model parameters. Given the set of Gaussian mixture models (GMM) $\{p(y|e)\}$, the task of frame-dependent feature compensation is to estimate the compensated feature vector $\hat{x}$ from the original feature vector y by applying the environment-dependent transformation $\mathcal{F}(y;\Theta^{(e_y)})$, where $\Theta^{(e_y)}$ represents the trainable parameters of the transformation and e_y denotes the corresponding environment class to which y belongs. However, for the simplicity of notation, we will hereinafter simply use e to denote the environment class to which y belongs, if no confusion will be caused according to the context.

So far we have studied five forms of FT functions [7, 8, 5, 11]. The first one is defined as follows:

$$\hat{x} \triangleq \mathcal{F}_1(y;\Theta^{(e)}) = y + \sum_{k=1}^{K} p(k|y,e)b_k^{(e)} \ , \qquad (1)$$

where

$$p(k|y,e) = \frac{p(k|e)p(y|k,e)}{\sum_{j=1}^{K} p(j|e)p(y|j,e)} \ ,$$

and $\Theta^{(e)} = \{b_k^{(e)}\}_{k=1}^{K}$. The second FT function is defined as

$$\hat{x} \triangleq \mathcal{F}_2(y;\Theta^{(e)}) = y + b_k^{(e)} \ , \qquad (2)$$

where, for the environment class e which y belongs to,

$$k = \arg\max_{k'=1,\ldots,K} p(k'|y,e) \ . \qquad (3)$$

The third one is defined as

$$\hat{x} \triangleq \mathcal{F}_3(y;\Theta^{(e)}) = A^{(e)}y + b^{(e)} \ , \qquad (4)$$

where $A^{(e)}$ is a nonsingular $D \times D$ matrix, $b^{(e)}$ is a D-dimensional vector, and $\Theta^{(e)} = \{A^{(e)}, b^{(e)}\}$. The fourth FT function is defined as

$$\hat{x} \triangleq \mathcal{F}_4(y;\Theta^{(e)}) = A^{(e)}y + \sum_{k=1}^{K} p(k|y,e)b_k^{(e)} \ , \qquad (5)$$

where $\Theta^{(e)} = \{A^{(e)}, b_k^{(e)}, k = 1,\ldots,K\}$. The fifth FT function is defined as

$$\hat{x} \triangleq \mathcal{F}_5(y;\Theta^{(e)}) = A^{(e)}y + b_k^{(e)} \ , \qquad (6)$$

where k is calculated by using Eq. (3).

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 $\lambda = \{\pi_s, a_{ss'}, c_{sm}, \mu_{sm}, \Sigma_{sm}; s, s' = 1,\cdots,S; m = 1,\cdots,M\}$, 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, $\mu_{sm} = [\mu_{sm1},\cdots,\mu_{smD}]^T$ is a D-dimensional mean vector, and $\Sigma_{sm} =$

1-4244-1484-9/08/$25.00 ©2008 IEEE

$diag\{\sigma^2_{sm1}, \cdots, \sigma^2_{smD}\}$ is a diagonal covariance matrix. Our environment compensated training approach is to adjust FT function parameters $\Theta = \{\Theta^{(e)}, e = 1, \cdots, E\}$ and CDHMM parameters $\Lambda = \{\lambda\}$ to optimize a training objective function. For example, the ML training approaches of $\mathcal{F}_1$ and $\mathcal{F}_2$ are presented in [7, 8]. The ML training approaches of $\mathcal{F}_3$ and $\mathcal{F}_4$ are presented in [5]. The ML training procedure of $\mathcal{F}_5$ is described in [11].

In recognition, given an unknown utterance Y, the most similar training environment class e is identified first (e.g. [9]). Then, the corresponding GMM and the mapping function are used to derive a compensated version of $\hat{X}$ from Y. For the convenience of notation, we also use hereinafter $\mathcal{F}(Y; \Theta^{(e)})$ to denote the compensated version of the utterance Y by transforming individual feature vector y_t as defined in the previous FT functions. After feature compensation, $\hat{X}$ is finally recognized by an HMM-based recognizer trained as described in [8], [5] or [11].

2. WHAT'S NEW

2.1. New Feature Transformation Function

In this paper, we study yet another new FT function $\mathcal{F}_6(y; \Theta^{(e)})$ which is defined as follows:

$$\hat{x} \triangleq \mathcal{F}_6(y; \Theta^{(e)}) = A^{(e)}_k y + b^{(e)}_k \ , \qquad (7)$$

where $A^{(e)}_k$ is a nonsingular $D \times D$ matrix, $b^{(e)}_k$ is a D-dimensional vector, $\Theta^{(e)} = \{A^{(e)}_l, b^{(e)}_l; l = 1, \ldots, K\}$, and k is calculated by using Eq. (3).

In recognition, similar to CMLLR [3], the likelihood of each frame of observation has to be evaluated as follows:

$$p(y|\Lambda, \Theta) = \mathcal{N}(\mathcal{F}_6(y; \Theta^{(e)}); \mu_{sm}, \Sigma_{sm})|\det(A^{(e)}_k)| \ , \qquad (8)$$

where $\det(A^{(e)}_k)$ denotes the determinant of matrix $A^{(e)}_k$.

2.2. ML/MAP Training of $\mathcal{F}_6(y; \Theta^{(e)})$ and CDHMMs

For ML training of parameters of $\mathcal{F}_6(y; \Theta^{(e)})$ and CDHMMs, it is similar to the procedure for speaker adaptive training described in [3]. The main difference is that the relevant sufficient statistics are accumulated according to the labeling of feature vectors in our case rather than the grouping of Gaussian components of CDHMMs as described in [3] and implemented in HTK [10]. However, given the large number of parameters in FT function $\mathcal{F}_6(y; \Theta^{(e)})$, there may exist a serious numerical problem when the training data assigned to $(A^{(e)}_k, b^{(e)}_k)$ is insufficient. In this paper, we propose to use maximum *a posteriori* (MAP) estimation for FT function parameters to deal with this problem. An objective function is defined as follows:

$$\mathcal{L}(\Theta, \Lambda) = \prod_{i=1}^{I} p(\mathcal{F}_6(Y_i; \Theta)|\Lambda) \prod_{e=1}^{E} \prod_{k=1}^{K} p(A^{(e)}_k, b^{(e)}_k) \qquad (9)$$

where $p(A^{(e)}_k, b^{(e)}_k)$ is the prior PDF (probability density function) of $(A^{(e)}_k, b^{(e)}_k)$. For notational convenience, let's use $W^{(e)}_k$ to denote the extended transformation matrix $[b^{(e)}_k A^{(e)}_k]$, and o to denote the extended observation vector, $[1 \ y^T]^T$. Eq. (7) can then be rewritten as $\hat{x} = W^{(e)}_k o$. The prior PDF of $W^{(e)}_k$ is defined as a matrix variate normal PDF (e.g., [6]):

$$\begin{aligned} p(W) \quad \propto \quad & |\Xi|^{-(D+1)/2}|\Phi|^{-D/2} \\ & \exp\left[-\frac{1}{2}\mathrm{tr}(W-U)^T \Xi^{-1} (W-U) \Phi^{-1}\right] \qquad (10) \end{aligned}$$

where U, Ξ, and Φ are the hyperparameters, with $U \in \mathbb{R}^{D\times(D+1)}$, $\Xi \in \mathbb{R}^{D\times D}$, $\Xi \geq 0$, and $\Phi \in \mathbb{R}^{(D+1)\times(D+1)}$, $\Phi \geq 0$. In this study, we set $U^{(e)}_k$ as the ML estimate of $W^{(e)}$ by using FT function $\mathcal{F}_3(y; W^{(e)})$ [5]. We fix $\Xi^{(e)}_k = cI$ and $\Phi^{(e)}_k = I$, where c is a scalar control parameter and I is an identity matrix. It is noted that a full version of ML training for both Θ and Λ can be obtained by specifying a noninformative prior PDF for $W^{(e)}_k$, i.e., setting $c = \infty$.

The following *method of alternating variables* can then be used to maximize the above objective function:

Step 1: *Initialization*

A set of CDHMMs, Λ, are trained from multi-condition training data $\mathcal{Y}$ and used as the initial values of HMM parameters. The initial values of transformation matrices $A^{(e)}_k$'s are set to be identity matrices and the initial values of bias vectors $b^{(e)}_k$'s are set to be zero vectors.

Step 2: *Estimating FT Function Parameters Θ by Fixing CDHMM Parameters Λ*

Given the CDHMM parameters Λ, for each environment class e, we estimate the environment-dependent mapping function parameters $\bar{\Theta}^{(e)}$ by using MAP estimation with several (1 in our experiments) EM iterations. The updating formula for the r-th row of $W^{(e)}_k$ (hereinafter denoted as $w^{(e)}_{kr}$) is as follows:

$$w^{(e)}_{kr} = (\alpha^{(e)}_{kr} p^{(e)}_{kr} + v^{(e)}_{kr} + \frac{1}{c}U^{(e)}_{kr})(G^{(e)}_{kr} + \frac{1}{c}I)^{-1} \ , \qquad (11)$$

where $p^{(e)}_{kr}$ is the extended cofactor row vector $[0, c^{(e)}_{kr1} \ldots c^{(e)}_{krD}]$ with $c^{(e)}_{krl} = cof(A^{(e)}_{krl})$, and

$$G^{(e)}_{kr} \quad = \quad \sum_{i\in I_e}\sum_t\sum_s\sum_m \frac{\zeta_{it}(s,m)}{\sigma^2_{smr}} o_{it} o_{it}^T \delta_{kit} \qquad (12)$$

$$v^{(e)}_{kr} \quad = \quad \sum_{i\in I_e}\sum_t\sum_s\sum_m \frac{\zeta_{it}(s,m)}{\sigma^2_{smr}} \mu_{smr} o_{it}^T \delta_{kit} \qquad (13)$$

$$\alpha^{(e)}_{kr} \quad = \quad -\frac{\varepsilon_2}{2\varepsilon_1} \pm \frac{\sqrt{\varepsilon_2^2 + 4\beta^{(e)}_k \varepsilon_1}}{2\varepsilon_1} \qquad (14)$$

$$\delta_{kit} \quad = \quad \begin{cases} 1 & \text{if } k = \arg\max_{k'} p(k'|y_{it}, e) \\ 0 & \text{otherwise} \end{cases} \qquad (15)$$

$$\beta^{(e)}_k \quad = \quad \sum_{i\in I_e}\sum_t\sum_s\sum_m \zeta_{it}(s,m)\delta_{kit} \qquad (16)$$

$$\varepsilon_1 \quad = \quad p^{(e)}_{kr}(G^{(e)}_{kr} + \frac{1}{c}I)^{-1} p^{(e)T}_{kr} \qquad (17)$$

$$\varepsilon_2 \quad = \quad p^{(e)}_{kr}(G^{(e)}_{kr} + \frac{1}{c}I)^{-1}(v^{(e)}_{kr} + \frac{1}{c}U^{(e)}_{kr})^T \ . \qquad (18)$$

The value of $\alpha^{(e)}_{kr}$ is selected that maximizes

$$\mathcal{Q}_e = \beta^{(e)}_k \log|\alpha^{(e)}_{kr}\varepsilon_1 + \varepsilon_2| - \frac{1}{2}\alpha^{(e)2}_{kr}\varepsilon_1 \ . \qquad (19)$$

In the above equations, $\zeta_{it}(s,m)$ is the occupation probability of Gaussian component m in state s at time t of the current compensated observation, where the likelihood for each frame of observation is evaluated by using Eq. (8).

As we mentioned above, ML estimation is a special case of the MAP estimation. By setting $c = \infty$, we obtain the following updating formula for $w^{(e)}_{kr}$:

$$w^{(e)}_{kr} = (\alpha^{(e)}_{kr} p^{(e)}_{kr} + v^{(e)}_{kr}) G^{(e)-1}_{kr} \ . \qquad (20)$$

Eqs. (17) and (18) are also simplified accordingly as follows:

$$\varepsilon_1 = p_{kr}^{(e)} G_{kr}^{(e)-1} p_{kr}^{(e)T} , \quad (21)$$

$$\varepsilon_2 = p_{kr}^{(e)} G_{kr}^{(e)-1} v_{kr}^{(e)T} . \quad (22)$$

Step 3: *Estimating CDHMM Parameters Λ by Fixing FT Function Parameters Θ*

Given the estimated FT function parameters $\bar{\Theta}$, we re-estimate CDHMM parameters $\bar{\Lambda}$ by using the following updating formulas:

$$\bar{\mu}_{sm} = \frac{\sum_{i,t,s,m} \zeta_{it}(s,m)\hat{x}_{it}}{\sum_{i,t,s,m} \zeta_{it}(s,m)} \quad (23)$$

$$\bar{\Sigma}_{sm} = \frac{\sum_{i,t,s,m} \zeta_{it}(s,m)(\hat{x}_{it} - \bar{\mu}_{sm})(\hat{x}_{it} - \bar{\mu}_{sm})^T}{\sum_{i,t,s,m} \zeta_{it}(s,m)} \quad (24)$$

where $\zeta_{it}(s,m)$ is again the occupation probability of Gaussian component m in state s at time t of the current compensated observation with the likelihood for each frame of observation evaluated using Eq. (8). Five EM iterations are performed in our experiments.

Step 4: *Repeat* **Step 2** *and* **Step 3** *several times if necessary*.

In our experiments, we skipped this step.

3. EXPERIMENTS AND RESULTS

3.1. Experimental Setup

We use Finnish Aurora3 database [1] to verify our algorithm. Aurora3 contains utterances of connected digits that were recorded by using both close-talking (CT) and hands-free (HF) microphones in cars under several driving conditions to reflect some realistic scenarios for typical in-vehicle ASR applications. There are roughly three conditions: *quiet*, *low noise*, and *high noise*. The database is divided into three subsets according to matching degree between training data and test data. In this paper we conduct our experiments on the so-called *Well-Matched (WM) condition*, where both training and testing data include utterances recorded by both CT and HF microphones from all conditions.

In our experiments, the ETSI Advanced Front-End (AFE) as described in [2] is used for feature extraction from a speech utterance. A feature vector sequence is extracted from the input speech utterance via a sequence of processing modules that include noise reduction, waveform processing, cepstrum calculation, blind equalization, and "server feature processing". Each frame of feature vector has 39 features that consists of 12 MFCCs (C_1 to C_{12}), a combined log energy and C_0 term, and their first and second order derivatives. Although all the feature vectors are computed from a given speech utterance, the feature vectors that are sent to the speech recognizer and the training module are those corresponding to speech frames, as detected by a VAD module described in Annex A of [2]. After the VAD processing, the training data consists of 3080 utterances with the average utterance duration of 4.87 seconds, and the testing data consists of 1320 utterances with the average utterance duration of 4.95 seconds. In FT-based experiments, all the training data are clustered into 8 different environment classes (i.e. $E = 8$), of which each is modeled by a GMM consisting of K Gaussian components.

Each digit is modeled as a whole-word left-to-right CDHMM with 16 emitting states, 3 Gaussian mixture components with diagonal covariance matrices per state. Besides, two pause models, "sil" and "sp", are created to model the silence before/after the digit string and the short pause between any two digits, respectively. The "sil" model is a 3-emitting state CDHMM with a flexible transition structure as described in [4]. Each state is modeled by a mixture of 6 Gaussian components with diagonal covariance matrices. The "sp" model consists of 2 dummy states and a single emitting state which is tied with the middle state of "sil". During recognition, an utterance can be modeled by any sequence of digits with the possibility of a "sil" model at the beginning and at the end and a "sp" model between any two digits. The baseline system is trained from multi-condition training data without using feature compensation.

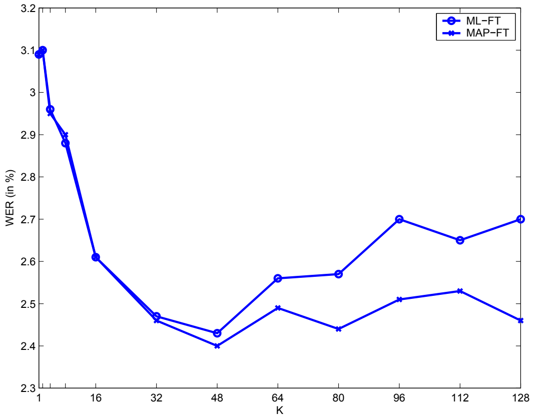

Fig. 1. Comparison of word error rates (WERs in %) achieved by using ML- and MAP-trained $\mathcal{F}_6(y; \Theta^{(e)})$ with different values of K ($E = 8$).

3.2. Comparison of ML and MAP Training

In order to compare the performance of ML and MAP estimation of parameters of FT function $\mathcal{F}_6(y; \Theta^{(e)})$ in all cases of sufficient and insufficient training data, we train a series of environmental condition dependent GMMs with the number of Gaussian components K varying from 2 to 128. With the increase of K, the number of transformation parameters increases quickly and the training data becomes severely insufficient to estimate these parameters reliably. For example, the number of FT function parameters $\{A_k^{(e)}, b_k^{(e)}\}$, in the setting of $E = 8$ and $K = 32$, is $(D \times D + D) \times K \times E = 10240 \times D$, which is much bigger than the number of parameters of baseline CDHMMs, by counting only the means and variances of 10 digit CDHMMs and "sil" CDHMM, i.e., $(16 \times 3 \times 10 + 3 \times 6) \times 2 \times D) = 996 \times D$.

In MAP estimation of transformation parameters, the scale factor c controls the degree of information "borrowed" from prior PDF. The influence of the prior information increases when c decreases. On the other hand, the influence of the training data increases when c increases. It is observed that a similar performance is achieved when c varies between 0.1 and 1000. We set $c = 100$ in our experiments.

Fig. 1 compares word error rates (WERs in %) achieved by using ML and MAP estimation of transformation parameters with different values of K. When K is smaller than 32, ML and MAP estimations yield very close performance because the amount of training data is still sufficient. For example, when $K = 32$, ML- and MAP-trained FT-based systems achieve a WER of 2.47% and 2.46% respectively. However, when $K = 48$, ML- and MAP-trained FT-based systems achieve a WER of 2.43% and 2.40% respectively. Apparently, given the same amount of training data, more FT transforms can be used for MAP estimation than ML estimation, which in turn offers hopefully a chance to achieve a better performance. In current experimental setup, it is observed that when $K > 48$, the performance of the ML estimation degrades severely because the training data becomes insufficient while the performance of MAP estimation suffers less severely.

3.3. Comparison with Other Methods

For the convenience of reference, let's use FT1, FT2, FT3, FT4, FT5 and FT6 to refer to FT-based approaches with 6 different FT functions respectively. Table 1 summarizes a comparison of FT-based approaches in terms of word error rate (WER in %), the number of parameters (counting both FT and CDHMM parameters), and relative error rate reduction (in %) compared to that achieved by the CDHMM baseline system without feature compensation. All the FT-based systems use the same default setting of $E = 8$, $K = 32$ and ML training except for FT6 systems. To evaluate the potential for performance improvement by using more complicated CDHMMs, we also increase the number of Gaussian components in each state of digit CDHMMs from 3 to 4 and 5, yielding CDHMM(4) and CDHMM(5), respectively. In Table 1, comparisons have also been made to the systems using CMLLR-based adaptive training (CMLLR-AT) [3, 10]. CMLLR-AT(1) refers to an CMLLR-AT based system with one global regression class, while CMLLR-AT(8) refers to an CMLLR-AT based system with 8 CMLLR transformations associated with 8 regression classes by clustering Gaussian components of CDHMMs as implemented in HTK [10]. From the results in Table 1, we made the following observations:

- The performance of traditional CDHMM-based systems saturates at CDHMM(4) with a relative error rate reduction of 12.91% versus baseline system;
- CMLLR-AT(1) can achieve a similar performance as that of CDHMM(4) with fewer parameters while CMLLR-AT(8) can achieve a better performance with the same number of parameters;
- FT3 method differs from CMLLR-AT(8) mainly in the selection of transformation classes and achieves slightly better performance than CMLLR-AT(8) with the same number of parameters;
- FT6 systems achieve much better performance than all the other systems. Among them, MAP-trained FT6 system with $K = 48$ yields a WER of 2.40%, which is the best among all the approaches compared and represents a relative error rate reduction of 39.24% over the baseline performance.

4. SUMMARY

In this paper, we have studied one more feature transformation (FT) approach to robust ASR which can compensate for possible "distortions" caused by factors irrelevant to phonetic classification in both training and recognition stages. Maximum *a posteriori* (MAP) estimation is used for estimating FT function parameters to deal with the possible problem of insufficient training data caused by the increased number of model parameters. The effectiveness of the proposed approach is confirmed by evaluation experiments on Finnish Aurora3 database.

Table 1. Summary of comparisons of several methods in terms of word error rate (WER in %), the number of model parameters, and relative error rate reduction (RERR in %) compared to that achieved by the CDHMM baseline system without feature compensation.

Methods	# of Parameters	WER	RERR
Baseline	996D	3.95	-
CDHMM(4)	1316D	3.44	12.91
CDHMM(5)	1636D	3.44	12.91
CMLLR-AT(1)	1036D	3.46	12.41
CMLLR-AT(8)	1316D	3.17	19.75
FT1 (ML, $K = 32$)	1252D	3.33	15.70
FT2 (ML, $K = 32$)	1252D	3.34	15.44
FT3 (ML, $K = 32$)	1316D	3.09	21.77
FT4 (ML, $K = 32$)	1564D	2.92	26.08
FT5 (ML, $K = 32$)	1564D	2.88	27.09
FT6 (ML, $K = 32$)	11236D	2.47	37.47
FT6 (MAP, $K = 32$)	11236D	2.46	37.72
FT6 (ML, $K = 48$)	16356D	2.43	38.48
FT6 (MAP, $K = 48$)	16356D	2.40	39.24

5. REFERENCES

[1] Aurora document AU/217/99, "Availability of Finnish speechdat-car database for ETSI STQ WI008 front-end standardisation," Nokia Nov 1999.

[2] ETSI standard document, "Speech processing, transmission and quality aspects (STQ); distributed speech recognition; advanced front-end feature extraction algorithm; compression algorithms," ETSI ES 202 050 v1.1.1 (2002-10), 2002.

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

[4] H. G. Hirsch and D. Pearce, "The AURORA experimental framework for the performance evaluations of speech recognition systems under noisy conditions," *ISCA ITRW ASR-2000*, Paris, France, 2000, pp.181-188.

[5] Q. Huo and D. Zhu, "A maximum likelihood training approach to irrelevant variability compensation based on piecewise linear transformations," *Proc. Interspeech 2006 – ICSLP*, pp.1129-1132.

[6] O. Siohan, T. A. Myrvoll, and C.-H. Lee, "Structural maximum *a posteriori* linear regression for fast HMM adaptation," *Computer, Speech and Language*, vol. 16, pp.5-24, 2002.

[7] J. Wu and Q. Huo, "An environment compensated minimum classification error training approach based on stochastic vector mapping," *IEEE Trans. on Audio, Speech and Language Processing*, Vol. 14, No. 6, pp.2147-2155, 2006.

[8] J. Wu, Q. Huo, and D. Zhu, "An environment compensated maximum likelihood training approach based on stochastic vector mapping," *Proc. ICASSP 2005*, pp. 429-432.

[9] J. Wu, D. Zhu, and Q. Huo, "A study of minimum classification error training for segmental switching linear Gaussian hidden Markov models," *Proc. ICSLP 2004*, pp.2813-2816.

[10] S. J. Young, *et al.*, *The HTK Book* (for HTK Version 3.3), 2005.

[11] D. Zhu and Q. Huo, "A maximum likelihood approach to unsupervised online adaptation of stochastic vector mapping function for robust speech recognition," *Proc. ICASSP-2007*, pp.IV-773-776.