Flexible Discriminative Training Based On Equal Error Group Scores Obtained From An Error-Indexed Forward-Backward Algorithm

Erik McDermott & Atsushi Nakamura

NTT Communication Science Laboratories,
NTT Corporation, Kyoto-fu 619-0237, Japan

{mcd,ats}@cslab.kecl.ntt.co.jp

Abstract

This article presents a new approach to discriminative training that uses equal error groups of word strings as the unit of weighted error modeling. The proposed approach, Minimum Group Error (MGE), is based on a novel error-indexed Forward-Backward algorithm that can be used to generate group scores efficiently over standard recognition lattices. The approach offers many possibilities for group occupancy scaling, enabling, for instance, the boosting of error groups with low occupancies. Preliminary experiments examined the new approach using both uniformly and non-uniformly scaled group scores. Results for the new approach evaluated on the Corpus of Spontaneous Japanese (CSJ) lecture speech transcription task were compared with results for standard Minimum Classification Error (MCE), Minimum Phone Error (MPE) and Maximum Mutual Information (MMI), in tandem with I-smoothing. It was found that non-uniform scaling of group scores outperformed MPE when no I-smoothing is used.

Index Terms: speech recognition, discriminative training

1. Introduction

In recent years, discriminative training of hidden Markov models (HMMs) has become a standard part of state-of-the-art speech recognition systems. Common approaches to discriminative training include Minimum Phone Error (MPE), Maximum Mutual Information (MMI), and Minimum Classification Error (MCE) [1][2]. While MCE [2] directly addresses the notion of correct vs. incorrect word sequence discrimination, MPE [3] and Minimum Phone Frame Error (MPFE) [4] are solutions for minimizing not just binary classification error but classification error *weighted* by a fine grained error count such as word, phone or phone frame error. The latter family of weighted-error discriminative training methods appear to require the use of "I-smoothing" [3] to yield good results.

The MPE derivative (see Equ. (8)), relies on the occupancy of a particular arc or string to scale the magnitude of the derivative. Strings with low error ("good" strings) often do not have high occupancies; they are dominated by the many "bad" strings in the lattice; as a result MPE derivatives will stay low for those strings. This might be one of the factors behind the need for I-smoothing to, e.g., Maximum Likelihood (ML) statistics, which specifically weights the reference strings.

This point can be illustrated by considering group statistics for a recognition lattice for a given utterance. Using techniques described in this article, probability masses were obtained for groups of strings with equal error (here phone frame error). The resulting occupancies and MPE derivatives for each group are plotted in Fig. 1. One can see that groups with low phone frame
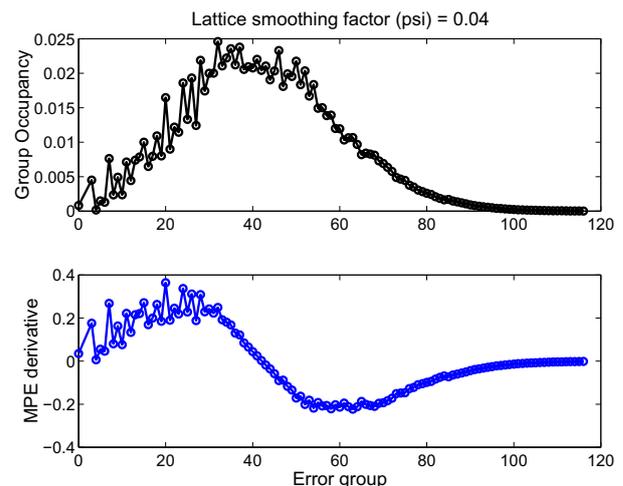

Figure 1: *Group occupancies and MPE derivatives; lattice smoothing factor $\psi = 0.04$.*

error counts have a low overall occupancy. As a result, MPE derivatives for those groups are low too. This suggests that though MPE will work to decrease overall average error, it is not using the good strings to maximal effect. Nor can these low occupancies be remedied by lattice smoothing ("acoustic scaling") very effectively. Too much smoothing (low value for lattice smoothing factor ψ in the framework presented here) results in even greater domination of good strings by the many bad strings in the lattice; too little smoothing (high value for ψ) means too few competitor strings will be in play.

In efficiently propagating average error through the lattice, MPE avoids the need for explicit modeling of total string scores and total string error counts. This article presents an alternative approach to representing weighted word, phone or phone frame error over recognition lattices, the "Minimum Group Error" (MGE) approach, more general than MPE, significantly more computation-intensive, but offering more control over the weight given to strings during training. The central idea is to use a novel error-indexed Forward-Backward algorithm to group all strings with a given error into the same modeling unit, the *equal error group*, that can then be used for discriminative training. Different kinds of scaling can be applied at the group level, boosting strings in (good) low-error groups with low occupancy (and low string cardinality). Several methods for group scaling are proposed and evaluated on the Corpus of Spontaneous Japanese 100k word LVCSR lecture transcription task.

September 22 – 26, Brisbane Australia

2. Generalized Error-Weighted Discriminative Training

2.1. Discriminant functions for Equal Error String Groups

The j-th *equal error group* is defined as the set of all strings S_i that have the same error (frame, phone, word, etc.), j, with reference to the correct reference transcription S_r for a given training token $\mathcal{X}_r$.

The discriminant function for a given error group j, and a given set of model parameters $\mathbf{\Lambda}$ is defined as:

$$G_j(\mathcal{X}_r, \mathbf{\Lambda}) = \sum_{i|\mathcal{A}(S_i,S_r)=j} P(S_i)^{\eta\psi} p_{\mathbf{\Lambda}}(\mathcal{X}_r|S_i)^{\psi}. \quad (1)$$

Here ψ is the lattice smoothing factor ("acoustic scaling factor"), $P(S_i)$ is provided by the Language Model (LM), η is the LM scaling factor, and $\log p_{\mathbf{\Lambda}}(\mathcal{X}_r|S_i)$ is defined in terms of an HMM-based log probability, summed over the Viterbi path for string S_i. This function represents a ψ-smoothed probability mass calculated over a lattice using an error-indexed Forward-Backward algorithm, described in the next section. Following common notation, $\mathcal{A}(S_i, S_r)$ denotes the error (phone, phone frame, word, etc.) between hypothesis S_i and reference S_r.

2.2. Error-indexed Forward-Backward Algorithm

The error-indexed Forward-Backward algorithm performs the usual propagation and merging of forward backward probability masses, but in an error-dependent manner, as illustrated in Fig. 2. Either the last forward score α_j^{final} or the first backward score β_j^{start} can be used for the overall group score $G_j(\mathcal{X}_r, \mathbf{\Lambda})$. Subsequent error-dependent arc occupancy calculation requires score merging; this is the computational bottleneck.

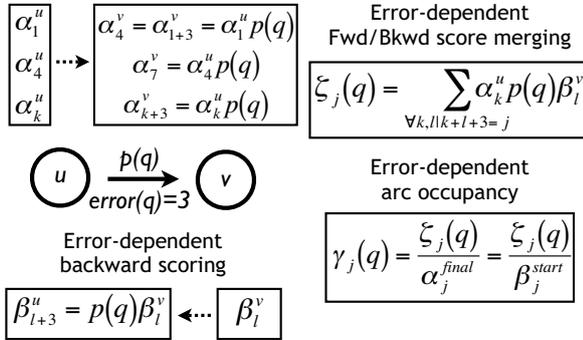**Error-dependent forward scoring**

$$\begin{bmatrix} \alpha_1^u \\ \alpha_4^u \\ \alpha_k^u \end{bmatrix} \cdots \begin{bmatrix} \alpha_4^v = \alpha_{1+3}^v = \alpha_1^u p(q) \\ \alpha_7^v = \alpha_4^u p(q) \\ \alpha_{k+3}^v = \alpha_k^u p(q) \end{bmatrix}$$

Error-dependent Fwd/Bkwd score merging

$$\zeta_j(q) = \sum_{\forall k,l | k+l+3=j} \alpha_k^u p(q) \beta_l^v$$

$$u \xrightarrow[\text{error}(q)=3]{p(q)} v$$

Error-dependent arc occupancy

$$\gamma_j(q) = \frac{\zeta_j(q)}{\alpha_j^{final}} = \frac{\zeta_j(q)}{\beta_j^{start}}$$

Error-dependent backward scoring

$$\beta_{l+3}^u = p(q)\beta_l^v \leftarrow \cdots \quad \beta_l^v$$

Figure 2: *Error-indexed Forward-Backward Algorithm for an arc q connecting nodes u and v and incurring an error of 3. Here α_k^u denotes the forward score up to node u for all partial strings with error k; β_l^v denotes the backward score up to node v for all partial strings with error l.*

2.3. The Minimum Group Error Criterion

The group-based error criterion for a token $\mathcal{X}_r$ is defined as:

$$F_{MGE}(\mathcal{X}_r, \mathbf{\Lambda}) = \sum_j j\ell_j(\mathcal{X}_r, \mathbf{\Lambda}), \quad (2)$$

where j indexes a group of strings S_i which all have the same error $\mathcal{A}(S_i, S_r) = j$, and where $\ell_j(\mathcal{X}_r, \mathbf{\Lambda})$ is an error group
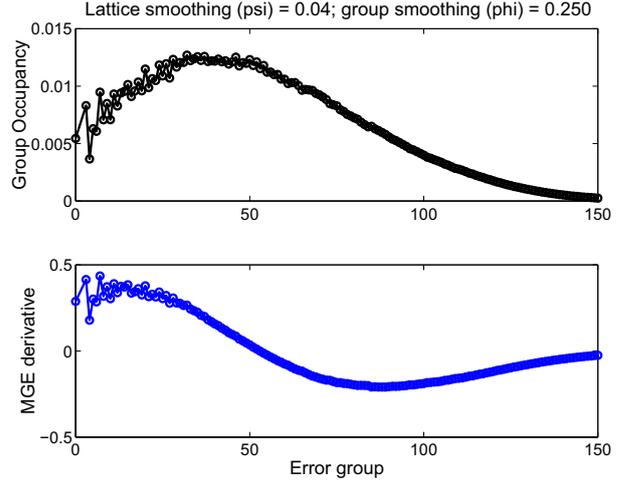

Figure 3: *Group occupancies and MGE derivatives for lattice smoothing factor $\psi = 0.04$ and group scaling factor $\phi = 0.25$.*

selector or occupancy function described in the following. (The occupancy function can also be seen as a general MCE style loss function comparing a particular group with all other groups).

2.4. Scaled error group occupancies

2.4.1. Uniformly scaled group occupancy

A straightforward occupancy function is

$$\ell_j(\mathcal{X}_r, \mathbf{\Lambda}) = \frac{G_j(\mathcal{X}_r, \mathbf{\Lambda})^{\phi}}{\sum_i G_i(\mathcal{X}_r, \mathbf{\Lambda})^{\phi}}, \quad (3)$$

analogous to the use of scaled likelihood ratios to weight errors in standard MPE (Equ. (4)). The difference here is that scaling occurs at both the string-level (*within* an error group, using ψ) and at the group-level, using ϕ. A large value of ϕ will select the group with the largest score. Conversely, small values of ϕ will *flatten* occupancies across groups, raising small occupancies and lowering large occupancies. This can be used to boost the derivatives of strings in groups with low errors.

Comparing to the usual MPE loss function for token $\mathcal{X}_r$,

$$F_{MPE}(\mathcal{X}_r, \mathbf{\Lambda}) = \frac{\sum_i P(S_i)^{\psi\eta} p_{\mathbf{\Lambda}}(\mathcal{X}_r|S_i)^{\psi} \mathcal{A}(S_i, S_r)}{\sum_i P(S_i)^{\psi\eta} p_{\mathbf{\Lambda}}(\mathcal{X}_r|S_i)^{\psi}}, \quad (4)$$

one can see by plugging Equ. (1) and Equ. (3) into Equ. (2) that F_{MPE} is equivalent to F_{MGE} with $\phi = 1$.

2.4.2. Non-uniform group scaling around a target error

It may be desirable to apply *non-uniform* scaling of group scores. In particular, one may want to enhance the importance of low-error groups compared to higher-error groups.

A non-uniform "partitioned" error criterion for a token $\mathcal{X}_r$ can be obtained from Equ. (2) with a different set of occupancy functions $\ell_j(\cdot)$. For an error group whose error j is less than a target error t, a "partitioned" occupancy can be defined as:

$$\ell_j = \left(\frac{G_j^{\phi}}{\sum_{i<t} G_i^{\phi}}\right) \frac{\left(\sum_{i<t} G_i^{\phi}\right)^{\nu}}{\left(\sum_{i<t} G_i^{\phi}\right)^{\nu} + \left(\sum_{i\geq t} G_i^{\phi}\right)^{\nu}}, \quad (5)$$
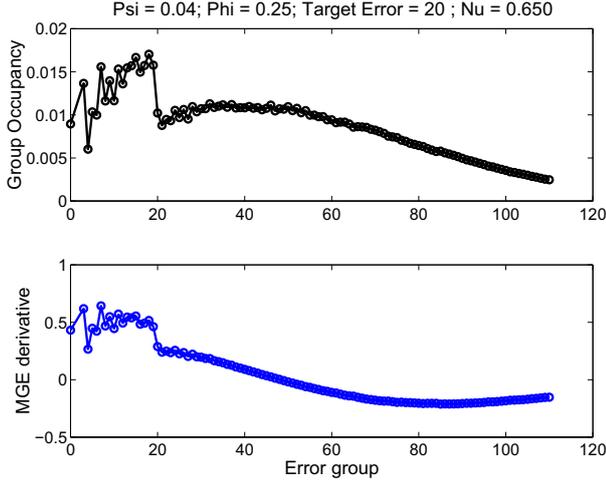
Figure 4: *Group occupancies and MGE derivatives for lattice smoothing $\psi = 0.04$, group scaling factor $\phi = 0.25$, target error = 20, and in/out partition smoothing $\nu = 0.65$.*
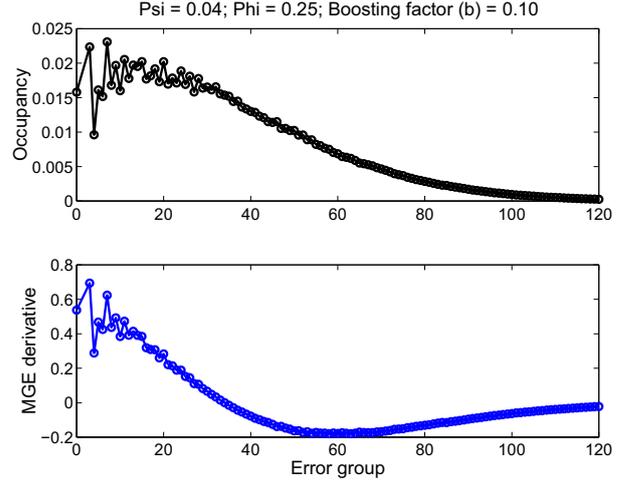
where dependency on $\mathcal{X}_r$ and $\boldsymbol{\Lambda}$ has been dropped. This expression corresponds to a "within-partition" occupancy of G_j within all "acceptable" error groups $i < t$, multiplied by a within vs. total partition occupancy. If $\nu = 1$, the partitioning is irrelevant and Equ. (5) reduces to Equ. (3). The corresponding occupancy for groups $j \geq t$ is defined similarly to Equ. (5), switching sum ranges $i < t$ and $i \geq t$. Setting a fixed target for any training utterance is of course arbitrary; it would be preferable to fix the target as, e.g., a fraction of the utterance length.

2.4.3. Exponential scaling/boosting

Another approach to non-uniform scaling that seems reasonable and less heavy-handed than the partitioning just described is to introduce exponential scaling into error group occupancy (Equ. (3)) using, for instance,

$$\ell_j(\mathcal{X}_r, \boldsymbol{\Lambda}) = \frac{G_j(\mathcal{X}_r, \boldsymbol{\Lambda})^\phi e^{-bj}}{\sum_i G_i(\mathcal{X}_r, \boldsymbol{\Lambda})^\phi e^{-bi}}, \qquad (6)$$

where b is a biasing (or boosting) parameter. This relates to the "boosted" MMI approach proposed in [5]. If no group scaling is applied ($\phi = 1$), such biasing can be accomplished by a simple addition of local arc error to log-likelihoods during Forward-Backward scoring; no explicit group scores are needed. However, group scaling with $\phi < 1$ may amplify the effect of this technique, which might not be sufficient by itself to boost good string scores.

2.5. MGE gradient and optimization

The following derivatives are to be summed over all tokens $\mathcal{X}_r$ in the training set, for subsequent gradient-based or Extended Baum-Welch optimization [1] [2].

2.5.1. Group derivatives

Given Equ. (5), it is easy to find the error group derivatives of the MGE criterion, Equ. (2):

$$\frac{\partial F_{MGE}(\mathcal{X}_r)}{\partial \log G_j(\mathcal{X}_r)} = \phi \nu \ell_j(\mathcal{X}_r) \left(j - F_{MGE}(\mathcal{X}_r) \right), \qquad (7)$$

Figure 5: *Group occupancies and MGE derivatives for lattice smoothing $\psi = 0.04$, group scaling factor $\phi = 0.25$, exponential group boosting factor $b = 0.1$.*
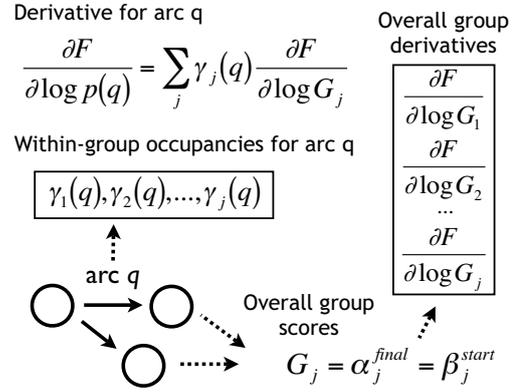

Figure 6: *Propagation of error group derivatives to arc derivatives. Overall error group scores are obtained from the error-indexed forward or backward scores (see Fig. 2).*

(Dependency on $\boldsymbol{\Lambda}$ has now been permanently dropped from the notation). If $\phi = \nu = 1$, the group level picture factorizes to the standard MPE derivative for a lattice arc q:

$$\frac{\partial F_{MPE}(\mathcal{X}_r)}{\partial \log p(q, \mathcal{X}_r)} = \gamma(q, \mathcal{X}_r) \left(c(q, \mathcal{X}_r) - F_{MPE}(\mathcal{X}_r) \right), \quad (8)$$

for an arc q with forward-backward occupancy $\gamma(q, \mathcal{X}_r)$ and average error $c(q, \mathcal{X}_r)$ [3]. The group derivatives for Equ. (6) are straightforward and not detailed here.

2.5.2. Arc derivatives

The group derivatives then have to be propagated over the lattice arcs. For each arc q, the derivative Equ. (7) for each error group j is multiplied by the arc occupancy $\gamma_j(q, \mathcal{X}_r)$ within that group; this in turn is summed over all groups, yielding a total arc derivative:

$$\frac{\partial F_{MGE}(\mathcal{X}_r)}{\partial \log p(q, \mathcal{X}_r)} = \sum_j \frac{\partial F_{MGE}(\mathcal{X}_r)}{\partial \log G_j(\mathcal{X}_r)} \frac{\partial \log G_j(\mathcal{X}_r)}{\partial \log p(q, \mathcal{X}_r)}, \qquad (9)$$

where $\frac{\partial \log G_j(\mathcal{X}_r)}{\partial p(q, \mathcal{X}_r)}$ can be identified with $\gamma_j(q, \mathcal{X}_r)$, the occupancy of arc q within the ψ-smoothed probability mass for the error group j. Rewriting this gives:

$$\frac{\partial F_{MGE}(\mathcal{X}_r)}{\partial \log p(q, \mathcal{X}_r)} = \sum_j \gamma_j(q, \mathcal{X}_r) \frac{\partial F_{MGE}(\mathcal{X}_r)}{\partial \log G_j(\mathcal{X}_r)}. \qquad (10)$$

Fig. 6 illustrates the overall derivative propagation process.

2.6. Visualization of group-level statistics

The effect of uniform group scaling ($\phi = 0.25$) is illustrated in Fig. 3 for the same utterance used in Fig. 1. The group occupancies have been "flattened", raising the low-error group occupancies, which can be seen as desirable, but also raising many high-error group occupancies.

The overall result is that uniform scaling shifts the "reference" average error level around which the MGE derivative (Equ. (7)) is defined (positive or negative) to a higher value, which seems like an undesirable consequence. The effect of the non-uniform scaling proposed in Section 2.4.2, explicitly setting a target error, is shown for the same utterance in Fig. 4. With this method, the average for the utterance gravitates towards that target. The effect of exponential boosting of low error groups (Section 2.4.3) is shown in Fig. 5. Here the low-error groups clearly receive more relative weight (and a larger total derivative) than when using uniform scaling. (Not illustrated here: setting $\phi = 1$ (equivalent to boosted MPE) results in a similar, but more jagged plot, where some low-error groups still do not have large occupancies/derivatives).

3. Experiments

This work extends previous work on the 100k word Corpus of Spontaneous Japanese (CSJ) lecture speech transcription task, with the standard training (230 hours) and testing sets (130 minutes) [6]. A 5000 state, 32 Gaussians/state HMM model trained with ML was used as the baseline for the following experiments.

MMI, MCE, MPE and MGE were evaluated, with and without ML I-smoothing [3] [4]. Experiments also evaluated "boosted" MPE (BMPE), which can be taken either as the MPE version of boosted MMI [5], or as the special case of boosted MGE using Equ. (6) with $\phi = 1$, i.e. with no group-level scaling and hence no need for the error-indexed Forward-Backward algorithm described in this article. In all experiments, the lattice smoothing factor was set at $\psi = 0.04$. The error used in MPE is phone frame error [4]. For all the training configurations examined, batch-mode Rprop [2] was used to optimize the HMM parameters; a simple Rprop-oriented version of I-smoothing was used.

Table 1 shows the test results. The value for τ refers to the degree of I-smoothing used. The Rprop iteration for the model being tested is shown in parentheses. The tag "fe/4" refers to the use of "reduced" frame error, where frame errors on lattice arcs were first pushed, and then divided by 4. This was a simple way of reducing the computation time for error-indexed Forward-Backward in MGE, which then runs around 3-5 times slower than MPE/MMI/MCE. (Future experiments will address the use of phone or word error rather than phone frame error). "MGE-uniform-fe/4" refers to MGE with uniform occupancy (Equ. (3)) and $\phi = 0.125$; "MGE-partition-fe/4" refers to the use of the partitioned, non-uniform occupancy defined in Equ. (5), with $\phi = 0.25$, $\nu = 0.01$, and target=3, heavily favoring errors less than 3 (roughly corresponding to 12 frame errors).

Standard MPE + I-smoothing is the best method tested here, but when no I-smoothing is used, both boosted MPE and MGE with non-uniform partitioning outperform standard MPE. This supports the claims made here concerning MPE's insufficient weighting of low error strings. Future experiments will evaluate boosted MGE (for general values of $\phi \neq 1$).

Table 1: *CSJ word error rates for different approaches*

Method	τ	Word Error (Iteration)
ML	-	21.6 %
MPE	-	19.3 % (4)
MPE	50	18.4 % (8)
MPE	100	18.2 % (11)
MPE-fe/4	-	19.2 % (4)
MPE-fe/4	100	18.5 % (9)
BMPE ($b = 0.1$)	-	19.0 % (4)
MMI	-	18.9 % (4)
MCE	-	18.8 % (4)
MCE	50	18.6 % (8)
MGE-uniform-fe/4	-	19.3 % (6)
MGE-partition-fe/4	-	18.8 % (4)

4. Conclusion

This article presented an original and general framework for error-weighted discriminative training, explicitly based on equal error string group scores, and allowing flexible, non-uniform scaling of error group statistics. Though this approach requires more computational effort, and involves the tuning of additional scaling parameters, it constitutes an important proof-of-concept. It might also lead to significantly simpler variants. This framework may help overcome limitations of current approaches to discriminative training, as suggested by the positive results obtained for non-uniform scaling without I-smoothing.

5. Acknowledgment

6. References

[1] W. Macherey, L. Haferkamp, R. Schlueter, and H. Ney, "Investigations on Error Minimizing Training Criteria for Discriminative Training in Automatic Speech Recognition," in *Proc. Eurospeech*, 2005.

[2] E. McDermott, T. Hazen, J. Le Roux, A. Nakamura, and S. Katagiri, "Discriminative training for large vocabulary speech recognition using Minimum Classification Error," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 1, pp. 203–223, January 2007.

[3] D. Povey and P. Woodland, "Minimum Phone Error and I-smoothing for improved discriminative training," in *Proc. IEEE ICASSP*, vol. 1, 2002, pp. 105–108.

[4] J. Zheng and A. Stolcke, "Improved Discriminative Training Using Phone Lattices," in *Proc. Interspeech*, 2005, pp. 2125–2128.

[5] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted MMI for Model and Feature-Space Discriminative Training," in *Proc. IEEE ICASSP*, 2008, pp. 4057–4060.

[6] E. McDermott and A. Nakamura, "String and Lattice based Discriminative Training for the Corpus of Spontaneous Japanese Lecture Transcription Task," in *Proc. Interspeech*, 2007, pp. 2082–2085.