# Semi-supervised Maximum Mutual Information Training of Deep Neural Network Acoustic Models

*Vimal Manohar*, Daniel Povey*†, Sanjeev Khudanpur*†

*Center for Language and Speech Processing
† Human Language Technology Center of Excellence
The Johns Hopkins University, Baltimore, MD 21218, USA
vmanoha1@jhu.edu, danielpovey@gmail.com, khudanpur@jhu.edu

## Abstract

Maximum Mutual Information (MMI) is a popular discriminative criterion that has been used in supervised training of acoustic models for automatic speech recognition. However, standard discriminative training is very sensitive to the accuracy of the transcription and hence its implementation in a semi-supervised setting requires extensive filtering of data. We will show that if the supervision transcripts are not known, the natural analogue of MMI is to minimize the conditional entropy of the lattice of possible transcripts of the data. This is equivalent to the weighted average of MMI criterion over different reference transcripts, taking those reference transcripts and their weighting from the lattice itself. In this paper we describe experiments where we applied this method to the semi-supervised training of Deep Neural Network acoustic models. In our experimental setup, the proposed method gives up to 0.5% absolute WER improvement over a DNN trained with sMBR only on the transcribed part of the data. This is 37% of the improvement that we would get from doing sMBR training if we had the transcripts for the untranscribed part of the data.

**Index Terms**: Semi-supervised Learning, Lattice Entropy, Deep Neural Network, Acoustic Modeling, Speech Recognition

## 1. Introduction

A discriminative criterion encourages the model to be maximally discriminative of the reference transcript against the competing hypotheses. A number of discriminative criteria such as MMI [1], MCE[2], MPE [3], sMBR [4, 5] and bMMI [6] have been developed and used in HMM-based speech recognition [7, 8, 9, 10].

Of late, there has been an lot of effort devoted to semi-supervised learning due to the availability of large amount of acoustic data and emphasis on speech recognition on low-resource languages. One of the most common appoaches to semi-supervised learning of DNN-based acoustic models is self-training [11, 12, 13] , where a seed system trained with only transcribed data is used to decode the untranscribed data and the predicted hypotheses are selected as the training transcripts, usually based on confidence-based filtering schemes. In [11], lattice posterior probabilities are used as frame-confidence measure for filtering frames during the Cross-Entropy training of DNN. Although that work demonstrates improvements for the Babel [14] languages, we could not replicate that work in our setup. However, we were able to get improvements from untranscribed data for cross-entropy training using a slightly different method which we will described here. We avoid the untranscribed data "polluting" the last layer of the network by giving it a separate final layer, using ideas inspired by multilingual DNN training [15, 16].

Discriminative training is very sensitive to the accuracy of the transcripts [17, 18, 19]. Therefore sequence-discriminative self-training methods do not work well without some form of confidence-based filtering, as used in [20, 21, 17, 22]. However, we show in this paper that by using an alternative objective function, Negative Conditional Entropy (NCE) on the untranscribed portion of the data, we can obtain improvements from untranscribed data without filtering. Entropy minimization has previously been used as an objective for semi-supervised learning in a facial recognition problem [23] and for sequence-discriminative training of GMM acoustic models for speech recognitinon [24]. In this paper, we apply the idea to semi-supervised training of DNN acoustic models for speech recognition. We also introduce a multilingual-inspired training architecture for semi-supervised training, which is more effective than the normal approach with or without confidence filtering.

This paper is organized as follows. Section 2 introduces the problem of semi-supervised training of acoustic models and describes the proposed sequence-discriminative training method for DNN acoustic models. Section 3 describes the experiments we conducted. Section 4 discusses the results of the experiments. Section 5 presents the conclusions.

## 2. Semi-supervised Training via Lattice Entropy

In the following sections, we show why minimization of lattice entropy is the natural extension of the MMI objective to the semi-supervised setting (Section 2.1); and we describe efficient ways to compute the objective and its gradients using lattices (Section 2.2). We then describe how these gradients are used to update the parameters of the DNN acoustic models in Section 2.3 and propose a multilingual architecture for semi-supervised training of DNN in Section 2.4.

### 2.1. Conditional Maximum Likelihood training and lattice entropy

The Condition Maximum Likelihood (CML) objective [25] is the conditional log-likelihood of the transcript $W$ given the acoustic features $\mathbf{O}$, summed over the training examples. For historical reasons this is known in the speech recognition community as Maximum Mutual Information (MMI) estimation, or

This work was partially supported by NSF Grant No IIS 0963898 and DARPA BOLT Contract No HR0011-12-C-0015.

MMIE [1]:

$$\mathcal{F}_{\text{MMI}}(\lambda) = \sum_r \log \mathbb{P}(W^{(r)} \mid \mathbf{O}^{(r)}; \lambda) \tag{1}$$

where the index $r$ ranges over all training utterances, and $\lambda$ is the parameters of the model. In the semi-supervised learning setting, we propose to take a weighted average of the above expression for all possible reference transcripts $W^{(r)}$, weighted by their probability in the lattice:

$$\begin{aligned}\mathcal{F}_{\text{NCE}}(\lambda) &\triangleq \sum_r \sum_W \mathbb{P}(W \mid \mathbf{O}^{(r)}; \lambda) \log \mathbb{P}(W \mid \mathbf{O}^{(r)}; \lambda) \\ &= -\sum_r \mathbb{H}(W \mid \mathbf{O}^{(r)}; \lambda),\end{aligned} \tag{2}$$

where $\mathbb{H}(W \mid \mathbf{O}; \lambda)$ is the conditional entropy of the transcript $W$ given the acoustic feature sequence $\mathbf{O}^{(r)}$ and the acoustic model parameters $\lambda$. This criterion was defined as "Negative Conditional Entropy (NCE)" in [24].

### 2.2. Lattice Entropy Computation

Lattice-based methods for discriminative training have been developed for many discriminative objective functions including MMI [7, 8]. The conditional entropy in (2) and its gradients can be computed using an algorithm reminiscent of the forward-backward algorithm. Our approach for computing the lattice entropy and its derivatives is based on the ideas in [26], but we present it in a form that does not require the concept of a semiring.

We generate lattices in the WFST framework using the "exact lattice" procedure. [27]. Each path $\pi$ in such a lattice $\mathcal{L}$ represents the best (lowest-cost) state-level alignment of the utterance for a distinct word sequence. Each arc $a$ in the lattice has an associated probability score $p_a$, which is a suitably weighted combination of the acoustic likelihood, language model probability and transition and pronunciation probabilities (we use an acoustic scale of $\kappa = 0.1$ throughout). Each path $\pi$ through the lattice has a probability score $P(\pi) = \sum_{a \in \pi} p_a$.

The entropy of the lattice $H_{\mathcal{L}} = \mathbb{H}(W \mid \mathbf{O}; \lambda)$ can be computed as follows:

$$\begin{aligned}H_{\mathcal{L}} &= -\sum_{\pi \in \mathcal{L}} \frac{P(\pi)}{Z} \log \frac{P(\pi)}{Z} \\ &= \log Z - \frac{\bar{r}}{Z}\end{aligned} \tag{3}$$

where $Z = \sum_{\pi \in \mathcal{L}} P(\pi)$ and $\bar{r} = \sum_{\pi \in \mathcal{L}} P(\pi) \log P(\pi)$. Its gradient wrt to $p_a$ can be computed as:

$$\frac{\partial H_{\mathcal{L}}}{\partial p_a} = \frac{1}{Z}\frac{\partial Z}{\partial p_a} - \frac{1}{Z}\frac{\partial \bar{r}}{\partial p_a} + \frac{\bar{r}}{Z^2}\frac{\partial Z}{\partial p_a}. \tag{4}$$

Algorithm 1 shows how to compute these quantities efficiently over a lattice. The $\alpha_p$ and $\alpha_r$ quantities correspond to the $Z$ and $\bar{r}$ quantities for sub-lattices starting at the start node and ending at each node, and the $\beta_p$ and $\beta_r$ are the same thing for sub-lattices starting at each node and ending at the end node of the lattice. Due to the limited dynamic range of floating point, the $\alpha_p$ and $\beta_p$ must be stored in log form; and $\alpha_r$ and $\beta_r$, which may be positive or negative, must be stored in log form with their sign stored separately. To explain the notation: $s(a)$ refers to the starting node of arc $a$, $e(a)$ refers to the end node of arc $a$, $\text{pre}(n)$ refers at arcs ending in node $n$, $\text{post}(n)$ refers to arcs following node $n$ and $r_a \triangleq p_a \log p_a$.

**Algorithm 1** Forward-Backward Algorithm over lattice $\mathcal{L}$

**Require:** $\mathcal{L}$ is topologically sorted.
**Require:** $\mathcal{L}$ has a single start-state and a single end-state.
**procedure** FORWARD-BACKWARD($\mathcal{L}$)
  $N \leftarrow |\mathcal{L}|$
  Initialize $\alpha_p(1 \ldots N)$, $\alpha_r(1 \ldots N)$, $\beta_p(1 \ldots N)$, $\beta_r(1 \ldots N)$, $Z$, $\bar{r}$ to all 0s
  $\alpha_p(1) \leftarrow 1, \beta_p(N) \leftarrow 1$
  **for** $n \leftarrow 2$ **to** $N$ **do**
    $\alpha_p(n) \leftarrow \sum_{a \in \text{pre}(n)} \alpha_p(s(a)) p_a$
    $\alpha_r(n) \leftarrow \sum_{a \in \text{pre}(n)} \alpha_p(s(a)) r_a + p_a \alpha_r(s(a))$
  **end for**
  $Z \leftarrow \alpha_p(N), \bar{r} \leftarrow \alpha_r(N)$
  **for** $n \leftarrow N-1$ **to** $1$ **do**
    $\beta_p(n) \leftarrow \sum_{a \in \text{post}(n)} \beta_p(e(a)) p_a$
    $\beta_r(n) \leftarrow \sum_{a \in \text{post}(n)} \beta_p(e(a)) r_a + p_a \beta_r(e(a))$
  **end for**
  **for each** arc $a \in \mathcal{L}$ **do**
    $\partial Z / \partial p_a \leftarrow \alpha_p(s(a)) \beta_p(e(a))$
    $\partial \bar{r} / \partial p_a \leftarrow \alpha_r(s(a)) \beta_p(e(a)) + \alpha_p(s(a)) \beta_r(e(a)) + \alpha_p(s(a)) \beta_p(e(a)) (1 + \log p_a)$
  **end for**
**end procedure**

### 2.3. Optimization of DNN Acoustic Model parameters

Since the objective function value depends on the neural network weights only through the DNN outputs $y_t(j)$, it is enough to find the gradients of the objective function with respect to the DNN outputs. The rest of this section describes this process.

In a HMM-DNN hybrid system, the DNN is used to provide the emission probability or the pseudo-likelihood [28] of an acoustic feature vector $\mathbf{o}_t$ at time $t$ from a pdf $j$:

$$p(\mathbf{o}_t \mid j) = \frac{y_t(j)}{P(j)}, \tag{5}$$

where $y_t(j) = P(j \mid \mathbf{o}_t)$ is the DNN output at $j^{th}$ node of output layer and $P(j)$ is the prior probability of pdf $j$.

To compute the gradients of the objective function w.r.t. the DNN outputs, we define an "NCE posterior" for each arc $a$ of the lattice as $\gamma_a \triangleq \frac{\partial H_{\mathcal{L}}}{\partial \log p_a}$, which can be computed using (4). The derivative w.r.t. the log DNN-outputs $\log p(\mathbf{o}_t \mid j)$, is just the sum of the $\gamma_a$ quantities over all arcs in the lattice at time $t$ that have the pdf $j$. We call these quantities state-level "NCE posteriors" $\gamma_t^{\text{NCE}}(j)$; they are analogous to "MBR posteriors" [29][1]. The derivative w.r.t. the DNN outputs can then be computed as:

$$\frac{\partial \mathcal{F}_{\text{NCE}}}{\partial y_t^{(r)}(j)} = \frac{\gamma_t^{\text{NCE}}(j)}{y_t^{(r)}(j)}, \tag{6}$$

These derivatives are backpropagated to find the gradients w.r.t. all the weights in the neural network, and the weights are updated using stochastic gradient descent (SGD). The randomization for SGD is performed at the part-of-lattice level: where we find "pinch points" in the lattice to split them up

[1] Actually, what we said about the derivative being the sum of selected $\gamma_a$ quantities is not quite true. The factor $\kappa$ should appear here, and we ignore it. This is just for consistency with prior work in discriminative training [9], in which that factor is ignored. It is absorbed into the learning rate

into the smallest possible pieces, discarding parts of lattices that would only produce zero gradients. As for our implementation of cross-entropy training, we update with a Natural Gradient extension of Stochastic Gradient Descent (NG-SGD) and parallelize over multiple machines via model averaging [30].

The prior probability $P(j)$ in (5) is usually computed from alignments [28]. Here, we use an alternate method that computes the priors by marginalization of DNN posteriors over all acoustic feature vectors, assuming they are drawn from an empirical distribution:

$$P(j) = \frac{1}{N} \sum_{i=1}^{N} p(j \mid \mathbf{o}_i). \quad (7)$$

We found this to give better WERs than the usual approach.

### 2.4. Multilingual training architecture

In the multilingual training architecture [15, 16], two (or more) DNNs are trained sharing all the layers except the last one. We can use this architecture for semi-supervised training by viewing the untranscribed data as the "second language". One of the final layers is used for transcribed training examples, and the other is used for untranscribed training examples. At the end of training, we discard the final layer that was trained on the untranscibed examples. The gradients arising from the untranscibed data can be scaled down to give that data less weight in the optimization. This architecture even allows a different context-dependency trees for the different final layers ; but this is not considered in this paper. In addition, filtering of untranscribed data frames using say, frame-level confidence [11], can be incorporated easily.

## 3. Experiments

In this paper, we report experiments on a subset of the Fisher English corpus [31] and several Babel languages in the *LimitedLP* condition. We compare our method with several baseline systems. These include cross-entropy and sMBR trained DNNs with only the transcribed data, in addition to self-training methods. All experiments[2] are conducted using Kaldi Speech Recognition toolkit [32].

### 3.1. Experimental Setup

The Fisher English corpus has a total of 1600 hours of telephone speech. The first 5000 utterances (about 3.3 hours) in the corpus was selected as the *dev* set for tuning hyperparameters and the next 5000 utterances (about 3.2 hours) was selected as the *test* set for evaluation. Out of remaining data, 100 hours was selected as transcribed data and the remaining part was selected as untranscribed data by ignoring the corresponding transcripts. In this paper, we show results with only a 250 hour subset of untranscribed data.

The Babel languages under the *LimitedLP* condition have 10 hours of transcribed data and 50-65 hours of untranscribed data after automatic segmentation. In this paper, we show results on four of the Babel languages – Assamese, Bengali, Zulu and Tamil. We use the fixed lexicon provided under the *LimitedLP* condition. We evaluate our systems on the 10 hour *dev10h* set, while tuning on a 2 hour subset *dev2h*. But we don't tune hyperparameters for different languages separately.

[2]The recipe used for these experiments can be found at `https://github.com/vimal-manohar91/kaldi-unsupervised/commit/137f0f12726a382529552ee68d75092f939413c3`

The language models used for the experiments are trained only on the transcripts of respective transcribed data. For sMBR training, we use a weak language model (unigram) to increase the number of alternative hypotheses for discrimination. But for NCE training, we use a trigram language model to produce a compact lattice with only the most likely hypotheses. This is in line with the empirical results in [33] that show that a stronger model is better for semi-supervised learning. The decoding of the test sets is also done using the same trigram language model.

### 3.2. System Description

All our experiments use the $p$-norm DNN with $p = 2$ and the same basic architecture as in [34]. For Fisher, the DNN had 4 hidden layers of p-norm nonlinearity with input and output dimensions of 3000 and 300 respectively. For Babel, the DNN had 3 hidden p-norm layers with input and output dimensions 2000 and 200 respectively. The features used are the Type IV acoustic features defined in [35], i.e. fMLLR features spliced over $\pm 4$ frames and then decorrelated and globally mean-subtracted with a matrix transform. For the experiments on Fisher English, we use MFCC as the base features. For the Babel experiments, we use PLP as the base features, but we additionally append pitch features [36]. The neural networks are trained using Natural Gradient SGD [30]. The alignments and context-dependency tree for the Cross-Entropy DNN training are obtained using a HMM-GMM model trained using only transcribed data.

In Fisher, we found the prior adjustment (7) to improve performance of the DNNs over the traditional method of prior estimation from alignments [28]. We used a subset (3 hours) of transcribed data for prior adjustment. In Babel, the sMBR objective is modified to penalize insertions [37].

#### *3.2.1. Supervised baseline systems*

The baseline DNN system *nnet2_CE* is trained with Cross-Entropy as objective for 20 epochs with an exponentially decreasing learning rate. The baseline discriminative system *nnet2_sMBR* is initialized with *nnet2_CE* and trained with sMBR as objective for 4 epochs.

#### *3.2.2. Self-training systems*

For the self-training systems, the untranscribed data was decoded using the *nnet2_CE* system and the best paths through the lattices were chosen as the transcripts. The system *nnet2_CE_semisup* has the same architecture as *nnet2_CE* and is trained from scratch using Cross-Entropy as objective with transcribed and untranscribed data frames combined together. The system *nnet2_CE_semisup:0.8* is as *nnet2_CE_semisup* but only selecting frames with confidences [11] greater than 0.8.

The systems *multilang2_CE* and *multilang2_CE:0.8* use the multilingual architecture (Section 2.4). The DNN is initialized from from a partially trained (after the mix-up stage) *nnet2_CE* neural network. The final layer corresponding to the untranscribed data is initialized randomly. The system is trained with an exponentially decreasing learning rate for 20 epochs as measured on transcribed data[3]. In *multilang2_CE:0.8*, frame-confidence-based selection is additionally done.

The system *multilang2_sMBR* is the sMBR self-training system in the multilingual architecture. The DNN is initialized

[3] It roughly corresponds to the same number of epochs on transcribed and untranscribed datasets because the number of parallel jobs for each dataset is varied in proportion to the amount of data available.

with the final *nnet2_CE* model; the last layer is cloned to make separate copies for transcribed and untranscribed training examples. The system is trained with sMBR criterion for 4 epochs[3] with a fixed learning rate. The learning rate on untranscibed data was reduced by a factor of 10 in order to give less weight to the corresponding gradients. Using equal learning rate for both datasets worsened the results.

#### 3.2.3. *Proposed system*

The system *multinnet2_sMBR+NCE* uses the multilingual architecture just like *multilang2_nnet2_sMBR*; it is trained using sMBR objective on the transcribed data, but $\mathcal{F}_{\text{NCE}}$ as objective on the untranscribed data. The training is done for 4 epochs[3] with a fixed learning rate. The learning rate on untranscribed data was reduced by a factor of 3. The resulting parameter updates using untranscribed data were found to be about 10 times smaller than those using transcribed data; this is because of NCE gradients being smaller than sMBR gradients in general.

We also show an oracle system *nnet2_sMBR_oracle* as an upper bound on the performance of the semi-supervised systems. This oracle system is similar to the *nnet2_sMBR* system, but does sMBR training using true transcripts of the untranscribed data. For the purpose of comparision with *multinnet2_sMBR+NCE*, the language model for the oracle system is trained using only the *LimitedLP* data.

## 4. Results and Discussion

The results on Fisher English with 250 hours of untranscribed data are given in Table 1. The self-learning CE system *nnet2_CE_semisup* has a WER worse than the baseline CE system *nnet2_CE* even with frame-filtering. This might be because we did not add multiple copies of supervised data as suggested in [11]. In contrast, self-learning in the multilingual architecture *multilang2_CE* gives nearly 1.4% absolute improvement over supervised CE system *nnet2_CE* with and without frame filtering. This suggests that the multilingual architecture is an effective framework for doing semi-supervised training of DNN.

Table 1: *WER (%) results on Fisher English (100 hrs transcribed + 250 hrs untranscribed) for DNN acoustic models*

| System | *dev* | *test* |
|---|---|---|
| *nnet2_CE* | 31.98 | 31.18 |
| *nnet2_sMBR* | 29.58 | 28.49 |
| *nnet2_CE_semisup* | 32.40 | – |
| *nnet2_CE_semisup:0.8* | 32.46 | – |
| *multilang2_CE* | 30.61 | 29.84 |
| *multilang2_CE:0.8* | 30.53 | 29.81 |
| *multilang2_sMBR* | 29.87 | 28.77 |
| *multinnet2_sMBR+NCE* | 29.44 | 28.11 |
| *nnet2_sMBR_oracle* | 28.50 | 27.46 |

Table 2: *WER (%) results on Fisher English (100 hrs transcribed + 250 hrs untranscribed) for GMM acoustic models*

| System | *dev* | *test* |
|---|---|---|
| *gmm_ML* | 39.58 | 38.33 |
| *gmm_MMI* | 38.97 | 36.88 |
| *gmm_MMI+NCE* | 38.15 | 35.84 |
| *gmm_ML_oracle* | 38.67 | 37.33 |
| *gmm_MMI_oracle* | 37.47 | 35.47 |

But even the best CE system (*multilang2_CE:0.8*) is more than 1% worse than the system with supervised discriminative training, *nnet2_sMBR*. This shows that in order to compete with a discriminatively trained system, the semi-supervised learning must involve discriminative training. The discriminatively self-trained system, *multilang2_sMBR*, in the multilingual architecture is shown to be slightly worse than the supervised baseline *nnet2_sMBR* even though the learning rate of $\text{nnet}_{\text{U}}$ was reduced by a factor of 10. This suggests that discriminative self-training might require filtering of untranscribed data as was suggested in several works in the literature.

On the other hand, our proposed system *multinnet2_sMBR+NCE* gives 0.16% and 0.38% absolute improvements on *dev* and *test* sets respectively without any explicit filtering of data. Comparing with the oracle system results (*nnet2_sMBR_oracle*), we see that these results of the proposed system correspond respectively to a recovery of 15% and 37% of the possible improvements if we had the true transcripts. We believe that the loss in accuary is due to a combination of inaccuracy in the decoding, mismatch in features because of using unsupervised speaker adaptation for untranscribed data and the choice of MMI as the criterion over sMBR.

Table 2 presents analogous results with GMM acoustic models This demonstrates that the method is not restricted to only DNN acoustic models.

We got similar WER improvements from 0.1% absolute on Zulu to 0.6% absolute on Assamese. Improvements in Bengali and Tamil are also in this range as detailed in Table 3.

Table 3: *WER (%) results on Babel*

| Language | System | *dev2h* | *dev10h* |
|---|---|---|---|
| Assamese | *nnet2_sMBR* | 63.9 | 62.2 |
| Assamese | *multinnet2_sMBR+NCE* | 63.4 | 61.6 |
| Bengali | *nnet2_sMBR* | 66.3 | 64.1 |
| Bengali | *multinnet2_sMBR+NCE* | 65.8 | 63.8 |
| Zulu | *nnet2_sMBR* | 65.9 | 67.3 |
| Zulu | *multinnet2_sMBR+NCE* | 65.7 | 67.2 |
| Tamil | *nnet2_sMBR* | 76.3 | 74.8 |
| Tamil | *multinnet2_sMBR+NCE* | 76.1 | 74.6 |

## 5. Conclusions

In this paper, we proposed a semi-supervised sequence-discriminative training method for DNN acoustic models using conditional entropy as the criterion in a multilingual-inspired DNN architecture. We show through experiments on Fisher English and Babel that the method gives improvements over sequence-discriminatively trained supervised DNN systems. Without needing explict filtering of data, the method can also outperform self-training methods. On Fisher English, the proposed method is shown to recover 37% of the WER improvement possible if the transcripts were available for the untranscribed data.

We also described a multilingual-inspired method of semi-supervised training, where the untranscribed portion of the data has its own version of the final layer, not shared with the final layer used for the supervised part, and which is discarded after training. We found this to work better than simply combining transcribed and untranscribed data, whether or not confidence filtering was used.

## 6. Acknowledgements

This work was partially supported by NSF Grant No IIS 0963898 and DARPA BOLT Contract No HR0011-12-C-0015.

# 7. References


[1] L. Bahl, P. Brown, P. de Souza, and R. Mercer, "Maximum Mutual Information Estimation of Hidden Markov Model parameters for Speech Recognition," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP '86.*, vol. 11, Apr 1986, pp. 49–52.

[2] B.-H. Juang, W. Hou, and C.-H. Lee, "Minimum Classification Error rate methods for Speech Recognition," *Speech and Audio Processing, IEEE Transactions on*, vol. 5, no. 3, pp. 257–265, 1997.

[3] D. Povey. and P. C. Woodland, "Minimum Phone Error and I-smoothing for Improved Discriminative Training," in *ICASSP*, 2002.

[4] J. Kaiser, B. Horvat, and Z. Kacic, "A novel loss function for the overall risk criterion based discriminative training of HMM models," in *Sixth International Conference on Spoken Language Processing*, 2000.

[5] M. Gibson and T. Hain, "Hypothesis spaces for minimum bayes risk training in large vocabulary speech recognition." in *INTERSPEECH*. Citeseer, 2006.

[6] D. Povey, D. Kanevsky *et al.*, "Boosted MMI for Feature and Model Space Discriminative Training," in *ICASSP*, 2008.

[7] V. Valtchev, J. Odell, P. Woodland, and S. Young, "Lattice-based discriminative training for large vocabulary speech recognition," in *Acoustics, Speech, and Signal Processing, 1996. ICASSP-96. Conference Proceedings., 1996 IEEE International Conference on*, vol. 2, May 1996, pp. 605–608 vol. 2.

[8] V. Valtchev, J. Odell, P. C. Woodland, and S. J. Young, "MMIE training of large vocabulary recognition systems," *Speech Communication*, vol. 22, no. 4, pp. 303–314, 1997.

[9] D. Povey, "Discriminative Training for Large Voculabulary Speech Recognition," Ph.D. dissertation, Cambridge University, 2004.

[10] P. C. Woodland and D. Povey, "Large scale discriminative training of Hidden Markov Models for Speech Recognition," *Computer Speech & Language*, vol. 16, no. 1, pp. 25–47, 2002.

[11] K. Vesely, M. Hannemann, and L. Burget, "Semi-supervised training of Deep Neural Networks," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*. IEEE, 2013, pp. 267–272.

[12] F. Grezl and M. Karafiat, "Semi-supervised bootstrapping approach for neural network feature extractor training," in *Automatic Speech Recognition and Understanding (ASRU), 2013 IEEE Workshop on*, Dec 2013, pp. 470–475.

[13] P. Zhang, Y. Liu, and T. Hain, "Semi-supervised dnn training in meeting recognition," in *Proceedings of.* Sheffield, 2014.

[14] M. Harper, "IARPA Babel Program," 2014.

[15] G. Heigold, V. Vanhoucke, A. Senior, P. Nguyen, M. Ranzato, M. Devin, and J. Dean, "Multilingual acoustic models using distributed deep neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, May 2013, pp. 8619–8623.

[16] J.-T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, May 2013, pp. 7304–7308.

[17] L. Mathias, G. Yegnanarayanan, and J. Fritsch, "Discriminative training of acoustic models applied to domains with unreliable transcripts." in *ICASSP (1)*, 2005, pp. 109–112.

[18] K. Yu, M. Gales, L. Wang, and P. C. Woodland, "Unsupervised training and directed manual transcription for LVCSR," *Speech Communication*, vol. 52, no. 7, pp. 652–663, 2010.

[19] X. Cui, J. Huang, and J.-T. Chien, "Multi-view and multi-objective semi-supervised learning for large vocabulary continuous speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*, May 2011, pp. 4668–4671.

[20] L. Lamel, J.-L. Gauvain, and G. Adda, "Lightly supervised and unsupervised acoustic model training," *Computer Speech & Language*, vol. 16, no. 1, pp. 115–129, 2002.

[21] H. Y. Chan and P. C. Woodland, "Improving broadcast news transcription by lightly supervised discriminative training," in *IN PROC. IEEE INT. CONF. ACOUST., SPEECH, SIGNAL PROCESS*, 2004.

[22] S.-H. Liu, F.-H. Chu, S.-H. Lin, and B. Chen, "Investigating data selection for minimum phone error training of acoustic models," in *Multimedia and Expo, 2007 IEEE International Conference on*, July 2007, pp. 348–351.

[23] Y. Grandvalet and Y. Bengio, "Semi-supervised learning by entropy minimization," in *Advances in Neural Information Processing Systems 17*, L. K. Saul, Y. Weiss, and L. Bottou, Eds. Cambridge, MA: MIT Press, 2005, pp. 529–536.

[24] J.-T. Huang and M. Hasegawa-Johnson, "Semi-supervised training of gaussian mixture models by conditional entropy minimization," *Optimization*, vol. 4, p. 5, 2010.

[25] A. Nadas, "A decision theoretic formulation of a training problem in speech recognition and a comparison of training by unconditional versus conditional maximum likelihood," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 31, no. 4, pp. 814–817, 1983.

[26] Z. Li and J. Eisner, "First-and second-order expectation semirings with applications to minimum-risk training on translation forests," in *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing: Volume 1-Volume 1*. Association for Computational Linguistics, 2009, pp. 40–51.

[27] D. Povey *et al.*, "Generating exact lattices in the WFST framework," in *Proceedings of 2012 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE Signal Processing Society, 2012, pp. 4213–4216.

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

[29] K. Veselỳ, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks." in *INTERSPEECH*, 2013, pp. 2345–2349.

[30] D. Povey, X. Zhang, and S. Khudanpur, "Parallel training of deep neural networks with natural gradient and parameter averaging," *arXiv preprint arXiv:1410.7455*, 2014.

[31] C. Cieri, D. Miller, and K. Walker, "The fisher corpus: a resource for the next generations of speech-to-text." in *LREC*, vol. 4, 2004, pp. 69–71.

[32] D. Povey, A. Ghoshal *et al.*, "The Kaldi Speech Recognition Toolkit," in *Proc. ASRU*, 2011.

[33] J. T. Huang, "Semi-supervised learning for acoustic and prosodic modeling in speech applications," Ph.D. dissertation, 2012.

[34] X. Zhang, J. Trmal, D. Povey, and S. Khudanpur, "Improving Deep Neural Network Acoustic Models using Generalized Maxout Networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*, May 2014, pp. 215–219.

[35] P. S. Rath, D. Povey, K. Veselỳ, and J. Cernockỳ, "Improved Feature Processing for Deep Neural Networks," in *Proceedings of Interspeech 2013*, no. 8. International Speech Communication Association, 2013, pp. 109–113.

[36] P. Ghahremani, B. BabaAli, D. Povey, K. Riedhammer, J. Trmal, and S. Khudanpur, "A Pitch Extraction Algorithm tuned for Automatic Speech Recognition," in *ICASSP*, 2014.

[37] V. Peddinti, G. Chen, D. Povey, and S. Khudanpur, "Reverberation robust acoustic modeling using i-vectors with time delay neural networks," *submitted to Interspeech*, 2015. [Online]. Available: http://speak.clsp.jhu.edu/uploads/publications/papers/1049_pdf.pdf