ADAPTIVE BOOSTED NON-UNIFORM MCE FOR KEYWORD SPOTTING ON SPONTANEOUS SPEECH

Chao Weng, Biing-Hwang (Fred) Juang

Center for Signal and Image Processing, Georgia Institute of Technology, Atlanta, USA
{chao.weng,juang}@ece.gatech.edu

ABSTRACT

In this work, we present a complete framework of discriminative training using non-uniform criteria for keyword spotting, adaptive boosted non-uniform minimum classification error (MCE) for keyword spotting on spontaneous speech. To further boost the spotting performance and tackle the potential issue of over-training in the non-uniform MCE proposed in our prior work, we make two improvements to the fundamental MCE optimization procedure. Furthermore, motivated by AdaBoost, we introduce an adaptive scheme to embed error cost functions together with model combinations during the decoding stage. The proposed framework is comprehensively validated on two challenging large-scale spontaneous conversational telephone speech (CTS) tasks in different languages (English and Mandarin) and the experimental results show it can achieve significant and consistent figure of merit (FOM) gains over both ML and discriminatively trained systems.

Index Terms— discriminative training, keyword spotting, MCE, non-uniform criteria, WFST

1. INTRODUCTION

Keyword spotting deals with the detection of a set of keywords given the speech utterances. This technique becomes crucial for automatic speech recognition (ASR) when it is fairly intractable to fully transcribe the spoken words in some challenging large vocabulary continuous speech recognition (LVCSR) tasks, e.g., spontaneous conversational telephone speech (CTS), where it may be sufficient to extract underlying messages from only certain significant keywords. In our prior work [1], we generalized keyword spotting as a non-uniform error ASR problem, successfully applied our discriminative training (DT) algorithms using non-uniform criteria to it and proposed *non-uniform MCE*. The main idea is to adapt the fundamental DT criteria in a cost-sensitive way which leads optimizations to place emphasis on keywords. It was shown that even with quite simple error cost function, we could achieve considerable spotting performance gains.

In this work, to further boost the spotting performance and tackle the potential issue of over-training in the non-uniform MCE, we present a complete framework of DT using non-uniform criteria for keyword spotting, adaptive boosted non-uniform minimum classification error (MCE), and comprehensively validate it on two challenging large-scale spontaneous CTS tasks in different languages (English and Mandarin). Specifically, we first make two improvements to the fundamental MCE optimization procedure as in Boosted MMI [2], i.e., canceling any shared part of the numerator and denominator statistics on each frame and replacing I-smoothing to ML estimate with one to the previous iteration's value. With the two improvements, our MCE implementation in the weighted finite state transducer (WFST) framework as in [1] can obtain comparable word accuracy gains with both Boosted MMI and MPE [3]. On top of this boosted MCE and motivated by AdaBoost [4], we introduce an adaptive scheme to embed error cost functions, namely the adaptive adjustment of the error cost function depending on whether the current frame is classified correctly or not, together with model combinations during the decoding procedure. Evaluated on two large scale CTS tasks, the adaptive boosted non-uniform MCE achieves significant spotting performance gains consistently over both ML and discriminatively trained systems. The remainder of this paper is organized as follows: Section 2 gives a review of the non-uniform MCE for keywords spotting, which serves as the background of this work. The detailed algorithms and implementations of the adaptive boosted non-uniform MCE will be described in Section 3. We report experimental results in Section 4, draw conclusions and make a brief discussion on how the paper's contributions are related to prior work in Section 5.

2. NON-UNIFORM MCE FOR KEYWORD SPOTTING

General MCE training [5] is a DT method for pattern recognition with the aim of direct minimization of the empirical error rate. In speech recognition scenario, let $X_r, r = 1, \cdots, R$, be the utterances in the training set, W_r be the label word transcription for X_r and W be the certain selected hypothesis events. The discriminant function for a hypothesis W is defined as,

$$g_\Lambda(X_r, W) = \log P_\Lambda^\alpha(X_r|W)P_\Lambda^\beta(W). \tag{1}$$

Thus the misclassification measure takes the following form,

$$d_\Lambda(X_r) = -g_\Lambda(X_r, W_r) + \log \left[\frac{1}{|W|} \sum_{W \neq W_r} \exp[g_\Lambda(X_r, W)]\eta \right]^{\frac{1}{\eta}}. \tag{2}$$

$P_\Lambda(X_r|W)$,$P_\Lambda(W)$ denote the acoustic and language models, and α and β are scaling factors respectively. Finally, with proper smoothing using the sigmoid function, the objective function is formulated as,

$$\mathcal{L}_\Lambda = \sum_{r=1}^{R} \ell(d_\Lambda(X_r)), \tag{3}$$

where $\ell(d) = \frac{1}{1+\exp(-\gamma d+\theta)}$. Based on Eq.(3), the objective function of non-uniform MCE can be written as,

$$\mathcal{L}_\Lambda = \sum_{r=1}^{R} \epsilon_r(t)\ell(d_\Lambda(X_r)), \tag{4}$$

where $\epsilon_r(t)$ is the error cost function, which defines error cost over time (frames) through the rth utterance. To gain an insight into the

non-uniform MCE objective function, we write down its gradients as,

$$\nabla \mathcal{L}_\Lambda = \sum_{r=1}^{R}\sum_{t=1}^{T_r} \ell(d_\Lambda(X_r))[1-\ell(d_\Lambda(X_r))] \epsilon_r(t)(-\gamma_{jm}^{W_r}(t)+\gamma_{jm}^{W\neq W_r}(t))\frac{\partial \log \mathcal{N}_{jm}(x_t^r,\Lambda)}{\partial \Lambda}, \quad (5)$$

where $\mathcal{N}_{jm}(x_t^r,\Lambda)$ is the corresponding Gaussian of certain model and mixture. $\gamma_{jm}^{W_r}(t)$ and $\gamma_{jm}^{W\neq W_r}(t)$ are Gaussian specific *occupancy probabilities* at certain frame t among the label and hypothesized transcriptions respectively. The value of error cost function at tth frame can be absorbed into corresponding occupancy probabilities (state posteriors), which implies we will scale the occupancy probabilities over frame by frame with $\epsilon_r(t)$ in the optimization procedure. In our prior work, to fit a keyword spotting task, the $\epsilon_r(t)$ was designed as,

$$\epsilon_r(t) = \begin{cases} 2 & t \in \{t|W_r(t) \in \text{keywords or } W(t) \in \text{keywords}\} \\ 1 & \text{otherwise} \end{cases}, \quad (6)$$

then we implemented it efficiently by taking advantage of WFST difference operations under a special semiring [6] as $\text{FST}_r^{\text{MCE}} = \text{FST}_{\text{compact}}(W) - \text{FST}(W_r)$. For more details, please consult [1].

3. ADAPTIVE BOOSTED NON-UNIFORM MCE

3.1. Improvements to MCE updates

In this work, we use extended Baum-Welch (EBW) to do the parameter updates. Furthermore, we make two improvements to it as in Boosted MMI [2]: The first is we cancel any shared part of the numerator and denominator posteriors (occupancy probabilities in reference and hypothesis) on each frame,

$$\gamma_{jm}^{W_r}(t) := \gamma_{jm}^{W_r}(t) - \min(\gamma_{jm}^{W_r}(t), \gamma_{jm}^{W\neq W_r}(t)). \quad (7)$$

$$\gamma_{jm}^{W\neq W_r}(t) := \gamma_{jm}^{W\neq W_r}(t) - \min(\gamma_{jm}^{W_r}(t), \gamma_{jm}^{W\neq W_r}(t)). \quad (8)$$

Note that with the canceling the accumulated statistics remain unchanged, while it changes the Gaussian specific learning rate D_{jm} in EBW updates; After canceling the shared part, the numerator statistics can not be directly used in the ML estimate for I-smoothing. Another modification is we do I-smoothing to the previous iteration rather that ML estimates. The rule for calculating D_{jm} is simply changed to $D_{jm} = \max(\tau + E\gamma_{jm}^{den}, 2D_{jm}^{\min})$, where τ is the I-smooth factor, $D_{jm}^{\min}$ is the smallest value that makes the covariance matrix be positive definite. These two modifications were reported to boost the word accuracy considerably in [2], and we will show in the Section 4 with these two improvements our fundamental MCE implementation in the WFST framework can achieve comparable performance with both Boosted MMI and MPE.

3.2. Adaptive Error Cost Function and Model Combination

On top of the boosted MCE above, we can adapt it to non-uniform MCE with the embedding of the error cost function $\epsilon_r(t)$ as in Eq.(4). The simple error cost function we used in Eq.(13) imposes the same error cost on certain training frame during the different optimization iterations, which could lead to severe overtraining when we use fairly large error cost. Additionally, the error cost function with no normalization over the whole training set can lead to too aggressive learning rate for each EBW updates when the number of frames corresponding to keywords is large. If we examine non-uniform MCE from another perspective, as in Eq.(5), it is actually equivalent to employing the regular MCE on a resampled training set in which each frame is weighted according to $\epsilon_r(t)$. Thus, the boosting based techniques can be applied here naturally which typically consist of iteratively learning weak classifiers with respect to a resampled data distribution and combining them to a final strong classifier. And adaptive boosting (AdaBoost) appears to be a perfect candidate since during each iteration it will adjust the cost (weight) corresponding to each data sample adaptively. After Freund and Schapire proposed AdaBoost for binary classification, they also generalized it for multiclass problems, AdaBoost.M1 and AdaBoost.M2 [7], which can be summarized in Algorithm 1.

Algorithm 1 Multiclass AdaBoost

Input: sequence of T training examples $\{(x_t, y_t)\}_{t=1}^T$, $x_t \in \mathcal{X}$, with class labels $y \in \{1, ..., C\}$, and weak classifiers $h_k \in \mathcal{H}$

1: **for** $t = 1, ..., T$ **do**
2: $D_1(t) = 1/T$
3: **end for**
4: **for** $k = 1, ..., K$ **do**
5: Train weak classifier h_k using distribution D_k.
6: Calculate the error of h_k: $\varepsilon_k = \sum_{t:h_k(x_t)\neq y_t} D_k(t)$.
7: If $\varepsilon_k > 1/2$, abort.
8: Set $\beta_k = \varepsilon_k/(1-\varepsilon_k)$.
9: **for** $t = 1, ..., T$ **do**
10: Update distribution:

$$D_{k+1}(t) = \frac{D_k(t)}{Z_k} \cdot \begin{cases} \beta_k, & \text{if } h_k(x_t) = y_t \\ 1, & \text{otherwise} \end{cases}, \quad (9)$$

where Z_k is the normalization factor such that $D_{k+1}(t)$ will be a distribution.
11: **end for**
12: **end for**

Output: $H(x) = \arg\max_{y\in\{1,...,C\}} \sum_{k=1}^K \log(1/\beta_k) I(h_k(x) = y)$

However, several issues need to be addressed before multiclass AdaBoost can be applied: how we define the class in this problem, in what level (utterance/phoneme/frame) we manipulate the sample distribution and how we combine the models trained from each iteration to a final stronger one. Previously, there are several works on boosting techniques for ASR. In [8] and [9], both utterance level and frame level boosting for ASR were investigated. Boosting phoneme HMMs and Gaussian mixtures were proposed in [10] and in [11], and a new method for model combination, multiple stream decoding, was also presented. Recently, boosting has been applied in discriminatively trained system with the re-estimated phonetic decision trees in model combination [12]. Below we describe how we embed the error cost function adaptively in a similar way as AdaBoost and explain how iteratively trained models are combined to a final stronger one in our framework. Firstly, we work on the frame level as our error cost function $\epsilon_r(t)$ imposes cost over frame by frame. And $\epsilon_r(t)$ would not be initialized uniformly as in Line 2 of Algorithm 1. As in non-uniform MCE for keywords spotting, we will use higher value for frames corresponding to keywords as in Eq.(13), while different values can be assigned asymmetrically where keyword frames occur in reference and hypothesis to achieve desirable compromise between the detection miss and false alarm rate, one can also accordingly enlarge the error cost for the frames near key-

word boundaries. Most of boosting techniques for ASR works on phoneme classification level, in this work, we choose frame level as the classification granularity, mainly for two reasons: First, as we impose error cost on the frame level which implies the data distribution is resampled at frame level during boosting iterative training procedure, classification on frames gives us fine-grained and consistent system; Second, this is also more convenient for model combination stage later on. Therefore, in our AdaBoost-like system, the class of acoustic frames is represented by the probability density function (pdf) corresponding to HMM state. (e.g., the corresponding GMM for a GMM-HMM system.) So the number of classes is equal to the number of leaf nodes (distinct acoustic states) of the phonetic decision trees which is easily beyond several thousand for a LVCSR system. Thus we make several modifications to the original multiclass AdaBoost algorithms, in each iteration we calculate the empirical error cost for each individual class, namely we will use class-specific ε_k^y, and at each frame, we consider it is a misclassification error if the value of accumulated state posteriors in hypothesis (denominator lattice) whose corresponding GMM's indices are different from the reference is beyond 0.5, $\sum_{j\neq y_t}\gamma_j^{W\neq W_r}(t) > 0.5$, note that $\gamma_j^{W\neq W_r}(t) = \sum_m \gamma_{jm}^{W\neq W_r}(t)$, so the class-specific *empirical error cost* over the whole training set is given by,

$$\varepsilon_k^y = \sum_{t:y_t\in y} \mathbb{1}\left\{\sum_{j\neq y}\gamma_j^{W\neq W_r}(t) > 0.5\right\}\cdot\epsilon_r(t). \qquad (10)$$

With the error cost available, we can evaluate class-specific β_k^y and use it to do the model combination for each class. For the model combination part, instead of doing ROVER [13], what we do is more like state-locked multiple-stream decoding as in [10] but implement in a more efficient way under WFST framework because it does not need multiple-pass decoding. As we keep the phonetic decision tree and HMM transition probabilities the same during non-uniform MCE iterations, in our framework, we use unified GMM indexing and compile transition probabilities into decoding WFST graph before we decode utterances. The model combination occurs in the acoustic score generation stage: during decoding, when the acoustic score over certain frame is demanded, instead generated from only one model, we calculate the acoustic score (log-likelihood) as the log-linear interpolation between models ,

$$\log p(x_t|\mathcal{M}^j) = \sum_{k=1}^{K}\frac{1}{Z_j}\log(1/\beta_k^j)\cdot\log p(x_t|\mathcal{M}_k^j), \qquad (11)$$

where Z_j is the normalization factor such that $\sum_{k=1}^{K}\frac{1}{Z_j}\log\frac{1}{\beta_k^j} = 1$. However, we find the values of $\log(1/\beta_k^j)$ are too flat over models trained from each iterations. So we change Eq.(11) to,

$$\log p(x_t|\mathcal{M}^j) = \sum_{k=1}^{K}\mathbb{1}\left\{k = \arg\min_k \varepsilon_k^j\right\}\cdot\log p(x_t|\mathcal{M}_k^j), \quad (12)$$

typically we just pick the model for each class with minimum empirical error cost during iterations. Finally we summarize our adaptive boosted non-uniform MCE in Algorithm 2.

4. EXPERIMENTS

We comprehensively validate the proposed adaptive boosted non-uniform MCE framework for keyword spotting on two challenging large-scale spontaneous CTS tasks, Switchboard-1 Release 2 and HKUST Mandarin Telephone Speech (LDC2005S15).

Algorithm 2 Adaptive Boosted Non-uniform MCE

Input: sequence of T training examples (acoustic frames) $\{(x_t, y_t)\}_{t=1}^T$, $x_t \in \mathcal{X}$, with class labels $y \in \{1, .., j, .., C\}$, initial model $\mathcal{M}_0 \in \mathcal{M}$.

for $t = 1, ..., T$ **do**

$$\epsilon_r^0(t) = \begin{cases} K_1 & t \in \{t|W_r(t) \in \text{keywords}\} \\ K_2 & t \in \{t|W(t) \in \text{keywords}\} \\ 1 & \text{otherwise} \end{cases}, \qquad (13)$$

end for

for $k = 1, ..., K$ **do**

  for $t = 1, ..., T$ **do**

    Collecting $\gamma_j^{W_r}(t)$ and $\gamma_j^{W\neq W_r}(t)$ using model $\mathcal{M}_{k-1}$.

    Update Error Cost function:

$$\epsilon_r^k(t) = \frac{\epsilon_r^{k-1}(t)}{Z_{k-1}}\cdot\begin{cases} 1, & \text{if } \sum_{j\neq y_t}\gamma_j^{W\neq W_r}(t) > 0.5 \\ \beta, & \text{otherwise} \end{cases}, \qquad (14)$$

    where Z_{k-1} is to guarantee $\sum_{t=1}^T \epsilon_r^k(t) = T$.

  end for

  Calculate the class-specific error cost ε_k^j using Eq.(10).

  Train $\mathcal{M}_k$ using boosted Non-uniform MCE with $\epsilon_r^k(t)$

end for

Output: Combine the models $\mathcal{M}$ using Eq.(12)

Method	Iteration	WER (LM scale)
MLE (Baseline)	-	33.4% (13)
Boosted MMI ($b = 0.1$)	4	30.6% (12)
MPE	4	30.8% (13)
MCE (boosted)	4	**30.3% (12)**

Table 1. WERs of different DT methods on HUB5 English test set

4.1. Experiments on Switchboard

The baseline ASR system is built using Kaldi Speech Recognition Toolkit [14], cross-word triphone models represented by 3-state left-to-right HMMs (5-state HMMs for silence) are trained using MLE on about half the data of whole Switchboard Corpus and a tri-gram language model is trained for decoding. The input features are MFCCs coupled with their linear discriminant analysis (LDA) and maximum likelihood linear transform (MLLT) and feature-space maximum likelihood linear regression (fMLLR) for speaker adaptation during later iterations. The WER of the baseline system on HUB5 English evaluation set is 33.4%. We first list WER results (best ones with LM scales from 9 to 20) on HUB5 for the comparisons of the different fundamental DT methods in Table 1, which shows after two improvements introduced in EBW updates for MCE as in Section 3.1, our implementation can achieve best word accuracy compared to Boosted MMI and MPE.

For the keywords spotting evaluations, we use credit card use subset of the Switchboard and 18 keywords are selected: "bank", "card", "cash", "charge", "check", "month", "account", "balance", "credit", "dollar", "hundred", "limit", "money", "percent", "twenty", "visa", "discover", "interest". We conduct both MCE (basic and boosted) and adaptive boosted non-uniform MCE in 4 iterations. We report FOMs w.r.t the decaying factor β, initial error cost for keywords frames in reference K_1 and in hypothesis K_2 in Table 2. In the experiments with adaptive boosted non-uniform MCE, we found that better spotting performance is achieved with increasing K_1 and K_2, while the influence of the decaying factor β

Method	K_1	K_2	β	FOM
MLE (Baseline)	-	-	-	83.59%
MCE	-	-	-	85.34%
MCE (boosted)	-	-	-	86.99%
Adaptive Boosted Non-uniform MCE	7	7	0.3	**88.45%**
	7	7	0.5	88.29%
	7	7	0.7	88.22%

Table 2. Keyword spotting evaluations on Credit Card Use subset

Method	Iteration	CER (LM scale)
MLE (Baseline)	-	49.67% (13)
Boosted MMI ($b = 0.1$)	4	**44.24% (11)**
MPE	4	44.96% (12)
MCE (boosted)	4	44.74% (11)

Table 3. CERs of different DT methods on HKUST Mandarin Telephone Dev Set

becomes more significant when K_1 and K_2 are fairly large. (Due to space limits, we will list more results in Section 4.2 only). The setup with $K_1 = K_2 = 7$ and $\beta = 0.3$ achieved 88.45% FOM which is 4.86% and 1.46% absolute improvements over baseline and boosted MCE system respectively.

4.2. Experiments on HKUST Mandarin Telephone

HKUST mandarin telephone (LDC2005S15) is a 150+ hours of Mandarin Chinese CTS collected by the Hong Kong University of Science and Technology (HKUST), this release contains the training and development sets with 873 and 24 calls respectively. Since there is no lexicon provided with the corpus and it contains both Chinese and English words (it is highly likely English words occur in spontaneous mandarin speech) below we briefly describe how we prepare the bilingual lexicon. For the Chinese word pronunciations (word to Pinyin), we use one available online dictionary CEDICT [15] for in-vocabulary Chinese words. For OOVs, we do Chinese characters mapping and enumerate all possible pronunciations for each word. We map all Pinyin initials and finals (with tones) to Arpabet phonemes which are widely used in English via IPA rules (not listed due to space limits). For the English word pronunciations, we use CMU dictionary [16] for in-vocabulary words. For OOVs, we use a pre-trained one grapheme to phoneme tools, Sequitur G2P [17]. Since there are several Arpabet phonemes missing for English words pronunciations, what we do is we first mapping the Arpabets to Pinyin (we omit the mapping rules here), and map them back to Arpabets again but with different phonemes that are within the Arpabet phonemes we use. Finally, a bilingual lexicon is built based on a unified phoneme set. We let each phoneme with the different tones to share the same root in the decision tree while making extra tonal questions for them. We use a open-source tools mmseg [18] to do the Chinese word segmentation and then a tri-gram language model is trained on all transcriptions from training set. For other components of baseline ASR setup, they are similar to the one in Section 4.1. The character error rate (CER) of the baseline system on the development set is 49.67%, which is comparable to the results reported in [19]. We also list CERs of the different fundamental DT methods in Table 3.

For the keywords spotting evaluations, we use the development set and 20 Chinese keywords are selected: 喜欢 (like), 中国 (China), 大学 (university), 生活 (life), 朋友 (friend), 国家 (country), 足球 (football), 黄山 (Huangshan), 锻炼 (exercise), 篮球 (basketball), 唱歌 (sing),工作 (job), 专业 (major), 运动 (sports), 电视 (television), 体育 (sports), 学习 (study), 问题 (problem), 台湾 (Taiwan), 学生 (student). We conducted the keywords spotting experiments with similar setups as in Section 4.1 and reported the results in Table 4. We find interesting FOMs results, which are shown in the first four rows of Table 4 that the FOMs with MCE and Boosted MMI systems are even slightly worse than one got from MLE baseline system. The results show that although those systems of these fundamental DT methods can achieve significant character accuracy gains in general as in Table 3, they fail to reduce the errors w.r.t. keywords. This substantially illustrates the advantage of our non-uniform MCE. The setup listed achieved 61.57% FOM which is 4.38% absolute improvements over baseline system. To gain an insight of the significance of the adaptive adjustment for the error cost functions, we list several typical FOMs of non-uniform MCE with and without the adaptive decaying schemes in Table. 5 for the comparisons. We can see there exists considerable absolute FOMs difference between two cases from 1.12% to 2.27%. With larger K_1 and K_2, the effect of the adaptive error cost adjustment scheme becomes more significant.

Method	K_1	K_2	β	FOM
MLE (Baseline)	-	-	-	57.19%
Boosted MMI	-	-	-	56.86%
MPE	-	-	-	59.11%
MCE (boosted)	-	-	-	57.14%
Adaptive Boosted Non-uniform MCE	7	7	0.3	**61.57%**
	7	7	0.5	60.77%
	7	7	0.7	59.90%

Table 4. Keyword spotting evaluation on Mandarin HKUST CTS

Method	K_1	K_2	β	FOMs	Improvements
Adaptive	3	3	1	59.10%	-
	3	3	0.3	60.22%	1.12%
Boosted	5	4	1	59.74%	-
	5	4	0.3	61.03%	1.29%
Non-uniform	5	4.5	1	59.26%	-
	5	4.5	0.3	60.76%	1.50%
MCE	7	6.5	1	59.55%	-
	7	6.5	0.3	61.44%	1.89%
	7	7	1	59.30%	-
	7	7	0.3	61.57%	**2.27%**

Table 5. Influence of the adaptive error cost function embedding, $\beta = 1$ corresponds to the case with no adaptive scheme.

5. CONCLUSIONS

We present a complete framework of DT using non-uniform criteria for keyword spotting, adaptive boosted MCE for keyword spotting. This work is based on the previously proposed non-uniform MCE in our prior work [1], to further boost its spotting performance and tackle its potential issue of over-training, motivated by Adaboost [4], we introduce an adaptive scheme to embed error cost functions together with model combinations during the decoding stage. Although boosting techniques have been applied in the ASR [8] [9] [10] [11] [12], the specific problem we solve and the implementation details in this work are quite different from them, please find the details in Section 3.2. Comprehensively validating the proposed framework on two challenging large-scale spontaneous CTS tasks, we show it can achieve significant and consistent FOM gains over both ML and discriminatively trained systems.

6. REFERENCES

[1] C. Weng, B.-H. Juang, and D. Povey, "Discriminative training using non-uniform criteria for keyword spotting on spontaneous speech," in *Proc. InterSpeech2012*, 2012.

[2] D. Povey, D. Kanevsky, B. Kingsbury, B. Ramabhadran, G. Saon, and K. Visweswariah, "Boosted MMI for model and feature space discriminative training," in *Proc. ICASSP2008*, 2008, pp. 4057–4060.

[3] D. Povey, "Discriminative learning for large vocabulary speech recognition," Ph.D. dissertation, Univ. of Cambridge, 2004.

[4] Y. Freund and R. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," in *Proc. EuroCOLT95*, 1995, pp. 23–27.

[5] B.-H. Juang and S. Katagiri, "Discriminative learning for minimum error classification," *IEEE Trans. Signal Process.*, vol. 40, pp. 3043–3054, Dec. 1992.

[6] D. Povey and M. H. et. al, "Generating exact lattices in the WFST framework," in *Proc. ICASSP2012*, 2012, pp. 4213–4216.

[7] Y. Freund and R. E. Schapire, "Experiments with a new boosting algorithm," 1996.

[8] R. Zhang and E. I. Rudnicky, "Comparative study of boosting and non-boosting training for constructing ensembles of acoustic models," in *Proc. EuroSpeech2003*, 2003.

[9] ——, "A frame level boosting training scheme for acoustic modelling," in *Proc. of ICSLP 2004*, 2004.

[10] C. Dimitrakakis and S. Bengio, "Boosting HMMs with an application to speech recognition," in *Proc. ICASSP2004*, 2004.

[11] G. Zweig and M. Padmanabhan, "Boosting Gaussian mixtures in an LVCSR system," in *Proceedings of ICASSP 2000*, 2000.

[12] G. Saon and H. Soltau, "Boosting systems for large vocabulary continuous speech recognition," *Speech Communications*, vol. 54, pp. 212–218, Feb. 2012.

[13] J. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER)," in *Proc. IEEE International Workshop on Automatic Speech Recognition and Understanding.*, 1997, pp. 347–354.

[14] D. Povey, A. Ghoshal *et al.*, "The Kaldi speech recognition toolkit," in *Proc. ASRU2011*, 2011.

[15] CEDICT - On-line Chinese Tools. [Online]. Available: http://www.mdbg.net/chindict/chindict.php?page=cedict

[16] The CMU Pronouncing Dictionary. [Online]. Available: http://www.speech.cs.cmu.edu/cgi-bin/cmudict

[17] Sequitur G2P - A trainable Grapheme-to-Phoneme converter. [Online]. Available: http://www-i6.informatik.rwth-aachen.de/web/Software/g2p.html

[18] MMSeg - Chinese Segment On MMSeg Algorithm. [Online]. Available: http://pypi.python.org/pypi/mmseg/1.3.0

[19] M.-Y. Hwang, X. Lei *et al.*, "Progress on mandarin conversational telephone speech recognition," in *in International Symposium on Chinese Spoken Language Processing*, 2004.