CTC LOSS FUNCTION WITH A UNIT-LEVEL AMBIGUITY PENALTY

Ryoichi Takashima, Sheng Li and Hisashi Kawai

National Institute of Information and Communications Technology, Japan
ryoichi.takashima.@nict.go.jp

ABSTRACT

This paper presents a modified loss function for training connectionist temporal classification (CTC)-based acoustic models. CTC-based acoustic models have been studied as alternatives to conventional hidden Markov models (HMMs), but have often shown worse performance than conventional deep neural network (DNN)-HMM hybrid models. In this paper, we attempt to identify the primary factor preventing CTC-based models from achieving their full potential, and hypothesize this constraint lies in the ambiguity in the identification boundaries among unit-level labels (phonemes or characters). In accordance with this hypothesis, we propose a modified CTC loss function using an ambiguity penalty. This penalty is defined by the conditional entropy and works to increase the separation metrics among unit-level labels. We evaluate the proposed method on the WSJ and CHiME4 tasks, and demonstrate that our modification improves the word error rate compared with that of the conventional CTC-based model when the training dataset is small.

Index Terms— Speech recognition, connectionist temporal classification, loss function, conditional entropy, acoustic model

1. INTRODUCTION

Deep neural networks (DNNs) have enabled significant improvements in the accuracy of automatic speech recognition (ASR) and have replaced Gaussian mixture models (GMMs) as the de facto standard [1, 2, 3]. One reason for this improved accuracy is that DNNs do not depend on the strong assumptions used in GMMs (i.e., assumptions based on Gaussian distributions), thus enabling more flexible acoustic models to be learned. However, most DNN-based acoustic models are not completely free of conventional assumptions, because they are often applied with conventional hidden Markov models (HMMs) (i.e., DNN-HMM hybrids), which assume that the observations are independent. Therefore, it is expected that the ASR performance could be further improved if more flexible acoustic models were trained without these assumptions.

As alternatives to DNN-HMMs, acoustic models based on the connectionist temporal classification (CTC) [4] have been developed. CTC makes it possible to convert a sequence of the network outputs of each frame to a sequence of labels without using HMMs by introducing the deletion of repeated labels and the insertion of blank labels (i.e., "no label"). Additionally, whereas training DNN-HMMs requires the training data to be pre-segmented for each label (i.e., alignments), which is usually estimated using HMMs, CTC-based acoustic models are trained using a forward–backward algorithm, and do not require any alignment information. This means that the CTC framework does not require the HMM for either training or decoding. Large vocabulary continuous speech recognition (LVCSR) using CTC-based models instead of DNN-HMMs has been studied [5, 6, 7, 8, 9, 10, 11]. CTC has also been applied in language-free end-to-end speech recognition systems, which recognize speech without any language information such as a lexicon or language model [5, 12, 13]. However, in previous research, CTC-based models have often shown lower ASR accuracies than DNN-HMMs when both are integrated with language models [5, 8, 10, 11]. In an experimental study, Kanda et al. [11] showed that CTC-based models require a large amount of training data to obtain higher ASR accuracy than DNN-HMMs. Pundak et al. [14] also experimentally showed the sensitivity of CTC-based models to the amount of training data. These observations indicate that CTC-based models have the potential to outperform DNN-HMMs; however, they perform below their full potential unless they have a large amount of training data. The goal of our research is to explain the primary factor preventing CTC-based models from achieving their full potential, and improve their performance with limited training data (i.e., data-efficiency).

For the purpose of our research, this paper focuses on the posterior probabilities of labels obtained from CTC-based models (i.e., network outputs of each frame). From our experiments, we have found that the posterior probabilities related to some labels are close to each other when there were few training data. This means that the identification boundaries among these labels are ambiguous in the CTC-based model. Thus, we hypothesize that one of the factors restricting the CTC-based model is that, whereas the CTC framework trains the model such that the likelihood of the correct sentence is maximized, the identification boundaries among unit-level labels (i.e., phonemes or characters) are not clear when there are insufficient training data. Therefore, we modify the loss

function for training CTC-based acoustic models so that we not only maximize the likelihood of the correct sentence, but also minimize the ambiguity among unit-level labels. In our proposed method, we add an ambiguity penalty (AP), defined by the conditional entropy, to the original CTC loss function. We evaluate the performance of the proposed method on the Wall Street Journal (WSJ) and CHiME4 tasks, and demonstrate that our proposed method improves the word error rate (WER) compared with that of the conventional CTC-based model when the training dataset is small.

2. CONNECTIONIST TEMPORAL CLASSIFICATION

In this section, we describe the conventional CTC framework [4]. For general speech recognition, we need to map a sequence of the label probabilities calculated for each frame into a label sequence of length equal to or less than the number of frames. CTC makes it possible to convert a sequence of RNN outputs for each frame to a label sequence by introducing the deletion of repeated labels and the insertion of blank labels (i.e., "no label"). For example, when a seven-frame observation is assigned as "$\pi = \{a - abba-\}$" or "$\pi = \{a - -abaa\}$" with labels modified by adding the blank label '-' (we call the assignment path π), both paths are mapped to an identical label sequence $\mathbf{l} = \mathcal{B}(\pi) = \mathcal{B}(\{a - abba - |a - -abaa\})$ ="aaba", where $\mathcal{B}$ is the mapping function. Thus, because there are multiple possible paths mapped to an identical label sequence, the conditional probability of the label sequence $\mathbf{l}$ given the observation sequence $\mathbf{x}$ is defined as the sum of the probabilities of all possible corresponding paths:

$$Pr(\mathbf{l}|\mathbf{x}) = \sum_{\pi \in \mathcal{B}^{-1}(\mathbf{l})} Pr(\pi|\mathbf{x}). \tag{1}$$

The probability of the modified label π_t at frame t is modeled with an RNN. Therefore, the conditional probability of path π is calculated as follows:

$$Pr(\pi|\mathbf{x}) = \prod_{t}^{T} y^t_{\pi_t}, \tag{2}$$

where $y^t_{\pi_t}$ denotes the output of the RNN node corresponding to modified label π_t at frame t, and T denotes the number of frames.

The CTC-based acoustic model is trained by maximizing the likelihood, that is, Eq. (1), for all training data. Practically, because the training target is the RNN, the aim is to minimize the loss function defined as

$$\mathcal{L}_{CTC} = -\sum_{(\mathbf{x},\mathbf{l}) \in Z} \ln Pr(\mathbf{l}|\mathbf{x}), \tag{3}$$

where Z denotes the training data-set. The conditional probability is efficiently computed using the forward-backward algorithm. After computing the conditional probability, the derivatives, that is, $\frac{\partial \mathcal{L}_{CTC}}{\partial y^t_k}$ and $\frac{\partial \mathcal{L}_{CTC}}{\partial u^t_k}$ (where $y^t_k = \frac{\exp(u^t_k)}{\sum_{k'} \exp(u^t_{k'})}$), are computed and the RNN is trained using backpropagation (for more details, see [4]).

3. PROPOSED METHOD

3.1. Focal point: posterior probabilities of labels

We focus on the posterior probabilities of labels obtained from CTC-based models (i.e., network outputs of each frame). The top figure in Figure 1 shows the network outputs of a training sample for each frame that correspond to some phoneme labels. As shown in this figure, the posterior probabilities of multiple labels (circled by red dashes) are close to each other, and it is relatively difficult to identify them. This indicates that, at least in these frames, the identification boundaries among the labels are ambiguous. Thus, we hypothesize that this ambiguity among unit-level labels is one of the factors restricting the CTC-based model. To verify our hypothesis, we attempt to train the CTC-based model to not only maximize the likelihood of the correct sentence, but also minimize the unit-level ambiguity by modifying the loss function. We then confirm whether our modification improves the ASR accuracy.

3.2. CTC loss function with an AP

As mentioned in Section 3.1, we modify the CTC loss function by adding a penalty based on the unit-level ambiguity (ambiguity penalty; AP). The proposed method uses the conditional entropy to define the AP. The minimization of conditional entropy is often used to increase discriminative ability in clustering tasks [15, 16, 17] and the training of generative models [18, 19]. In our proposed method, the conditional entropy is defined as the expectation of the log conditional probability of a modified label k given an observation x_t at frame t:

$$Entropy(k|x_t) = -\sum_{k} Pr(k|x_t) \ln Pr(k|x_t). \tag{4}$$

Lower values of the conditional entropy mean that x_t tends to be recognized as a specific label more clearly. Using the conditional entropy, we define the AP as

$$\mathcal{L}_{AP} = -\sum_{\mathbf{x} \in Z} \sum_{t=1}^{T_{\mathbf{x}}} \sum_{k} Pr(k|x_t) \ln Pr(k|x_t), \tag{5}$$

where $T_{\mathbf{x}}$ denotes the number of frames of a training data sample $\mathbf{x} = \{x_1, \ldots, x_{T_{\mathbf{x}}}\}$. Note that unlike using cross entropy, Eq. (5) does not require the correct label, that is, frame-level alignment. The conditional probability of a modified label k is defined as the output of the RNN node corresponding

to k:

$$Pr(k|x_t) = y_k^t. \quad (6)$$

Then, we compute the derivatives of $\mathcal{L}_{AP}$. The derivative with respect to the network output y_k^t is derived as follows:

$$\begin{aligned} \frac{\partial \mathcal{L}_{AP}}{\partial y_k^t} &= -\frac{\partial}{\partial y_k^t} \sum_{\mathbf{x} \in Z} \sum_{t=1}^{T_{\mathbf{x}}} \sum_k y_k^t \ln y_k^t \\ &= -(\ln y_k^t + 1). \end{aligned} \quad (7)$$

The derivative with respect to the *unnormalized* output u_k^t, where $y_k^t = \frac{\exp(u_k^t)}{\sum_{k'} \exp(u_{k'}^t)}$, is derived as follows:

$$\begin{aligned} \frac{\partial \mathcal{L}_{AP}}{\partial u_k^t} &= \sum_{k'} \frac{\partial \mathcal{L}_{AP}}{\partial y_{k'}^t} \frac{\partial y_{k'}^t}{\partial u_k^t} \\ &= -\sum_{k'} (\ln y_{k'}^t + 1)(y_{k'}^t \delta_{k',k} - y_{k'}^t y_k^t) \\ &= -\Big(y_k^t (\ln y_k^t + 1) - y_k^t \sum_{k'} y_{k'}^t (\ln y_{k'}^t + 1) \Big), \end{aligned} \quad (8)$$

where $\delta_{k',k}$ is Kronecker's delta.

The proposed CTC loss function is defined as the interpolation between the conventional CTC loss function and the AP:

$$\mathcal{L}_{CTC+AP} = (1 - \lambda)\mathcal{L}_{CTC} + \lambda \mathcal{L}_{AP}, \quad (9)$$

where $\lambda (0 \leq \lambda \leq 1)$ denotes a tunable weighting parameter.

4. EXPERIMENTS

4.1. Experimental conditions

To evaluate the CTC-based acoustic models, we used the EESEN software [8], and implemented our proposed loss function in EESEN. We set the training conditions according to the approach in [8], written by the developers of the software, as follows. We used a bi-directional long short-term memory [20, 21] with four hidden layers and 320 memory cells in each hidden layer for building a CTC network. We extracted 40-dimensional mel-filterbank features with their first- and second-order derivatives (FBANK+Δ+$\Delta\Delta$, 120 dimensions in total) as acoustic features, and defined the target labels to include 69 phonemes, two noise marks, and a blank (72 labels in total). The model parameters were optimized using stochastic gradient descent (SGD) with a momentum of 0.9, and the learning rate was initially set to 0.00004. For decoding, the EESEN framework integrated the CTC-based acoustic model, lexicon, and language model using the weighted finite-state transducer [22, 23]. The decoding framework is detailed in [8].

To evaluate the DNN-HMMs, we used the Kaldi ASR toolkit [24]. We set the training conditions following standard Kaldi recipe, although we used the same base acoustic features as for training the CTC-based model: We extracted 120-dimensional FBANK+Δ+$\Delta\Delta$ features and spliced 11 neighboring frames for the inputs of the DNN (a total of 1,320 nodes in the input layer). The DNN had four hidden layers and 1,024 nodes in each hidden layer. The model parameters were optimized under the cross-entropy (CE) criterion (CE-DNN-HMM) using the standard SGD without momentum.

The experiments were conducted on the WSJ and CHiME4 tasks. For the WSJ task, we performed the experiments on two types of training dataset: (1) using only "WSJ0 (LDC93S6B)" (15 hours, known as "train_si84" in the Kaldi recipe); and (2) using "WSJ0" and "WSJ1 (LDC94S13B)" (81 hours, known as "train_si284" in the Kaldi recipe). For both experiments, we used 95% of the training data for learning the model parameters and the remaining 5% for validation, and we used the "dev93" and "eval92" datasets for evaluation. Note that we did not use dev93 as the validation set, unlike many other studies. We used the CMU dictionary as the lexicon and the 20,000-word vocabulary WSJ pruned language model, known as "lm_tgpr" in the Kaldi recipe, as the language model. The experimental conditions using train_si284 were the same as those in [8].

The CHiME4 corpus [25] was recorded in noisy environments, such as a cafe, street junction, public transport, and pedestrian area. We used this corpus to evaluate the effect of noisy training data on our proposed method. We used the "tr05_multi_noisy" (18 hours) dataset to learn the model parameters, "dt05_multi_noisy" (5.6 hours) dataset for validation, and "dt05_real_isolated_1ch_track" and "et05_real_isolated_1ch_track" datasets for evaluation. We used the CMU dictionary as the lexicon and the 5,000-word vocabulary WSJ pruned language model, known as "lm_tgpr_5k" in the Kaldi recipe.

4.2. Results

Table 1 presents the WERs for each task. "CTC-AP" denotes our proposed method (i.e., CTC with AP). On the WSJ task using train_si84 (15 hours), increasing the weight of the AP λ to 0.050 improved the WERs for the conventional CTC-based model (this is equivalent to the case of $\lambda = 0.000$). However, increasing λ above 0.050 worsened the WERs. This tendency was observed for both dev93 and eval92.

Figure 1 shows the posterior label probabilities of the CTC models trained using train_si84. The top figure shows the probabilities on the conventional CTC model and the bottom figure shows those on the CTC-AP model with $\lambda = 0.050$. As described in Section 3.1, for the conventional CTC model, the posterior probabilities of multiple labels (circled by red dashes) were close to each other, and it was relatively difficult to identify them. In contrast, for the CTC-AP model, those probabilities converged to the correct labels. This indicates that our proposed method clarified the

Table 1. Word error rates (WERs) for each task.

Acoustic Model	WER	
WSJ–train_si84 (15hrs)	dev93	eval92
CTC (baseline)	20.18	13.11
CTC-AP ($\lambda = 0.010$)	18.74	12.95
CTC-AP ($\lambda = 0.025$)	18.42	11.70
CTC-AP ($\lambda = 0.050$)	18.03	11.41
CTC-AP ($\lambda = 0.075$)	18.86	12.53
CTC-AP ($\lambda = 0.100$)	21.59	13.75
CE-DNN-HMM	13.69	8.36
WSJ–train_si284 (81hrs)	dev93	eval92
CTC [8]	11.39	7.87
CTC (our work, baseline)	11.70	8.05
CTC-AP ($\lambda = 0.010$)	11.71	7.83
CTC-AP ($\lambda = 0.025$)	11.72	7.92
CTC-AP ($\lambda = 0.050$)	11.93	7.83
CTC-AP ($\lambda = 0.075$)	12.19	8.12
CTC-AP ($\lambda = 0.100$)	12.67	8.38
CE-DNN-HMM	11.03	7.02
CHiME4-tr05_multi_noisy (18hrs) + dt05_multi_noisy (5.6hrs)	dt05_real	et05_real
CTC (baseline)	29.12	45.50
CTC-AP ($\lambda = 0.010$)	28.83	44.85
CTC-AP ($\lambda = 0.025$)	28.47	44.38
CTC-AP ($\lambda = 0.050$)	28.03	43.59
CTC-AP ($\lambda = 0.075$)	31.58	47.44
CTC-AP ($\lambda = 0.100$)	33.48	49.81
CE-DNN-HMM	22.45	38.86

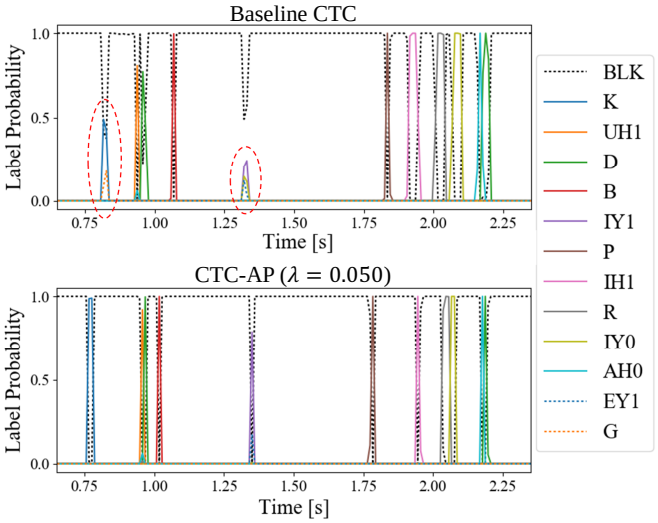

Fig. 1. Network outputs (i.e., posterior probabilities) of some phoneme labels on a training data sample uttered "COULD BE. PERIOD." The correct phoneme label sequence is "K/UH1/D/B/IY1/P/IH1/R/IY0/AH0/D." "BLK" denotes the blank label. The circled areas explain that multiple labels are relatively difficult to be identified in these frames.

unit-level identification boundaries on a CTC-based model, and that clarification could improve the WERs of the CTC-based model. Conversely, this result confirms our hypothesis described in Section 3.1.

As described above, we have confirmed that our proposed method improves the WERs of the conventional CTC-based model when train_si84 is used as the training set. However, as shown in Table 1, our proposed method did not produce significant improvement when train_si284 (80 hours) was used. These results indicate that our proposed method has a positive effect with relatively few training data.

On the CHiME4 task, as for the experiments on the WSJ-train_si84 training set, our proposed method improves the WERs compared with the conventional CTC-based model. Additionally, the WER tendency when increasing λ is the same as for the train_si84 task. Because the size of the training set for the CHiME4 task is approximately as small as train_si84 and the training set includes noisy data, these results indicate that the proposed method has a positive effect with relatively few training data, even when the training data include noisy speech.

For all tasks, the CE-DNN-HMMs showed the best WERs, as reported in previous studies [5, 8, 11], even though our proposed method improves the WERs of CTC-based models.

5. CONCLUSION

In this paper, we attempted to identify the primary factor that prevents CTC-based models from achieving their full potential when there are insufficient training data. To verify our hypothesis that one of the factors lies in the ambiguity among unit-level labels, we proposed a modified CTC loss function with a unit-level ambiguity penalty. In experiments using multiple corpora, our proposed modification did not outperform conventional CE-DNN-HMMs, but did improve the WERs of conventional CTC-based models when there were few training data. This means that the proposed method did not realize the full potential of CTC-based models, but did improve their data-efficiency. From these results, we conclude that unit-level ambiguity is not the primary factor, but is one of the sub-factors preventing CTC-based models from achieving their full potential.

Because our method is effective in scenarios with relatively few training data, it might be helpful for ASR of low resource languages. In future work, we will evaluate our proposed method in these tasks, and continue efforts to achieve the full potential of CTC-based models.

6. REFERENCES

[1] Geoffrey Hinton, Li Deng, Dong Yu, et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[2] Frank Seide, Gang Li, and Dong Yu, "Conversational speech transcription using context-dependent deep neural networks.," in *Interspeech*. ISCA, 2011, pp. 437–440.

[3] Jinyu Li, Li Deng, R. Haeb-Umbach, and Y. Gong, *Robust Automatic Speech Recognition - A Bridge to Practical Applications*, Elsevier, Amsterdam, Netherlands, October 2015.

[4] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning (ICML2006)*. ACM, 2006, pp. 369–376.

[5] Alex Graves and Navdeep Jaitly, "Towards end-to-end speech recognition with recurrent neural networks.," in *The 31st International Conference on Machine Learning (ICML 2014)*, 2014, vol. 14, pp. 1764–1772.

[6] Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, et al., "Deep speech 2: End-to-end speech recognition in english and mandarin," in *The 33rd International Conference on Machine Learning (ICML 2016)*, 2016, pp. 173–182.

[7] Haşim Sak, Andrew Senior, Kanishka Rao, and Françoise Beaufays, "Fast and accurate recurrent neural network acoustic models for speech recognition," in *Interspeech*. ISCA, 2015, pp. 1468–1472.

[8] Yajie Miao, Mohammad Gowayyed, and Florian Metze, "Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding," in *The 2015 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU 2015)*. IEEE, 2015, pp. 167–174.

[9] Yajie Miao, Mohammad Gowayyed, Xingyu Na, et al., "An empirical exploration of ctc acoustic models," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP2016)*. IEEE, 2016, pp. 2623–2627.

[10] Naoyuki Kanda, Xugang Lu, and Hisashi Kawai, "Maximum-a-posteriori-based decoding for end-to-end acoustic models," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 5, pp. 1023–1034, 2017.

[11] Naoyuki Kanda, Xugang Lu, and Hisashi Kawai, "Mininum bayes risk training of ctc acoustic models in maximum a posteriori based decoding framework," in *2017 IEEE International Conference on Acoustc, Speech and Signal Processing (ICASSP2017)*. IEEE, 2017, pp. 4855–4859.

[12] Suyoun Kim, Takaaki Hori, and Shinji Watanabe, "Joint ctc-attention based end-to-end speech recognition using multi-task learning," in *2017 IEEE International Conference on Acoustc, Speech and Signal Processing (ICASSP2017)*. IEEE, 2017, pp. 4835–4839.

[13] Takaaki Hori, Shinji Watanabe, Yu Zhang, and William Chan, "Advances in joint ctc-attention based end-to-end speech recognition with a deep cnn encoder and rnn-lm," in *Interspeech*. ISCA, 2017, pp. 949–953.

[14] Golan Pundak and Tara N Sainath, "Lower frame rate neural network acoustic models," in *Interspeech*. ISCA, 2016, pp. 22–26.

[15] Haifeng Li, Keshu Zhang, and Tao Jiang, "Minimum entropy clustering and applications to gene expression analysis," in *2004 IEEE Computational Systems Bioinformatics Conference (CSB2004)*. IEEE, 2004, pp. 142–151.

[16] Yves Grandvalet and Yoshua Bengio, "Semi-supervised learning by entropy minimization," in *Advances in Neural Information Processing Systems 18 (NIPS 2005)*, 2005, pp. 529–536.

[17] Andrew Rosenberg and Julia Hirschberg, "V-measure: A conditional entropy-based external cluster evaluation measure.," in *The 2007 Joint Meeting of the Conference on Empirical Methods on Natural Language Processing and the Conference on Natural Language Learning (EMNLP-CoNLL2007)*, 2007, vol. 7, pp. 410–420.

[18] Jost Tobias Springenberg, "Unsupervised and semi-supervised learning with categorical generative adversarial networks," in *4th International Conference on Learning Representations (ICLR 2016)*, 2016.

[19] Xun Huang, Yixuan Li, Omid Poursaeed, et al., "Stacked generative adversarial networks," in *IEEE Conference on Computer Vision and Pattern Recognition 2016 (CVPR2016) (on appear)*, Honolulu, HI, 2017.

[20] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[21] Haşim Sak, Andrew Senior, and Françoise Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling," in *Interspeech*. ISCA, 2014.

[22] Mehryar Mohri, Fernando Pereira, and Michael Riley, "Weighted finite-state transducers in speech recognition," *Computer Speech & Language*, vol. 16, no. 1, pp. 69–88, 2002.

[23] Cyril Allauzen, Michael Riley, Johan Schalkwyk, et al., "Openfst: A general and efficient weighted finite-state transducer library," in *12th International Conference on Implementation and Application of Automata (CIAA2007)*. Springer, 2007, pp. 11–23.

[24] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, et al., "The kaldi speech recognition toolkit," in *The 2011 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU 2011)*. IEEE Signal Processing Society, 2011.

[25] Emmanuel Vincent, Shinji Watanabe, Aditya Arie Nugraha, et al., "An analysis of environment, microphone and data simulation mismatches in robust speech recognition," *Computer Speech & Language*, 2016.