SPEAKER-AWARE TRAINING OF LSTM-RNNS FOR ACOUSTIC MODELLING

Tian Tan [1], *Yanmin Qian*[1,2], *Dong Yu*[3], *Souvik Kundu*[4], *Liang Lu*[5], *Khe Chai SIM*[4], *Xiong Xiao*[6], *Yu Zhang*[7]

[1]Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China
[2] Cambridge University Engineering Department, Cambridge, UK
[3]Microsoft Research, Redmond, USA
[4]School of Computing, National University of Singapore, Republic of Singapore
[5]Centre for Speech Technology Research, University of Edinburgh, UK
[6]Temasek Labs, Nanyang Technological University, Singapore.
[7]CSAIL, MIT, Cambridge, USA

ABSTRACT

Long Short-Term Memory (LSTM) is a particular type of recurrent neural network (RNN) that can model long term temporal dynamics. Recently it has been shown that LSTM-RNNs can achieve higher recognition accuracy than deep feed-forword neural networks (DNNs) in acoustic modelling. However, speaker adaption for LSTM-RNN based acoustic models has not been well investigated. In this paper, we study the LSTM-RNN speaker-aware training that incorporates the speaker information during model training to normalise the speaker variability. We first present several speaker-aware training architectures, and then empirically evaluate three types of speaker representation: I-vectors, bottleneck speaker vectors and speaking rate. Furthermore, to factorize the variability in the acoustic signals caused by speakers and phonemes respectively, we investigate the speaker-aware and phone-aware joint training under the framework of multi-task learning. In AMI meeting speech transcription task, speaker-aware training of LSTM-RNNs reduces word error rates by 6.5% relative to a very strong LSTM-RNN baseline, which uses FMLLR features.

Index Terms— speaker-aware training, LSTM-RNNs, speaker adaptation, i-vector, speaking rate

1. INTRODUCTION

Deep learning has achieved tremendous success in acoustic modelling. With multiple hidden layers, the hybrid neural network hidden Markov model (NN/HMM) [1] can obtain significant improvement in terms of recognition accuracy compared to the conventional Gaussian mixture models (GMMs) [2]. Previous studies mainly focus on the feed-forward neural networks using the acoustic features from a fixed-size context window, while recently, the recurrent neural networks (RNNs) have been demonstrated to be able to achieve higher recognition accuracy. The RNN has recurrent-connections on its hidden layers, which is expected to capture much longer temporal dynamics. However, training an RNN turns to be difficult due to the well-known gradient vanishing and exploding problems [3]. In this context, Long Short-Term Memory (LSTM) [4] is proposed to overcome these problems by introducing separate gate functions to control the flow of the information. For acoustic modelling, LSTM-RNNs are reported to outperform the DNNs on the large vocabulary tasks [5, 6].

However, a long standing problem in acoustic modelling is the mismatch between training and test data caused by speakers and environmental differences. Although DNNs are more robust to the mismatch compared to GMMs, significant performance degradation has been observed [7]. There have been a number of studies to improve the robustness of DNN-HMM acoustic models, which can be divided into three categories [8]: transformation based adaptation [9, 10], conservative training [11] and speaker-aware training [12, 13, 14]. In particular, speaker dependent linear transformations for input or output layer were used for speaker adaptation in [9, 10], while in [11], KL divergence based regularization was exploited to control overfitting in DNN acoustic model adaptation. Finally, in [12, 13], a speaker auxiliary vector was introduced to allow DNNs to normalise the speaker variability automatically. However, most of these studies have been limited to DNNs, and to our best knowledge, only Miao and Metze [15] investigated speaker adaptation for LSTM-RNNs.

The approaches investigated in [15] fall in the category of transformation based adaptation, where linear input feature transformation and hidden activation transformation were applied to adapt LSTM-RNNs. In this work, we focus on the speaker-aware training for LSTM-RNNs. We show in our study that this is not trivial. Different from DNNs, LSTM-RNNs are dynamic systems that are sensitive to a static input. To deal with this problem, we investigated different model architectures in order to make the speaker-aware training effective for LSTM-RNNs. We also evaluated these model architectures with three different speaker representations, namely, I-vector [16], bottleneck speaker vector (BSV) [17] and the speaking rate. In addition, we managed to incorporate the phone information in our adaptation network, which helped to further reduce the word error rate. Our experiments were performed on the AMI meeting speech transcription dataset, and we obtained 6.5% relative improvement over a strong LSTM-RNN baseline.

2. SPEAKER-AWARE TRAINING ON LSTM-RNNS

2.1. LSTM-RNNs

Compared to the feedforward neural networks, RNNs have the advantage of learning complex temporal dynamics in sequential data.

The work reported here was started at JSALT 2015 in UW, Seattle, and was supported by JHU via grants from NSF (IIS), DARPA (LORELEI), Google, Microsoft, Amazon, Mitsubishi Electric, and MERL. Tian Tan is supported by the Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning, the China NSFC project No. 61222208, JiangSu NSF project No. 201302060012.

However, in practice, training RNNs to learn long-term temporal dependency can be difficult due to the gradient vanishing and exploding problem [3]. The LSTM architecture provides a solution that partially overcomes the weakness of simple RNNs and achieves the-state-of-the-art performance in speech recognition [5, 6]. More on standard form of LSTM is presented in [4], while in [5], a projection layer is applied to project the memory cells' outputs to a lower-dimensional vector which is particularly effective for speech recognition when the amount of training data is not abundant. In this work, we used the same LSTM-projection (LSTMP) architecture.

2.2. Adaptation architectures for speaker-aware training

Speaker-aware training is an approach that incorporates the speaker information into the network training process in order to normalise the speaker variability. Typical speaker representations are I-vector [16], bottleneck speaker vector (BSV) [17] and jointly trained speaker-code [12]. For feed-forward neural networks, it usually works well by simply concatenating the speaker representations with the acoustic features. Similar approach may be applied for LSTM-RNNs as shown in Figure 1(a). However, this approach may not work well because speaker auxiliary vectors are constant for a given speaker, adding a static vector into a dynamical system is ineffective on ASR as shown in[18, 19].

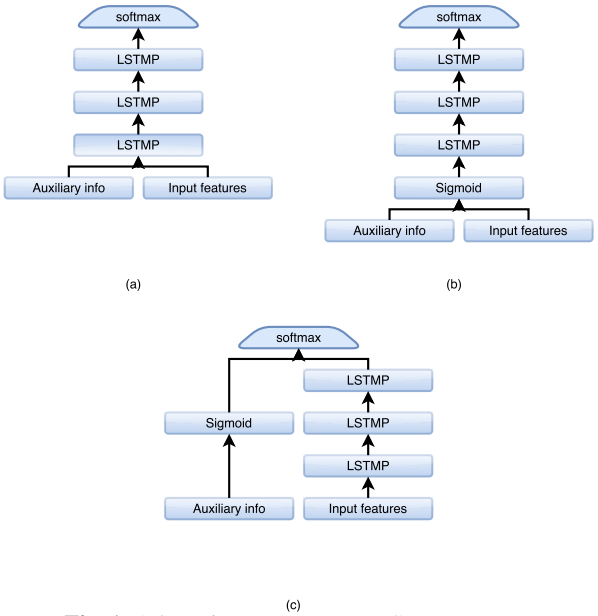

Fig. 1. Adaptation structures on LSTM-RNNs

To solve this problem, two structures are explored. In figure 1(b), the auxiliary vector and acoustic feature pass through a non-linear transformation first so that the input for LSTM-RNN becomes different among frames. In figure 1(c), the auxiliary vector goes through a shallow NN first and is then concatenated to the output of the LSTM-RNN.

3. SPEAKER AUXILIARY VECTOR

In this paper, we investigate three kinds of auxiliary speaker representations: i-vector, bottleneck speaker vector and speaking rate. The methods to obtain these speaker representations are described below.

3.1. I-vector

I-vector is a popular technique for speaker verification and recognition [16]. It can capture the most important information of a speaker in a low-dimensional representation. Furthermore it has been shown that i-vectors can be used in speaker-aware training on DNNs for speech recognition [13, 20].

3.2. Bottleneck speaker vector

Bottleneck speaker vector, proposed in [7], has obtained competitive performance to the i-vector based adaptation on DNNs. Figure 2(a) is the standard network for extracting BSVs. It is a 3-layer DNN trained to classify speakers. A bottleneck layer is set at the third layer to extract the bottleneck vectors. To obtain the BSV, we averaged all bottleneck vectors of a speaker and normalise its L2-norm to 1.

Motivated by the work in [21] which uses an ASR-DNN system to integrate the speech content and to extract a more powerful i-vector, two more structures are proposed to exploit phone information when extracting BSV.

- Figure 2(b) shows the phone-aware training (PAT) used in this paper. The posterior of a mono-phone based DNN is fed as input to the speaker recognition network. Different from the traditional speaker-aware training, in this work, these two networks are trained jointly. The criterion used for model optimization is

$$E(\theta) = \lambda E_{\texttt{mono}}(\theta) + E_{\texttt{spk}}(\theta) \quad (1)$$

  where θ denotes model parameters and $E_{\texttt{mono}}(\theta)$, $E_{\texttt{spk}}(\theta)$ are the cross-entropy for the mono-phone and speaker DNNs respectively. λ is a *mixing factor*.
- Figure 2(c) illustrates another structure, in which the mono-phone and speaker DNNs are trained jointly Eq. (1). Multi-task joint-training, which optimizes more than one criteria in model training, can yield better performance than normal cross entropy training. In [22], triphones and trigraphemes are used to train acoustic models. In [23] phone and speaker joint training is conducted.

Note that the criteria used in previous two methods are the same. However, the model architectures are different. In PAT, phone information acts as an auxiliary information to help the network classify speaker better. While in multi-task joint training, phone information serves to regularize the network.

3.3. Speaking rate

It has been known that the speaking rate impacts the accuracy of automatic speech recognition (ASR). A relatively low or high speaking rate usually increases the word error rate [24, 25]. The speaking rate may influence the speech signal in two ways: on the temporal properties or on the spectrum distortion. In previous works, frame rate normalization [26] was applied on GMM-HMM and speaking rate assisted training was evaluated on DNN-HMM [27]. In this work, speaking rate was used for speaker-aware training on LSTM-RNNs. Considering that LSTM-RNNs are both dynamic systems and neural networks, they may compensate for the speaking rate in both temporal properties and the spectrum distortion. The speaking rate is fed into LSTM-RNNs with acoustic feature and is extracted as follows.

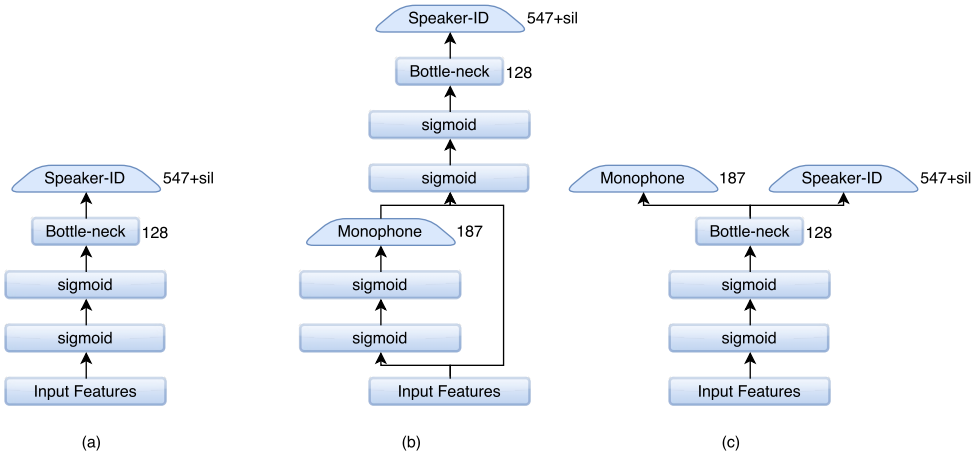

Fig. 2. Structures for extracting bottleneck speaker vectors

- An alignment is generated with a baseline DNN system.
- The speaking rate is then calculated by dividing the number of phones by the duration of all phones.

$$f(i) = N / \sum_{j=1}^{N} t_{i,j}$$

where $f(i)$ is the speaking rate for utterance i, $t_{i,j}$ is the duration of phone j, and there are N phones in utterance i.

4. EXPERIMENT

Our experiments are conducted on AMI corpus, which contains around 100 hours of meetings recorded in specifically equipped instrumented meeting rooms at three sites in Europe (Edinburgh, IDIAP, TNO) [28]. Acoustic signal is captured and synchronized by multiple microphones including individual head microphones (IHM, close-talk), lapel microphones, and one or more microphone arrays. In this work, the IHM (close-talk) data are used. Our experiments adopted the suggested AMI corpus partition that contains about 80 hours and 8 hours in training and evaluation sets respectively [29]. For fast turnarounds, we also selected 10000 utterances from the training set and created a lighter set-up. The training procedures and test sets are identical in the sub- and full-set experiments. For decoding, the 50K-word AMI dictionary and a trigram language model interpolated from the one created using the AMI training transcripts and the Fisher English corpus were used.

4.1. Baseline set up

The GMM-HMM system was built using the standard Kaldi AMI recipe [30]. 39-dimensional MFCC (plus deltas and double deltas) features with CMVN was used to train the initial ML model. Then 7 frames of MFCCs were spliced and projected to 40 dimensions with linear discriminant analysis (LDA). A maximum likelihood linear transform (MLLT) was estimated on the LDA features to generate the LDA+MLLT model. After that, speaker adaptive training was performed with one FMLLR transform per speaker. 3962 tied-states were used in GMM-HMM. FMLLR features were then used for training DNN, LSTM-RNN, i-vector and BSV.

CNTK [31] was used to train the DNN and LSTM-RNN. The DNN has 6 hidden layers each of which contains 2048 neurons. The input feature for DNNs contains 11 frames (5 frames on each side of the center frame). Cross-entropy (CE) was used. The learning rate started from 0.1 per minibatch and changed to 1 for the second epoch, then it was decayed by a factor of 0.5 when the cross entropy on a cross-validation set between two consecutive epochs increases.

The LSTM-RNN has 3 projected LSTM layers which are followed by the soft-max layer. Each LSTM layer has 1024 memory cells and 512 output units in projection. Input to the LSTM-RNN is a single acoustic frame with 5 frames shift, the truncated version of BPTT was used for training LSTM-RNN[5]. 40 utterances were processed in parallel and the BPTT truncation size was set to 20. To ensure training stability, the gradient was clipped to the range of [-1, 1] when updating parameters.

Data set	Model	WER
Sub Set	DNN	34.9
	LSTM-RNN	32.4
Full Set	DNN	26.5
	LSTM-RNN	26.0

Table 1. WER (%) of DNNs and LSTM-RNN on AMI IHM condition

Table 1 shows the results of baseline DNNs and LSTM-RNNs trained on the sub- and full-set. On both sets, the LSTM-RNN performs better than the DNN, demonstrating its advantage in acoustic modeling. However, the improvement becomes smaller when more data was used, which probably because the FMLLR feature is not so suitable for LSTM-RNNs. The transformation may break the inherent temporal dependency between neighbor frames and this also indicates that other speaker adaptation methods exploration are necessary for LSTM-RNNs.

4.2. Experiment on different structure for LSTM adaptation

Since the efficacy of i-vector for adaptation has been shown on DNNs, we firstly choose i-vector as the speaker vector for speaker-aware training on LSTM-RNNs. In this work, a 128 dimensional i-vector was extracted for each speaker by using GMM-UBM (the universal background model) with 2048 components. The i-vector was length normalized to one. Three different structures presented in Section 2 for speaker adaptation on LSTM-RNNs were investigated.

In Table 2, (a),(b),(c) are structures shown in Figure 1. These structures did not perform according to our expectation. The

Model	Struct.	WER
LSTMP	—	32.4
+i-vector	(a)	31.1
	(b)	33.2
	(c)	34.5
LSTMP+DNN	—	35.1

Table 2. WER (%) of different structures for LSTM-RNN adaptation on AMI IHM Subset

straightforward structure gained the best performance, outperforming all other structures. This is partly because the additional sigmoid function introduced into LSTM-RNNs makes the network hard to train well. We verified our conjecture using an LSTM-RNN followed by a 1-hidden-layer DNN. This structure performed worser than normal LSTM-RNNs. We will investigate ReLU and advanced learning rate strategy in the future work.

4.3. Experiment on different BSVs for speaker adaptation

In this section, we compare the WER of the DNN and LSTM-RNN adaptation using different bottleneck speaker vectors obtained by standard training (ST), phone aware training (PAT) and multi-task training (MT). The mixing factor λ for BSV-MT and BSV-PAT were set to 0.1 in our experiments.

Model	Struct.	WER
DNN	—	34.9
+i-vector	—	34.4
+BSV	ST	34.6
	PAT	34.7
	MT	34.3
LSTMP	—	32.4
+i-vector	—	31.1
+BSV	ST	31.9
	PAT	31.8
	MT	31.7

Table 3. WER (%) of different BSV networks trained on AMI IHM Subset

From Table 3, we can observe that using BSV feature can lead to improvements on both DNN and LSTM-RNN systems. BSV with multi-task training is slightly better than standard BSV on both DNNs and LSTM-RNNs. On DNNs, it was even slightly better than using i-vector. However, phone-aware training is not helpful for BSV extraction on both DNN and LSTM models.

4.4. Experiment on different speaker auxiliary vectors

In this subsection we compare different speaker vectors. As shown in Table 4, all auxiliary vectors achieved 1.5%-4% relative improvement over the baseline LSTM-RNN model. Moreover these auxiliary vectors are complementary. The best performance is obtained using the i-vector+BSV-MT+spk-rate setup. The WER is improved from 32.4% to 30.6% (relative 6%).

4.5. Experiment on full set

Finally speaker-aware training on LSTM-RNNs is evaluated on the AMI IHM full corpus. For adaptation on LSTM-RNNs, we adopt the best configurations discovered in the previous section, i.e., BSV-MT and combining all auxiliary vectors.

Feature	WER
FMLLR	32.4
+i-vector	31.1
+BSV-MT	31.7
+spk-rate	31.9
+i-vector+BSV-MT	30.9
+i-vector+spk-rate	30.9
+i-vector+BSV-MT+spk-rate	**30.6**

Table 4. WER (%) of different auxiliary information on AMI IHM Subset

Feature	WER
FMLLR	26.0
+i-vector	24.3
+BSV-MT	25.0
+spk-rate	25.7
+i-vector+BSV-MT+spk-rate	**24.3**

Table 5. WER (%) of different auxiliary information on AMI IHM Full set

From Table 5, we can see that for each auxiliary vector, the adapted LSTM-RNN performed consistently better than the baseline LSTM-RNN. However, when using all speaker representative vectors together, no further improvement can be obtained. The reason may be that the amount of speakers on full set is large enough to capture all variability, and on full set the LSTM-RNN model can encode richer speaker variability by itself and thus the efficacy of speaker adaptation decreases. Even under this condition the speaker-aware trained LSTM-RNN still outperforms the baseline LSTM-RNN using FMLLR feature with 6.5% relative improvement.

5. CONCLUSION

In this paper, the speaker-aware training was studied on LSTM-RNNs. A simple but effective structure was found for LSTM-RNNs adaptation and two structures are proposed to incorporate phone information into BSV. In addition different auxiliary vectors were explored to test the efficiency for LSTM-RNN adaptation. Our experiments with the AMI corpus show that adaptation with i-vector significantly improves the performance of LSTM-RNN models and the improvement also can be obtained by using BSV and speaking rate. For the future work, we will focus on joint training of the speaker information extractor and the speech recognition model.

6. REFERENCES

[1] Herve A Bourlard and Nelson Morgan, *Connectionist speech recognition: a hybrid approach*, vol. 247, Springer Science & Business Media, 2012.

[2] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N Sainath, et al., "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[3] Yoshua Bengio, Patrice Simard, and Paolo Frasconi, "Learning long-term dependencies with gradient descent is difficult," *Neural Networks, IEEE Transactions on*, vol. 5, no. 2, pp. 157–166, 1994.

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

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

[6] Xiangang Li and Xihong Wu, "Constructing long short-term memory based deep recurrent neural networks for large vocabulary speech recognition," in *ICASSP*. IEEE, 2015, pp. 4520–4524.

[7] Yan Huang, Dong Yu, Chaojun Liu, and Yifan Gong, "A comparative analytic study on the Gaussian mixture and context dependent deep neural network hidden Markov models," in *INTERSPEECH*, 2014.

[8] Yu Dong and Deng Li, *Automatic speech recognition, A deep learning approach*, Springer.

[9] Frank Seide, Gang Li, Xie Chen, and Dong Yu, "Feature engineering in context-dependent deep neural networks for conversational speech transcription," in *ASRU*. IEEE, 2011, pp. 24–29.

[10] Kaisheng Yao, Dong Yu, Frank Seide, Hang Su, Li Deng, and Yifan Gong, "Adaptation of context-dependent deep neural networks for automatic speech recognition.," in *SLT*, 2012, pp. 366–369.

[11] Dong Yu, Kaisheng Yao, Hang Su, Gang Li, and Frank Seide, "Kl-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," in *ICASSP*. IEEE, 2013, pp. 7893–7897.

[12] Ossama Abdel-Hamid and Hui Jiang, "Fast speaker adaptation of hybrid nn/hmm model for speech recognition based on discriminative learning of speaker code," in *ICASSP*. IEEE, 2013, pp. 7942–7946.

[13] George Saon, Hagen Soltau, David Nahamoo, and Michael Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *ASRU*. IEEE, 2013, pp. 55–59.

[14] Romain Serizel and Diego Giuliani, "Vocal tract length normalisation approaches to dnn-based children's and adults' speech recognition," in *Spoken Language Technology Workshop (SLT), 2014 IEEE*. IEEE, 2014, pp. 135–140.

[15] Yajie Miao and Florian Metze, "On speaker adaptation of long short-term memory recurrent neural networks," in *INTERSPEECH*, 2015.

[16] Najim Dehak, Patrick Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 788–798, 2011.

[17] Hengguan Huang and Khe Chai Sim, "An investigation of augmenting speaker representations to improve speaker normalisation for dnn-based speech recognition," in *ICASSP*. IEEE, 2015, pp. 4610–4613.

[18] Xie Chen, Tian Tan, Xunying Liu, Pierre Lanchantin, Moquan Wan, Mark Gales, and Phil Woodland, "Recurrent neural network language model adaptation for multi-genre broadcast speech recognition," in *INTERSPEECH*, 2015.

[19] Tomas Mikolov and Geoffrey Zweig, "Context dependent recurrent neural network language model.," in *SLT*, 2012, pp. 234–239.

[20] Yajie Miao, Hao Zhang, and Florian Metze, "Towards speaker adaptive training of deep neural network acoustic models," in *INTERSPEECH*, 2014.

[21] Yun Lei, Nicolas Scheffer, Luciana Ferrer, and Moray McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *ICASSP*. IEEE, 2014, pp. 1695–1699.

[22] Dongpeng Chen, Brian Mak, Cheung-Chi Leung, and Sunil Sivadas, "Joint acoustic modeling of triphones and trigraphemes by multi-task learning deep neural networks for low-resource speech recognition," in *ICASSP*. IEEE, 2014, pp. 5592–5596.

[23] Nanxin Chen, Yanmin Qian, and Kai Yu, "Multi-task learning for text-dependent speaker verification," in *INTERSPEECH*, 2015.

[24] Matthew Siegler, Richard M Stern, et al., "On the effects of speech rate in large vocabulary speech recognition systems," in *ICASSP*. IEEE, 1995, vol. 1, pp. 612–615.

[25] Nelson Morgan, Eric Fosler-Lussier, and Nikki Mirghafori, "Speech recognition using on-line estimation of speaking rate.," in *Eurospeech*, 1997, vol. 97, pp. 2079–2082.

[26] Stephen M Chu and Daniel Povey, "Speaking rate adaptation using continuous frame rate normalization," in *ICASSP*. IEEE, 2010, pp. 4306–4309.

[27] Xiangyu Zeng, Shi Yin, and Dong Wang, "Learning speech rate in speech recognition," in *INTERSPEECH*, 2015, pp. 528–532.

[28] Thomas Hain, Luká Burget, John Dines, Philip N Garner, František Grézl, Asmaa El Hannani, Marijn Huijbregts, Martin Karafiat, Mike Lincoln, and Vincent Wan, "Transcribing meetings with the amida systems," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 20, no. 2, pp. 486–498, 2012.

[29] Pawel Swietojanski, Arnab Ghoshal, and Steve Renals, "Hybrid acoustic models for distant and multichannel large vocabulary speech recognition," in *ASRU*. IEEE, 2013, pp. 285–290.

[30] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukáš Burget, Ondřej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlíček, Yanmin Qian, Petr Schwarz, et al., "The kaldi speech recognition toolkit," 2011.

[31] Dong Yu, Adam Eversole, Mike Seltzer, Kaisheng Yao, Zhiheng Huang, Brian Guenter, Oleksii Kuchaiev, Yu Zhang, Frank Seide, Huaming Wang, et al., "An introduction to computational networks and the computational network toolkit," Tech. Rep., Microsoft Research, 2014.