Proceedings of the **4th** International Workshop on **Speech Processing in Everyday Environments**

CHiME 2016

San Francisco, 13th September 2016

Published by:
Emmanuel Vincent

ISBN: 978-2-9557814-0-1

Credits:
Editors: Emmanuel Vincent, Shinji Watanabe, Jon Barker, Ricard Marxer
Cover photo: By Christian Mehlführer, User:Chmehl - Own work, CC BY 2.5,
https://commons.wikimedia.org/w/index.php?curid=3971050
Proceedings assembled by: Ricard Marxer

Workshop took place in San Francisco, CA, USA — September 13, 2016

Published online at http://spandh.dcs.shef.ac.uk/chime/ — November 10, 2016

Workshop Organisation

Organising Committee

Emmanuel Vincent Inria, France

Shinji Watanabe MERL, USA

Jon Barker Univ. of Sheffield, UK

Ricard Marxer Univ. of Sheffield, UK

Local Organiser

Kean Chin Google

Scientific Committee

Jen-Tzung Chien National Chiao Tung Univ.

Kean Chin Google

Hakan Erdogan Sabanci Univ.

Qiang Fu Inst. of Acoustics, Chinese Acad. of Sciences

Stefan Goetze Fraunhofer IDMT

Reinhold Haeb-Umbach Univ. of Paderborn

Michael I. Mandel City Univ. of New York

Marco Matassoni FBK

Bernd T. Meyer Carl von Ossietzky Univ. Oldenburg

Vikramjit Mitra SRI International

Arun Narayanan Google

Franz Pernkopf Graz Univ. of Technology

Björn Schuller Univ. of Passau

Masahito Togami Hitachi

Dat Huy Tran Inst. for Infocomm Research

Stavros Tsakalidis Raytheon BBN Technologies

Hugo van Hamme KU Leuven

Tuomas Virtanen Tampere Univ. of Technology

Xiong Xiao Nanyang Technological Univ.

Sponsors

Google

MITSUBISHI ELECTRIC
RESEARCH LABORATORIES

Conference Program

Poster Session 1

Oral Session

Poster Session 2

The MELCO/MERL System Combination Approach for the Fourth CHiME Challenge

Yuuki Tachioka[1], *Shinji Watanabe*[2], *Takaaki Hori*[2]

[1]Information Technology R&D Center, Mitsubishi Electric Corporation
[2]Mitsubishi Electric Research Laboratories
Tachioka.Yuki@eb.MitsubishiElectric.co.jp, watanabe@merl.com, thori@merl.com

Abstract

This paper describes our approach for all three tracks of the fourth CHiME challenge. Front-end process prepared two speech enhancements. Back-end process extracted three types of different features and after decoding, it used neural network based rescoring. Finally, the hypotheses of the multiple systems were combined and the word error rate of our best system became less than half of that of the state-of-the-art baseline.

1. Background

The 4th CHiME challenge provides three tracks: 1ch, 2ch, and 6ch track [1]. We entered all three tracks. For all tracks, state-of-the-art baseline scripts were prepared. They employed discriminatively trained deep neural network (DNN) acoustic models and recurrent neural network (RNN) based rescoring with advanced speech enhancement. There are four different environments in the tasks and for these kinds of tasks, system combination was effective. To realize more effective combination, we prepared multiple systems with different speech enhancement and different feature extractions. This paper separately confirmed the effectiveness of our approach in terms of the word error rate (WER).

2. Front-end process

For single-channel track, sparse non-negative matrix factorization (NMF) [2] was used to suppress noise. To reduce distortions, enhanced speech was mixed with original noisy speech. For multi-channel track, in addition to the provided beamformer (BeamformIt), minimum variance distortionless response (MVDR) beamformer with precise steering vector estimation [3] was employed.

3. Back-end process

In addition to the provided 13-dimensional MFCC $+\Delta+\Delta\Delta$ with feature-space maximum likelihood linear regression (fM-

Table 1: System description for Table 2. All systems used DNN acoustic model.

{m,p,f}-{s,m}-{n,s,b,m}-{u,a,a2}+{r,l}	
{m,p,f}	MFCC / PLP / fbank
{s,m}	Single / multi-channel data training
{n,s,b,m}	Noisy / sparse NMF / BeamformIt / MVDR
{u,a,a$_2$}	Unadapted / adapted / adapted-2 DNN
{r,l}	RNN / LSTM-LM rescoring

Table 2: Average WER [%] for the tested systems. For 1ch, “baseline1” was “m-s-n-u” and “baseline2” was “m-s-n-u+r”. For 2ch and 6ch, “baseline1” was “m-s-b-u” and “baseline2” was “m-s-b-u+r”. “best” combined asterisk-marked systems.

Track	System	Dev		Test	
		real	simu	real	simu
1ch	baseline1	14.67	15.67	27.69	24.15
	baseline2	11.69	15.43	23.71	20.95
	m-m-n-u	12.67	13.55	22.17	20.29
	m-m-n-u+l*	7.76	8.92	15.66	15.12
	p-m-n-u+l*	7.74	9.23	16.03	15.31
	f-m-n-u+l*	5.60	7.60	11.76	12.75
	f-m-n-a+l*	5.58	7.70	11.85	12.72
	m-m-s-u+l*	7.78	8.86	15.49	15.08
	p-m-s-u+l*	7.60	9.33	15.47	15.61
	f-m-s-u+l*	5.56	7.30	11.64	12.76
	f-m-s-a+l*	5.41	7.48	11.64	12.90
	best	5.15	7.15	11.13	12.15
2ch	baseline1	10.90	12.36	20.44	19.03
	baseline2	9.63	10.72	18.08	16.88
	m-m-b-u	9.90	10.60	16.89	16.27
	m-m-b-u+l*	5.59	6.33	11.43	10.55
	p-m-b-u+l*	5.51	6.48	11.71	10.77
	f-m-b-u+l*	4.19	5.23	8.38	9.10
	f-m-b-a+l*	3.96	5.15	8.23	8.49
	m-m-m-u+l*	5.34	6.09	11.21	11.55
	p-m-m-u+l*	5.03	6.40	11.11	11.61
	f-m-m-u+l*	3.96	5.23	8.45	9.62
	f-m-m-a+l*	3.80	5.06	7.99	9.10
	best	3.50	4.63	7.28	8.03
6ch	baseline1	8.14	9.07	15.04	14.20
	baseline2	5.75	6.77	11.47	10.91
	m-m-b-u	7.69	8.23	12.57	12.66
	m-m-b-u+r	4.99	5.72	9.22	8.96
	m-m-b-u+l*	3.94	4.49	7.77	7.51
	p-m-b-u+l*	3.90	4.62	7.64	7.71
	f-m-b-u+r	4.18	4.95	7.20	7.47
	f-m-b-u+l*	3.10	3.63	5.94	6.28
	f-m-b-a+l*	3.05	3.60	5.71	5.94
	m-m-m-u+r	4.45	4.19	7.45	7.51
	m-m-m-u+l*	3.47	3.06	6.42	6.39
	p-m-m-u+l*	3.43	2.99	6.36	6.23
	f-m-m-u+r	3.72	3.66	6.11	6.67
	f-m-m-u+l*	2.75	2.61	5.19	5.72
	f-m-m-a+l*	2.60	2.53	5.06	5.01
	f-m-m-a$_2$+l*	2.47	2.45	4.75	4.39
	best	2.30	2.32	4.31	4.18

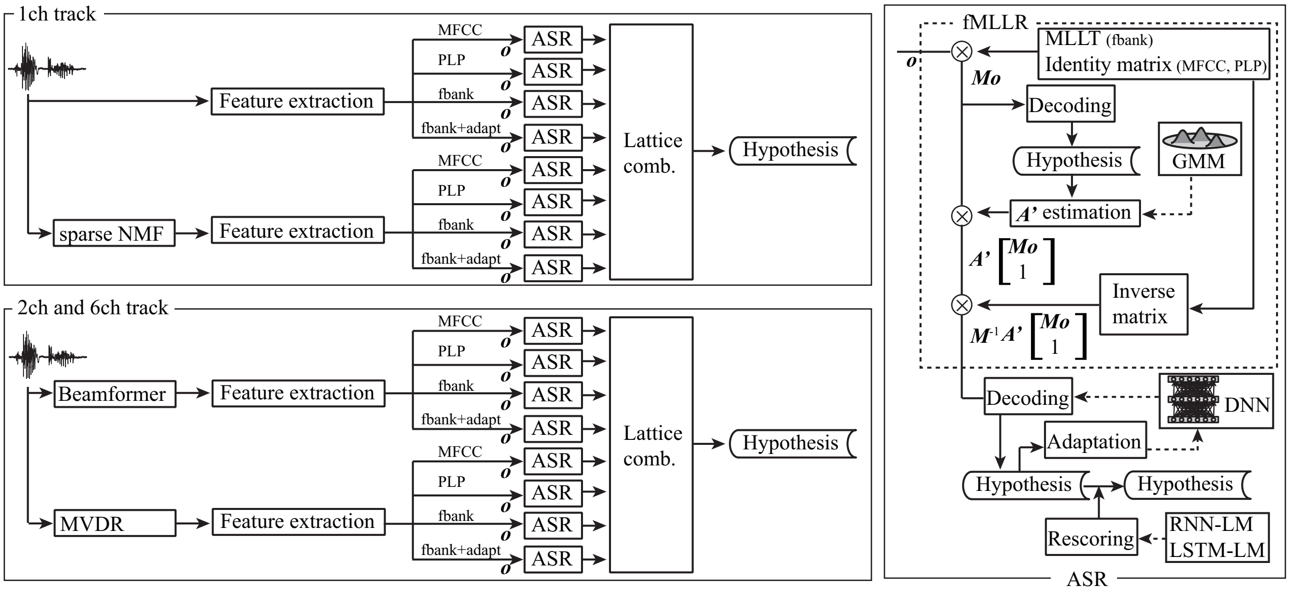


Figure 1: Schematic diagram of the proposed ASR systems.

Table 3: WER [%] per environment for the best system.

Track	Envir.	Dev		Test	
		real	simu	real	simu
1ch	BUS	7.15	6.24	18.00	8.55
	CAF	5.19	9.81	11.73	13.93
	PED	3.05	4.97	7.81	11.71
	STR	5.19	7.57	6.99	14.40
2ch	BUS	4.54	3.92	11.42	5.08
	CAF	3.63	6.28	7.08	9.41
	PED	2.21	3.38	5.59	8.33
	STR	3.63	4.96	5.04	9.28
6ch	BUS	3.07	2.01	5.16	2.95
	CAF	2.40	2.99	3.90	4.63
	PED	1.64	1.76	4.00	4.18
	STR	2.11	2.51	4.17	4.97

LLR) transformation, we employed 13-dimensional PLP $+\Delta+\Delta\Delta$ with fMLLR transformation and 40-dimensional filterbank (fbank) feature $+\Delta+\Delta\Delta$ with maximum likelihood linear transformation (MLLT) and fMLLR transformation [4]. Features in the consecutive 11 frames were input to the DNN.

In addition to the feature-space adaptation, model-space adaptation of DNN [5] was also used where the second layer of DNN was switched for each speaker. To train DNN acoustic models, multi-channel (6ch) data were all used whereas baseline only used single-channel data. These modification increased the training data size [3]. All training data were noisy without any speech enhancement, i.e., noisy data training.

After decoding, we used long short-term memory (LSTM)-language model (LM) rescoring [6] instead of the baseline recurrent neural network (RNN)-LM. Figure 1 shows the schematics of the proposed method. In each track, there were two types of speech enhancement. For each enhancement, three different features were used; and for fbank feature, model-space speaker adaptation was performed. In total, hypotheses of eight systems are combined by using lattice combination.

4. Experimental evaluation

Table 2 shows the WERs of the challenge. Descriptions of the system ID is shown in Table 1. Comparison of baseline1 and "m-m-n-u" shows the effectiveness of multi-channel data training, which was especially effective for 1ch track and improved the WERs by around 2–5%. Comparison of baseline1 and baseline2 and that of "m-m-n-u" and "m-m-n-u+l" show the effectiveness of LSTM-LM rescoring, which improved WER more than RNN-LM rescoring. The performances of MFCC and PLP features were almost equivalent but fbank feature significantly improved the WERs. DNN model adaptation was also effective. MVDR beamformer shows its effectiveness for the 6ch track more than 2ch track, compared with the baseline beamformer. Combining multiple systems additionally improved WERs by around 0.3–0.6%. WERs of the best system were less than half of those of "baseline2" except one case (Test and simu in the 1ch track).

Table 3 shows the WER of the best system per environment in Table 2. Increasing the number of microphones was effective for all conditions. In real data, "BUS" was the most difficult task.

5. Conclusion

This paper showed our approach for the fourth CHiME challenge. Multi-channel data training, fbank feature, and LSTM-LM based rescoring were the most effective. System combination gave additional improvements for all conditions.

6. References

[1] E. Vincent, S. Watanabe, A. A. Nugraha, J. Barker, and R. Marxer, "An analysis of environment, microphone and data simulation mismatches in robust speech recognition," *Computer Speech and Language*, to appear, 2016.

[2] J. Eggert and E. Komer, "Sparse coding and NMF," in *Proceedings of the IEEE International Joint Conference on Neural Networks*, vol. 4. IEEE, 7 2004, pp. 2529–2533.

[3] T. Yoshioka, N. Ito, M. Delcroix, A. Ogawa, K. Kinoshita, M. Fujimoto, C. Yu, W. J. Fabian, M. Espi, T. Higuchi, S. Araki, and

T. Nakatani, “The NTT CHiME-3 system: Advances in speech enhancement and recognition for mobile multi-microphone devices,” in *Proceedings of ASRU*. IEEE, 12 2015, pp. 436–443.

[4] T. N. Sainath, B. Kingsbury, A. R. Mohamed, G. E. Dahl, G. Saon, H. Soltau, T. Beran, A. Y. Aravkin, and B. Ramabhadran, “Improvements to deep convolutional neural networks for LVCSR,” in *Proceedings of ASRU*. IEEE, 12 2013, pp. 315–320.

[5] T. Ochiai, S. Matsuda, H. Watanabe, X. Lu, C. Hori, and S. Katagiri, “Speaker adaptive training for deep neural networks embedding linear transformation networks,” in *Proceedings of ICASSP*. IEEE, 4 2015, pp. 4605–4609.

[6] T. Hori, C. Hori, S. Watanabe, and J. Hershey, “Minimum word error training of long short-term memory recurrent neural network language models for speech recognition,” in *Proceedings of ICASSP*. IEEE, 3 2016.

LSTM Network Supported Linear Filtering For The CHiME 2016 Challenge

Xiaofei Wang, Ziteng Wang, Xu Li, Yueyue Na, Qiang Fu, Yonghong Yan

Insitute of Acoustics, Chinese Academy of Sciences

xiaofei.wang1987@gmail.com, wangziteng@hccl.ioa.ac.cn

Abstract

This paper explores the combination of the emerging long short-term memory (LSTM) and the well established linear filtering techniques, parametric multi-channel Wiener filtering (PMWF) as well as single-channel minimum variance distortionless response (MVDR), for robust front-end signal processing in a speech recognition system. LSTM is employed for the estimation of speech and noise statistics, which are then used to compute the filter coefficients. PMWF is utilized in a novel way that the residual noise power remains constant along the frequency axis, while single-channel MVDR exploits inter-frame correlation coefficient vector, taking advantage of LSTM network based mask prediction, for linear filter estimation. With the baseline recognition system, our proposed methods reach a final word error rates (WER) of 5.69% on the 6ch real evaluation set of CHiME-4 challenge.

Keywords: CHiME 2016 Challenge, Supervised Time-frequency Masking, Parametric Multichannel Wiener Filtering, Single-channel MVDR

1. Introduction

The technique of neural network has greatly promoted speech recognition in everyday environments. It also quickly expands its scope to the signal processing area. Articles apply deep neural network (DNN) for spectral mask estimation [1] or predicting the clean spectrum [2]. Both tasks report promising results. However, most neural network based approaches only deal with problems in the signal channel case.

While multi-channel algorithms are more capable of extracting the desired source and suppressing undesired components at the same time, microphone arrays are becoming commonplace in modern human-machine interaction systems. The well established minimum variance distortionless response (MVDR) and multi-channel Wiener filter (MWF), which have solid theoretical foundations, arose new interests.

MVDR filter is also proposed for single-microphone noise reduction [3]. This filter takes the speech correlations of consecutive time frames into account. Under the assumption that noise spectrum is known previously, the MVDR filter could achieve promising performance in terms of speech distortion which is a key factor that affects speech recognition accuracy rate.

For the task of robust speech recognition of CHiME-4 [4], one practical front-end signal processing solution is the combination of the above two techniques [5][6]. DNN deals well with the noisy data and makes no extra assumptions as in conventional methods. Meanwhile, the multi-channel algorithms and single-channel MVDR provide optimized solutions.

Specifically, long short-term memory (LSTM) is employed for the estimation of speech and noise masks as originally suggested in [6][7]. With short-time Fourier transform performed in 1024 points, the network input is of 513 nodes. We have the following one bi-directional LSTM layer of 256 nodes and two feed-forward layers of 513 nodes. The training targets are ideal binary masks of both speech and noise, which are calculated by weighting the local signal-to-noise ratio (SNR) and the local threshold (LC)

$$\mathcal{M} = \begin{cases} 1, & \text{SNR} > LC \\ 0, & else \end{cases} \tag{1}$$

The Adam optimization algorithm [8] is used for tuning the network. Dropout and batch normalization techniques are also employed for improving the generalization performance.

In the testing phase, the predicted masks $\mathcal{M}'_c$ ($c = speech$ and $noise$) are used to calculate the power spectral density (PSD) matrixes that are needed by our proposed PMWF and MVDR.

$$\mathbf{\Phi}_{cc} = \sum \mathcal{M}'_c \, \mathbf{y}\mathbf{y}^H \tag{2}$$

where $\mathbf{y}$ is the observation vector and superscript H denotes Hermitian transpose.

2. Parametric multi-channel Wiener filter

In the 2ch and 6ch tasks, the multi-channel processing problem is formulated in the frequency domain. With an array of M microphones, we have

$$Y_p(j\omega) = X_p(j\omega) + N_p(j\omega), \quad p = 1, 2, ..., P \tag{3}$$

In order to extract the desired source $X(j\omega)$ from the noisy observations, we apply an optimal filter $\mathbf{h}(j\omega)$

$$X(j\omega) = \mathbf{h}^H(j\omega)\mathbf{y}(j\omega) \tag{4}$$

where $\mathbf{y}(j\omega) = [Y_1(j\omega)...Y_p(j\omega)...Y_P(j\omega)]^T$.

The solution of PMWF [9][10] is known as

$$\mathbf{h}(j\omega) = \frac{\mathbf{\Phi}_{nn}^{-1}(j\omega)\mathbf{\Phi}_{xx}(j\omega)}{\mu + \lambda(\omega)}\mathbf{u}_{ref} \tag{5}$$

where Φ_{nn}, Φ_{xx} are respectively the noise and speech PSD matrixes which can be derived by (2), $\mathbf{u}_{ref}$ is one zero vector except for the index of reference channel being one (The first channel was used as reference in CHiME-4). $\lambda(\omega) = \text{tr}\{\mathbf{\Phi}_{nn}^{-1}(j\omega)\mathbf{\Phi}_{xx}(j\omega)\}$, μ is the hyper-parameter that controls the tradeoff between speech distortion and noise reduction. With a higher value, we get more noise reduction at the expense of more distortion.

In speech recognition applications, it is still unclear how the speech distortion and noise reduction factors will influence the final recognition performance. Here, we propose a novel parameter control strategy that proves quite effective. Particularly, the residual noise power (RNP) in the filter output is constrained to be constant along the frequency axis. From Eq.(5), the output RNP is

$$\mathbf{h}^H(j\omega)\mathbf{\Phi}_{nn}(j\omega)\mathbf{h}(j\omega) = \frac{\phi_{x_{ref}x_{ref}}\lambda(\omega)}{[\mu+\lambda(\omega)]^2} \quad (6)$$

We denoted the desired RNP as r_{nn}. Hence, we have

$$\mu(\omega) = \sqrt{\phi_{x_{ref}x_{ref}}(\omega)\lambda(\omega)/r_{nn}} - \lambda(\omega) \quad (7)$$

It should be noted that the value of r_{nn} only scales the output rather than changes the spectral shape of speech. It is set to 1.0. By regulating the RNP, a bin-wise controller $\mu(\omega)$ is derived. The reason why r_{nn} is constant across frequencies is that the spectrums of filtered signals would be preferred flat, avoiding transient changes between adjacent bins.

3. Single-channel MVDR

In the 1ch task, the single-channel problem is formulated as follows in the frequency domain. The complex spectral noisy observation $Y(k,m)$ is thus given by

$$Y(k,m) = X(k,m) + N(k,m) \quad (8)$$

where k is the frequency bin number and m is the frame index. The estimate of the clean speech spectral component $X(k,m)$ is obtained by applying an FIR filter

$$\hat{X}(k,m) = \mathbf{h}^H(k,m)\mathbf{y}(k,m) \quad (9)$$

where L is the order of the filter (set 20), and

$$\mathbf{h}(k,m) = [H(k,m,0)...H(k,m,L-1)]^T \quad (10)$$

$$\mathbf{y}(k,m) = [Y(k,m)...Y(k,m-L+1)]^T \quad (11)$$

By introducing the speech inter-frame correlation (IFC) coefficient vector $\gamma_x(k,m)$, which is defined by (The operator $E[\cdot]$ denotes the expectation),

$$\gamma_x(k,m) = \frac{E[\mathbf{x}(k,m)X(k,m))]}{E[\|X(k,m)\|^2]} \quad (12)$$

Therefore, from [3] the single-channel MVDR filter is

$$\mathbf{h}_{mvdr}(k,m) = \frac{\mathbf{\Phi_y}^{-1}(k,m)\gamma_x^*(k,m)}{\gamma_x^T(k,m)\mathbf{\Phi_y}^{-1}(k,m)\gamma_x^*(k,m)} \quad (13)$$

$$\mathbf{\Phi_y}(k,m) = \lambda_y\mathbf{\Phi_y}(k,m)+(1-\lambda_y)\mathbf{y}(k,m)\mathbf{y}^H(k,m) \quad (14)$$

where λ_y is the forgetting factor(set 0.95). Also, to calculate $\mathbf{\Phi_y}^{-1}(k,m)$, the regularization is used,

$$\mathbf{\Phi_y}^{-1}(k,m) = \{\mathbf{\Phi_y}(k,m) + \frac{\delta \cdot tr[\|\mathbf{\Phi_y}(k,m)\|]}{L}\mathbf{I}_{L\times L}\}^{-1} \quad (15)$$

where $\delta > 0$ is the regularization parameter (set 0.04).

Specifically, IFC $\gamma_x(k,m)$ can be estimated as follows,

$$\begin{aligned}\gamma_x(k,m) &= \frac{\Phi_Y(k,m)}{\Phi_Y(k,m)-\Phi_N(k,m)}\gamma_y(k,m)\\ &- \frac{\Phi_N(k,m)}{\Phi_Y(k,m)-\Phi_N(k,m)}\gamma_n(k,m) \quad (16)\end{aligned}$$

$\Phi_Y(k,m)$ and $\Phi_N(k,m)$ represent the second-order statistics of observed signal $Y(k,m)$ and noise $N(k,m)$, respectively.

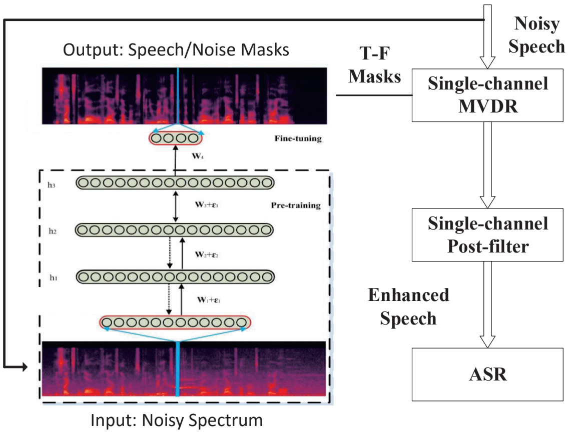


Figure 1: Diagram for the 1ch recognition task.

We use the speech soft mask $\mathcal{M}'_c$ to get the estimated noise component $\hat{N}(k,m)$ as follows,

$$\hat{N}(k,m) = (1-\mathbf{max}(\epsilon, 1-\mathbf{max}(\sqrt{\mathcal{M}'_c}, \epsilon)))Y(k,m) \quad (17)$$

where ϵ is an extremely small number to avoid sudden changes between frames.

Following the single-channel MVDR filtering, a stationary noise reduction algorithm [11][12] is applied to the filtered signal as a post-filter shown in Fig.1.

4. Experimental evaluation

4.1. 2ch and 6ch results

For all the recognition tasks, we always apply matched training. In the case of 2ch track, we randomly select two channels from all six channels to compose the training set. The channels selected for development and evaluation are kept unmodified. In the back-end (2ch and 6ch tasks), only one modification is made to the standard scripts. We make use of the fact that we have all six channels data available. Besides the enhanced data, we also use all six channel real and one channel simulated recordings in the training stage.

In the front-end, LSTM is trained with all the six channel simulated data [7]. The mask estimation is actually single-channel based, so we get separate outputs for each channel. For 2ch and 6ch tasks, the masks are then taken median between specific channels for robustness to outliers.

The results of 2ch and 6ch tasks using sequence training and RNN language model rescoring are given in Table 1, 2. The WERs of real test data in the 2ch and 6ch tasks are 9.64% and 5.69%, respectively.

4.2. 1ch results

In the 1ch task, we use 6 channels' data for matched training. The results are given by Table 3. A relative 15.59% WER decrease on real test data using GMM acoustic model is achieved compared to the official baseline, in which both training and testing data are noisy signals. Single-channel MVDR and post-filtering achieve the best performance since MVDR could filter the non-stationary noise without speech distortion, meanwhile, post-filtering is good at suppressing the stationary noise. Single-channel MVDR and post-filtering benefit each other.

Table 1: Average WER (%) for the multi-channel tested systems.

Track	System	Dev		Test	
		real	simu	real	simu
2ch	Baseline	8.23	9.50	16.58	15.33
	GMM	12.95	16.06	21.08	20.53
	DNN+sMBR	8.34	9.54	12.16	13.27
	DNN+RNNLM	**5.58**	**7.18**	**9.64**	**8.77**
6ch	Baseline	5.76	6.77	11.51	10.90
	GMM	9.25	9.24	12.70	10.49
	DNN+sMBR	5.43	5.19	8.25	6.51
	DNN+RNNLM	**3.65**	**3.71**	**5.69**	**4.38**

Table 2: WER (%) per environment for the current multi-channel best system.

Track	Envir.	Dev		Test	
		real	simu	real	simu
2ch	BUS	6.73	5.68	12.82	6.16
	CAF	5.97	10.10	10.59	10.29
	PED	4.34	6.03	8.56	9.15
	STR	5.26	6.92	6.57	9.47
6ch	BUS	4.81	3.33	7.35	3.46
	CAF	3.20	4.69	5.27	4.76
	PED	2.99	3.07	5.66	4.28
	STR	3.58	3.75	4.50	5.01

Table 3: Average WER (%) for the 1ch tested systems using the GMM-HMM acoustic model. Baseline(official) used only CH5 for training. Baseline(all channels) used all the 6 channels' data for training. sMVDR/sMVDR + Postfilter means all the 6 channels' data and test data are processed with single-channel MVDR/MVDR + Postfilter.

System	Dev		Test	
	real	simu	real	simu
Baseline(official)	22.15	24.49	37.54	33.3
Baseline(all channels)	20.87	23.07	35.01	31.65
sMVDR	19.71	**20.76**	34.64	28.96
sMVDR + Postfilter	**19.27**	20.92	**31.69**	**27.70**

Table 4: Average WER (%) for the 1ch tested systems using the DNN acoustic model (without sMBR and RNNLM rescore).

System	Dev		Test	
	real	simu	real	simu
Baseline(official)	14.86	15.47	27.27	24.09
Baseline(all channels)	**14.10**	**15.25**	25.74	22.79
sMVDR	15.24	16.10	25.21	**21.73**
sMVDR + Postfilter	15.90	16.92	**25.05**	22.34

Besides GMM acoustic model, results of DNN acoustic model are also given in Table 4. The best results of test set are achieved using single-channel MVDR or single-channel MVDR + postfiltering, however, the best results of development set are achieved using unprocessed data. This phenomenon is different from the consistent improvements using GMM acoustic model. It is still expected to achieve a better tradeoff between noise reduction and distortion.

5. Conclusion

The main contributions of the submitted systems were two proposed front-end processing methods, which were multi-channel and single-channel noise reductions for specific recognition tasks, respectively. With a fine-tuning parametric multi-channel Wiener filter, WERs on 2ch and 6ch Real Test sets of CHiME-4 were reduced to 9.64% and 5.69%. Meanwhile, supervised time-frequency masking based single-channel MVDR filter with a post-filter performed well in the 1ch task. The results showed that WER of Real Test set decreased much on GMM acoustic model but slightly on DNN model. Experimental results also showed that enlarging the training data could bring benefits for CHiME-4 tasks.

6. Acknowledgement

This work is supported by the National Natural Science Foundation of China (No. 61601453) and the China Scholarship Council (No. 201604910007).

7. References

[1] Y. Wang, A. Narayanan, and D. Wang, "On training targets for supervised speech separation," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 22, no. 12, pp. 1849–1858, 2014.

[2] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE Signal Processing Letters*, vol. 21, no. 1, pp. 65–68, 2014.

[3] J. Benesty and Y. Huang, "A single-channel noise reduction mvdr filter," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2011, pp. 273–276.

[4] E. Vincent, S. Watanabe, A. A. Nugraha, J. Barker, and R. Marxer, "An analysis of environment, microphone and data simulation mismatches in robust speech recognition," *Computer Speech and Language, to appear.*

[5] S. Sivasankaran, A. A. Nugraha, E. Vincent, J. A. Morales-Cordovilla, S. Dalmia, I. Illina, and A. Liutkus, "Robust asr using neural network based speech enhancement and feature simulation," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2015, pp. 482–489.

[6] J. Heymann, L. Drude, A. Chinaev, and R. Haeb-Umbach, "Blstm supported gev beamformer front-end for the 3rd chime challenge," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. IEEE, 2015, pp. 444–451.

[7] J. Heymann, L. Drude, and R. Haeb-Umbach, "Neural network based spectral mask estimation for acoustic beamforming," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 196–200.

[8] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[9] J. Benesty, J. Chen, Y. Huang, and B. Rafaely, "Microphone array signal processing," *Journal of the Acoustical Society of America*, vol. 125, no. 6, pp. 4097–4098, 2009.

[10] M. Souden, J. Benesty, and S. Affes, "On optimal frequency-domain multichannel linear filtering for noise reduction," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 2, pp. 260–276, 2010.

[11] Y. Ephraim and D. Malah, "Speech enhancement using a minimum mean-square error log-spectral amplitude estimator," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 33, no. 2, pp. 443–445, 1985.

[12] R. Martin, "Noise power spectral density estimation based on optimal smoothing and minimum statistics," *IEEE Transactions on speech and audio processing*, vol. 9, no. 5, pp. 504–512, 2001.

The THU-SPMI CHiME-4 system : Lightweight design with advanced multi-channel processing, feature enhancement, and language modeling

Hongyu Xiang, Bin Wang, Zhijian Ou

Speech Processing and Machine Intelligence (SPMI) Lab, Tsinghua University, Beijing, China
Contact: ozj@tsinghua.edu.cn

Abstract

In this paper, we describe our lightweight system designed for CHiME-4. For multi-channel processing, we experiment with a bundle of beamforming methods, including minimum variance distortionless response (MVDR), parameterized multi-channel wiener filter (PMWF), generalized sidelobe canceller (GSC), spectral mask estimation (ME), and compare these techniques with the same back-end. Combining MVDR's distortionless and reliable estimation of the steering vector by ME is found to be most effective. We propose to applying histogram equalization (HEQ) to compensate for the residual noise in the MVDR beamformed speech. We apply the recently introduced trans-dimensional random field (TRF) language model and confirm its superiority in rescoring. In combination these techniques are surprisingly effective in the CHiME-4 task, achieving 6.55% word error rate (WER) for the real evaluation data while keeping low system complexity. Applying multi-channel training further reduces the WER to 5.81%.

1. Background

The performance of automatic speech recognition (ASR) has been significantly improved in recent years. However, robust ASR in everyday environments remains a challenge. Research efforts can be roughly decomposed into developing more powerful front-ends (e.g. microphone array signal processing, feature enhancement) and back-ends (e.g. acoustic modeling, language modeling).

For front-ends, some widely used beamforming techniques are minimum variance distortionless response (MVDR) [1], parameterized multi-channel wiener filter (PMWF) [2], generalized sidelobe canceller (GSC) [3], and weighted delay and sum (WDAS) [4]. Beamforming filters could be designed based on different criteria, representing different trade-offs between distortion and noise reduction. For example, MVDR minimizes the output energy subject to no distortion in the desired direction. It is known that the effectiveness of beamformers heavily relies on the estimation of the spatial correlation matrix, the steering vector or time delays, which are usually difficult to estimate in practice. Researchers have explored to estimate the spatial correlation matrix using time-frequency masks, which are obtained either by complex Gaussian mixture models (GMMs) [5] or advanced neural networks [6]. For back-ends, neural network based acoustic models have become the state-of-the-art in speech recognition [7]. Neural network based language models (LMs) have also begun to surpass the classic n-gram LMs [8,9].

The CHiME-4 challenge [10] revisits the CHiME-3 data [11], i.e., WSJ0 corpus sentences spoken by talkers situated in challenging noisy environments recorded via a 6-microphone tablet device. The aim is to provide a new benchmark task for evaluating and promoting far-field speech recognition in everyday environments.

The CHiME-3 baseline uses MVDR beamformer with diagonal loading [12] as the front-end. The back-end is based on the Kaldi toolkit [13] and consists of a GMM-HMM using fMLLR transformed features to provide senone state alignment and a DNN using fbank features. The DNN is trained using sequence discriminative training with state-level mimimum Bayes risk (sMBR) criterion. After CHiME-3, an upgraded Kaldi-based baseline script was made available for CHiME-4 task, which further incorporates multichannel enhancement using WDAS based BeamformIt [4], fMLLR features for the DNN stage, interpolated 5-gram LM and RNN LM for rescoring. The CHiME-4 baseline produces an average WER of 11.57% for the real evaluation data (obtained by our own run).

This paper presents the THU-SPMI system designed for CHiME-4. For time constraint, we only submit results for 6-channel track, although the techniques developed in this submission could be applied to 1-channel track and 2-channel track.

2. Contributions

The goal of this study is to create a lightweight advanced system for far-field multi-channel speech recognition, which can achieve a good trade-off between system complexity and system performance, and is practically useful. To this end, we do not rely on feature fusion (e.g. extracting multiple types of features) or hypothesis fusion (e.g. training multiple systems and doing ROVER), though these are provably beneficial. We are selective to integrate front-end and back-end techniques and stay simple. Specifically, we identify the following three key techniques which enable us to significantly improve over the baseline while keeping low system complexity.

1) For multi-channel processing, after experiments with a bundle of beamforming methods, the MVDR beamformer with the steering vector being estimated by time-frequency masks, as proposed in [5], is found to be most effective. Our contribution is extensive comparisons between various beamformers with the same back-end.

2) Note that the MVDR beamformer reduces the noise under the distortionless constraint of any signal from the source direction. There are few artifacts in the beamformed speech, but there still exists considerable residual noise. We propose to apply histogram equalization (HEQ) technique for feature normalization, which is originally studied for single-channel feature enhancement [14]. WERs are found to be significantly reduced by using HEQ after the MVDR beamformer, which is an important empirical finding from this study.

3) Recently, we have shown in previous work [15, 16] with open source code [17] that a new trans-dimensional random

This work is supported by NSFC grant 61473168.

Table 1: Average WER (%) for the CHiME-4 baseline system obtained by our own run.

Track	System	Dev		Test	
		real	simu	real	simu
6ch	GMM	12.90	14.35	21.55	21.09
	DNN sMBR	8.12	9.37	14.84	14.38
	KN5+RNN	5.89	6.97	11.57	10.66

field (TRF) LM achieves superior performance. In the CHiME-4 task, we confirm that interpolated TRF and LSTM performs better than using LSTM alone, and produces significantly better rescoring performance than interploted 5-gram-KN and RNN provided in the CHiME-4 baseline. This represents an advance of the state-of-the-art of language model rescoring.

3. Experimental evaluation

3.1. System overview

Basically, the proposed system follows the pipeline of the CHiME-4 baseline, and is strengthened with the three techniques which are highlighted before and will be introduced and evaluated in the following. Table 1 shows the WER results for the baseline, which are obtained by our own run. Starting from the baseline, we incrementally investigate the relative contribution of each technique from front-end to back-end, and show that in combination they are surprisingly effective for the CHiME-4 task, ultimately achieving 6.55% WER for the real evaluation data. Applying multi-channel training further reduces the WER to 5.81%, which was conducted after the CHiME-4 submission.

3.2. Beamforming

We experiment with a bundle of beamforming methods, which will be briefly introduced below. The experimental results are shown in Table 2, with the same back-end.

3.2.1. Signal model

In the time domain, most beamforming methods assume the following signal model:

$$x_i(t) = s(t) * h_i(t) + n_i(t) \tag{1}$$

where $x_i(t)$ is the i-th microphone signal, $s(t)$ is the source signal, $h_i(t)$ is the impulse response from the source to the i-th microphone, and $n_i(t)$ is the additive noise.

In frequency domain, we have

$$\mathbf{X}(t,\omega) = S(t,\omega)\mathbf{d}(\omega) + \mathbf{N}(t,\omega) = \mathbf{G}(t,\omega) + \mathbf{N}(t,\omega) \tag{2}$$

where $S(t,\omega)$, $\mathbf{X}(t,\omega)$, $\mathbf{N}(t,\omega)$ are the STFT coefficients of the desired source signal, the microphone signal vector and the noise signal vector respectively. $\mathbf{d}$ denotes the steering vector. For convenience, we omit t and ω in the following description.

3.2.2. Weighted delay and sum (WDAS)

WDAS simply aligns different channels in time and sums them together as follows:

$$y(t) = \sum_i w_i x_i(t - \tau_i) \tag{3}$$

where τ_i is the time delay from the source to the i-th micophone, w_i is the weight. The CHiME-4 baseline BeamformIt [4] is based on WDAS, where time delays are estimated by use of generalized cross correlation with phase transform (GCC-PHAT) [18] and two-step Viterbi postprocessing.

3.2.3. Minimum variance distortionless response (MVDR)

MVDR is designed to minimize the output energy subject to no distortion in the desired direction:

$$\min_{\mathbf{W}} \mathbf{E}||\mathbf{W^H X}||^2 \; s.t. \mathbf{W^H d} = 1 \tag{4}$$

which has the well-known closed-form solution

$$\mathbf{W} = \frac{\mathbf{\Phi_{NN}^{-1} d}}{\mathbf{d^H \Phi_{NN}^{-1} d}} \tag{5}$$

where $\mathbf{\Phi_{NN}}$ is the noise correlation matrix, $\mathbf{H}$ denotes conjugate transposition.

The performance of MVDR relies heavily on the estimation of the noise correlation matrix $\mathbf{\Phi_{NN}}$ and the steering vector $\mathbf{d}$. The steering vector could be estimated by time delays τ_i, $\mathbf{d} = [e^{-j\omega\tau_1}, e^{-j\omega\tau_2}, ...]$, as did in the CHiME-3 baseline. A recent method studied in [5], denoted as MVDR-EV, is to obtain the steering vector from the principal eigenvector of the estimated spatial corelation matrix of clean signal $\mathbf{\Phi_{GG}} = \mathbf{\Phi_{XX}} - \mathbf{\Phi_{NN}}$, and use complex GMM based spectral mask estimation (ME) method to esimate $\mathbf{\Phi_{NN}}$.

Allowing the desired direction gain to be the reference component of $\mathbf{d}$, we obtain MVDR with relative transfer function (MVDR-RTF) [2]. Assuming the first channel to be the reference channel, MVDR-RTF can be expressed as

$$\min_{\mathbf{W}} \mathbf{E}||\mathbf{W^H X}||^2 \; s.t. \mathbf{W^H d} = d_1 \tag{6}$$

where d_1 is the first component of $\mathbf{d}$. The solution is

$$\mathbf{W} = \frac{\mathbf{\Phi_{NN}^{-1} \Phi_{GG}}}{tr(\mathbf{\Phi_{NN}^{-1} \Phi_{GG}})} \mathbf{u_1} \tag{7}$$

where $\mathbf{u_1}$ is vector $[1, 0, 0, ..., 0]$.

3.2.4. Generalized sidelobe canceller (GSC)

Generalized sidelobe canceller is composed of three parts: a fixed beamformer, a block matrix and a noise canceller. The fixed beamfomer and the block matrix are normally fixed filters. Using $\mathbf{b}$ and $\mathbf{z}$ to represent the output of the fixed beamformer and block matrix respectively, GSC aims at finding the filter minimizing the output of the noise canceller,

$$\min_{\mathbf{R}} ||\mathbf{b} - \mathbf{R^H z}||^2 \tag{8}$$

where $\mathbf{R}$ is the noise canceller filter and is normally implemented by an adaptive filter.

3.2.5. Parameterized multi-channel Wiener filter (PMWF)

PMWF explicitly expresses the trade-off between noise reduction and distortion. The PMWF filter is defined by

$$\min_{\mathbf{W}} \mathbf{E}(||\mathbf{W^H X} - G_1||^2 + \beta||\mathbf{W^H N}||^2) \tag{9}$$

where G_1 is the first element of $\mathbf{G}$, assuming the first microphone to be the reference microphone, and β is the parameter. The first term $\mathbf{E}(||\mathbf{W^H X} - G_1||^2$ represents distortion and the second term $\mathbf{E}||\mathbf{W^H N}||^2$ represents noise reduction. The solution is

$$\mathbf{W} = (\mathbf{\Phi_{XX}} + \beta\mathbf{\Phi_{NN}})^{-1}\mathbf{\Phi_{GG} u_1} \tag{10}$$

Table 2: Average WER (%) of different beamformers with the CHiME-4 baseline back-end but without RNN.

Track	System	Dev		Test	
		real	simu	real	simu
6ch	WDAS	8.19	9.40	15.59	15.61
	MVDR	14.31	5.97	25.89	6.99
	GSC	10.99	15.77	19.79	24.17
	GSC+WDAS	9.46	11.73	16.61	19.00
	PMWF	10.82	9.90	19.58	14.18
	ME+direct	8.75	6.98	15.05	7.74
	ME+PMWF	8.87	6.51	15.52	7.33
	ME+MVDR-EV	8.04	6.07	13.59	7.32
	ME+MVDR-RTF	11.07	6.90	18.99	8.53

3.2.6. Results and Discussions

In Table 2, WDAS denotes the BeamformIt in the CHiME-4 baseline [4]; MVDR denotes the one released at CHiME-3 [11]; GSC is a standard one with a fixed beamformer and a simple fixed block matrix. GSC+WDAS means using WDAS to relpace the beamformer block of GSC. When applying MVDR, MVDR-RTF and PMWF, noise correlation matrix is estimated using a limited context immediately before the utterance as in the CHiME-3 baseline. After complex GMM based mask estimation (ME), we apply the estimated masks directly to separate the source (”ME+direct”) or to estimate the spatial correlation matrices which are fed to different beamformers (the last three rows in Table 2). We use all 6 channels with energy based microphone failure detection, except in the case of running WDAS where we do not use channel 2.

Several points can be drawn from Table 2. (1) CHiME-3 baseline MVDR performs best on the simulated data but worst on the real data. Presumably this is because that the steering vector estimation in the CHiME-3 baseline MVDR is similiar to the generation of the simulated data and is not matched to the real data. The CHiME-4 baseline WDAS (BeamformIt) performs well. (2) Different beamforming methods pursue tradeoff between reducing noise and avoiding source distortion from different perspectives. MVDR-RTF and PMWF contain distortion even if with perfect estimation of spatial correlation matrix. The MVDR-EV beamformer is attractive since it explicitly enforces distortionless in the desired source direction. (3) The MVDR-EV beamformer relies on the estimation of the spatial correlation matrices of clean and noise signals, which in turn are used to estimate the steering vector $\mathbf{d}$ and the beamformer coefficients $\mathbf{W}$. Complex GMM based spectral mask estimation is found to be superior for this purpose. (4) Replacing the fixed beamformer for GSC is not able to improve the performance of GSC. The block matrix and noise canceller may play a more important role than the fixed beamformer for GSC.

In summary, among those beamforming techniques show in Table 2, ME works well for its ability to reduce noise; WDAS (BeamformIt) performs well for its robustness; ME+MVDR-EV is found to be most effective, which combines MVDR's distortionless and reliable estimate of the steering vector by ME. Noise reduction, distortionless and robustness should be considered together when designing a beamformer.

The Table 2 results are obtained by training back-end GMMs and DNNs over the enhanced speech. Results in all later Tables (starting from Table 3) are obtained by 1) using cross-correlation based mic failure detection, 2) training back-end acoustic models over only channel 5 but testing over the enhanced speech from ME+MVDR-EV.

Table 3: Average WER (%) for the ME+MVDR-EV enhanced speech with the CHiME-4 baseline back-end.

Track	System	Dev		Test	
		real	simu	real	simu
6ch	GMM	10.89	10.45	16.42	12.10
	DNN sMBR	7.20	6.44	11.10	8.02
	KN5+RNN	5.16	4.70	8.21	5.79

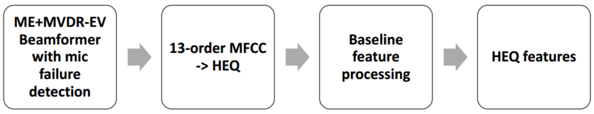

ME+MVDR-EV Beamformer with mic failure detection
13-order MFCC -> HEQ
Baseline feature processing
HEQ features

Figure 1: HEQ feature enhancement flow chart in testing.

3.3. Microphone failure detection

For the 6-ch speech recognition, there exists microphone failure, which hurts the recognition performance. Energy based microphone failure detection does not work well, so we propose to use segmental cross-correlation to detect microphone failure.

Microphone failure is mainly caused by microphones not working or touched by the speaker, thus there may have small or large energies. Considering that cross-correlation is influenced by speech magnitudes, we first normalize the 6-ch signals to have equal energies for each channel. Then we calculate the summed segmental maximum cross-correlation:

$$\operatorname{corr}[i,m] = \sum_{j,j\neq i} \max_{n} \operatorname{corr}[i,j,m,n] \tag{11}$$

where $\operatorname{corr}[i,j,m,n]$ denotes the cross-correlation between the m-th segment from ch-i and the m-th segment from ch-j with n-point shift. The $\operatorname{corr}[i,m]$ is further scaled by the median as follows:

$$\operatorname{scorr}[i,m] = \frac{\operatorname{corr}[i,m]}{\underset{i}{median}\ \operatorname{corr}[i,m]} \tag{12}$$

When $\operatorname{scorr}[i,m]$ is smaller than the threshold α, the ch-i's m-th segment is considered as a failure segment. If one channel contains more than β failure segments, this channel is thrown away. In our experiments, a segment is of 128ms duration, α is set to be 0.6 and β is set to be 2.

3.4. Histogram equalization (HEQ)

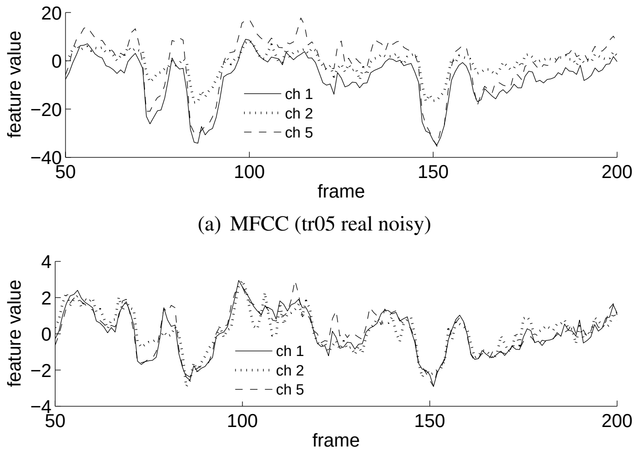
The baseline acoustic features are 13-order MFCCs. HEQ is to warp each component of the cepstral vector over a specified time interval to match the standard Gaussian. While HEQ is applied in sentence level in [14], HEQ over sliding 3-second windows performs better in our experiments. After HEQ, other feature transformations are applied as in the CHiME-4 baseline. In training, HEQ is applied to the MFCCs of channel 5[1]. In testing, HEQ is applied to the enhanced speech, as shown in the flow chart in Figure 1.

It is worthwhile to compare the well-known CMVN and the HEQ. While both are for feature normalization, HEQ is potentially more effective to compensate for additive noise due to the nonlinear nature of the distortion caused by additive noise in the cepstral domain. Comparing Table 3 and 4, it is clear to see the benefit of applying HEQ to compensate for the residual noise in the MVDR beamformed speech, espeically for the real data.

[1] In multi-channel training, HEQ is applied to all six channels.

Table 4: Average WER (%) for the stack-HEQ features with the CHiME-4 baseline back-end.

Track	System	Dev		Test	
		real	simu	real	simu
6ch	GMM	10.39	10.37	13.53	12.01
	DNN sMBR	6.73	6.12	9.95	8.19
	KN5+RNN	4.64	4.22	7.15	5.51



(a) MFCC (tr05 real noisy)

(b) HEQ (tr05 real noisy)

Figure 2: Effect of HEQ over the second component of MFCC feature vectors.

For illustration purpose, Figure 2 plots the second component of the MFCC feature vectors and the corresponding HEQ features for utterance 011_011C0201_PED in real training set. Three channels (ch 1, ch 5 and ch 6) are plot separately.

HEQ reduces variations in noisy signals but may lose details. We stack two types of fMLLR features with and without HEQ as the input of the DNN for information fusion (called stack-HEQ).

3.5. Trans-dimensional random field (TRF) LM

In addition to the 5-gram LM and RNN LM provided in the baseline, a TRF LM is trained on the official training corpus with 200 word classes and the features ”w+c+ws+cs+wsh+csh” [15]. “w”/“c” denotes the word/class n-gram up to order 4 and “ws”/“cs” denotes the word/class skipping n-gram up to order 4. “wsh”/“csh” denotes the higher-order long-skipping features. The definition of feature types is shown in Table 1 of [15].

Here is a brief introduction to TRF LMs. Denote by $x^l = (x_1, \dots, x_l)$ a sentence (i.e., word sequence) of length l ranging from 1 to m. Each element of x^l corresponds to a single word. D denotes the whole training corpus and D_l denotes the collection of length l in the training corpus. n_l denotes the size of D_l and $n = \sum_{l=1}^{m} n_l$.

As defined in [15], a trans-dimensional random field model represents the joint probability of the pair (l, x^l) as

$$p(l, x^l; \lambda) = \frac{n_l/n}{Z_l(\lambda)} e^{\lambda^T f(x^l)}, \qquad (13)$$

where n_l/n is the empirical probability of length l. $f(x^l) = (f_1(x^l), \dots f_d(x^l))^T$ is the feature vector, which is usually defined to be position-independent and length-independent, e.g. the n-grams. d is the dimension of the feature vector $f(x)$. λ is the corresponding parameter vector of $f(x^l)$. $Z_l(\lambda) = \sum_{x^l} e^{\lambda^T f(x^l)}$ is the normalization constant of length l. By making explicit the role of length in model definition, it is clear that the model is a mixture of random fields on sentences of different lengths (namely on subspaces of different dimensions), and hence will be called a trans-dimensional random field (TRF).

Table 5: Average WER (%) for different language models.

Track	System	Dev		Test	
		real	simu	real	simu
6ch	KN5	5.57	5.11	8.25	6.42
	RNN	5.24	4.82	7.92	5.91
	TRF	5.09	4.56	7.92	6.04
	LSTM	5.35	4.20	7.08	5.28
	KN5+RNN	4.64	4.22	7.15	5.51
	KN5+LSTM	4.68	3.74	6.79	5.15
	TRF+RNN	4.48	4.06	6.96	5.26
	TRF+LSTM	4.58	3.78	6.55	4.95

Table 6: WER (%) comparison w/o multi-channel training (enhanced speech with HEQ and TRF+LSTM back-end).

Track	System	Dev		Test	
		real	simu	real	simu
6ch	trained on only ch 5	4.58	3.78	6.55	4.95
	multi-channel	4.32	3.47	5.81	4.41

In the joint SA training algorithm [15], another form of mixture distribution is defined as follows:

$$p(l, x^l; \lambda, \zeta) = \frac{n_l/n}{Z_1(\lambda) e^{\zeta_l}} e^{\lambda^T f(x^l)} \qquad (14)$$

where $\zeta = \{\zeta_1, \dots, \zeta_m\}$ with $\zeta_1 = 0$ and ζ_l is the hypothesized value of the log ratio of $Z_l(\lambda)$ with respect to $Z_1(\lambda)$, namely $\log \frac{Z_l(\lambda)}{Z_1(\lambda)}$. $Z_1(\lambda)$ is chosen as the reference value and can be calculated exactly. An important observation is that if and only if ζ were equal to the true log ratios, then the marginal probability of length l under distribution equals to n_l/n. This property is then used to construct the augmented SA algorithm, which jointly estimates the model parameters λ and normalization constants ζ.

TRF LMs have the potential to integrate a richer set of features, and as shown in [15], outperform the traditional 4-gram LM significantly with the relative WER reduction 9.1%. Moreover TRF LMs also achieve slightly better WER results than RNN LMs, but with much faster speed in computing sentence probabilities.

In this experiment, the RNN LM is trained using the CHiME-4 baseline script with 300 hidden units. The LSTM LM is trained using the open source toolkit provided by [19] with 2 hidden layers and 500 hidden units of each layer. 10 epoch iterations are performed before early stop and no dropout is used. Following the challenge instructions, we tune the LM weight and interpolation weight over the whole development set including all noisy environments and data types. The experiment scripts can be found in [17]. As shown in Table 5, TRF alone performs as good as RNN; TRF+RNN further reduces the WER from KN5+RNN; TRF+LSTM performs even better.

3.6. Multi-channel training

After the CHiME-4 submission, we perform multi-channel training as a straightforward way to expose the acoustic model to larger training data, as did in [20], and obtain further significant improvement, as shown in Table 6.

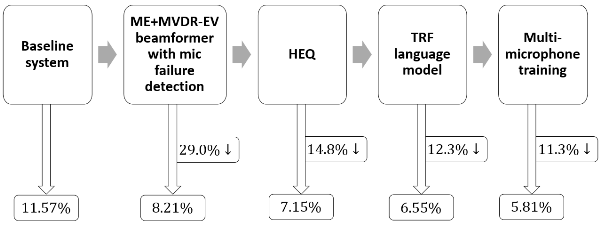


Figure 3: WERs on the real evaluation data, showing the relative contribution of each technique.

Table 7: Average WER (%) for the CHiME-4 baseline front-end (BeamformIt) with our submitted back-end (stack-HEQ).

Track	System	Dev		Test	
		real	simu	real	simu
6ch	GMM	12.56	14.37	18.82	19.84
	DNN sMBR	7.75	8.74	13.84	13.57
	KN5+RNN	5.46	6.29	10.35	9.97
	TRF+LSTM	5.23	5.66	9.36	9.16

Table 8: WER (%) per environment for the submitted system w/o TRF LM.

Track	Envir.	Dev		Test	
		real	simu	real	simu
6ch without TRF (KN5+RNN)	BUS	5.80	4.13	10.36	4.09
	CAF	3.78	4.90	6.13	5.12
	PED	3.85	3.63	5.08	5.60
	STR	5.12	4.22	7.04	7.23
6ch with TRF (TRF+LSTM)	BUS	5.68	4.56	9.67	4.74
	CAF	3.98	4.06	5.60	4.22
	PED	3.78	3.14	4.17	4.65
	STR	4.90	3.36	6.76	6.18

4. Summary

In this paper, we build a lightweight advanced system for CHiME-4 far-field multi-channel speech recognition challenge, with three key techniques. After experiments with a bundle of beamforming methods, the MVDR beamformer with the steering vector being estimated by time-frequency masks is found to be most effective. HEQ is successfully applied to compensate for the residual noise in the MVDR beamformed speech. Interpolated TRF+LSTM LMs perform significantly better than the baseline KN5+RNN LMs and are also superior to the state-of-the-art interpolated KN5+LSTM LMs in language model rescoring. In combination these techniques are surprisingly effective, achieving 6.55% WER for the real evaluation data while keeping low system complexity. Applying multi-channel training further reduces the WER to 5.81%.

Figure 3 shows how the system performance is incrementally improved over the CHiME-4 baseline with the introduced techniques from front-end to back-end. Following the challenge instructions, Table 7 shows the results of the CHiME-4 baseline front-end (BeamformIt) with our submitted back-end (stack-HEQ, TRF LM, without multi-channel training); Table 8 shows the WER per environment for our submitted system.

5. References

[1] J. Bitzer and K. U. Simmer, “Superdirective microphone arrays,” in *Microphone arrays*. Springer, 2001, pp. 19–38.

[2] M. Souden, J. Benesty, and S. Affes, “On optimal frequency-domain multichannel linear filtering for noise reduction,” *IEEE Transactions on audio, speech, and language processing*, vol. 18, no. 2, pp. 260–276, 2010.

[3] L. Griffiths and C. Jim, “An alternative approach to linearly constrained adaptive beamforming,” *IEEE Transactions on antennas and propagation*, vol. 30, no. 1, pp. 27–34, 1982.

[4] X. Anguera, C. Wooters, and J. Hernando, “Acoustic beamforming for speaker diarization of meetings,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 2011–2022, 2007.

[5] T. Higuchi, N. Ito, T. Yoshioka, and T. Nakatani, “Robust mvdr beamforming using time-frequency masks for online/offline asr in noise,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[6] J. Heymann, L. Drude, and R. Haeb-Umbach, “Neural network based spectral mask estimation for acoustic beamforming,” in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016.

[7] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *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.

[8] T. Mikolov, M. Karafiát, L. Burget, J. Cernockỳ, and S. Khudanpur, “Recurrent neural network based language model.” in *Interspeech*, 2010.

[9] M. Sundermeyer, R. Schlüter, and H. Ney, “Lstm neural networks for language modeling.” in *Interspeech*, 2012, pp. 194–197.

[10] E. Vincent, S. Watanabe, A. A. Nugraha, J. Barker, and R. Marxer, “An analysis of environment, microphone and data simulation mismatches in robust speech recognition,” *Computer Speech and Language*, to appear.

[11] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, “The third chime speech separation and recognition challenge: Dataset, task and baselines,” in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2015.

[12] X. Mestre and M. A. Lagunas, “On diagonal loading for minimum variance beamformers,” in *IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, 2003.

[13] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, “The kaldi speech recognition toolkit,” in *IEEE workshop on automatic speech recognition and understanding (ASRU)*, 2011.

[14] A. De La Torre, A. M. Peinado, J. C. Segura, J. L. Pérez-Córdoba, M. C. Benítez, and A. J. Rubio, “Histogram equalization of speech representation for robust speech recognition,” *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 3, pp. 355–366, 2005.

[15] B. Wang, Z. Ou, and Z. Tan, “Trans-dimensional random fields for language modeling,” in *Annual Meeting of the Association for Computational Linguistics (ACL)*, 2015.

[16] B. Wang, Z. Ou, Y. He, and A. Kawamura, “Model interpolation with trans-dimensional random field language models for speech recognition,” *arXiv preprint arXiv:1603.09170*, 2016.

[17] “https://github.com/wbengine/spmilm.”

[18] M. S. Brandstein and H. F. Silverman, “A robust method for speech signal time-delay estimation in reverberant rooms,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1997.

[19] W. Zaremba, I. Sutskever, and O. Vinyals, “Recurrent neural network regularization,” *arXiv preprint arXiv:1409.2329*, 2014.

[20] T. Yoshioka, N. Ito, M. Delcroix, A. Ogawa, K. Kinoshita, M. Fujimoto, C. Yu, W. J. Fabian, M. Espi, T. Higuchi *et al.*, “The ntt chime-3 system: Advances in speech enhancement and recognition for mobile multi-microphone devices,” in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2015.

Wide Residual BLSTM Network with Discriminative Speaker Adaptation for Robust Speech Recognition

Jahn Heymann, Lukas Drude, Reinhold Haeb-Umbach

Paderborn University
Department of Communications Engineering
Paderborn, Germany
{heymann, drude, haeb}@nt.uni-paderborn.de

Abstract

We present a system for the 4th CHiME challenge which significantly increases the performance for all three tracks with respect to the provided baseline system. The front-end uses a bidirectional Long Short-Term Memory (BLSTM)-based neural network to estimate signal statistics. These then steer a Generalized Eigenvalue beamformer. The back-end consists of a 22 layer deep Wide Residual Network and two extra BLSTM layers. Working on a whole utterance instead of frames allows us to refine Batch-Normalization. We also train our own BLSTM-based language model. Adding a discriminative speaker adaptation leads to further gains. The final system achieves a word error rate on the six channel real test data of $3.48\,\%$. For the two channel track we achieve $5.96\,\%$ and for the one channel track $9.34\,\%$. This is the best reported performance on the challenge achieved by a single system, i.e., a configuration, which does not combine multiple systems. At the same time, our system is independent of the microphone configuration. We can thus use the same components for all three tracks.

1. Introduction

Automatic speech recognition has become part of everyday life, not the least because current systems start to achieve remarkable results even in adversarial environments with severe noise conditions. These advances can mainly be attributed to stronger back-ends relying on Deep Neural Networks (DNNs) and by the processing of multiple input channels which can provide spatial selectivity to extract the signal of interest.

Indeed, today's powerful mobile devices are often equipped with multiple microphones, and thus multi-channel signal processing has become a more and more relevant approach to counterfeit more severe signal impairments due to noise or reverberation. The majority of multi-channel speech enhancement systems now consists of some kind of frequency domain beamforming approach. These beamforming systems traditionally relied on model based masking of time frequency (tf) bins [1, 2, 3, 4, 5, 6, 7] but more recently, more data driven and discriminatively trained masking approaches have been proposed [8].

Still, the dispute whether more emphasis should be put on a stronger front-end or just a deeper back-end remains. The latter has been pushed to an extreme, where the multi-channel waveforms or their short time Fourier representations are directly input to the DNN for acoustic modeling. It is then left to the network training to learn that fusion of the channels from the data, which is most effective for ASR performance [9, 10, 11].

We argue that the front-end should be as unobstrusive to the signal as possible. The danger of adding processing artifacts is worse than limited noise reduction and therefore we recommend to only use (masking based) linear beamforming. This still leverages all information contained in correlations between the individual channels and therefore leave all non-linear feature extraction to a deeper and more advanced back-end. Consequently, we employ an explicit acoustic beamforming component, thus taking advantage of recent progress in this field, e.g., by avoiding an explicit speaker localization component and by giving up the assumption of an anechoic environment. Still, we value the power of DNNs by using them for mask estimation: The beamforming coefficients, are estimated from signal statistics, more precisely, from the power spectral density matrices of the target speech and of the distortions. These statistics are obtained from spectral masks, which indicate for each tf bin, whether it is dominated by speech or by noise. And it is this mask estimation which is carried out by means of a DNN.

Nevertheless, a strong back-end is essential for good ASR performance. We employ a Wide Residual Network (WRN) with 22 layers for acoustic modeling. While this network architecture has been used successfully on image recognition tasks [12], it is adapted and used here for the first time for ASR.

The paper is organized as follows. In Sec. 2.1 we give an overview of our front-end design while the back-end is described in Sec. 2.2. The following section (3) shortly describes the database. Detailed experimental results are presented in Sec. 4. At the end we draw conclusions.

2. System Overview

2.1. Front-end

We use the Generalized Eigenvalue (GEV) beamformer which maximizes the signal-to-noise ratio (SNR) of the beamformer output in each frequency bin separately, leading to the beamformer coefficients [13]:

$$\mathbf{F}_{\mathrm{GEV}}(f) = \underset{\mathbf{F}(f)}{\operatorname{argmax}} \frac{\mathbf{F}(f)^{\mathrm{H}} \mathbf{\Phi}_{\mathbf{XX}}(f) \mathbf{F}(f)}{\mathbf{F}(f)^{\mathrm{H}} \mathbf{\Phi}_{\mathbf{NN}}(f) \mathbf{F}(f)}. \tag{1}$$

Here, $\mathbf{\Phi}_{\mathbf{XX}}(f)$ is the target and $\mathbf{\Phi}_{\mathbf{NN}}(f)$ the noise Cross-Power Spectral Density (PSD) matrix for the f-th frequency band. Please note that this does not require any assumptions (e.g., assuming an anechoic environment) regarding the nature of the Acoustic Transfer Function (ATF) from the speech source to the sensors or regarding the spatial correlation of the noise.

The maximization of the coefficient given in Eq. (1) is achieved by solving a generalized eigenvalue problem:

$$\mathbf{\Phi}_{\mathbf{XX}} \mathbf{F} = \lambda \mathbf{\Phi}_{\mathbf{NN}} \mathbf{F}, \tag{2}$$

where the eigenvector corresponding to the largest eigenvalue is the solution to Eq. (1).

This equation, however, does not impose a constraint on the norm of $\mathbf{F}$, and since each frequency is considered independently, this can introduce arbitrary speech distortions.

We handle these distortions by applying the following single channel post filter to the GEV output signal [13]:

$$g_{\mathrm{BAN}}(f) = \frac{\sqrt{\mathbf{F}_{\mathrm{GEV}}(f)^{\mathrm{H}}\boldsymbol{\Phi}_{\mathbf{NN}}(f)\boldsymbol{\Phi}_{\mathbf{NN}}(f)\mathbf{F}_{\mathrm{GEV}}(f)/D}}{\mathbf{F}_{\mathrm{GEV}}(f)^{\mathrm{H}}\boldsymbol{\Phi}_{\mathbf{NN}}(f)\mathbf{F}_{\mathrm{GEV}}(f)}, \tag{3}$$

where D is the number of microphones. This filter performs a so-called Blind Analytic Normalization (BAN) to obtain a distortionless response in the direction of the speaker: The overall ATF from the target source to the post filter output should have unit gain for every frequency bin. If this were achieved perfectly, speech distortions would be removed and one would eventually arrive at the Minimum Variance Distortionless Response (MVDR) beamformer [14, 15].

Another option is to normalize each beamforming vector to unit length. This leaves some distortions in the target signal but those can be handled by the acoustic model if the same kind of distortions occur both in training and test. Indeed, we found out that this matched training scenario even leads to slightly better results, compared to a training on the beamformer output signal after applying BAN. Here, however, we choose to use BAN because we want to train the acoustic model on all channels, and not only on the single beamformer output signal. Then BAN is necessary to reduce the mismatch between the six channels used for training and the beamformer output, which is used for recognition. The benefit of the six times larger training set size more than compensated for the slight loss due to using BAN.

To solve Eq. 2, signal statistics, namely the PSD matrices, are required. We estimate these using a mask based approach. Given non-overlapping masks, $M_{\mathbf{X}}$ for the target signal and $M_{\mathbf{N}}$ for the distortion, we estimate the PSD matrix by calculating the weighted sum of outer products of the microphone signals [16]:

$$\boldsymbol{\Phi}_{\nu\nu}(f) = \sum_{t=1}^{T} M_{\nu}(t,f)\mathbf{Y}(t,f)\mathbf{Y}(t,f)^{\mathrm{H}}, \tag{4}$$

where $\nu \in \{\mathbf{X}, \mathbf{N}\}$ and $\mathbf{Y}(t,f)$ is the vector of microphone signals at time frame t and frequency bin f.

To obtain an estimation of these masks given our observed signals, we utilize a neural network. Tbl. 1 details its configuration. The network operates on each channel independently yielding D masks for the target and D for the distortions. For each source the masks are condensed into a single mask by median pooling. We opted for this pooling operation because it makes the mask estimation more robust against channel failures compared to computing the average of the masks.

We do not force the values of the estimated masks to be one or zero. Rather, we restrict them to be in the range between one and zero using a Sigmoid non-linearity activation function for both estimates, i.e. we work with soft-masks.

We employ ADAM [17] for training. A fixed learning-rate of 0.001 and full backpropagation through time [18] is used. Additionally, if the norm of a gradient for this network is greater than one, we divide the gradient by its norm [19].

To achieve a better generalization, we use Dropout [20] for the input-hidden connection of the bi-directional Long Short-Term Memory (BLSTM) units [21] and for the input of the Rectified Linear Unit (ReLU) layers [20]. The dropout rate is fixed at $p = 0.5$ for every layer during the whole training. We do not use dropout for the last layer. Additionally we modified the SNR randomly in a range of $0\,\mathrm{dB}$ to $-7\,\mathrm{dB}$. We use the development data for cross-validation, stopping the training when the loss does not decrease anymore after 5 epochs of patience.

Table 1: Network configuration for mask estimation

	Units	Type	Non-Linearity	p_{dropout}
L1	256	BLSTM	Tanh	0.5
L2	513	FF	ReLU	0.5
L3	513	FF	ReLU	0.5
L4	1026	FF	Sigmoid	0.0

We apply Batch-Normalization (BN) [22] for each layer. Statistics for the BN are summarized along the time frame dimension. In contrast to the method proposed in [22], we do not use the population estimates obtained from the training or development data for the mean and variance at test time. Rather, we use the statistics of each utterance for each channel individually also for the test data.

The ideal binary masks used as training targets are defined as:

$$\mathrm{IBM}_{\mathbf{N}}(t,f) = \begin{cases} 1, & \frac{||\mathbf{X}(t,f)||}{||\mathbf{N}(t,f)||} < 10^{\mathrm{th}_{\mathbf{N}}(f)}, \\ 0, & \text{else}, \end{cases} \tag{5}$$

and

$$\mathrm{IBM}_{\mathbf{X}}(t,f) = \begin{cases} 1, & \frac{||\mathbf{X}(t,f)||}{||\mathbf{N}(t,f)||} > 10^{\mathrm{th}_{\mathbf{X}}(f)}, \\ 0, & \text{else}, \end{cases} \tag{6}$$

respectively.

The two thresholds $\mathrm{th}_{\mathbf{X}}$ and $\mathrm{th}_{\mathbf{N}}$ are not identical. Their values range from -5 to 10 depending on the frequency and are hand-tuned. They are chosen such that a decision in favor of speech/noise is only taken if the instantaneous SNR is high/low enough to ensure a low false acceptance rate. The network is trained on all utterances and all channels using the binary cross-entropy cost.

2.2. Back-end

2.2.1. Network configuration

For the back-end network we combine a slightly modified design of a WRN [12] with BLSTM layers. This configuration is motivated by the fact that each layer type has its own distinct advantages which complement each other in a unified architecture [23] and by recent findings about Convolutional Neural Networks (CNNs) by the image community [12, 24, 25]. An overview of the structure, which we call Wide Residual BLSTM Network (WRBN), is given in Fig. 3 while Fig. 2 and Fig. 1 detail the building blocks.

The first part of the network is composed of a WRN. The WRN consists of three residual building blocks which again consist of smaller building blocks (BlockA and BlockB). The difference between BlockA and BlockB is rather small. BlockA can reduce the frequency resolution by having a stride ≥ 1 and it increases the number of channels. Due to this change of the size of the tensor, a direct residual connection to the output of the block is not possible. BlockA therefore has an

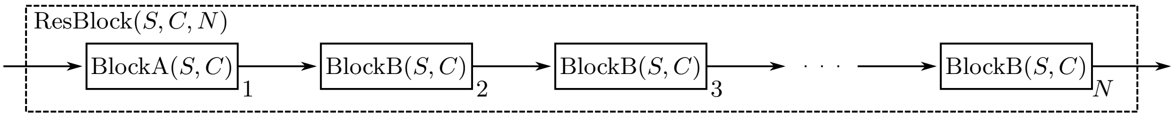


Figure 1: Detailed view of a ResBlock. A ResBlock is parameterized by its striding S, the number of output channels C and the number of inner blocks N. Accordingly, BlockB is repeated $N - 1$ times.

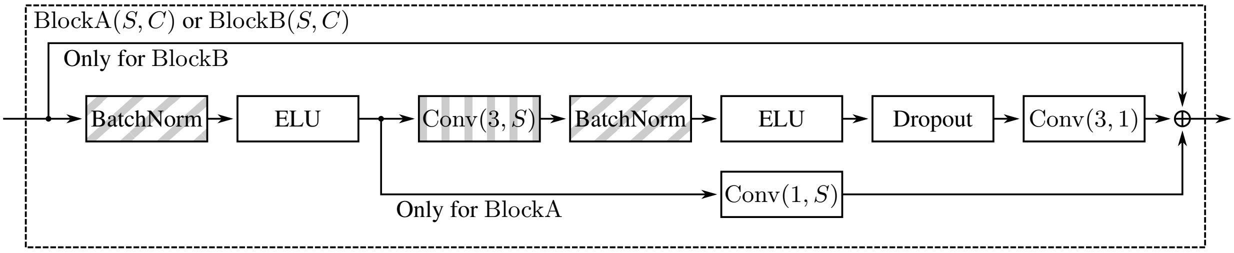


Figure 2: Detailed view of the building blocks BlockA or BlockB. The batch normalization collects statistics along the frequency band axis and along the time frame axis. A convolution block $\mathrm{Conv}(A, S)$ is parameterized by the filter size $A \times A$, the zero padding $(A - 1)/2$ in both directions and the consecutive striding S.

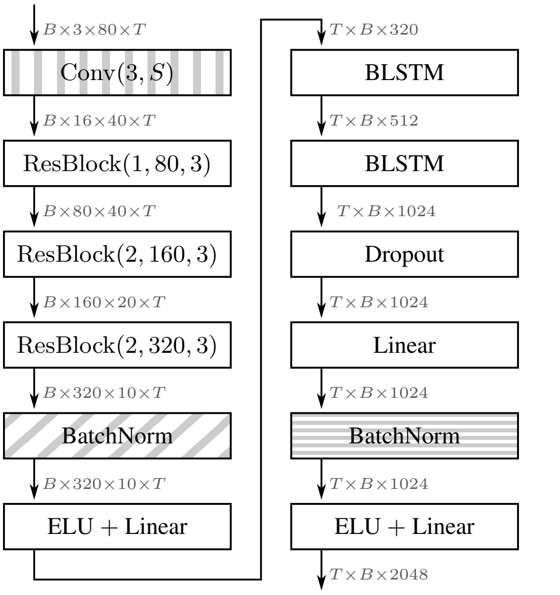


Figure 3: Overview of the back-end structure. The annotations in gray indicate the dimension of the tensors where B is the mini-batch size and T is the number of frames of the largest utterance within the batch. The building blocks are explained in Fig. 1 and Fig. 2. The convolution and the diagonally striped batch normalization is defined as in in Fig. 2. The horizontally striped batch normalization just collects statistics along the time frame axis.

additional convolution operation with filter size 1×1 which acts as the residual connection but also changes the size accordingly. Other than that, the two blocks are identical. A Batch-Normalization [22] normalizes the output of the preceding convolution and an Exponential Linear Unit (ELU) non-linearity [26] is applied afterwards. Before the last convolution of a block we use Dropout [20] with $p = 0.5$.

After the residual blocks, we get 320, each with a dimension of $10 \times T$ where T describes the number of frames and the first dimension can be interpreted as frequency bands. These are then weighted and combined for each channel with learnable weights resulting in a feature dimension of 320 per frame. These frames are used as the input for two consecutive BLSTM layers with 512 units for each direction. The output of the directions is merged by a sum after the first BLSTM layer and by a concatenation after the second BLSTM layer. To prevent overfitting we use Dropout on the input of each layer. Additionally we also use Dropout for the hidden-hidden transitions. Instead of sampling the dropout masks individually per frame, however, we sample the mask once per sequence with $p = 0.5$ [27]. This sampling strategy avoids losing temporal information as a result of Dropout.

The last part of the network consists of two feed-forward layers with Batch-Normalization and an ELU non-linearity. The final output are the posterior probabilities for the 2042 context-dependent states for each frame.

2.2.2. *Training*

We first extract the alignments with the baseline back-end and our front-end using all six channels (*Kaldi+GEV*). We then train our network with a cross-entropy criterion and Adam [17] with $\alpha = 10^{-4}$ on the unprocessed training data from all six microphones. We use 80 dimensional mean-normalized log-mel filterbank features as input. Their delta and delta-delta features are used for two additional input channels. We do not train the network on a window of n frames with truncated backpropagation. Instead, we train it on a whole utterance with full backpropagation through time. We see two main advantages in this strategy. First, the CNN and especially the BLSTM is able to exploit the full temporal context and we can avoid zero-padding within the utterance. Second, we can make efficient use of Batch-Normalization as described in the following.

2.2.3. Batch-Normalization

Batch-Normalization was first proposed in the context of image recognition and has been shown to improve convergence as well as generalization [22]. However, a drawback of this approach is that it relies on statistics accumulated on training and/or development data at test time. Calculating the statistics during test would lead to a dependency on the mini-batch constellation since the statistics are aggregated over the batch dimension. Here, we treat the batch dimension as an independent dimension. We can then calculate the statistics also at test time without losing determinism. This is possible because using the whole utterance we can get a reliable estimate of the statistics without including other utterances. Thus each utterance is normalized separately. For the tensors within the WRN we calculate the statistics over the height (frequency) and width (time) for each channel separately. For the other tensors we calculate the statistics over time and normalize the feature dimension.

2.2.4. Adaptation

For (speaker) adaptation, we train an additional layer consisting of a 80×80 weight matrix for each speaker and each track [28]. That layer with tied weights is applied to all three feature channels equally. Although CNNs can provide some translation invariance, we found that the additional transformation of the input features improves performance. It helps to reduce the mismatch between the unprocessed data at training time and the beamformed data at test time. We opted for the single layer since preliminary results got worse when we adapted the whole network or parts of it. Training is done by first decoding the utterances with our best speaker-independent model to get an alignment for each utterance for each track. We then prepend the layer to the network and train it with backpropagation for 5 epochs and $\alpha = 10^{-5}$.

2.2.5. Language model

The baseline system features three different language models. First, the search graphs are created using a standard 3-gram model provided by the WSJ database [29]. The graph is then rescored with a 5-gram Kneser-Ney [30] language model trained on the provided training data. Finally, the scores are interpolated (rescored) with a recurrent neural network language model [31]. Here, we aim to replace the latter by a stronger one. To this aim, we employ a two layer Long Short-Term Memory (LSTM) language model with 650 hidden units each – similar to the example provided by [32].

Instead of training on an endless word stream (initial state of next batch is end state of current batch), we found that training on complete sentences from the provided language model training data in a random mini-batch improved cross validation scores slightly (6 % relative word error rate (WER) improvement compared to an endless stream).

Again we use Adam [17] with the parameters proposed in the aforementioned paper for optimization for 39 epochs. The main benefit of using Adam besides a slightly improved WER was the fact, that a learning rate did not have to be tuned manually.

We experimented with ZoneOut [33] as a regularization technique for recurrent neural networks but ended up using regular dropout in the vertical connections only.

Global gradient clipping with a maximum value of 5 is used. All weight matrices and bias vectors, including the embedding matrix, are initialized with random weights sampled from a uniform distribution in $[-0.1, 0.1]$.

We experiment with restricted training sets limiting the maximum number of unknown symbols during training. This yielded reduced cross validation perplexities. Nevertheless, we finally selected a model trained on unrestricted training data, since this resulted in the lowest development test WERs.

Although, the training objective for the language model was perplexity, it turned out to beneficial to select the final language model based on the actual WER on the development set.

3. Database

The dataset from the fourth CHiME challenge [34] features three different tracks with real and simulated audio data of prompts taken from the 5k WSJ0-Corpus [29] with 4 different types of real-world background noise. The noise as well as the real utterances were recorded in a pedestrian, in a cafe, on the street and in a bus. The recording device was a tablet with six microphones mounted on its frame. The tracks were differentiated by the number of microphones used at test time. All were used in the six channel track, while in the two and one channel track the microphones were sampled randomly.

4. Experimental evaluation

Tbl. 2 gives an overview of all experiments and their results.

Concentrating on the effect of the front-end first, we can conclude that for the two channel track using our front-end (*Kaldi+GEV*) instead of the baseline front-end (*Baseline*) gives noticeable improvements in terms of WER. For the six channel track, just exchanging the front-end even decreases the WER by about 50 %, clearly showing the effectiveness of our approach. Nevertheless, there is still a big gap between the six channel and the two channel track. While this shows that our front-end is able to leverage additional microphones, it also shows that there is still room for improvements.

The advances in acoustic modelling are best visible for the one channel track. Compared to the Baseline, our proposed acoustic model achieves significantly lower WERs. Especially when comparing the results on the development and the test data, we can see that the gap is much smaller for our model, indicating its ability to generalize to different noise conditions. Looking at the six channel track we can conclude that the gap between the baseline model and our model gets smaller as the quality of the input signal improves (*Basline* vs. *WRBN+BFIT* and *Kaldi+GEV* vs. *WRBN+GEV*). This tendency is also visible for the two channel track.

For all tracks, we are able to further improve the results employing different methods presented in Sec. 2.2. The biggest gain here can be attributed to Batch-Normalization at test time.

Detailed results for the best system for each track are shown in Tbl. 3. Here, the results are splitted according to the four different environments. We can see that the more microphones we use, the less sensitive the result is to a specific environment. Especially for the one channel track the bus environment performs worse with the street environment having nearly half of the WER for the real test set.

Table 2: Average WER (%) for the tested systems. Bold results correspond to the officially submitted results. The individual abbreviations mean: "Kaldi": baseline backend, "WRBN": our WRBN (Section 2.2.1), "+BN": with Batch-Normalization (Sec. 2.2.3), "+SA": with additional linear speaker adaptation layer (Sec. 2.2.4) "+NTLM": with own language model (Sec. 2.2.5), "+GEV": with GEV beamformer (Sec. 2.1), "+BFIT": with baseline front-end beamformer

Track	System	Dev		Test	
		real	simu	real	simu
1ch	Baseline	11.57	12.98	23.70	20.84
	WRBN	6.64	9.09	11.8	13.78
	+BN	5.69	7.53	10.4	12.67
	+SA	5.5	7.18	9.88	11.68
	+NTLM	**5.19**	**6.69**	**9.34**	**11.11**
2ch	Baseline	8.23	9.50	16.58	15.33
	Kaldi+GEV	6.93	8.03	13.76	9.9
	WRBN+GEV	4.67	5.38	7.65	6.53
	+BN	4	4.76	6.96	6.22
	+SA	3.8	4.45	6.44	5.38
	+NTLM	**3.54**	**4.05**	**5.96**	**5.16**
6ch	Baseline	5.76	6.77	11.51	10.90
	Kaldi+GEV	3.7	3.72	5.66	4.34
	WRBN+BFIT	4.43	5.27	7.33	7.85
	WRBN+GEV	3.16	3.2	4.52	3.41
	+BN	3.06	2.99	4.07	3.51
	+SA	2.84	2.75	3.85	3.11
	+NTLM	**2.73**	**2.34**	**3.48**	**2.76**

5. Conclusions

Comparing the presented system with the baseline system, two components can be identified which provided significant improvements (on the order of 20 % – 50 %): first the neural network supported GEV beamformer turned out to be more effective than the baseline BeamformIt! [35] beamformer, and, second, the WRBN acoustic model significantly improved over the standard DNN backend. Further, the proposed batch normalization per utterance, the additional linear layer at the WRBN input for speaker adaptation, and the LSTM language model delivered additional improvements (each on the order of 5 % – 10 %). It is further worth mentioning that, up to the speaker adaptation, this is a single-pass recognition system. The described setup can be considered light-weight, as it is a single system and not a combination of multiple systems. While it achieved the best reported single-system results on the CHiME-4 challenge, even better error rates can be achieved by a system combination, as can be seen, e.g., in a companion paper [36].

6. Acknowledgments

This research was supported by the Deutsche Forschungsgemeinschaft (DFG) under Contract No. Ha3455/11-1. Computational resources were provided by the Paderborn Center for Parallel Computing.

Table 3: WER (%) per environment for the best system.

Track	Env	Dev		Test	
		real	simu	real	simu
1ch	BUS	6.82	5.41	13.22	8.07
	CAF	5.28	9.29	9.45	13.17
	PED	3.7	5.21	7.75	10.22
	STR	4.96	6.86	6.93	12.98
2ch	BUS	4.23	3.2	7.85	3.88
	CAF	3.61	5.4	5.79	5.85
	PED	2.86	3.67	4.97	5.21
	STR	3.44	3.92	5.23	5.7
6ch	BUS	2.92	2.14	3.76	2.71
	CAF	2.65	2.63	3.25	2.88
	PED	2.67	2.14	3.33	2.97
	STR	2.67	2.45	3.57	2.48

7. References

[1] H. Sawada, S. Araki, and S. Makino, "Underdetermined Convolutive Blind Source Separation via Frequency Bin-wise Clustering and Permutation Alignment," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 3, pp. 516–527, 2011.

[2] N. Ito, S. Araki, T. Yoshioka, and T. Nakatani, "Relaxed Disjointness Based Clustering for Joint Blind Source Separation and Dereverberation," in *Acoustic Signal Enhancement (IWAENC), 2014 14th International Workshop on*, Sept 2014, pp. 268–272.

[3] D. H. T. Vu and R. Haeb-Umbach, "Blind Speech Separation Employing Directional Statistics in an Expectation Maximization Framework," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, March 2010, pp. 241–244.

[4] N. Ito, S. Araki, and T. Nakatani, "Permutation-free Convolutive Blind Source Separation via Full-band Clustering based on Frequency-independent Source Presence Priors," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, May 2013, pp. 3238–3242.

[5] T. Yoshioka, N. Ito, M. Delcroix, A. Ogawa, K. Kinoshita, M. Fujimoto, C. Yu, W. J. Fabian, M. Espi, T. Higuchi, S. Araki, and T. Nakatani, "The NTT CHiME-3 system: Advances in speech enhancement and recognition for mobile multi-microphone devices," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Dec 2015, pp. 436–443.

[6] S. Araki and T. Nakatani, "Hybrid Approach for Multichannel Source Separation Combining Time-frequency Mask with Multichannel Wiener Filter," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2011, pp. 225–228.

[7] S. Araki, M. Okada, T. Higuchi, A. Ogawa, and T. Nakatani, "Spatial Correlation Model based Observation Vector Clustering and MVDR Beamforming for Meeting Recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 385–389.

[8] J. Heymann, L. Drude, A. Chinaev, and R. Haeb-Umbach, "BLSTM supported GEV beamformer front-end for the 3RD CHiME challenge," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Dec 2015, pp. 444–451.

[9] Y. Hoshen, R. J. Weiss, and K. W. Wilson, "Speech Acoustic Modeling from Raw Multichannel Waveforms," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, April 2015, pp. 4624–4628.

[10] T. N. Sainath, R. J. Weiss, K. W. Wilson, A. Narayanan, M. Bacchiani, and A. Senior, "Speaker Location and Microphone Spacing Invariant Acoustic Modeling from Raw Multichannel Waveforms," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Dec 2015, pp. 30–36.

[11] T. N. Sainath, R. J. Weiss, K. W. Wilson, A. Narayanan, and M. Bacchiani, "Factored Spatial and Spectral Multichannel Raw Waveform CLDNNs," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, 2016.

[12] S. Zagoruyko and N. Komodakis, "Wide Residual Networks," *CoRR*, vol. abs/1605.07146, 2016. [Online]. Available: http://arxiv.org/abs/1605.07146

[13] E. Warsitz and R. Haeb-Umbach, "Blind Acoustic Beamforming Based on Generalized Eigenvalue Decomposition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 5, pp. 1529–1539, July 2007.

[14] B. D. Van Veen and K. M. Buckley, "Beamforming Techniques for Spatial Filtering," *Digital Signal Processing Handbook*, 1997.

[15] U. K. Simmer, J. Bitzer, and C. Marro, "Post-filtering Techniques," in *Microphone Arrays*. Springer, 2001, pp. 39–60.

[16] M. Souden, S. Araki, K. Kinoshita, T. Nakatani, and H. Sawada, "A Multichannel MMSE-Based Framework for Speech Source Separation and Noise Reduction," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 9, pp. 1913–1928, Sept 2013.

[17] D. Kingma and J. Ba, "Adam: A Method for Stochastic Optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[18] P. J. Werbos, "Backpropagation through time: what it does and how to do it," *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, Oct 1990.

[19] R. Pascanu, T. Mikolov, and Y. Bengio, "Understanding the Exploding Gradient Problem," *CoRR*, vol. abs/1211.5063, 2012. [Online]. Available: http://arxiv.org/abs/1211.5063

[20] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A Simple Way to Prevent Neural Networks from Overfitting," *Journal of Machine Learning Research*, vol. 15, pp. 1929–1958, 2014. [Online]. Available: http://jmlr.org/papers/v15/srivastava14a.html

[21] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent Neural Network Regularization," *CoRR*, vol. abs/1409.2329, 2014. [Online]. Available: http://arxiv.org/abs/1409.2329

[22] S. Ioffe and C. Szegedy, "Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift," *CoRR*, vol. abs/1502.03167, 2015. [Online]. Available: http://arxiv.org/abs/1502.03167

[23] T. N. Sainath, O. Vinyals, A. Senior, and H. Sak, "Convolutional, Long Short-Term Memory, fully connected Deep Neural Networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, April 2015, pp. 4580–4584.

[24] K. He, X. Zhang, S. Ren, and J. Sun, "Deep Residual Learning for Image Recognition," *CoRR*, vol. abs/1512.03385, 2015. [Online]. Available: http://arxiv.org/abs/1512.03385

[25] ——, "Identity Mappings in Deep Residual Networks," *CoRR*, vol. abs/1603.05027, 2016. [Online]. Available: http://arxiv.org/abs/1603.05027

[26] D. Clevert, T. Unterthiner, and S. Hochreiter, "Fast and Accurate Deep Network Learning by Exponential Linear Units (ELUs)," *CoRR*, vol. abs/1511.07289, 2015. [Online]. Available: http://arxiv.org/abs/1511.07289

[27] S. Semeniuta, A. Severyn, and E. Barth, "Recurrent Dropout without Memory Loss," *CoRR*, vol. abs/1603.05118, 2016. [Online]. Available: http://arxiv.org/abs/1603.05118

[28] V. Abrash, H. Franco, A. Sankar, and M. Cohen, "Connectionist Speaker Normalization and Adaptation," in *in Eurospeech*. Citeseer, 1995.

[29] J. Garofalo *et al.*, "CSR-I (WSJ0) complete," *Linguistic Data Consortium, Philadelphia*, 2007.

[30] R. Kneser and H. Ney, "Improved Backing-off for M-gram Language Modeling," in *Acoustics, Speech, and Signal Processing, 1995. ICASSP-95., 1995 International Conference on*, vol. 1, May 1995, pp. 181–184 vol.1.

[31] T. Mikolov, S. Kombrink, A. Deoras, L. Burget, and J. Černocký, "RNNLM - Recurrent Neural Network Language Modeling Toolkit," in *Proceedings of ASRU 2011*. IEEE Signal Processing Society, 2011, pp. 1–4. [Online]. Available: http://www.fit.vutbr.cz/research/view_pub.php?id=10087

[32] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent Neural Network Regularization," *arXiv preprint arXiv:1409.2329*, 2014.

[33] D. Krueger, T. Maharaj, J. Kramár, M. Pezeshki, N. Ballas, N. R. Ke, A. Goyal, Y. Bengio, H. Larochelle, A. C. Courville, and C. Pal, "Zoneout: Regularizing rnns by randomly preserving hidden activations," *CoRR*, vol. abs/1606.01305, 2016. [Online]. Available: http://arxiv.org/abs/1606.01305

[34] E. Vincent, S. Watanabe, A. A. Nugraha, J. Barker, and R. Marxer, "An Analysis of Environment, Microphone and Data Simulation Mismatches in Robust Speech Recognition," *Computer Speech and Language*, 2016, to appear.

[35] X. Anguera, C. Wooters, and J. Hernando, "Acoustic Beamforming for Speaker Diarization of Meetings," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 2011–2022, Sept 2007.

[36] T. Menne, J. Heymann, A. Alexandridis, K. Irie, A. Zeyer, M. Kitza, P. Golik, I. Kulikov, L. Drude, R. Schlüter, H. Ney, R. Haeb-Umbach, and A. Mouchtaris, "The RWTH/UPB/FORTH System Combination for the 4th CHiME Challenge Evaluation," 2016, to appear.

Deep Beamforming and Data Augmentation for Robust Speech Recognition: Results of the 4th CHiME Challenge

Tobias Schrank, Lukas Pfeifenberger, Matthias Zöhrer, Johannes Stahl, Pejman Mowlaee, Franz Pernkopf

Signal Processing and Speech Communication Laboratory
Graz University of Technology, Graz, Austria
lukas.pfeifenberger@alumni.tugraz.at,
{tobias.schrank,matthias.zoehrer,johannes.stahl,pejman.mowlaee,pernkopf}@tugraz.at

Abstract

Robust automatic speech recognition in adverse environments is a challenging task. We address the 4th CHiME challenge [1] multi-channel tracks by proposing a deep eigenvector beamformer as front-end. To train the acoustic models, we propose to supplement the beamformed data by the noisy audio streams of the individual microphones provided in the real set. Furthermore, we perform data augmentation by modulating the amplitude and time-scale of the audio. Our proposed system achieves a word error rate of 4.22% on the real development and 8.98% on the real evaluation data for 6-channels and 6.45% and 13.69% for 2-channels, respectively.

1. Background

This report describes our proposed ASR system for the 6- and 2-channel task of the 4th CHiME challenge. The proposed modifications of the baseline system are:

- As multi-channel front-end we employ an optimal multi-channel Wiener filter, which consists of an eigenvector GSC beamformer and a single-channel postfilter. Both components depend on a speech presence probability mask, which we learn using a deep neural network (DNN).
- In addition to the beamformed signals we use noisy multi-channel real data to train the acoustic model of the ASR, i.e. we perform *multi-channel* training.
- We perform data augmentation by modulating the signal amplitude (volume perturbation) and time-scale modifications (speed perturbation).
- We perform sequential language model rescoring using (gated) RNNs.
- We combine multiple systems with a lattice-based approach which uses minimum Bayes risk decoding.

A detailed introduction of the individual components and relevant literature are provided in the next section.

2. Robust Multi-Channel ASR System

Figure 1 shows the block diagram of the proposed multi-channel ASR system including the data augmentation and multi-channel training of the recognizer. Each processing step is detailed in the following sections.

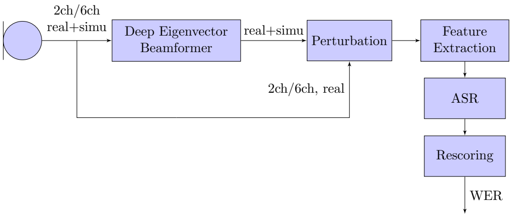


Figure 1: System overview.

2.1. Deep Eigenvector Beamformer

As multi-channel speech enhancement front-end we employ a *deep eigenvector beamformer*, which consists of a generalized sidelobe canceller (GSC) beamformer [2–6], followed by a single-channel postfilter. The GSC consists of a steering vector $\boldsymbol{F}$, a blocking matrix $\boldsymbol{B}$, and an adaptive interference canceller, such that: $\boldsymbol{W} = \boldsymbol{F} - \boldsymbol{B}\boldsymbol{H}_{AIC}$. The GSC block diagram is given in Figure 2. The steering vector $\boldsymbol{F}$ has to model the *acoustic transfer functions* (ATFs) from the speaker to the microphones [7]. Usually this is done by a *direction of arrival* (DOA) estimation. However, this method does not include the complex propagation paths present in the CHiME4 data. Therefore we use the dominant eigenvector of the speech PSD matrix $\hat{\boldsymbol{\Phi}}_{SS}$ as steering vector $\boldsymbol{F}$, such that the beamformer is directed towards the speech source in signal subspace. This allows the beamformer to account for early echoes and reverberation of the speaker signal [7–9]. Hence, we refer to this beamformer as *eigenvector GSC* (EV-GSC).

Using the steering vector $\boldsymbol{F}$, the blocking matrix is given as $\boldsymbol{B} = \boldsymbol{I} - \boldsymbol{F}\boldsymbol{F}^H$. The adaptive interference canceller $\boldsymbol{H}_{AIC}$ is learned using an adaptive NLMS filter [10]. The single-channel postfilter consists of a real-valued gain mask $G = \frac{\xi}{1+\xi}$, which is obtained from the SNR ξ at the beamformer output. It is given as $\xi = \frac{\boldsymbol{W}^H \hat{\boldsymbol{\Phi}}_{SS} \boldsymbol{W}}{\boldsymbol{W}^H \hat{\boldsymbol{\Phi}}_{NN} \boldsymbol{W}}$. The SNR depends on both the speech and noise PSD matrices, which are estimated using a time and frequency dependent *speech presence probability* p_{SPP}.

We use a DNN to learn p_{SPP} from the dominant eigenvector of the PSD matrix of the noisy inputs. As we are operating in the frequency domain, each frequency bin k is assigned to a kernel as shown in Figure 3. The feature vector $\boldsymbol{x}_k$ for each kernel consists of the cosine distance between the eigenvectors of 5 consecutive frames. This introduces some context-

This work was supported by the LEAD project, the Austrian Science Fund (FWF) under the project number P25244-N15 and P27803-N15 and the K-Project ASD. Furthermore, we acknowledge NVIDIA for providing GPU computing resources.

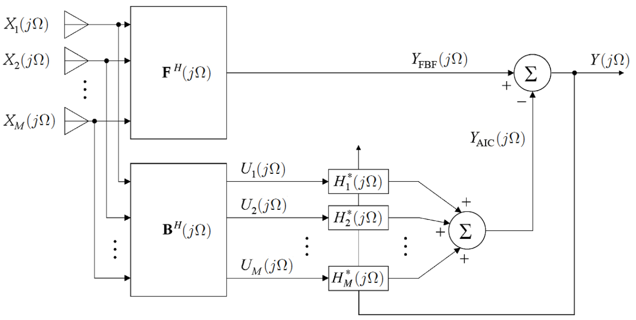


Figure 2: GSC beamformer

sensitivity into our model. The DNN of each kernel uses a hybrid model with a generative and a discriminative component [11]. The generative component consists of two autoencoder layers, which perform unsupervised clustering of the input data $\boldsymbol{x}_k$. The autoencoder kernels operate independently for each frequency bin. We used 20 neurons in the first layer, and 10 neurons in the second layer. The discriminative component consists of a regression layer which fuses the activations of all autoencoder kernels, in order to exploit information which is distributed across the frequency. The regression layer predicts the K output labels $p_{SPP}(\boldsymbol{x}_k)$). Figure 3 illustrates the kernelized DNN used in our system.

For more details on the EV-GSC beamformer and the kernelized DNN, we refer the reader to [12]. We use the same architecture for the 2ch and 6ch track, as the training data is the same for both tracks.

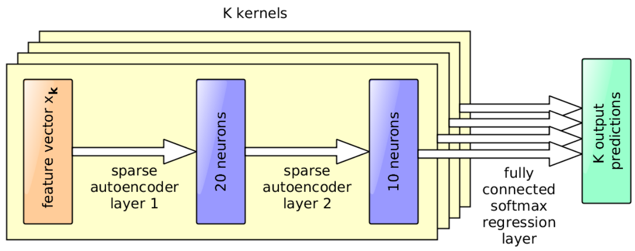

K kernels
feature vector x_k
sparse autoencoder layer 1
20 neurons
sparse autoencoder layer 2
10 neurons
fully connected softmax regression layer
K output predictions

Figure 3: Kernelized DNN to estimate the speech presence probability p_{SPP}

2.2. ASR

The ASR system employs a hybrid DNN architecture which is implemented with the Kaldi toolkit [13]. We do not only use the beamformed data for training but add the noisy channels of the real data (except for channel 2 which faces backwards). With this *multi-channel training (MC)* we can both compensate for the small amount of training data and make the acoustic model less sensitive to noise that might be left over in the evaluation data. In the evaluation stage we still use only the beamformed signals.

The GMM system uses 13 MFCCs and their deltas and delta-deltas. The DNN uses 40 fMLLR features extracted from this GMM system. For the DNN the data is augmented with speed-perturbed copies of the original data. Additionally, the data is volume-perturbed for greater robustness (*pert*). The DNN is then generatively pre-trained using restricted Boltzmann machines. The DNN has 6 hidden layers and is trained with a state-level minimum Bayes risk (*sMBR*) criterion. The results which have been obtained in this way are then rescored with a Kneser-Ney smoothed 5-gram model (*5-gram*), a recurrent neural network language model (*RNNLM*) and a gated RNNLM (*GRNNLM*). The two RNNLMs consist of a single hidden layer with 300 and 500 neural units, respectively.

We perform system combination by first combining the lattices of the system with perturbed training data (*pert*), the system with multi-channel training (*MC*) and the system that uses both (*MC + pert*). We then decode the resulting lattices with an sMBR criterion.

3. Experimental Evaluation

Table 1 shows the results of our systems for the 6-channel and 2-channel tasks of the 4th CHiME challenge. For each data set the best score for a single system and for system combination is in boldface. Due to time constraints we report only those results for the 2-channel task which uses the system architecture that we have found to be optimal for the 6-channel task (S_C). Therefore the following comparison focuses on the 6-channel task.

On average over the test sets, our proposed EV-GSC beamformer of S2 performs 2% WER better than the baseline *BeamformIt* beamformer of S1, i.e. 7.95% WER vs. 9.98% WER for the RNNLM-rescored DNN. However, this performance improvement is the least pronounced for the real evaluation data. Data augmentation through speed perturbation and volume perturbation (*pert*) of S3 results in an improvement of .74% WER on average, i.e. 7.20% WER vs. 7.95% WER. Multi-channel (MC) training of S4 leads to an improvement of 0.80% WER on average, i.e. 7.15% WER vs. 7.95% WER. Both multi-channel training and amplitude and time-scale perturbation (MC+pert) of S5 results in an improvement of 1.19% WER on average, i.e. 6.75% WER vs. 7.95% WER. Further rescoring with the gated RNNLM leads to a small improvement of 0.04% WER. The best results for 6-channels are achieved by a combination of systems S3, S4, and S5 as S6. In particular, we obtain a WER of 8.98% and 7.02% on the real and simulated test set, respectively.

Table 2 shows the individual results for each environment of our best system for the 6- and 2-channel track. For both systems, performance on the real evaluation data set is considerably worse for BUS than for any other environment.

4. References

[1] E. Vincent, S. Watanabe, A. A. Nugraha, J. Barker, and R. Marxer, “An analysis of environment, microphone and data simulation mismatches in robust speech recognition,” *Computer Speech and Language*, to appear.

[2] O. Hoshuyama, A. Sugiyama, and A. Hirano, “A robust adaptive beamformer for microphone arrays with a blocking matrix using constrained adaptive filters,” *IEEE Transactions on Signal Processing*, vol. 47, no. 10, Oct. 1999.

[3] J. Benesty, M. M. Sondhi, and Y. Huang, *Springer Handbook of Speech Processing*. Berlin–Heidelberg–New York: Springer, 2008.

[4] S. Gannot, D. Burshtein, and E. Weinstein, “Signal enhancement using beamforming and nonstationarity with applications to speech,” *IEEE Transactions on Signal Processing*, vol. 49, no. 8, Aug. 2001.

[5] W. Herbordt and W. Kellermann, “Analysis of blocking matrices for generalized sidelobe cancellers for non-stationary broadband signals,” *IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, May 2002.

Table 1: Average WER (%) for the tested systems.

Track	System				Dev		Test	
	Tag	ASR	Data	BF	real	simu	real	simu
2ch	S_A	GMM	–	EV-GSC	14.16	15.13	26.33	24.12
	S_B	GMM	MC	EV-GSC	13.41	15.36	23.46	23.49
	S_C	DNN	MC + pert	EV-GSC	9.38	11.33	17.92	18.10
		+sMBR			9.24	10.91	17.16	17.46
		+5-gram			7.63	9.60	15.29	15.81
		+RNNLM			6.66	8.54	14.02	14.46
		+GRNNLM			**6.45**	**8.29**	**13.69**	**14.33**
6ch	S1	GMM	–	beamformit	12.78	14.87	23.13	23.06
		DNN			9.41	10.43	17.26	17.14
		+sMBR			8.33	9.21	15.72	15.88
		+5-gram			6.91	7.96	13.75	13.63
		+RNNLM			5.99	7.16	12.21	12.42
		+GRNNLM			6.03	7.21	12.07	12.50
	S2	GMM	–	EV-GSC	11.21	11.92	23.41	16.13
		DNN			8.32	8.32	17.36	11.75
		+sMBR			7.37	7.52	15.55	10.83
		+5-gram			6.01	6.14	14.05	9.35
		+RNNLM			5.14	5.48	12.60	8.56
		+GRNNLM			5.16	5.51	12.64	8.35
	S3	DNN	pert	EV-GSC	7.82	7.96	16.13	11.01
		+sMBR			6.83	6.86	14.34	10.16
		+5-gram			5.66	5.76	12.78	8.70
		+RNNLM			4.71	5.13	11.53	7.44
		+GRNNLM			4.74	**5.05**	11.45	**7.34**
	S4	GMM	MC	EV-GSC	11.05	11.77	19.65	15.93
		DNN			8.15	7.94	14.38	11.37
		+sMBR			7.30	7.49	13.38	10.56
		+5-gram			5.82	6.17	11.55	9.51
		+RNNLM			4.96	5.27	10.23	8.14
		+GRNNLM			4.86	5.29	10.08	8.06
	S5	DNN	MC + pert	EV-GSC	7.65	8.03	13.53	10.89
		+sMBR			6.81	7.24	12.50	10.01
		+5-gram			5.53	6.08	10.94	8.57
		+RNNLM			**4.65**	5.35	9.63	7.38
		+GRNNLM			4.66	5.28	**9.54**	7.38
	S6	combination		EV-GSC	**4.22**	**4.73**	**8.98**	**7.02**

Table 2: WER (%) per environment for the best system.

Track	Envir.	Dev		Test	
		real	simu	real	simu
2ch	BUS	8.35	7.24	19.46	9.28
	CAF	5.78	10.80	13.41	16.92
	PED	4.23	5.86	12.07	15.00
	STR	7.45	9.25	9.81	16.12
6ch	BUS	5.25	3.79	13.72	4.20
	CAF	3.98	5.99	7.12	7.73
	PED	2.79	3.58	7.31	8.29
	STR	4.85	5.56	7.79	7.86

[6] S. Gannot and I. Cohen, “Speech enhancement based on the general transfer function GSC and postfiltering,” *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 6, Nov. 2004.

[7] J. Benesty, J. Chen, and Y. Huang, *Microphone Array Signal Processing*. Berlin–Heidelberg–New York: Springer, 2008.

[8] Y. Huang, J. Benesty, and J. Chen, *Acoustic MIMO Signal Processing*. Berlin–Heidelberg–New York: Springer, 2006.

[9] E. Warsitz and R. Haeb-Umbach, “Blind acoustic beamforming based on generalized eigenvalue decomposition,” *IEEE Transactions on audio, speech, and language processing*, vol. 15, no. 5, Jul. 2007.

[10] P. Vary and R. Martin, *Digital Speech Transmission*. West Sussex: Wiley, 2006.

[11] M. Zöhrer, R. Peharz, and F. Pernkopf, “Representation learning for single-channel source separation and bandwidth extension,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 12, pp. 2398–2409, 2015.

[12] L. Pfeifenberger, M. Zöhrer, and F. Pernkopf, “Dnn-based speech mask estimation for eigenvector beamforming,” in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 2017, submitted.

[13] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, “The kaldi speech recognition toolkit,” in *ASRU*, 2011.

The FBK system for the CHiME-4 challenge

Marco Matassoni, Mirco Ravanelli, Shahab Jalalvand, Alessio Brutti, Daniele Falavigna

Fondazione Bruno Kessler, Trento, Italy
{matasso,mravanelli,jalalvand,brutti,falavi}@fbk.eu

Abstract

This paper describes the ASR system submitted by FBK to the CHiME-4 challenge for the single channel track. The proposed solution employs multiple subsystems, whose DNNs are trained with different training criteria and strategies (i.e. diverse training material, with and without batch normalization). A "self" adaptation of acoustic models is applied to each subsystem, relying on a blind estimate of the accuracy of automatic transcriptions. This adaptation, performed in a batch fashion over the entire evaluation set, significantly improves the performance of each subsystem. The final output is obtained by combining the multiple transcriptions through ROVER, which provides a further improvement, reducing the average WER on the evaluation set from 22.3% to 16.5%.

1. Introduction

In a number of application scenarios (e.g., home automation, smart cars, robots), performance of automatic speech recognition (ASR) is heavily affected by noises of various types, competing speakers and reverberation effects. The CHiME challenges [1, 2, 3, 4] represent an excellent framework to evaluate signal enhancement and noise-robust acoustic models for ASR in such realistic conditions. Built upon the previous CHiME-3 challenge, the CHiME-4 dataset comprises utterances recorded by a 6-channel tablet-based microphone array. The recognition task is the automatic transcription of read sentences from the Wall Street Journal (WSJ) corpus, acquired in four noisy conditions; [4] illustrates training, development and evaluation data sets released for the competition. The results in [3] proved the effectiveness of signal enhancement approaches combined with the use of hybrid acoustic models based on deep neural networks hidden Markov models (DNN-HMMs) [5, 6, 7, 8].

In this submission we consider the *1ch*-track of the challenge, focusing specifically on deep learning techniques and building upon our previous submission for the CHiME-3 challenge [9], where an effective two-pass strategy was explored. In that work the DNNs employed to recognize each input stream (beamformed or single channels) were re-trained using the corresponding automatic transcription generated with the baseline acoustic models. A MAP selection procedure, at sentence level, produced the improved final transcriptions.

For the current *1ch*-track CHiME-4 challenge, only a single channel is available in the decoding pass and the multiple hypotheses generated for a final ROVER combination are derived from systems exploiting not only different training material, as done in [9], but also introducing a variety of DNN architectures. Secondly, we improve the model adaptation stage, replacing the standard retraining on the whole adaptation set with a more sophisticated solution, which enhances the adaptation with effective instance weighing and selection criteria. Finally, the combination of the hypotheses provided by the sub-systems is based on our previous work on driving ROVER with segment-based ASR quality estimation [10].

The paper presents in Section 2 the approach and the main features of the proposed system while Section 3 describes the steps of the processing pipeline and Section 4 reports the corresponding WER results. Section 5 concludes the work, presenting possible future directions.

2. Main characteristics

The main features explored in our current submission are the introduction of diverse DNN architectures in order to be able to rank, select and combine multiple hypotheses after an effective DNN adaptation stage; Figure 1 shows the blocks detailed in Section3.

In particular, we explored the use of *batch-normalized DNNs*. Training DNNs is indeed complicated by the fact that the distribution of each layer's inputs changes during training, as the parameters of the previous layers change. This problem, known as internal covariate shift, slows down the training of deep neural networks. Batch normalization [11] addresses this issue by normalizing the mean and the variance of each layer for each training mini-batch, and back-propagating through the normalization step. It has been long known that the network training converges faster if its inputs are properly normalized [12] and, in such a way, batch normalization extends the normalization to all the layers of the architecture. However, since a per-layer normalization may impair the model capacity, a trainable scaling parameter γ and a trainable shifting parameter β are introduced in

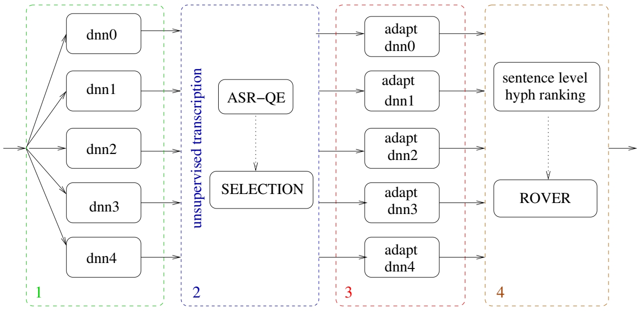


Figure 1: The architecture of the proposed CHiME-4 automatic transcription system, characterized by a four-steps pipeline.

each layer to restore the representational power of the network. The above-mentioned systems, implemented with Theano [13], are coupled with the Kaldi toolkit [14] to form a context-dependent DNN-HMM speech recognizer.

Another technique explored in this work is *DNN adaptation*. The usual way to adapt a DNN trained on a large set of data, given a much smaller set of adaptation data, is to retrain the DNN over the adaptation set, which could lead to overfitting the model on the adaptation data. A solution to prevent these detrimental effects is to adopt a conservative learning procedure by adding a regularization component to the loss function. The adaptation technique proposed here is based on a Kullback-Leibler divergence (KLD) regularization [15]. KLD regularization can be implemented through cross-entropy minimization between a new target probability distribution and the current probability distribution. Moreover, this regularization binds directly the DNN output probabilities rather than the model parameters; as a consequence, the method can be easily implemented with any software tool based on back-propagation, without introducing any modification.

In addition, we evolved our previous system by exploiting a recently developed *automatic quality estimator* (QE), which is able to provide (sentence by sentence) a confidence score related to the expected word error rate (WER%). Automatic assessment methods can be used to select audio data for unsupervised training [16], active learning of acoustic models [17, 18], combination of multiple transcription hypotheses into a single and more accurate one [19]. The proposed technique, which has shown promising in both ASR and machine translation applications [20, 10], contributed to this submission in two ways. First, we used the confidence scores to automatically select the best subset of utterance for the unsupervised adaptation step. Secondly, we exploit such a confidence score to rank multiple hypothesis prior to a standard system combination based on ROVER, as done in our previous submission.

3. System implementation

The architecture of our proposed system, depicted in Fig.1, is based on four steps: generation of preliminary transcriptions using the models trained on the noisy channels; quality estimation of the resulting hypotheses and selection of suitable adaptation sentences according to WER predictions; DNN adaptation using KLD regularization; systems combination through ROVER.

3.1. Step 1: multiple DNN-based speech recognizers

With the final purpose of improving system diversity, different DNNs have been considered. All the DNNs use the standard 40 fMLLR features used in the CHiME-4 baseline recipe [4]. Such features are then gathered into a context windows of 11 consecutive frames prior to feeding a feed-forward DNN. A Stochastic Gradient Descend (SGD) algorithm is used as DNN optimizer.

A first system (*dnn0*) based on the CHiME-4 baseline has been trained using one single channel (CH5), as originally proposed. A second DNN (*dnn1*), is trained following again the baseline recipe but exploiting all the six channels available in the training-set (CH1-CH6).

In addition, a set of batch-normalized DNNs are trained (*dnn2-4*). For these systems (due to time and computational restrictions) the standard training-set (based on channel 5 only) was used. The adopted batch-normalized DNNs are based on Rectified Linear Units (ReLU) and employ drop-out (with a drop-out rate of $\rho = 0.2$). Moreover, to further improve the system performance, the labels for DNN training are derived by a forced-alignment over the close-talking signals. Such an approach has been studied in [21]. The first batch-normalized DNN (*dnn2*) is based on six hidden layers composed of 2048 neurons. A second batch-normalized DNN (*dnn3*) is trained with the same architecture, but exploiting features derived by an automatic classification of the environment. More specifically, a DNN is trained using the environmental labels in the training set and the posterior probabilities generated by such a network are concatenated with the standard fMLLR features. The last batch-normalized DNN (*dnn4*), inspired by our recent work on joint training [22], concatenates a speech enhancement and a speech recognition deep neural network, whose parameters are jointly updated as if they were within a single bigger network. More precisely, in the joint training framework we perform a forward pass, compute the loss functions at the output of each DNN (mean-squared error for speech enhancement and cross-entropy for speech recognition), compute the corresponding gradients, and back-propagate them though.

Particular attention should be devoted to the initialization of the γ parameter. Contrary to [11], where it was initialized to unit variance ($\gamma = 1$), in this work we have observed better performance and convergence properties with a smaller variance initialization ($\gamma = 0.1$). A similar outcome was found in [23, 24], where fewer vanishing gradient problems are empirically observed with small values of γ in the case of recurrent neural networks.

3.2. Step 2: Quality Estimation

The transcriptions generated by each hybrid DNN-HMMs systems are processed by a system that automatically estimates the WERs of each sentence. The approach makes use of a supervised regression method that effectively exploits a combination of "glass-box" and "black-box" features [20, 10]. Glass-box features, similar to confidence scores, refer to the one extracted when the ASR features such as lattice and confidence scores are available, and capture information inherent to the inner workings of the ASR system that produced the transcriptions. The black-box ones, instead, are extracted by looking only at the signal and the transcription. On one side, they try to capture the *difficulty* of transcribing the signal while, on the other side, they try to capture the *plausibility* of the output transcriptions. In both cases, the information used is independent of knowledge about the ASR system, making the approach of [20] ASR QE applicable to a wide range of scenarios in which the only elements available for quality prediction are the signal and the transcription. The extensive experiments in different testing conditions discussed in [20, 10] indicate that regression models based on Extremely Randomized Trees (XRT) [25] can achieve competitive performance, being able to outperform strong baselines and to approximate the true WER scores computed against reference transcripts. For the experiments reported here we trained two different XRT based regressor on the CHiME-4 development sets: dt05_simu and dt05_real, and used the resulting models on the related evaluation sets.

3.3. Step 3: DNN unsupervised adaptation

The WER predictions of the sentences in each evaluation set are hence used to build adaptation sets containing sentences of mid-high quality. In particular, for these experiments we selected all the sentences with a predicted WER below 20%. The selected material is used to perform "self" DNNs adaptation (i.e. we are using, as adaptation sets, selected subsets of the test data.

The KLD regularization introduced for the adaptation step is implemented through cross-entropy minimization between a new target probability distribution and the current probability distribution. The new target distribution is obtained as a linear interpolation of the original distribution and the distribution computed via forced alignment with the adaptation data:

$$P[s_i|o_t] = (1-\alpha)\hat{p}[s_i|o_t] + \alpha\overset{*}{p}[s_i|o_t] \quad 0 \leq \alpha \leq 1 \quad (1)$$

Note that, in Eq. 1, $\alpha = 0$ is equivalent to a "pure" retraining of the DNN over the adaptation data, while $\alpha = 1$ means that the output probability distribution of the adapted DNN is forced to follow that of the original DNN . What one can expect is that the optimal value of α is close to 0 when the size of the adaptation set is large and the transcriptions of the adaptation sentences are not affected by errors (i.e. in supervised conditions). Conversely, when the size of the adaptation set is small and/or its transcription can be affected by errors (i.e. in the case of unsupervised adaptation), α should increase.

Table 1: Average WER (%) for the each systems and the final combination

Track	System	Dev		Test	
		real	simu	real	simu
1ch	sys0	10.42	12.54	20.09	18.17
	sys1	9.02	10.98	17.21	16.52
	sys2	9.64	11.48	18.44	17.40
	sys3	9.65	11.52	18.26	16.99
	sys4	10.02	12.92	18.62	18.23
	comb	**9.02**	**9.51**	**16.87**	**16.09**

DNNs are adapted to the acoustic conditions of each evaluation set: we adapt a different DNN for each one of the two sets: dt05 and et05. The automatic supervision of each adaptation set is given by the ASR hypotheses generated in the first decoding pass of Figure 1.

A final decoding step is then carried out using the adapted DNNs, followed by the LM rescoring pass included in the CHiME-4 baseline (based on a linear combination of 5-gram LM and RNNLM).

3.4. Step 4: hypotheses combination

A common way to combine multiple ASR hypotheses is through ROVER [CITE]. However, the behaviour of ROVER strongly depends on the order of the hypotheses[CITE], and the overall performance could substantially improve if the ARS transcription are ranked according to they accuracy [10]. Therefore, the ASR transcriptions, obtained after the unsupervised DNN adaptation, are automatically ranked at sentence level using the QE system described in [10]. We train an automatic ranking system for each development data sets (dt05_simu and dt05_real), and used it to rank the sentence hypotheses of the evaluation sets: et05_simu and et05_real.

4. Experimental evaluation

4.1. Submitted system

Table 1 reports the results obtained with each subsystems and with their final combination. The systems labeled as "*sys0-4*" refer to five DNNs (*dnn0-4*) after the unsupervised adaptation. The system "*comb*" represents the final ROVER combination.

We can observe that, as expected, the best single sys-

Table 2: WER (%) per environment for the submitted system

Track	Envir.	Dev		Test	
		real	simu	real	simu
1ch	BUS	12.41	8.01	24.57	12.01
	CAF	8.70	12.12	18.36	18.36
	PED	6.23	7.30	13.60	15.57
	STR	8.73	10.59	10.96	18.23

tem is *sys1*, since it is trained with all the available channels. However, the performance obtained with batch-normalized DNNs (*sys2-sys4*) are rather competitive with *sys1*, even if such systems are training with a single channel only. However, the comparison between *sys0* (no batch-norm) and *sys2* (with batch norm) confirms the significant benefits obtained with such a technique. Results also reveals that the addition of the environmental features seems to give only minor benefits (compare *sys2* and *sys3*). We also found that, differently to what we experimented in [22], the joint training systems (*sys3*) performs slightly worse than a single DNN case. The last row of Table 1 reports the results obtained by combining all the considered systems. The performance obtained with the latter system for each noise conditions is reported in Table 2.

4.2. Updated system

The importance of the quality of automatic transcriptions for the adaptation pass suggested us to introduce a modification in the system architecture, i.e. to make use of an additional combination stage after the initial decoding step; indeed, it is possible to automatically rank [10] also the hypotheses generated in the pass-1 and select the ”best” one as supervision for all the systems in pass-3. Table 3 shows the results obtained with this new adaptation strategy, represented in Figure 2. An additional gain is achieved, indicating that the improved transcription obtained exploiting the diversity of multiple systems produces better adapted DNN models.

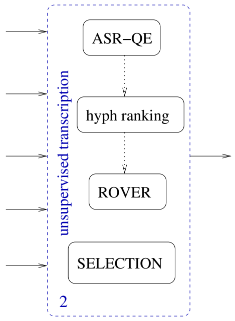


Figure 2: The updated pass-2: a unique QE-based supervision is derived for all the DNN systems.

Table 3: Average WER (%) for the updated system in which the pass-2 produces a single supervision for all the DNN systems.

Track	System	Dev		Test	
		real	simu	real	simu
1ch	comb (new)	**8.45**	**10.56**	**16.17**	**15.20**

5. Discussion and conclusions

In this work we have proposed a refinement of the system previously submitted to the CHiME-3 challenge [9]. The *two-pass decoding* combined with *automatic data selection* for DNN adaptation benefited from previous experience on quality estimation of ASR hypotheses in the framework of ASR system combination [10].

To perform data selection we applied ASR quality estimation, using automatic WER prediction as a criterion to isolate subsets of the adaptation data featuring variable quality. As a result, ASR QE-based data selection, in combination with KLD-based DNN adaptation, provides a significant advantage. Instead, the diversity of the hypotheses generated by DNNs trained on different channels or with different procedure (batch-normalization) is quite limited and the final combination step provides small improvements with respect to the single systems.

Overall, the experimental results confirm the effectiveness of the proposed approach that, using the provided training set and the baseline language models, allows to improve from 22.3% to 16.5% WER (average on the evaluation set).

Finally, note that the regularization coefficient α in Eq. 1 can be made dependent on the quality of each test sentence (e.g., by predicting the corresponding WER or by implementing a specific training phase for estimating sentence dependent α_k, being k the identifier of the k^{th} test utterance) allowing to implement a soft scheme for DNN adaptation: this approach has given promising results on the recognition of a data set of children speech [26].

A planned direction for further investigations is the introduction of more effective types of neural network architectures (Convolutional Neural Networks or Long-Short Term Memory Recurrent Neural Networks [27]), both for improving the overall performance of the related ASR systems and for augmenting the diversity of the hypotheses. In this way both the quality of the supervision and the efficacy of hypotheses combination are expected to increase.

6. References

[1] J. Barker, E. Vincent, N. Ma, H. Christensen, and P. Green, "The PASCAL CHiME Speech Separation and Recognition Challenge," *Computer Speech & Language*, vol. 27, no. 3, pp. 621–633, 2013.

[2] E. Vincent, J. Barker, S. Watanabe, J. Le Roux, F. Nesta, and M. Matassoni, "The Second CHiMEspeech Separation and Recognition Challenge: An Overview of Challenge Systems and Outcomes," in *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Olomouc, Czech Republic, 2013, pp. 162–167.

[3] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third 'CHiME' Speech Separation and Recognition Challenge: Dataset, Task and Baselines," in *Proc. of the 15th IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Scottsdale, Arizona, USA, 2015, pp. 1–9.

[4] E. Vincent, S. Watanabe, A. Nugraha, J. Barker, and R. Marxer, "An analysis of environment, microphone and data simulation mismatches in robust speech recognition," *Computer Speech and Language, to appear*, 2016.

[5] G. Hinton, L. Deng, D. Yu, and Y. Wang, "Deep Neural Networks for Acoustic Modeling in Speech Recognition," *IEEE Signal Processing Magazine*, vol. 9, no. 3, pp. 82–97, 2012.

[6] A. Mohamed, G. Dahl, and G. Hinton, "Acoustic Modeling Using Deep Belief Networks," *IEEE Trans. on Audio Speech and Language Processing*, vol. 20, no. 1, pp. 14–22, 2012.

[7] P. Swietojanski and S. Renals, "Hybrid Acoustic Models for Distant and Multichannel Large Vocabulary Speech Recognition," in *Proc. of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Olomuc, Czech Rep., 2013, pp. 285–290.

[8] S. Renals and P. Swietojanski, "Neural Networks for Distant Speech Recognition," in *Proc. of Hands-free Speech Communication and Microphone Arrays (HSCMA) Wokshop*, Villers-les-Nancy, 2014, pp. 172–176.

[9] S. Jalalvand, D. Falavigna, M. Matassoni, P. Svaizer, and M. Omologo, "Boosted Acoustic Model Learning and Hypotheses Rescoring on the CHiME-3 Task," in *Proc. of the IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Scottsdale, Arizona, USA, 2015, pp. 409–415.

[10] S. Jalalvand, M. Negri, D. Falavigna, and M. Turchi, "Driving ROVER With Segment-based ASR Quality Estimation," in *Proc. of ACL*, Beijing, China, July 2015.

[11] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. of ICML*, 2015, pp. 448–456.

[12] Y. LeCun, L. Bottou, G. Orr, and K. Müller, "Efficient backprop," in *Neural networks: Tricks of the trade*. Springer Berlin Heidelberg, 1998, pp. 9–50.

[13] Theano Development Team, "Theano: A Python framework for fast computation of mathematical expressions," *arXiv e-prints*, vol. abs/1605.02688, May 2016.

[14] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi Speech Recognition Toolkit," in *Proc. of ASRU*, 2011.

[15] D. Yu, K. Yao, H. Su, G. Li, and F. Seide, "KL-Divergence Regularized Deep Neural Network Adaptation for Improved Large Vocabulary Speech Recognition," in *Proc. of ICASSP*, Vancouver (Canada), May, 26-31 2013, pp. 7893–7897.

[16] L. Lamel, J.-L. Gauvain, and G. Adda, "Investigating Lightly Supervised Acoustic Model Training," in *Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1, Salt Lake City, USA, 2001, pp. 477–480.

[17] G. Riccardi and D. Hakkani-Tur, "Active Learning: Theory and Applications to Automatic Speech Recognition," *IEEE Transaction on Speech and Audio Processing*, vol. 13, no. 4, pp. 504–511, 2005.

[18] A. Facco, D. Falavigna, R. Gretter, and V. Vigano, "Design and Evaluation of Acoustic and Language Models for Large Scale Telephone Services," *Speech Communication*, vol. 48, no. 2, pp. 176–190, 2006.

[19] J. G. Fiscus, "A Post-processing System to Yield Reduced Word Error Rates: Recognizer Output Voting Error Reduction (ROVER)," in *Proc. of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*. Santa Barbara, CA, USA: IEEE, 1997, pp. 347–354.

[20] M. Negri, M. Turchi, D. Falavigna, and J. G. C. de Souza, "Quality Estimation for Automatic Speech Recognition," in *Proc. of COLING*, Dublin, Ireland, 2014.

[21] M. Ravanelli and M. Omologo, "Contaminated speech training methods for robust DNN-HMM distant speech recognition," in *Proc. of INTERSPEECH 2015*, pp. 756–760.

[22] M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio, "Batch-normalized joint training for DNN-based distant speech recognition," in *Proc. of SLT 2016*.

[23] T. Cooijmans, N. Ballas, C. Laurent, Ç. Gülçehre, and A. Courville, "Recurrent batch normalization," *arXiv preprint arXiv:1603.09025*, 2016.

[24] M. Ravanelli, P. Brakel, M. Omologo, and Y. Bengio, "A network of deep neural networks for distant speech recognition," in *submitted to ICASSP 2016*.

[25] P. Geurts, D. Ernst, and L. Wehenkel, "Extremely Randomized Trees," *Machine Learning*, vol. 63, no. 1, pp. 3–42, 2006.

[26] M. Matassoni, D. Falavigna, and D. Giuliani, "Cross and Self Adaptation of DNN for Recognition of Children Speech," in *Proc. of SLT*, San Diego (CA), Usa, December 2016.

[27] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition," *Interspeech*, 2014.

A Study of Learning Based Beamforming Methods for Speech Recognition

Xiong Xiao[1], *Chenglin Xu*[1], *Zhaofeng Zhang*[2], *Shengkui Zhao*[3], *Sining Sun*[4], *Shinji Watanabe*[5]
Longbiao Wang[6], *Lei Xie*[4], *Douglas L. Jones*[3], *Eng Siong Chng*[1], *Haizhou Li*[7,1]

[1]Nanyang Technological University (NTU), Singapore, [2]Nagaoka University of Technology, Japan,
[3]Advanced Digital Sciences Center, Singapore, [4]Northwestern Polytechnical University, China,
[5]Mitsubishi Electric Research Laboratories, USA, [6]Tianjin University, China,
[7]National University of Singapore, Singapore.

{xiaoxiong, xuchenglin}@ntu.edu.sg, s147002@stn.nagaokaut.ac.jp, shengkui.zhao@adsc.com.sg

Abstract

This paper presents a comparative study of three learning based beamforming methods that are specifically designed for robust speech recognition. The three methods are 1) neural network that predicts beamforming weights from generalized cross correlation (GCC) features; 2) neural network that predicts time-frequency (TF) mask which is used to estimate MVDR (minimum variance distortionless response) beamforming weights; 3) maximum likelihood estimation of beamforming weights to fit enhanced features to clean trained Gaussian mixture model. All three methods operate in frequency domain. They are evaluated on the CHiME-4 benchmarking speech recognition task and compared with traditional delay-and-sum and MVDR beamforming methods on the same speech recognition task. Discussions and future research directions are presented.

1. Introduction

Beamforming is an important approach to improve the performance of automatic speech recognition (ASR) in far field scenarios.. Traditional beamforming methods enhance the speech signals to improve signal level criteria, e.g. the signal-to-noise ratio (SNR) of output signal. As these criteria are not directly related to the ASR's performance measure, tradiitonal methods are usually not optimized for the ASR task.

Recently, several learning based beamforming methods are proposed for the ASR task. By learning based methods, we mean these methods learn from a large amount of training data (single or multi-channel), and apply the learned knowledge at run time to estimate parameters for ASR, e.g. beamforming weights. In one approach [1–3], multi-channel raw waveforms are fed into the neural network acoustic model directly. A temporal convolution layer at the bottom of the network is used to approximate the filter-and-sum beamforming operation. After training, the temporal convolution layer learnes a fixed bank of spatial and temporal filters, each with specific looking directions. We call this approach the spatial filter learning approach. In another approach, beamforming filter weights are predicted by neural networks that are jointly optimized with the acoustic model networks. Deep neural network (DNN) is used to predict beamforming weights in frequency domain from generalized cross correlation (GCC) features [4] or spatial covariance matrix (SCM) features [5]. In [6], long short-term memory (LSTM) networks are used to predict the beamforming weights in the time domain directly which has less number of free parameters than the frequency domain. We call this appraoch the spatial filter prediction approach. While the filter learning approach learns a fixed set of spatial filters, the filter prediction approach predicts spatial filters dynamically from the input data. In another approach, neural networks are used to predict time-frequency (TF) mask that specifies whether a TF bin is dominated by speech or noise. The TF mask is used to help estimating the speech and noise SCMs required by beamforming methods, such as the minimum variance distortionless response (MVDR) [7, 8] and generalized eigenvalue (GEV) [9, 10] beamformers. The mask predicting network can be trained by using ideal masks as target [11–13] or by minimizing the ASR cost function [14]. The filter learning, filter predicting, and mask predicting approaches are called discriminative approach in this paper, as the models are trained to minimize the ASR error.

Besides discriminative methods, there is also learning based beamforming methods based on generative modeling of speech features. In [15, 17], a method called LIMABEAM estimates time or frequency domain filter-and-sum weights to maximize the likelihood of the enhanced feature vectors on clean trained HMM/GMM acoustic model. In the unsupervised implementation, multi-pass decoding is required, where the first pass decoding provides the hypothesized text used to obtain HMM state alignment. Beamforming weights can be estimated iteratively to maximize the likelihood of the enhanced features given the state alignment. It is reported that LIMABEAM outperforms delay-and sum beamforming in several ASR tasks.

Although several learning based methods have been proposed in the past, they are usually implemented by different researchers and evaluated on different ASR tasks. As a result, it is difficult to compare their performance. In this paper, we attempt to study three learning based beamforming methods comparatively, with the implementation in the same toolkit, i.e. SignalGraph [25], and evaluation in the same task, i.e. the CHiME-4 speech recognition task [16]. The three methods include a maximum likelihood (ML) beamforming (a variant of LIMABEAM [15]), the spatial filter weight predicting network [4], and the mask predicting network [14].

2. Learning Based Beamforming Methods

2.1. Spatial Filter Weight Predicting Network

The system diagram of the spatial filter weight predicting network [4] is shown in Fig. 1. On the bottom left of the figure, a network is used to predict the beamforming weights in frequency domain. The weights are then applied on the multichannel inputs to generate enhanced speech, from which features are extracted for acoustic modeling.

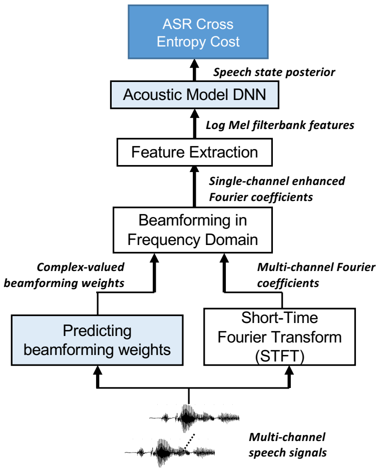

ASR Cross Entropy Cost
Speech state posterior
Acoustic Model DNN
Log Mel filterbank features
Feature Extraction
Single-channel enhanced Fourier coefficients
Beamforming in Frequency Domain
Complex-valued beamforming weights
Multi-channel Fourier coefficients
Predicting beamforming weights
Short-Time Fourier Transform (STFT)
Multi-channel speech signals

Figure 1: Discriminative beamforming weight prediction.

The weight prediction network and the acoustic model network are jointly optimized using the ASR cost function. The weight predicting network is initialized by learning from a delay-and-sum filter on simulated data. Specifically, if the true time difference of arrival (TDOA) of the different channels are known, which is the case for simulated data, we can use the ideal delay-and-sum filter weights as the target for the weight predicting network to learn. The network predicts the real and imaginary of the ideal weights independently. Mean square error (MSE) between the ideal weights and predicted weights is used as the cost function in initialization. After the initialization, the weight predicting network is jointly refined with the acoustic model using back propagation and ASR cost.

The details of the weight predicting network is illustrated in Fig. 2. From the waveforms, we extract feature vectors from GCC function between two channels [18] for every 0.2s long frame with 0.1s shift. The GCC feature vectors encode the phase information of channels and the features extracted from all channel pairs are concatenated to form a single feature vector for each frame. For the CHiME-4 data [16], the dimensionality of the GCC feature vector is 27 for each channel pair. This is because the maximum TDOA is less than 13 samples for the array geometry used in CHiME-4 and 16kHz sampling rate. The bottom right of Fig. 2 shows example GCC features. As different direction of arrival (DOA) angles have different GCC patterns, the GCC features contain information for DNN to determine spatial direction of the source and also TDOA [19]. In this work, a DNN is used to map the GCC features to the beamforming weights in frequency domain. For stable estimation of weights, we take the mean of predicted weight vectors of all frames for each sentence.

While the array geometry is assumed to be fixed in [4], in the two channel track of the CHiME-4 benchmarking task, the geometry of the array depends on the distance bewteen the two microphones randomly selected from a 6-microphone array. We will test whether one single weight predicting network is able to cover several array geometries.

2.2. Time Frequency Mask Predicting Network

The TF mask predicting network is illustrated in Fig. 3. The log power spectra of input signals are mean normalized on an utterance basis and used as features for mask prediction. The mask prediction is carried out for each channel independently, but shares the same LSTM mask predictor. For each channel, two TF masks are predicted by the LSTM network, one speech mask that specifies whether a TF bin is speech dominated and one noise mask. We call this splitted mask. We can also force the speech and noise masks to sum to 1 for each TF bin. This can be implemented by only predicting the speech mask and obtain the noise mask by 1-speech mask.

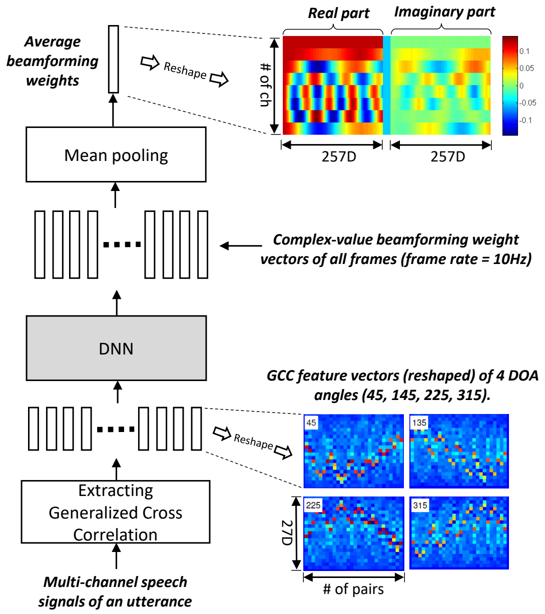

Real part
Imaginary part
Average beamforming weights
Reshape
# of ch
0.1
0.05
0
-0.05
-0.1
257D
257D
Mean pooling
Complex-value beamforming weight vectors of all frames (frame rate = 10Hz)
DNN
GCC feature vectors (reshaped) of 4 DOA angles (45, 145, 225, 315).
Reshape
45
135
225
315
27D
# of pairs
Extracting Generalized Cross Correlation
Multi-channel speech signals of an utterance

Figure 2: Details of weight predicting network. The size of the GCC feature matrix (bottom right) depends on the number of unique channel pairs.

The LSTM network contains one hidden layer, whose activation vector is projected to noise and speech mask vectors by using two independent projection layers. The sigmoid activation function is used for projection layers to ensure that the predicted masks will have value between 0 and 1. For both noise and speech masks, pooling is used to reduce the set of masks of all channels to a single mask. Four types of pooling functions are compared, including mean, median, min, and max. Note that during training, we only uses one channel (the first channel) to predict the mask, and hence pooling is not necessary. Only at testing, we may estimate the masks for all channels and use pooling.

Given the mask, the MVDR beamforming weights can be determined as follows [20],

$$\mathbf{w}(f) = \frac{\Phi_{nn}^{-1}(f)\Phi_{ss}(f)\mathbf{u}}{\mathrm{Tr}[\Phi_{nn}^{-1}(f)\Phi_{yy}(f)]} \tag{1}$$

where $\mathbf{u}$ is a vector with the element for reference channel being 1 and all others being 0. $\mathrm{Tr}[\cdot]$ denotes trace of a matrix. Φ_{nn} and Φ_{ss} are the noise and speech SCMs estimated as

$$\Phi_{ss}(f) = \frac{\sum_{t=1}^{T} \hat{m}_t^s(f)\mathbf{y}_t(f)\mathbf{y}_t^H(f)}{\sum_{t=1}^{T} \hat{m}_t^s(f)} \tag{2}$$

$$\Phi_{nn}(f) = \frac{\sum_{t=1}^{T} \hat{m}_t^n(f)\mathbf{y}_t(f)\mathbf{y}_t^H(f)}{\sum_{t=1}^{T} \hat{m}_t^n(f)} \tag{3}$$

where $\hat{m}_t^s$ and $\hat{m}_t^n$ are the estimated mask values at frame t and frequency f for speech and noise, respectively. $\mathbf{y}_t(f)$ is the observed signal in frequency domain.

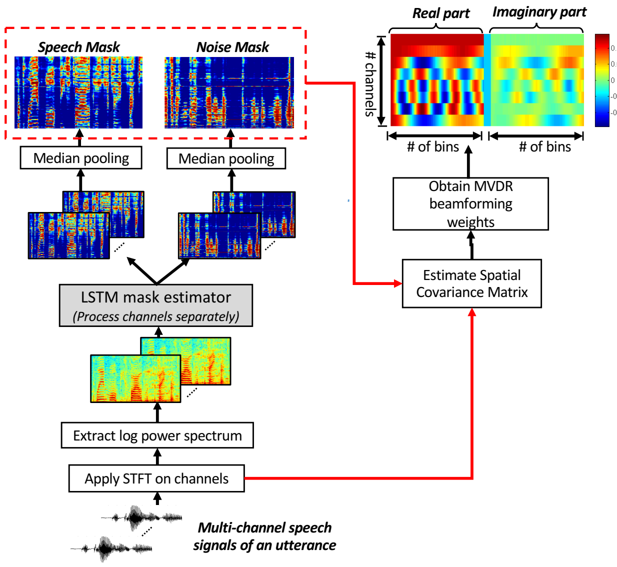

Real part
Imaginary part
Speech Mask
Noise Mask
# channels
# of bins
# of bins
Median pooling
Median pooling
Obtain MVDR beamforming weights
Estimate Spatial Covariance Matrix
LSTM mask estimator
(Process channels separately)
Extract log power spectrum
Apply STFT on channels
Multi-channel speech signals of an utterance

Figure 3: Details of mask predicting network and the estimating of MVDR weights.

The LSTM mask predicting network is initialized by learning from ideal binary mask (IBM) of speech. For simulated data, we can obtain the oracle local SNR for each TF bin. The speech IBM is set to 1 if the local SNR is larger than 0dB and vice versa. Then the LSTM network is trained to predict the speech IBM from single channel log power spectrum by minimizing the mean square error (MSE) between the predicted mask and the IBM. Once initialized, the network in Fig. 3 is used to replace the weight predicting module in Fig. 1, and the LSTM mask predictor is jointly refined with the acoustic model to minimize ASR cost function. The noise projection layer's weights and bias can be initialized as the negative weights and bias of the speech projection layer so the sum of noise and speech masks sum to one for each TF bin. Note that, after joint training, the noise and speech masks usually do not sum to 1.

2.3. Maximum Likelihood Spatial Filter Estimation

We also investigate a modified version of the LIMABEAM [15]. The beamforming parameters are estimated as follows:

$$\begin{aligned}\hat{\mathbf{W}}_{\text{ML}} &= \arg\max_{\mathbf{W}} \frac{1}{T}\log p\left(\mathbf{O}(\mathbf{W});\Theta\right) \\ &\quad + \frac{1}{2}\log\left|\Sigma_{\mathbf{O}}(\mathbf{W})\right| - \frac{\alpha}{2}|\mathbf{W}-\mathbf{W}_0|_F^2 \qquad (4)\end{aligned}$$

where $\mathbf{O}(\mathbf{W})$ is the enhanced feature vectors and is a function of the beamforming weights. Θ is the parameters of the acoustic model and T is the number of frames in the test utterance. The first term in (4) measures the likelihood of the enhanced features evaluated on the acoustic model, which can be an HMM/GMM or GMM. When the acoustic model represents clean features' distribution, it is a reasonable assumption that the higher the likelihood is, the higher the quality of the enhanced features [15]. The second and third terms are added in this work to the orginal LIMABEAM. The second term is the log determinant of the covariance matrix of the enhanced features (also a function of weights) and it acounts for the nonlinear transformation of the feature space [21, 22] due to the beamforming operation. The third term is the Frobenius norm between the weight matrix and its initial values. This term is used to impose L2 norm regularization on the parameters to prevent overfitting. The modified LIMABEAM is called maximum likelihood beamforming (MLBF) in this paper and illustrated in Fig. 4. The three terms in (4) are represented as three cost function nodes in blue color.

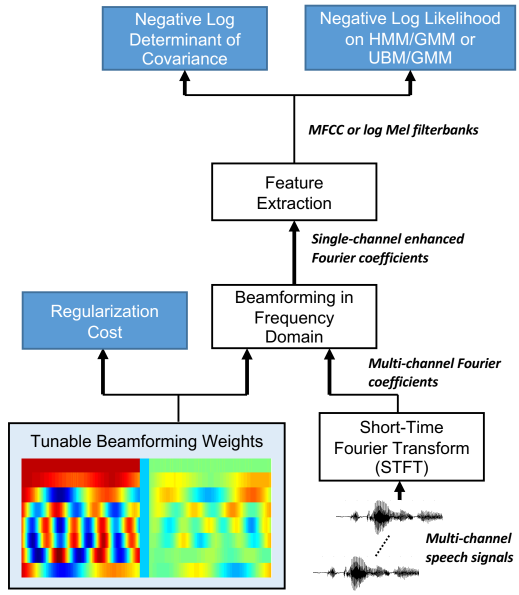

Negative Log Determinant of Covariance
Negative Log Likelihood on HMM/GMM or UBM/GMM
MFCC or log Mel filterbanks
Feature Extraction
Single-channel enhanced Fourier coefficients
Regularization Cost
Beamforming in Frequency Domain
Multi-channel Fourier coefficients
Tunable Beamforming Weights
Short-Time Fourier Transform (STFT)
Multi-channel speech signals

Figure 4: System diagram of maximum likelihood based beamforming weight estimation.

Instead of using HMM/GMM as the acoustic model, we use a single GMM to model the distribution of the clean MFCC features. The advantage of using GMM is that there is no need to perform one extra pass of decoding to obtain the HMM state alignment. However, it is possible that performance will degrade compared with using HMM/GMM.

There are two ways to represent the frequency domain beamforming weights. In the first way, we treat the real and imaginary parts of the weights as free parameters, hence there are $2IK$ free parameters, where I and K are the number of channels and frequency bins, respectively. In the second way, the weights are represented as follows

$$w_i(f) = g_i(f)\exp(j2\pi f\frac{\tau_i}{f_s}) \qquad (5)$$

where $w_i(f)$ and $g_i(f)$ are the weight and gain for channel i at frequency f, respectively. f_s is the sampling frequency, $\frac{\tau_i}{f_s}$ is the TDOA of channel i and assumed to be frequency independent. The first channel is always selected as the reference channel and its TDOA is set to 0. Hence, there are totally $I-1$ free parameters from TDOA, and IK free parameters from gain.

3. Experiments

3.1. Settings

We evaluate the three learning based beamforming methods on the 2-channel and 6-channel tracks of the CHiME-4 task [16]. The baseline DNN acoustic model is used, except that the fMLLR [23] features are replaced by 40D log Mel filterbank features, due to the fact that fMLLR needs to be dynamically estimated and makes it difficult to conduct joint training of beamforming networks and acoustic model. No pre-emphasis or

DC removal is applied. Delta and acceleration features are appended and then 11 frames of feature vectors are concatenated to form the input for the DNN acoustic model. Two types of DNN acoustic model is used, one is trained from the fifth channel (called ch5 model, channel 5 is the single best channel in the array), while the other is trained from all the 6 channels (called chall model). The baseline trigram language model is used if not otherwise specified. Speech recognition is performed using the sequentially trained DNN acoustic model, i.e. the state-level minimum Bayes risk (SMBR) model [24].

All the three learning based beamforming methods are implemented in SignalGraph, a Matlab based toolkit for applying deep learning to signal processing [25]. The beamforming weight predicting network uses either a 3 hidden layer DNN or an 1 hidden layer LSTM, both using 1024 hidden nodes. The input to the network is 27D (1 microphone pair) for 2-channel case and 405D (15 microphone pairs) for 6-channel case. The output dimension is 257x2x2=1028 for 2-channel case and 257x2x6=3084 for 6-channel case, as 257 complex numbers (512 FFT length) need to be predicted for each channel. The network is initilized on 71680 simulated sentences (10 times of the official simulated training data) generated by ourselves using the provided simulation tool. After initilization, the network is refined together with the ch5 acoustic model (trained with cross entropy, or CE, criterion) by using the frame level CE cost function. As we will use the SMBR model for decoding, we fixed the acoustic model during joint training to prevent the acoustic feature space from drifting too much from the one used to train the SMBR model.

The mask predicting network is implemented by using a one hidden layer LSTM containing 1024 memory cells. The memory cells' outputs are projected to noise and speech masks by using two 1024 to 257 affine transorms in projection layers. The network is initialized on the 71680 simulated sentences (same as the data used to initialize the weight predicting network). After initialization, the network is jointly refined with the ch5 acoustic model in the same way as the joint training of weight predicting network.

The GMM used in the ML beamforming is trained from the close talk version of the 1680 real training sentences. The GMM uses 39D MFCC features and diagonal covariance matrix, and contains either 512 or 1024 Gaussians. The beamforming parameters are estimated iteratively using the expectation-maximization (EM) algorithm [26]. At most 3 EM iterations are used. At the E step of each EM iteration, the posteriors of the Gaussians are re-estimated using the enhanced features. At the M step, the beamforming parameters are re-estimated given the updated Gaussian posteriors by using the L-BFGS algorithm [27]. Due to the iterative nature of the EM algorithm, the real time factor is usually 1-5 for the whole estimation process for each sentence, which is much slower than the other two methods. When L2 regualization is used, it is used on all parameters except for the TDOAs.

3.2. Results of Beamforming Weight Predicting Network

The performance of weight predicting network is shown in Table 1. Row 2 and 3 show the results of MSE training in which the neural networks learn from the ideal unweighted delay-and-sum beamforming and simulated data. Comparing with the weighted delay-and-sum implemented in BeamformIt [28] (row 1), the neural networks perform slightly worse in overall, and LSTM performs slightly better than DNN. Row 4 and 5 show the results of CE training in which the neural networks are refined using the ASR cross entropy cost function on the official training data. For the two channel case, moderate imporvement is obtained by CE training over MSE training for DNN model (17.2% versas 16.5% on real data), while the results of LSTM model is mixed which could be due to overfitting.

Table 1: Recognition word error rate (WER %) obtained by weight predicting network on the CHiME-4 task. "DNN*" and "LSTM*" refer to CE refined model only using 1680 real recorded training sentences. The 5 channel case does not include the second channel. "DS" refers to BeamformIt.

Row	Model	Cost	6 channels		5 channels		2 channels	
			Eval		Eval		Eval	
			Real	Simu	Real	Simu	Real	Simu
1	DS	-	14.8	12.6	13.6	14.2	17.2	18.2
2	DNN	MSE	15.8	13.8	13.5	16.5	17.2	18.5
3	LSTM		14.7	13.4	12.9	14.9	16.5	18.3
4	DNN	CE	15.0	11.4	15.9	11.6	16.5	16.8
5	LSTM		14.6	11.5	14.7	11.6	16.8	17.3
6	DNN*		13.6	16.0	-			
7	LSTM*		14.6	14.5				

For 6 channel case, CE training obtains significant improvement overal MSE training on simulated data, but not on real data. One possible reason is that the target signal's gain is not equal at different channels for real data. Sometimes, channels may even totally fail to receive signals. The neural networks takes GCC features as input where the gain information is largely removed. Hence, the neural networks are unable to predict the gains of channels properly. To investigate the issue, we conducted two more experiments. First, we train the neural networks without using the second channel (5 channel case) which is known to have poor signal quality for real data. This leads to better performance of MSE trained models (row 2 and 3) on real data (as the bad channel is removed), but worse results on simulated data (as a good channel is removed). This pattern is also observed for the BeamformIt results (row 1). However, the CE trained models still perform poorly on real data. Second, we refine the neural networks only on 1680 real sentences of the official training set (row 6 and 7 of the 6 channel case). WER on real data is improved for DNN model, however, WER on simulated data gets much worse for both models. This shows that the CE training should use data similar to the eval data.

In summary, we found that the weight predicting framework [4] do not consistently outperform BeamformIt on CHiME-4, while it does outperform BeamformIt significantly on the AMI corpus. We hypothesize that this is because the AMI is a far field scenario and the gains of the channels are similar, while CHiME-4 is a near field scenario where the gains of channels could be very different. As the network only uses GCC as input, it is not able to estimate the channel gains proporly.

3.3. Results of Mask Predicting Network

The performance obtained with mask predicting network and MVDR beamforming is shown in Table 3 for 2 channel case and Table 2 for 6 channel case. Let's go through the 6 channel case as the results of 2 channel case will be similar. A conventional MVDR beamforming [29] with TDOA tracking, frequency dependent channel gain estimation, and noise estimation using 0.5s noises prior to the test utterance obtains a WER of 12.0% on the real eval data (row 3). By comparison, the MVDR using masks predicted by IBM-initialized LSTM produces a WER of 12.8% (row 4). By using ASR cost function to

Table 2: Recognition word error rate (WER %) obtained by mask predicting network on the CHiME-4 6-channel track. "Split Mask" specifies whether we estimate speech and noise masks separately. LM: "3" means trigram, "5" means 5-gram, while "R" is RNN LM rescoring.

Row	Settings						Dev		Eval	
	#ch for mask	ASR cost	Split Mask	Pooling	#Pass	LM	Real	Simu	Real	Simu
1	1-channel track					3	12.4	14.8	21.6	22.0
2	Delay-and-sum (BeamformIt)						8.2	9.4	13.6	14.2
3	Traditional MVDR						7.6	6.6	12.0	8.2
4	First channel	No	No	No	1		8.3	7.1	12.8	19.5
5		Yes			1		7.3	6.4	10.9	15.2
6					3		6.4	6.1	9.4	11.1
7			Yes		1		6.5	6.1	10.1	11.9
8					3		6.1	6.0	9.0	9.9
9	Estimate masks for all 6 channels, then pool the masks			max	1		6.6	6.0	10.2	10.0
10				min	1		6.6	6.0	10.3	8.9
11				mean	1		6.4	5.9	9.8	9.2
12				median	1		6.2	6.0	9.5	8.9
13					3		6.1	5.9	8.9	9.6
14					3	5	4.8	4.9	7.4	7.9
15					3	R	4.1	4.3	6.3	6.9

fine tune the LSTM mask predictor, the WER reduces to 10.9% (row 5). The reason for poor performance on simulated eval data is that the first channel of this data set has much lower SNR than other channels and the network predicts the mask from the first channel only. In overall, the results show the effectiveness of using ASR cost function to fine tune mask predictor.

We investigated several approaches to further improve the performance on mask based MVDR. The first is to use multiple passes of mask estimation and beamforming. Specifically, the mask estimation (using enhanced speech) and beamforming can be performed alternately until converge. In row 6, applying the mask estimation and beamforming 3 times is found to reduce WER further to 9.4% (3 passes) from 10.9% (1 pass). The second approach we studied is the splitted mask, i.e. predicting the speech and noise masks independently. Comparing row 7 to row 5, using splitted masks consistently outperforms using unsplitted mask. Lastly, we investigated the use of mask pooling. From row 9 onwards, the masks of all the 6 channels are estimated and pooled. It is observed that median pooling produces the best performance, which agrees with findings in [11]. For the 2 channel case, no pooling is used. We investigated the mask predicting using concatenation of two channels' log power spectra. Comparing row 7 and 8 of Table 3, concatenated input outperforms the single channel input significantly. By combining all the methods together, we obtain the best WER on the real eval data in row 13 in Table 2, with a WER of 8.9%. This represents a 3.1% absolute WER reduction compared to conventional MVDR.

3.4. Results of Maximum Likelihood Weight Estimation

The performance of MLBF on the 6 channel track is shown in Table 4. Row 1 shows that by only estimating 5 TDOAs of channel 2-6 using the MLBF, a WER of 17.1% is obtained, which is significantly lower than 1 channel case (21.6%) shown in row 1 of Table 2. By only using TDOAs, the signals are aligned and added together, similar to unweighted delay-and-sum beamforming. If frequency dependent gains are also estimated and L2 norm is tuned, the WER can be further reduced to 16.1% (row 3). We also tried to use frequency independent gains (row 4), i.e. only uses one global gain for each channel, the same WER of 16.1% WER is obtained. We improve the frequency dependent gain estimation by using frequency independent gains as the initial gains $\mathbf{W}_0$ in equation (4). The L2 regularization ensures that the frequency dependent gains are not too far from the frequent independent gains. Results in row 5 show that this way of initialization and L2 regularization reduce the WER significantly to 14.5%.

Table 3: Recognition word error rate (WER %) obtained by mask predicting network on the CHiME-4 2-channel track.

Row	Settings						Dev		Eval	
	#ch for mask	ASR cost	Split Mask	#Pass	AM	LM	Real	Simu	Real	Simu
1	Not applicable for BeamformIt				ch5	3	10.9	12.4	20.4	19.0
2							11.9	13.1	20.8	20.2
3					chall		10.1	11.7	17.2	18.2
7	First channel	No	No	1			9.8	10.4	16.6	17.0
8	ch 1&2			1			9.2	10.2	15.5	14.9
9	First channel	Yes		1			9.4	10.1	15.7	16.2
10				3			9.1	10.0	15.0	15.0
11			Yes	1			8.9	10.0	15.2	15.3
12				3			8.8	9.9	14.5	14.3
13				3			8.4	9.5	14.4	14.2
14				3		5	7.0	8.1	12.3	12.1
15				3		R	6.1	7.1	10.8	10.7

Table 4: Recognition WER (%) obtained by MLBF on the CHiME-4 6-channel track.

Row	Settings				Eval	
	Parameters	Init.	Gain	#Gau.	Real	Simu
1	TDOA + Gain	No	None	512	17.1	17.6
2		No	Freq. Dependent	512	16.2	14.6
3		No	Freq. Dependent	1024	16.1	14.5
4		No	Freq. Independent	1024	16.1	14.5
5		Row 4	Freq. Dependent	1024	14.5	12.2
6	Real + Imag.	MVDR using mask prediction			9.5	8.9
7		Row 6	-	1024	9.2	8.3

We initialize the real and imaginary parts of the weights with the weights generated by the mask based MVDR (shown in row 6, also row 12 of Table 2). L2 regularization is applied to prevent big deviations of the weights from the initial weights. Results show that the WERs on both simulated and real data are improved moderately. It is worth noting that there is a big gap in performance between row 5 and 7. This could be due to different parameterization of weights and/or the MLBF may easily stuck in a local minimum of cost function.

3.5. Discussions and Future Works

In this paper, we conducted a comparative study of three learning based beamforming methods for far field speech recognition. We found that the MVDR beamformer using LSTM predicted time frequency masks perform the best, while the beamforming filter weight predicting network and MLBF also improve the ASR performance significantly compared to the single channel baseline. In terms of computational cost, the weight predicting network is the most efficient, followed by mask predicting network. Both of these networks are faster than real time. The MLBF is the slowest due to iterative weight optimization at run time.

The better performance of MVDR formulation could be due to that the noise information is important in this task. While the mask based MVDR explicitly makes use of noise estimation, the weight predicting network does not use noise information since only the phase-carrying GCC features are used as input. Although the MLBF has access to the raw noisy data in frequency domain, it does not find good weight solution similar to the MVDR's, possibly due to the local minimum problem of EM algorithm. Hence, the future works could be done to add noise information explicitly to these two types of methods. Another observation is that for near field scenario, it is important to estimate the channel gains as shown in the results of MLBF. The weight predicting network may be improved by explicitly predicting the gains and also use MVDR weights as the supervision during initialization. The MLBF could be integrated with traditional methods. For example, besides maximizing likelihood, one can also maximize the output SNR so more supervision information is used and better solution could be obtained.

4. Acknowledgments

Xiong Xiao and Chenglin Xu are supported by DSO funded project MAISON DSOCL14045.

5. References

[1] Y. Hoshen, R. J. Weiss, and K. W. Wilson, "Speech acoustic modeling from row multichannel waveforms," *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, pp. 4624–4628, 2015.

[2] T. N. Sainath, R. J. Weiss, K. W. Wilson, A. Narayanan, M. Bacchiani, and Andrew, "Speaker location and microphone spacing invariant acoustic modeling from raw multichannel waveforms," *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 30–36, 2015.

[3] T. N. Sainath, R. J. Weiss, K. W. Wilson, A. Narayanan, and M. Bacchiani, "Factored spatial and spectral multichannel raw waveform CLDNNs," *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 2016-May, pp. 5075–5079, 2016.

[4] X. Xiao, S. Watanabe, H. Erdogan, L. Lu, J. Hershey, M. L. Seltzer, G. Chen, Y. Zhang, M. Mandel, and D. Yu, "Deep beamforming networks for multi-channel speech recognition," in *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 2016-May, 2016, pp. 5745–5749.

[5] X. Xiao, S. Watanabe, E. S. Chng, and H. Li, "Beamforming Networks Using Spatial Covariance Features for Far-field Speech Recognition," *accepted by APSIPA ASC*, 2016.

[6] B. Li, T. N. Sainath, R. J. Weiss, K. W. Wilson, and M. Bacchiani, "Neural Network Adaptive Beamforming for Robust Multichannel Speech Recognition," *INTERSPEECH*, 2016.

[7] L. Griffiths and C. Jim, "An alternative approach to linearly constrained adaptive beamforming," *IEEE Transactions on Antennas and Propagation*, vol. 30, no. 1, pp. 27–34, 1982.

[8] J. Capon, "High-resolution frequency-wavenumber spectrum analysis," *Proceedings of the IEEE*, vol. 57, no. 8, pp. 1408–1418, 1969.

[9] E. Warsitz and R. Haeb-Umbach, "Blind acoustic beamforming based on generalized eigenvalue decomposition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 5, pp. 1529–1539, 2007.

[10] A. Krueger, E. Warsitz, and R. Haeb-Umbach, "Speech enhancement with a GSC-like structure employing eigenvector-based transfer function ratios estimation," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 1, pp. 206–219, 2011.

[11] J. Heymann, L. Drude, A. Chinaev, and R. Haeb-Umbach, "BLSTM supported GEV beamformer front-end for the 3RD CHiME challenge," *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 444–451, 2015.

[12] J. Heymann, L. Drude, and R. Haeb-Umbach, "Neural network based spectral mask estimation for acoustic beamforming," *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 2016-May, pp. 196–200, 2016.

[13] H. Erdogan, J. Hershey, S. Watanabe, and M. Mandel, "Improved MVDR Beamforming using Single-channel Mask Prediction Networks," *INTERSPEECH*, 2016. [Online]. Available: http://www.merl.com/publications/docs/TR2016-072.pdf

[14] X. Xiao, S. Zhao, D. L. Jones, E. S. Chng, and H. Li, "On Time-Frequency Mask Estimation for MVDR Beamforming With Application in Robust Speech Recognition," *submitted to ICASSP 2017*.

[15] M. L. Seltzer, B. Raj, and R. M. Stern, "Likelihood-maximizing beamforming for robust hands-free speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 5, pp. 489–498, 2004.

[16] E. Vincent, S. Watanabe, A. A. Nugraha, J. Barker, and R. Marxer, "An analysis of environment, microphone and data simulation mismatches in robust speech recognition," *to appear in Computer Speech and Language*.

[17] M. L. Seltzer, "Microphone Array Processing for Robust Speech Recognition," *PhD thesis, CMU*, no. July, p. 163, 2003.

[18] C. Knapp and G. C. Carter, "The generalized correlation method for estimation of time delay," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 24, no. 4, pp. 320–327, 1976.

[19] X. Xiao, S. Zhao, X. Zhong, D. L. Jones, E. S. Chng, and H. Li, "A learning-based approach to direction of arrival estimation in noisy and reverberant environments," *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 2015-Augus, pp. 2814–2818, 2015.

[20] M. Souden, J. Benesty, and S. Affes, "On optimal frequency-domain multichannel linear filtering for noise reduction," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 18, no. 2, pp. 260–276, 2010.

[21] D. H. H. Nguyen, X. Xiao, E. S. Chng, and H. Li, "Feature Adaptation Using Linear Spectro-Temporal Transform for Robust Speech Recognition," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 24, no. 6, pp. 1006–1019, 2016.

[22] Z. Zhang, X. Xiao, L. Wang, J. Dang, M. Iwahashi, E. S. Chng, and H. Li, "Multi-channel feature adaptation for robust speech recognition," *ISCSLP*, 2016.

[23] M. J. F. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech and Language*, vol. 12, no. 2, pp. 75–98, 1998.

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

[25] X. Xiao, "SignalGraph: a deep learning toolkit for signal processing," 2016. [Online]. Available: https://github.com/singaxiong/SignalGraph

[26] D. Dempster, A.P. and Laird, N.M. and Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society. Series B (Methodological)*, vol. 39, no. 1, pp. 1–38, 1977.

[27] D. C. Liu and J. Nocedal, "On the limited memory BFGS method for large scale optimization," *Mathematical Programming*, vol. 45, no. 1-3, pp. 503–528, 1989.

[28] X. Anguera, C. Wooters, and J. Hernando, "Acoustic beamforming for speaker diarization of meetings," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 15, no. 7, pp. 2011–2022, 2007.

[29] S. Zhao, X. Xiao, Z. Zhang, T. N. T. Nguyen, X. Zhong, B. Ren, L. Wang, D. L. Jones, E. S. Chng, and H. Li, "Robust speech recognition using beamforming with adaptive microphone gains and multichannel noise reduction," *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 460–467, 2016.

Evolution Strategy Based Neural Network Optimization and LSTM Language Model for Robust Speech Recognition

Tomohiro Tanaka[1], *Takahiro Shinozaki*[1],
Shinji Watanabe[2], *Takaaki Hori*[2]

[1]Tokyo Institute of Technology, Japan
[2]Mitsubishi Electric Research Laboratories, USA

Abstract

This paper reports our system for the 1-channel track task in the 4th CHiME challenge (CHiME4). A bottle-neck in developing neural network based systems is the tuning of meta-parameters. We automate it by using Covariance Matrix Adaptation Evolution Strategy (CMA-ES) so that high performance system is obtained without relying on human experts. We run two evolution experiments for the DNN acoustic model used in the official baseline system. One uses development set word error rate (WER) after the cross-entropy (CE) based training as the objective function for the evolution, and the other uses the WER after the sequential discriminative training. Additionally, we run an evolution experiment for a Long Short-Term Memory recurrent neural network based language model (LSTM-LM), replacing the original recurrent neural network language model (RNN-LM) used in the baseline system for N-best rescoring. All of these evolution experiments resulted in reduced WERs. To produce the final results, we augmented training data by pooling speech data from all the 6 channels and imported the optimized meta-parameter settings without modification. For the real test data, reduced WER of 17.40% and 16.58% were obtained compared to the baseline WER of 22.75% when the RNN and LSTM-LMs were used, respectively.

1. Background

Neural network based techniques have shown great performance in automatic speech recognition (ASR) tasks [1, 2]. To use neural network, various meta-parameters must be specified including model topology (e.g., the numbers of layers and hidden units), training configuration (e.g., the learning rate and the maximum number of iterations) and system organization (e.g., the choice of features). Properly tuning these meta-parameters is essential for building high performance systems. Usually, the tuning is manually performed. However, it requires expert knowledge and laborious effort. Thus there is a demand to automate the tuning process using computers.

We have previously investigated several automatic meta-parameter optimization frameworks for neural network acoustic models [3, 4, 5]. In the experiments, covariance matrix adaptation-evolution strategy (CMA-ES) [6, 7, 8] showed superior performance than Genetic Algorithm (GA) and Bayesian optimization [9, 10] giving better model with smaller or similar number of system evaluations. Further, we have applied CMA-ES to optimize neural network based language models and have shown that it works well to improve system performance [11]. Here, we apply it to neural network based acoustic and language models in the CHiME4 1-channel track task.

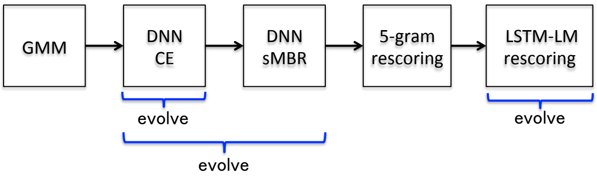


Figure 1: *Recognition system used for evolution of DNN-AM and LSTM-LM.*

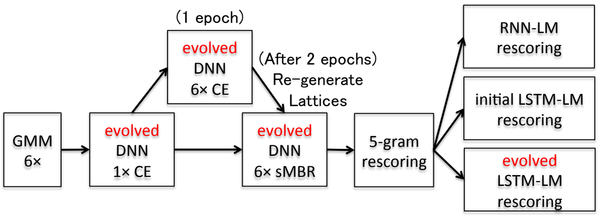


Figure 2: *Recognition system used with augmented acoustic model training data.*

2. Contributions

2.1. CMA-ES based tuning of neural networks

CMA-ES is a population based algorithm for black box optimization that has demonstrated superior performance in several benchmarking tasks. Similar to the GA, it encodes possible solutions as genes. It assumes that the value of an objective function $f(\boldsymbol{x})$ is available, while the functional form of f might be too complex to perform analytical optimization. More specifically, CMA-ES estimates parameters $\boldsymbol{\theta}$ of a Gaussian distribution for a gene $\boldsymbol{x}$ such that the distribution is concentrated in a region with high values of $f(\boldsymbol{x})$ as shown in Eq. (1).

$$\hat{\boldsymbol{x}} \sim \mathcal{N}(\boldsymbol{x}|\hat{\boldsymbol{\theta}}) \text{ s.t. } \hat{\boldsymbol{\theta}} = \arg\max_{\boldsymbol{\theta}} \underbrace{\int f(\boldsymbol{x})\mathcal{N}(\boldsymbol{x}|\boldsymbol{\theta})d\boldsymbol{x}}_{\triangleq \mathbb{E}[f(\boldsymbol{x})|\boldsymbol{\theta}]} . \tag{1}$$

The estimation of $\boldsymbol{\theta}$ is based on an iterative method, where in each iteration, a set of genes $\{\boldsymbol{x}\}$ is sampled from the Gaussian, their performance $f(\boldsymbol{x})$ is evaluated, and $\boldsymbol{\theta}$ is updated based on the results. In other words, while GA represents a distribution of genes in a generation by the samples themselves, CMA-ES uses a Gaussian distribution. In our case, a gene represents a set of meta-parameters of a neural network to optimize.

Table 1: WER after CE based DNN-AM training.

System	Dev		Test	
	real	simu	real	simu
Baseline	16.45	17.81	29.67	26.20
Evolved	15.40	16.88	29.16	25.28

Table 2: WER after sequential DNN-AM training.

System	Dev		Test	
	real	simu	real	simu
Baseline	14.90	15.70	27.24	24.34
Evolved	13.82	15.49	25.67	22.95

Table 3: WER after RNN/LSTM-LM based rescoring.

System	Dev		Test	
	real	simu	real	simu
Baseline RNN-LM	11.60	12.92	22.75	21.07
+ Evolved DNN-AM	10.98	12.74	21.29	19.74
Evolved LSTM-LM	10.20	12.23	21.09	19.66
+ Evolved DNN-AM	10.00	11.45	20.54	18.85

2.2. LSTM based language model

Neural network based language models have shown to be very effective for improving speech recognition performance [12]. In the CHiME4 baseline system [13], recurrent neural network language model (RNN-LM) [14] is used for final rescoring. The parameters of a RNN are trained using back-propagation through time (BPTT) so that the context dependency is modeled. However, RNNs cannot effectively use long context information due to the vanishing gradient problem [15]. To address the problem, Long Short-Term Memory RNN that utilizes LSTM blocks has been proposed [16]. A LSTM block has a memory cell and three gates (input, forget and output) to control the value stored in the memory cell. By replacing the unit in recurrent hidden layer of a RNN language model with the LSTM block, a LSTM RNN language model (LSTM-LM) [17] is obtained. We replace RNN-LM with LSTM-LM, which is known to perform better in various tasks [18].

3. Experimental evaluation

3.1. Evolution using single channel training data

Using the single channel (channel 5) multicondition training data, we ran two evolution experiments for the DNN acoustic model (DNN-AM) used in the official baseline system based on CMA-ES. One used development set WER after the CE based training as the objective function for the evolution, and the other used the WER after the sequential discriminative training based on state-level Minimum Bayes Risk (sMBR) criterion. Additionally, we ran an evolution experiment for a LSTM-LM replacing the original RNN-LM using WER after N-best rescoring as the objective for the evolution, where 100-best was generated from the decoding result using the 5-gram language model with Kneser-Ney smoothing [19]. These three evolutions were performed independently. Figure 1 shows where the evolutions were performed in the recognition system structure.

For the DNN-AM, 11 meta-parameters were optimized, which were the same as our previous work [5]. These included the number of hidden layers, the number of units per a hidden layer, the initial learning rate, and so on. The population size (e.g. the number of sampled genes from the Gaussian at each generation) was 36. The numbers of iterations (e.g. generations) were 6 and 4 for the two evolutions, respectively. Table 1 shows the results when WER after CE based DNN-AM training was used as the objective, and Table 2 shows the results when WER after the sequential training was used. In both cases, lower WERs were obtained by the evolution based automatic tuning. The sequential training gave some gain compared to the CE based training, and evolution based optimization gave further gain. The best performing DNN chosen by the development set WER had 9 hidden layers and 2461 units per a layer.

Table 4: WER after RNN/LSTM-LM based rescoring. DNN-AM was trained using the augmented training data.

System	Dev		Test	
	real	simu	real	simu
RNN-LM	9.09	10.86	17.40	16.49
Initial LSTM-LM	9.02	10.82	17.52	16.62
Evolved LSTM-LM	8.06	10.15	16.58	15.67

Table 5: Detailed WERs after RNN-LM rescoring. DNN-AM was trained using the augmented training data.

Env.	Dev		Test	
	real	simu	real	simu
BUS	12.27	9.63	26.51	12.27
CAF	9.23	14.69	19.18	19.11
PED	5.58	8.30	13.62	16.51
STR	9.28	10.83	10.29	18.06
AVG.	9.09	10.86	17.40	16.49

For the LSTM-LM, 19 meta-parameters were optimized including the vocabulary size, the number of layers, the number of units per a layer, the initial learning rate and the dropout ratio [20]. The maximum number of hidden layers were set to six and they were used depending on the number of hidden layer. The population size was 30 and the number of generations was 4. All LSTM-LMs were trained using the Chainer toolkit [1] [21]. The population sizes were decided based on our previous experiments and available computer resources for this experiment. Table 3 shows the results. By using LSTM-LM, lower WERs were obtained than the baseline RNN-LM. When the DNN-AM evolved by using the WER after the sequential training was combined, further reduction in WERs was obtained. The vocabulary size of the tuned LSTM-LM was 8112 and the number of hidden layers was 2. The list of the meta-parameters and their initial and optimized values are shown in appendix.

3.2. Single channel system with augmented training data

In the official single channel CHiME4 baseline system, only 5th channel data is used for training. We augmented the training data by 6 times by pooling speech data from all the 6 channels of the official data for further improvement. For the DNN-AM and LSTM-LM, the previously optimized meta-parameters by the evolutions using the original (1x) training data were imported and used as it is. To save the time for experiment, part of the CE based DNN training used the 1x data, and lattice regeneration was performed at slightly different timing from the baseline system as shown in Figure 2. The sequential training for DNN-AM was performed for 6 epochs. Table 4 shows

[1] http://chainer.org/

Table 6: Detailed WERs after LSTM-LM rescoring. LSTM-LM was trained importing the evolved meta-parameters. DNN-AM was trained using the augmented training data.

Env.	Dev		Test	
	real	simu	real	simu
BUS	10.93	9.22	26.00	11.39
CAF	8.29	13.86	18.58	18.77
PED	4.69	7.80	12.05	15.50
STR	8.32	9.73	9.68	17.02
AVG.	8.06	10.15	16.58	15.67

summary of WERs after the RNN-LM based rescoring and the LSTM-LM based rescoring using the initial and the evolved meta-parameters. As can be seen, the lowest WERs were obtained when the evolved LSTM-LM was used. Tables 5 and 6 show the details of the WERs when the RNN and the evolved LSTM-LM were used. By using the LSTM-LM, the averaged real environment WER for the development and evaluation sets were 8.06% and 16.58%, respectively.

4. Acknowledgments

The work of T. Tanaka and T. Shinozaki was supported by JSPS KAKENHI Grant Number 26280055.

5. References

[1] H. Schwenk and J.-L. Gauvain, “Connectionist language modeling for large vocabulary continuous speech recognition.” in *Proc. ICASSP*, 2002, pp. 765–768.

[2] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath, and B. Kingsbury, “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.

[3] S. Watanabe and J. Le Roux, “Black box optimization for automatic speech recognition,” in *Proc. ICASSP*. IEEE, 2014, pp. 3256–3260.

[4] T. Shinozaki and S. Watanabe, “Structure discovery of deep neural network based on evolutionary algorithms,” in *Proc. ICASSP*, 2015, pp. 4979–4983.

[5] T. Moriya, T. Tanaka, T. Shinozaki, S. Watanabe, and K. Duh, “Automation of system building for state-of-the-art large vocabulary speech recognition using evolution strategy,” in *Proc. ASRU*, 2015, pp. 610–616.

[6] N. Hansen, S. D. Müller, and P. Koumoutsakos, “Reducing the time complexity of the derandomized evolution strategy with covariance matrix adaptation (CMA-ES),” *Evolutionary Computation*, vol. 11, no. 1, pp. 1–18, 2003.

[7] Y. Akimoto, Y. Nagata, I. Ono, and S. Kobayashi, “Bidirectional relation between CMA evolution strategies and natural evolution strategies,” in *Proc. Parallel Problem Solving from Nature (PPSN)*, 2010, pp. 154–163.

[8] D. Wierstra, T. Schaul, T. Glasmachers, Y. Sun, J. Peters, and J. Schmidhuber, “Natural evolution strategies,” *J. Mach. Learn. Res.*, vol. 15, no. 1, pp. 949–980, 2014.

[9] J. Mockus, “On Bayesian methods for seeking the extremum,” in *Proceedings of the IFIP Technical Conference*. London, UK, UK: Springer-Verlag, 1974, pp. 400–404.

[10] J. Snoek, H. Larochelle, and R. P. Adams, “Practical Bayesian optimization of machine learning algorithms,” in *Advances in Neural Information Processing Systems 25*, 2012.

[11] T. Tanaka, T.Moriya, T. Shinozaki, S. Watanabe, T. Hori, and K. Duh, “Automated structure discovery and parameter tuning of neural network language model based on evolution strategy,” in *IEEE Workshop on Spoken Language Technology (SLT)*, 2016, (accepted).

[12] Y. Bengio, R. Ducharme, P. Vincent, and C. Jauvin, “A neural probabilistic language model,” *Journal of Machine Learning Reseach*, vol. 3, pp. 1137–1155, 2003.

[13] E. Vincent, S. Watanabe, A. A. Nugraha, J. Barker, and R. Marxer, “An analysis of environment, microphone and data simulation mismatches in robust speech recognition,” *Computer Speech and Language*, 2016, (submitted).

[14] T. Mikolov, M. Karafiát, L. Burget, J. Cernocký, and S. Khudanpur, “Recurrent neural network based language model,” in *Proc. INTERSPEECH*, 2010, pp. 1045–1048.

[15] Y. Bengio, P. Simard, and P. Frasconi, “Learning long-term dependencies with gradient descent is difficult,” *IEEE Transactions on Neural Networks*, vol. 5, no. 2, pp. 157–166, 1994.

[16] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] M. Sundermeyer, R. Schlüter, and H. Ney, “LSTM neural networks for language modeling,” in *Proc. INTERSPEECH*, 2012, pp. 194–197.

[18] T. Hori, C. Hori, S. Watanabe, and J. R. Hershey, “Minimum word error training of long short-term memory recurrent neural network language models for speech recognition,” in *Proc. ICASSP*, 2016, pp. 5990–5994.

[19] R. Kneser and H. Ney, “Improved backing-off for m-gram language modeling,” *Proc. ICASSP*, pp. 181–184, 1995.

[20] G. E. Dahl, T. N. Sainath, and G. E. Hinton, “Improving deep neural networks for LVCSR using rectified linear units and dropout,” in *Proc. ICASSP*. IEEE, 2013, pp. 8609–8613.

[21] S. Tokui, K. Oono, S. Hido, and J. Clayton, “Chainer: a next-generation open source framework for deep learning,” in *Neural Information Processing Systems (NIPS)*, 2015.

A. Appendix: Meta-parameters

Table 7 shows the initial and optimized meta-parameters for the DNN acoustic model using the development set WER after the sequential discriminative training as the objective for evolution. Similarly, table 8 lists the initial and optimized meta-parameters for the LSTM language model. For each table, the best gene of the meta-parameters was selected from the pool of all the generations based on the WER of the development set.

Table 7: Meta-parameters for DNN-AM.

Description	Initial value	Best value
feature type($\{$MFCC,FBANK,PLP$\}$)	FBANK	FBANK
splice (segment length for DNN	5	7
# of hidden layers	6	9
# of hidden layer units	2048	2461
initial parameters in 1st RBM	$1.00E-1$	$1.15E-1$
initial parameters in other RBMs	$1.00E-1$	$5.04E-2$
RBM learning rate	$4.00E-1$	$5.64E-1$
lower RBM learning rate	$1.00E-2$	$1.26E-2$
RBM Lasso regularization	$2.00E-4$	$1.61E-4$
learning rate for fine tuning	$8.00E-3$	$3.38E-4$
momentum for fine tuning	$1.00E-5$	$9.33E-6$

Table 8: Meta-parameters for LSTM-LM.

Description	Initial value	Best value
vocabulary size	5000	8112
# of hidden layers	2	2
# of projection layer units	300	399
# of 1st layer units	300	671
# of 2nd layer units	300	438
NNLM weight	0.50	0.52
acoustic weight	14.00	21.56
minibatch size	32	35
dropout ratio	0.50	0.44
initial learn rate	1	0.90
learn decay	0.50	0.48
learn decay epochs	6	7
momentum	$1.00E-10$	$1.03E-10$
gradient clipping	5.00	6.23
initial forget gate bias	1.00	1.18

The USTC-iFlytek System for CHiME-4 Challenge

Jun Du[1], Yan-Hui Tu[1], Lei Sun[1], Feng Ma[2],
Hai-Kun Wang[2], Jia Pan[2], Cong Liu[2], Jing-Dong Chen[3], Chin-Hui Lee[4]

[1]University of Science and Technology of China, Hefei, Anhui, P. R. China
jundu@ustc.edu.cn, {tuyanhui,sunlei17}@mail.ustc.edu.cn
[2]iFlytek Research, iFlytek Co., Ltd., Hefei, Anhui, P. R. China
{fengma,hkwang,jiapan,congliu2}@iflytek.com
[3]Northwestern Polytechnical University, Xian, Shaanxi, P. R. China
jingdongchen@ieee.org
[4]Georgia Institute of Technology, Atlanta, Georgia, USA
chl@ece.gatech.edu

Abstract

The submitted system for CHiME-4 this year includes significant improvements over the previous one for CHiME-3, including the front-end design, training data augmentation via different versions of the official training data, acoustic model fusion, and language model fusion. The final average WERs of the real test set are 2.24%, 3.91%, 9.15% for 6-channel, 2-channel, and 1-channel, respectively.

1. Background

For CHiME-4 [1], we participate all the tracks including 1 ch, 2 ch, and 6 ch tasks. In comparison to CHiME-3 challenge [2, 3], our new progress mainly includes: 1) a closed-loop optimization for beamforming by leveraging the information of deep neural network (DNN) based single-channel speech enhancement and the recognition results; 2) diversified training data using the noisy data of each channel, the multiple beamformers' outputs data of 6 channels and 2 channels; 3) the acoustic model upgrade via the deep convolutional neural networks (DCNNs) [4, 5]; 4) the long short-term memory (LSTM) based language modeling [6, 7]. In the next section, we will elaborate these contributions.

2. Contributions

The overall system flowchart is given in Fig. 1, where a unified framework for all three tasks, namely 1/2/6-channel cases, is designed. In the training stage, both the acoustic models with multiple front-ends and language models are built. In the recognition, multiple acoustic models are fused at the state-level first and then first-pass decoding is performed with the HMM and 3-gram to generate the lattice as the hypotheses , which are served for the second-pass decoding with a LSTM-based LM. The details can refer to the following subsections.

2.1. Beamforming

The beamforming approach showed in Fig. 2 is similar to the work in [8], namely the generalized sidelobe canceller (GSC) with a post-filtering. First, the time-frequency (T-F) masking is calculated via the complex Gaussian mixture model (CGMM) [9] to estimate the covariance matrices of noise and noisy speech. The relative transfer function is implemented by the eigenvector-based estimation in [10]. To further improve the estimation of the time-frequency masking, both the VAD information from the segmentation results of recognizer based on the beamformed speech and the ideal ratio mask (IRM) estimated using a DNN are used for a second-pass beamforming. The input of IRM-DNN is the log-power spectra (LPS) of the beamformed speech while the output is the masking values of T-F units calculated between the noisy speech of the channel 5 and the underlying clean speech. Obviously, the VAD and IRM information are based upon the beamforming results, which forms a feedback loop optimization [11] among them with multiple iterations. Experiments show that this new framework could significantly improve the recognition accuracy, yielding a remarkable gain over the best beamforming approach of CHiME-3 [2].

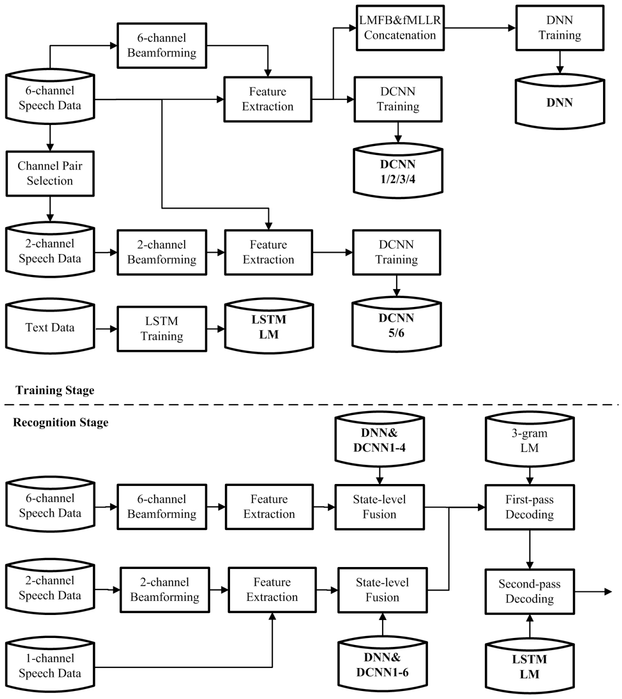


Figure 1: System overview.

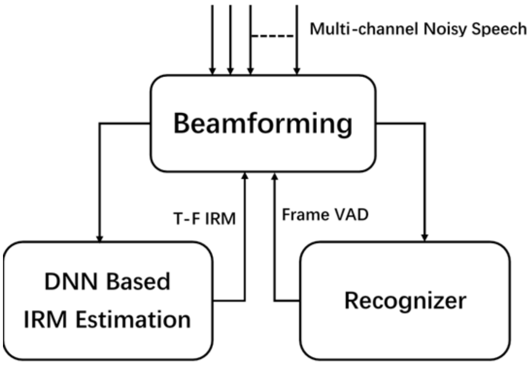


Figure 2: Beamforming.

2.2. Training data augmentation

The training data augmentation is a straightforward way to enlarge the data coverage, especially for 3 tasks with different settings of channel number in CHiME-4. Three data types are employed. First, the noisy speech of 5 channels (excluding the channel 2 with the most degraded speech) are used to simulate the 1-channel testing case. Then, the enhanced version using the beamforming approach applied to all 6 channels matches the 6-channel testing cases. Finally, we randomly select some channel pairs from 5 channels and the beamformed results of the corresponding channel pairs can correspond to the 2-channel testing cases. As illustrated in Fig. 1, both the noisy speech and 6-channel beamformed data are adopted to train the models (DNN and DCNN1/2/3/4) for all testing. Meanwhile, the noisy speech plus 2-channel beamformed data are combined to learn two other models (DCNN5/6) for 2-channel and 1-channel testing.

2.3. Acoustic models

We train mainly 2 types of neural networks. One is the conventional DNN and the other is DCNN. For 6-channel system, 5 models are built and fused via the state-level posterior average [3], including one DNN and 4 DCNNs (DCNN1/2/3/4). The DNN system concatenates the log mel-filterbank (LMFB) and fMLLR features. 4 DCNNs consist of LMFB-based one, fMLLR-based one and two others with different parameter settings. For 2-channel and 1-channel systems, two additional DCNNs (DCNN5/6) are used, namely 7 models in total. The DCNN system shows the strong complementarity when fused with the DNN system.

2.4. Language models

Besides the 5-gram and RNNLM provided officially, we also train an LSTM-based LM to further improve the recognition accuracy. According to our experiments, the LSTM-based LM alone could yield a relative WER reduction of more than 30% over the 5-gram+RNNLM based system.

3. Experimental evaluation

3.1. Beamforming

The Word Error Rates (WERs) on the evaluation data of the official and our proposed beamformers for the 2 ch and 6 ch track have been showed in Table 1. We adopted the DNN based official baseline system, 11 frames of 40-dimension fMLLR features. The DNN architecture is 440-2048*7-1987, namely 40*11 dimension for fMLLR input features, 7 hidden layers with 2048 nodes for each, and 1968 nodes for the output layers as our ASR model. The IRM-DNN is trained using 7 frames of 257-dimension LPS features of CH5. The IRM-DNN architecture is 1799-2048*3-257, namely 257*7 dimension for LPS input features, 3 hidden layers with 2048 nodes for each, and 257 nodes for the output T-F IRM. The significant reduction of WERs on the evaluation data for both the development and test sets can be found in Table 1, and our beamformer is more effective for more adverse environments and more microphones than official beamformer.

Table 1: WERs obtained with the proposed and official beamformers on the evaluation data for 2 ch and 6 ch tracks using the official DNN acoustic model.

Track	System	Dev		Test	
		real	simu	real	simu
2ch	Official	8.50	9.92	17.07	15.98
	Proposed	6.20	8.12	10.86	11.69
6ch	Official	6.25	7.15	11.82	11.43
	Proposed	4.18	4.17	6.13	5.23

Table 2: WERs between the baseline and data augmentation based systems on the evaluation data for 2 ch and 6 ch tracks.

Track	System	Dev		Test	
		real	simu	real	simu
2ch	Official	6.20	8.12	10.86	11.69
	Retrained	4.68	6.26	7.14	9.39
6ch	Official	4.18	4.17	6.13	5.23
	Retrained	3.24	3.33	4.33	4.21

3.2. Training data augmentation

The Word Error Rates (WERs) on the evaluation data of the official baseline and retrained by data augmentation DNN systems for the 1 ch, 2 ch and 6 ch tracks have been showed in Table 2. As for the retrained DNN system, 42-dimensional LMFB features and 40-dimensional fMLLR features with their first-order and second-order derivatives are used. The 20-dimensional i-vector features [3] are concatenated. The DNN architecture is 2234-2048*7-1965, namely (42+40)*3*9+20 dimension for LMFB+fMLLR+ivector combined input features, 7 hidden layers with 2048 nodes for each, and 1965 nodes for the output layer. The training data contains 1,3,4,5,6 channels data and 4 kinds of beamformered data, totally 78642 utterances(8738*9), and the beamformered data by our proposed method is used as our test set. Approximately 20% WERs reduction can be found between the official and our proposed systems in the all test sets.

3.3. Acoustic models

The Word Error Rates (WERs) on the real evaluation data of the different acoustic models for the 1 ch, 2 ch and 6 ch tracks have been showed in Table 3. The main difference of our DCNNs and conventional CNNs is the number and the size of the filters. The multi-layer small convolution kernels (3x3 and 3x5) are used, and the total number of convolutional layers is 25. And the learning rate is set to 0.002, and the batch size is 2048. Batch normalization is also used to speed up the training. In the Table 3, we can find that the performance of DCNNs is sig-

Table 3: WERs with the different acoustic models on the real evaluation data for 1 ch, 2 ch and 6 ch tracks.

Track	Set	System					
		DNN	DCNN1	DCNN2	DCNN3	DCNN4	Ensemble
1ch	Dev	8.29	7.70	7.71	9.87	9.86	6.10
	Test	14.58	15.47	14.72	17.05	17.45	11.12
2ch	Dev	4.68	4.05	4.13	5.24	5.43	3.55
	Test	7.14	6.87	6.94	8.34	8.36	5.40
6ch	Dev	3.24	2.88	2.99	3.37	3.50	2.61
	Test	4.33	3.87	4.09	4.67	4.90	3.22

Table 4: Average WER (%) for the tested systems.

Track	System	Dev		Test	
		real	simu	real	simu
1ch	Official LM	6.10	8.24	11.15	13.62
	LSTM LM	4.55	6.61	9.15	11.81
2ch	Official LM	3.56	4.89	5.41	7.30
	LSTM LM	2.33	3.46	3.91	5.74
6ch	Official LM	2.55	2.61	3.24	3.06
	LSTM LM	1.69	1.78	2.24	2.12

Table 5: WER (%) per environment for the best system.

Track	Envir.	Dev		Test	
		real	simu	real	simu
1ch	BUS	5.84	4.90	14.10	7.58
	CAF	5.09	9.84	9.64	14.98
	PED	2.66	4.84	6.89	11.58
	STR	4.63	6.86	5.98	13.09
2ch	BUS	2.74	2.83	5.16	3.83
	CAF	2.18	4.29	3.83	5.66
	PED	1.73	2.94	3.18	6.14
	STR	2.65	3.79	3.49	7.32
6ch	BUS	2.05	1.64	2.65	1.36
	CAF	1.50	1.99	2.09	1.87
	PED	1.50	1.55	1.74	2.35
	STR	1.71	1.93	2.48	2.91

nificantly better than DNN, and it can bring approximately 20% WERs reduction comparing to DNN on the real test set. Finally, the model ensemble is used by the state posterior average of single system output, it also can bring about 20% WERs reduction.

3.4. Language models

The Word Error Rates (WERs) on the evaluation data of the official and our language models for the 1 ch, 2 ch and 6 ch tracks have been showed in Table 4. The forward and backward LSTM models are trained for the combination of language models. We can find that the performance of LSTM-LM is more effective when the front-end and acoustic models are better in Table 4. Finally, Table 5 presents the results per environment for our best system, and we can find the improvement is significantly comparing to baseline system.

4. Acknowledgments

The team would like to thank other colleagues, Zi-Rui Wang, Tian Gao, Xiao Bao, Ye-Bo Bao, Di-Yuan Liu, and Shi-Liang Zhang for their contributions of building the final systems.

5. References

[1] E. Vincent, S. Watanabe, A. A. Nugraha, J. Barker, and R. Marxer, "An analysis of environment, microphone and data simulation mismatches in robust speech recognition," *Computer Speech and Language*, 2016.

[2] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third 'chime' speech separation and recognition challenge: Dataset, task and baselines," in *IEEE ASRU*, 2015.

[3] J. Du, Q. Wang, Y.-H. Tu, X. Bao, L.-R. Dai, and C.-H. Lee, "An information fusion approach to recognizing microphone array speech in the chime-3 challenge based on a deep learning framework," in *IEEE ASRU*, 2015, pp. 430–435.

[4] D. Yu, W. Xiong, J. Droppo, A. Stolcke, G. Ye, J. Li, and G. Zweig, "Deep convolutional neural networks with layer-wise context expansion and attention," in *INTERSPEECH*, 2016.

[5] T. Sercu, C. Puhrsch, B. Kingsbury, and Y. LeCun, "Very deep multilingual convolutional neural networks for lvcsr," in *ICASSP*, 2016.

[6] M. Sundermeyer, R. Schluter, and H. Ney, "Lstm neural networks for language modeling," in *INTERSPEECH*, 2012.

[7] R. Jozefowicz, O. Vinyals, M. Schuster, N. Shazeer, and Y. Wu, "Exploring the limits of language modeling," in *arXiv:1602.02410v2*, 2016.

[8] S. Gannot and I. Cohen, "Speech enhancement based on the general transfer function gsc and postfiltering," *IEEE Transactions on Speech Audio Processing*, vol. 12, no. 6, pp. 561–571, 2004.

[9] T. Higuchi, N. Ito, T. Yoshioka, and T. Nakatani, "Robust mvdr beamforming using time-frequency masks for online/offline asr in noise," in *ICASSP*, 2016.

[10] A. Krueger, E. Warsitz, and R. Haeb-Umbach, "Speech enhancement with a gsc-like structure employing eigenvector-based transfer function ratios estimation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 1, pp. 206–219, 2011.

[11] A. A. Nugraha, A. Liutkus, and E. Vincent, "Multichannel audio source separation with deep neural networks," *IEEE Transactions on Speech Audio Processing*, vol. 24, no. 9, pp. 1652–1664, 2016.

The RWTH/UPB/FORTH System Combination for the 4th CHiME Challenge Evaluation

Tobias Menne[1], *Jahn Heymann*[2], *Anastasios Alexandridis*[3,4], *Kazuki Irie*[1], *Albert Zeyer*[1], *Markus Kitza*[1], *Pavel Golik*[1], *Ilia Kulikov*[1], *Lukas Drude*[2], *Ralf Schlüter*[1], *Hermann Ney*[1], *Reinhold Haeb-Umbach*[2], *Athanasios Mouchtaris*[3,4]

[1]Human Language Technology and Pattern Recognition,
Computer Science Department, RWTH Aachen University, Aachen, Germany
[2]Paderborn University, Department of Communications Engineering, Paderborn, Germany
[3]FORTH-ICS, Signal Processing Laboratory, Heraklion, Crete, Greece, GR-70013
[4]University of Crete, Department of Computer Science, Heraklion, Crete, Greece, GR-70013
[1]<surname>@cs.rwth-aachen.de, [2]<surname>@nt.uni-paderborn.de,
[3],[4]{analexan,mouchtar}@ics.forth.gr

Abstract

This paper describes automatic speech recognition (ASR) systems developed jointly by RWTH, UPB and FORTH for the 1ch, 2ch and 6ch track of the 4th CHiME Challenge. In the 2ch and 6ch tracks the final system output is obtained by a Confusion Network Combination (CNC) of multiple systems. The Acoustic Model (AM) is a deep neural network based on Bidirectional Long Short-Term Memory (BLSTM) units. The systems differ by front ends and training sets used for the acoustic training. The model for the 1ch track is trained without any preprocessing. For each front end we trained and evaluated individual acoustic models. We compare the ASR performance of different beamforming approaches: a conventional superdirective beamformer [1] and an MVDR beamformer as in [2], where the steering vector is estimated based on [3]. Furthermore we evaluated a BLSTM supported Generalized Eigenvalue beamformer using NN-GEV [4]. The back end is implemented using RWTH's open-source toolkits RASR [5], RETURNN [6] and rwthlm [7]. We rescore lattices with a Long Short-Term Memory (LSTM) based language model. The overall best results are obtained by a system combination that includes the lattices from the system of UPB's submission [8]. Our final submission scored second in each of the three tracks of the 4th CHiME Challenge.

1. Background

This paper describes ASR systems for the 1ch, 2ch and 6ch tracks of the 4th CHiME Challenge. In contrast to the provided baseline system [9] the back end has been replaced completely and is described in Section 2.2. Furthermore we developed additional systems using different front ends. The front ends are described in Section 2.1. All experimental results presented in this work (Sections 3 and 4) are obtained with the official training set following the rules of the CHiME challenge.

2. Contributions

2.1. Front ends

In addition to the baseline (BL) front end we developed three other front ends that utilize different beamformers. The final enhanced signal at the output of each beamformer is given by:

$$\boldsymbol{Z}(k,l) = \boldsymbol{w}(k,l)^H \boldsymbol{X}(k,l) \tag{1}$$

where k, l denote the frequency index and time-frame, respectively, $\boldsymbol{w}(k,l)$ is the $M \times 1$ vector of beamformer filter coefficients for a given front end, $\boldsymbol{X}(k,l)$ is the $M \times 1$ vector of microphone array signals in the Short-time Fourier Transform (STFT) domain, and M denotes the number of microphones.

We also improved the microphone failure detection mechanism of [2], so as to better identify corrupted microphones. The enhanced microphone failure detection was used in the front ends described in Sections 2.1.2 & 2.1.3.

2.1.1. Microphone failure detection

Our microphone failure detection mechanism is based on measuring the consistency of the energies (calculated in each time frame) between the microphone signals. To do that, we construct M time-series $e_m(l), m = 1, \dots, M$, each one containing the energy of the signal for $l = 1, \dots, L$ frames, where L denotes the total number of frames in the utterance. Then, for each microphone m, the average correlation coefficient r_m^{AV} between $e_m(l)$ and $e_n(l)$ for $n \neq m$ is calculated. A microphone is considered to have failed if r_m^{AV} is less than a threshold δ, which was set empirically to 0.8. These microphones, in addition to the microphones which are considered to have failed by the system of [2], are excluded from further processing.

2.1.2. MVDR beamformer with steering vector estimation

This front end (MV) utilizes a minimum variance distortionless response (MVDR) beamformer with diagonal loading, similar to the one in [2]. The filter coefficients are calculated as:

$$\boldsymbol{w}_{\mathrm{MVDR}}(k,l) = \frac{\left[\boldsymbol{R}_{\mathrm{n}}(k) + \epsilon\, \mathrm{diag}(|\boldsymbol{X}(k,l)|^2)\right]^{-1} \boldsymbol{d}(k)}{\boldsymbol{d}(k)^H \left[\boldsymbol{R}_{\mathrm{n}}(k) + \epsilon\, \mathrm{diag}(|\boldsymbol{X}(k,l)|^2)\right]^{-1} \boldsymbol{d}(k)} \tag{2}$$

where $\epsilon = 10^{-3}$ is the diagonal loading term, $\boldsymbol{d}(k)$ is the steering vector, $\boldsymbol{R}_{\mathrm{n}}(k)$ is the spatial correlation matrix of noise, and diag($\mathbf{x}$) denotes the conversion of vector $\boldsymbol{x}$ to a diagonal matrix.

For the estimation of the unknown quantities $\boldsymbol{d}(k)$ and $\boldsymbol{R}_{\mathrm{n}}(k)$ we use the method of [3], which does not require knowledge of the array geometry or the speaker location. We assume that each frequency bin contains either speech and noise or is dominated only by noise. This assumptions allows the clustering of the STFT coefficients into two classes: the noisy (i.e., speech + noise) and the noise-only class. The clustering is performed by modeling the STFT coefficients at each frequency with a 2-component complex Gaussian mixture model. To associate each Gaussian component to its correct class, we measure the ratio of the first to second largest eigenvalues of the estimated covariance matrices. The component for which this ratio is the largest is assigned to the noisy class.

The spatial correlation matrices of speech, $\boldsymbol{R}_{\text{s}}(k)$, noise, $\boldsymbol{R}_{\text{n}}(k)$, and noisy signals, $\boldsymbol{R}_{\text{sn}}(k)$, are then estimated based on the posterior probabilities of each bin to belong to the noisy or noise-only class as:

$$\begin{aligned} \boldsymbol{R}_{\text{sn}}(k) &= \frac{1}{L}\sum_{l=1}^{L}\boldsymbol{X}(k,l)\boldsymbol{X}(k,l)^H && (3) \\ \boldsymbol{R}_{\text{n}}(k) &= \frac{1}{\sum_{l=1}^{L}\lambda_n(k,l)}\sum_{l=1}^{L}\lambda_n(k,l)\boldsymbol{X}(k,l)\boldsymbol{X}(k,l)^H && (4) \\ \boldsymbol{R}_{\text{s}}(k) &= \boldsymbol{R}_{\text{sn}}(k) - \boldsymbol{R}_{\text{n}}(k) && (5) \end{aligned}$$

where $\lambda_n(k,l)$ denotes the posterior probability that the time-frequency bin (k,l) is dominated by noise.

Finally, the steering vector for each frequency bin k is estimated as the principal component of $\boldsymbol{R}_{\text{s}}(k)$. For the 6ch track the spatial correlation matrix of noise which is used in Eq. (2) is estimated from Eq. (4), while for the 2ch track it is estimated from 400 ms to 800 ms of context immediately before the utterance, as it was shown to produce better recognition performance in the 2ch case. Each utterance was processed using frames of 512 samples with 50% overlap, windowed with sine windows and an FFT size of 512 samples, while channel 2 was excluded from processing.

2.1.3. Superdirective beamformer using time-delays

The superdirective beamformer maximizes the array gain, while maintaining a minimum constraint on the white noise gain [1]. The beamformer filter coefficients are computed as:

$$\boldsymbol{w}_{\text{SD}}(k,l) = \frac{\left[\boldsymbol{\Gamma}(k) + \epsilon\boldsymbol{I}\right]^{-1}\boldsymbol{d}(k,l)}{\boldsymbol{d}(k,l)^H\left[\boldsymbol{\Gamma}(k) + \epsilon\boldsymbol{I}\right]^{-1}\boldsymbol{d}(k,l)} \tag{6}$$

where $\boldsymbol{I}$ is the identity matrix and ϵ is the diagonal loading term which is used to control the white noise gain (WNG) constraint. $\boldsymbol{\Gamma}(k)$ is the noise coherence matrix for frequency bin k (assumed to be spherically isotropic diffuse [10]) whose elements are given by:

$$\Gamma_{ij}(k) = \text{sinc}\left(\frac{2\pi f d_{ij}}{c}\right) \tag{7}$$

where f is the frequency in Hz, $c = 343$ m/s is the speed of sound and d_{ij} denotes the distance between the ith and jth microphone. Finally, the steering vector is represented by:

$$\boldsymbol{d}(k,l) = \begin{bmatrix} e^{-j2\pi f\tau_1(l)} & \cdots & e^{-j2\pi f\tau_M(l)} \end{bmatrix} \tag{8}$$

where $\tau_i(l)$ denotes the time delay to the ith microphone for time-frame l, which was estimated using the nonlinear SRP-PHAT pseudo-spectrum [2]. To determine ϵ, we start from $\epsilon = 0$ and iteratively increase it by 0.05 until the WNG becomes equal or greater than -10 dB.

This front end (SD) is used in the 6ch track, as well as in the 2ch track. For both tracks, we used frames of 1024 samples with 50% overlap, windowed with sine windows and an FFT size of 1024, while channel 2 was excluded from processing.

2.1.4. BLSTM supported GEV

The Generalized Eigenvalue (GEV) front end (GE) maximizes the signal-to-noise ratio after the beamforming operation:

$$\boldsymbol{w}_{\text{GEV}}(k) = \underset{\boldsymbol{d}}{\text{argmax}}\,\frac{\boldsymbol{d}(k)^{\text{H}}\boldsymbol{R}_{\text{s}}(k)\boldsymbol{d}(k)}{\boldsymbol{d}(k)^{\text{H}}\boldsymbol{R}_{\text{n}}(k)\boldsymbol{d}(k)}. \tag{9}$$

Maximizing this equation leads to the generalized eigenvalue problem and its solution to the beamforming vector $\boldsymbol{w}_{\text{GEV}}(k)$ for each frequency. Similar to the MVDR beamformer described above, this beamformer only relies on the signal statistics, i.e. no assumptions on the microphone array configurations are made. In contrast to the MVDR however, the GEV can introduce arbitrary distortions because the magnitude of each beamforming vector can be chosen arbitrarily. We therefore normalize the steering vectors using Blind Analytic Normalization (BAN) [11]. This postfilter normalizes the Acoustic Transfer Function (ATF) from the target source to unit gain for each frequency.

The spatial correlation matrices needed for the beamforming operation are estimated using time-frequency masks from a neural network [12][4]. Here, we calculate two masks, one for the target and one for the distortion. These masks do not necessarily sum to one. We only want to take those time-frequency bins into consideration where the respective source is surely predominant. To calculate the masks we treat each microphone separately and then use median pooling to condense the masks into one for each source. This strategy makes the mask estimation immune to corrupted channels. It also allows us to use the same front end for the 6 channel, as well as for the 2 channel track without making any changes to the network. The network is the same as described in [12] and is trained using binary masks as targets.

2.2. Back end

2.2.1. Data sets

The participants of the CHiME 4 Challenge were given a training corpus that was derived from the WSJ0 SI-84 data set (approx. 18 hours) recorded with a close talk microphone (channel 0) and 6 distant microphones (channels 1-6). First off we trained a fairly standard GMM/HMM acoustic model on the quasi-clean data (channel 0 of the real training data as well as the booth training data and the original WSJ corpus) in order to use its alignments on all other channels without having to re-align the data for every subsequent experiment. We further created a flattened training set (referred to as a set of front facing microphones FC) by simply concatenating the channels $\{1, 3, 4, 5, 6\}$ of both real and simulated data into a 90 hours corpus. We mostly discard the second channel since the corresponding microphone points away from the speaker, resulting in a slightly worse quality. In order to investigate the effect of beamforming on the overall ASR performance, we further define an extension of the flattened set FC by adding the beamformed signal to the concatenation. The resulting 108 hours corpus is referred to as set FC+B.

For the processing of test data we followed the rules of the challenge. In the 1ch track, no beamforming is required. In the 2ch and 6ch tracks, we first beamform all available channels into a single signal before decoding. The recognition was done using the standard 5k lexicon and baseline 5-gram count LM, followed by lattice rescoring with a neural network language model.

2.2.2. RWTH's BLSTM acoustic model

The first back end was implemented using RWTH's open-source toolkits RASR [5] and RETURNN [6]. We will refer to this back end as *"R"* and the Kaldi baseline back end as *"K"*. The architecture and training algorithms for the speaker independent and speaker adapted AM are identical. The AM is a Deep Neural Network (DNN) with five BLSTM layers of size 600. The mini-batch training is carried out using stochastic gradient descent with Nadam [13] and the learning rate reduction is controlled by Newbob [14]. The initial learning rate is set to 10^{-3} and the gradient is distorted by Gaussian noise [15] with an initial variance of 0.3. The cross-entropy training is regularized by a dropout rate of 10% and L_2 norm of the weights with a factor of 0.01. The decoding pipeline is shown in Figure 1. It differs from a standard two-pass decoding strategy by an additional LM rescoring with a neural network LM after both passes.

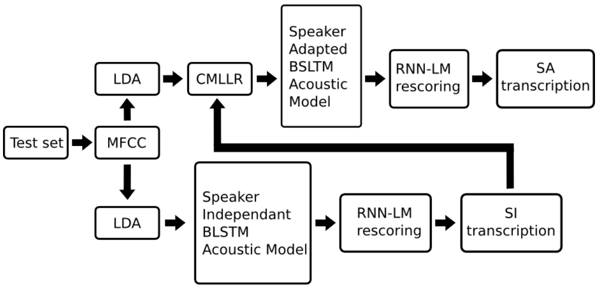


Figure 1: Decoding pipeline

2.2.3. UPB's wide residual BLSTM acoustic model

The lower part of the second back end follows a slightly modified design of a Wide Residual Network (WRN) [16] with $d = 22$ and $k = 5$, where d describes the depth of the network (i.e. number of layers) while k is a multiplicative factor for the number of channels (i.e. the width of the network). This number increases with the depth as follows: $16 \rightarrow k{\cdot}16 \rightarrow k{\cdot}32 \rightarrow k{\cdot}64$. We halve the frequency resolution by using a stride of 2 each time we increase the number of channels except for the first time. On top of these layers are two BLSTM layers with 512 units for each direction and a final fully connected layer. We call this configuration Wide Residual BLSTM Network (WRBN) (or "W" for short in Section 3).

For training, we first extract the alignments with the baseline back-end and the GEV front end using all six channels. We then train this network with a cross-entropy criterion and Adam [17]. To prevent overfitting we use dropout on the input of each layer. Additionally we use it on the hidden-hidden transitions of the BLSTM. Here, we sample the mask once per sequence [18]. We use 80 dimensional mean-normalized log-Mel filter bank features as input. Their delta and delta-delta features act as extra channels. The network is trained on the unprocessed training data from all six channels. Instead of training on a mini-batch of a few frames, we train it on a whole utterance with full backpropagation through time. This allows the WRN and BLSTM to exploit the full temporal context. Also, it enables us to use Batch-Normalization (BN) [19] in an effective way. Here, we do not rely on statistics estimated on the training or development data. We can normalize the networks activations using the utterance statistics during test time. This is not possible when working with frames because their high correlation due to their overlap prohibits a good estimation of the statistics. For (speaker) adaptation, we train an additional layer consisting of a 80×80 weight matrix for each speaker and each track. That layer with tied weights is applied to all three feature channels equally.

2.2.4. RNN language model lattice rescoring

For the RASR back end (R) we carried out lattice rescoring [20] with an LSTM-RNN language model [21, 22] as follows. For each of the two rescoring steps (speaker independent and adapted) shown in Figure 1, a specific model was used. The first pass lattices were always rescored with a small LSTM model we refer to as L1. For the rescoring of the second pass lattices we compared our own LSTM model L2 with the baseline RNN model LB. The model L1 is based on a one-layer standard LSTM while L2 is based on a 3-layer LSTM with highway connections. The size of all hidden layers is set to 500 for both models. For the training of model L2 we applied a dropout rate of 20% on the non-recurrent connections. The output layer is factorized with word classes trained using the exchange algorithm [23]. We used 100 classes for L1 and 70 for L2. For the training of the model L2, the sentences with high OOV rates were removed from the training data, exactly as described in [24]. The model L1 was trained without this pre-processing. The interpolation weights between the baseline 5-gram count model and the LSTM model were optimized w.r.t. the perplexity on the development data [25]. We used RWTH's open-source toolkit rwthlm [7] for both training and rescoring.

For the UPB back end (W) we employ a two layer LSTM language model with 650 hidden units each – similar to the example provided by [26]. Instead of training on an endless word stream (initial state of next batch is end state of current batch), we found that training on full sentences from the provided language model training data in a random mini-batch improved cross validation scores slightly. Again we use Adam [17] for optimization for 39 epochs and apply a dropout rate of 50% (this time in the vertical connections only). Global gradient clipping with a maximum value of 5 is used. All weight matrices and bias vectors, including the embedding matrix, are initialized with random weights sampled from a uniform distribution in $[-0.1, 0.1]$. We experiment with restricted training sets limiting the maximum number of unknown symbols during training. This yielded reduced cross validation perplexities. Nevertheless, we finally selected a model trained on unrestricted training data, since this lead to the lowest WER on the dev set.

2.3. System combination

For every track we obtain the final recognition result by performing confusion network combination (CNC) of multiple systems. The lattices are first converted to individual confusion networks [27, 28] and the combination is performed by aligning the confusion networks in the order of increasing word error rate. We optimize the system weights w.r.t. the WER on the real dev set using the downhill simplex algorithm. The frame-wise CN construction algorithm is described in greater detail in Section 4.4.4 of [29].

3. Experimental evaluation

The following section gives an overview over the effects of the different components of the final system in the 6 channel track. The following notation is used. The columns FE and BE describe the front end and back end used in the decoding step, respectively. The front end is either the baseline model (BL), the superdirective beamformer (SD), the minimum variance distortionless response beamformer (MV) or the GEV beamformer (GE). The RWTH back end (R) is described in Sections 2.2.2 and the UPB back end (W) is summarized in Section 2.2.3. See [9] for the description of the baseline Kaldi back end (K). All optimization have been done exclusively on the real dev set. The results on the simulated data are only provided for reference. All tables show absolute word error rates (WER) in percent.

3.1. MVDR configurations

The MVDR beamformer uses a slightly different configuration to estimate the spatial correlation matrix of noise used in Eq. (2) for the 2ch and 6ch track. As described in Section 2.1.2, for the 6ch track the spatial correlation matrix of noise is estimated using the time-frequency masks derived from the complex Gaussian Mixture Model (MV-NoiseMasks), while for the 2ch track the matrix is estimated from 400 ms to 800 ms of context immediately before the utterance (MV-NoiseContext).

The respective configurations have shown to yield better results on the real dev set, which has been used exclusively for selection and optimization. Table 1 shows the recognition performance of the MVDR beamformer with the two configurations. In the following, MV will denote the MVDR front end with the best performing configuration for each track, i.e., MV-NoiseMasks for the 6ch track and MV-NoiseContext for the 2ch track. These systems were trained on the FC+B data set.

Table 1: Configuration comparison for MVDR beamformer

Track	MVDR configuration	Dev		Test	
		real	simu	real	simu
6ch	MV-NoiseContext	3.69	4.44	5.56	6.26
	MV-NoiseMasks	3.57	4.73	5.58	6.28
2ch	MV-NoiseContext	4.94	7.09	8.77	10.17
	MV-NoiseMasks	5.09	7.61	10.53	12.01

3.2. Speaker adaptation and lattice rescoring

Table 2 shows the effect of speaker adaptation (SA) using Constrained Maximum Likelihood Linear Regression (CMLLR) and lattice rescoring using an RNN-LM. It can be seen that both components have a significant influence on the performance. A relative improvement of 40% can be reached by using the RWTH back end presented in Section 2.2.2 with speaker adaptation and lattice rescoring compared to the baseline back end (compare first and last row).

Table 2: Effect of speaker adaptation (SA) and lattice rescoring on the 6ch track evaluated on system trained on the FC training set.

System				Dev		Test	
FE	BE	SA	RNN-LM	real	simu	real	simu
BL	K	+	LB	5.75	6.76	11.49	10.89
	R	-	-	7.80	8.71	11.81	13.89
		-	L1	5.87	6.43	9.35	10.55
		+	-	6.22	8.10	9.69	11.70
		+	LB	5.76	7.37	8.92	10.67
		+	L2	4.34	5.65	6.83	8.16

3.3. Front end performance

In order to compare the front ends, we evaluate speaker adapted systems trained on the FC training set and apply LM rescoring with RNNs. Table 3 shows the performance of different beamformers on the 6ch track test data using both Kaldi and RWTH back ends. The results indicate that all front ends presented here have a positive effect on the ASR performance on the real data. The best front end (GE) leads to a further relative improvement of up to 41% over the baseline front end (BL).

Table 3: Comparison of front ends on a speaker adapted model with lattice rescoring for the 6ch track.

System		Dev		Test	
FE	BE	real	simu	real	simu
BL	K	5.75	6.76	11.49	10.89
SD		5.47	6.34	11.47	10.42
MV		4.63	5.44	8.73	8.62
GE		3.70	3.72	5.76	4.24
BL	R	4.34	5.65	6.83	8.16
SD		3.89	5.14	6.59	7.99
MV		3.90	5.23	5.65	8.36
GE		3.27	3.41	4.02	3.93

3.4. Including beamformed signal in the training

Table 4 shows the effect of extending the training set of the speaker adapted model by the pre-processed training data. The recognition is performed with the RWTH back end and includes lattice rescoring with the model L2. It can be seen that only minor improvements on the real data can be achieved. In the case of the baseline front end (BL) the performance on the simulated data even degrades. Nevertheless we decided to use the extended training set (FC+B) for further experiments.

Table 4: Effect of enhancing the training set FC consisting of channels 1,3-6 by the data obtained by pre-processing the training set with the matching front end on the 6ch track (FC+B).

System		Dev		Test	
FE	Training set	real	simu	real	simu
BL	FC	4.34	5.65	6.83	8.16
	FC+B	4.11	5.77	6.82	8.53
SD	FC	3.89	5.14	6.59	7.99
	FC+B	3.74	5.03	6.52	7.84
MV	FC	3.90	5.23	5.65	8.36
	FC+B	3.57	4.73	5.58	6.28
GE	FC	3.27	3.41	4.02	3.93
	FC+B*	3.05	2.79	3.77	2.67

* This system has not been available at time of evaluation and is only included here for completeness

3.5. System combination

Table 5 shows the single systems used for system combination and Table 6 shows the result of combining multiple systems. It can be seen that in case of using only the RWTH back end (R) each additional front end has a positive effect on the performance on the real data. However, the optimization algorithm reduces the weight of the system using the baseline front end (BL) to zero when we include UPB's back end W (last row). The best result obtained for the real evaluation data on the 6 channel track is 2.91%, which is a relative improvement of almost 75% over the baseline system.

Table 5: Single systems used for system combination in the 6 channel track

System				Dev		Test	
ID	FE	BE	Training set	real	simu	real	simu
1	BL	R	CH1,3-6+BL	4.11	5.77	6.82	8.53
2	SD		CH1,3-6+SD	3.74	5.03	6.52	7.84
3	MV		CH1,3-6+MV	3.57	4.73	5.58	6.28
4	GE		CH1,3-6	3.27	3.41	4.02	3.93
5	GE	W	CH1-6	2.73	2.34	3.48	2.76

Table 6: Results of CNC system combination of different systems for the 6 channel track

System weights					Dev		Test	
5	4	3	2	1	real	simu	real	simu
	0.50	0.50			2.75	3.13	3.57	3.56
	0.33	0.33	0.33		2.70	3.18	3.55	3.77
	0.40	0.25	0.25	0.10	2.61	3.07	3.40	3.46
0.45	0.55				2.48	2.47	3.12	2.90
0.43	0.33	0.34			2.25	2.30	2.98	2.61
0.35	0.20	0.20	0.25		2.19	2.34	2.91	2.68
0.35	**0.20**	**0.20**	**0.25**	**0.00**	**2.19**	**2.34**	**2.91**	**2.68**

3.6. Systems for 1 and 2 channel tracks

Table 7 shows the results of the single systems and the final system combination in the 1ch and 2ch track of the challenge. All shown systems have been included in the system combination. In the 1ch track no pre-processing has been used. Table 8 shows the breakdown of the results by environment for the best systems in each track.

Table 7: Single system, system combination results for the 1ch and 2ch track and best system combination result for the 6ch track.

Tr.	System			Dev		Test	
	FE	BE	Training set	real	simu	real	simu
1ch	-	K	CH5	11.58	12.99	23.77	20.82
	-	R	CH1,3-6	7.42	9.86	12.02	15.22
	-	W	CH1-6	5.19	6.69	9.34	11.11
	COM_1ch			**5.14**	**7.40**	**9.29**	**12.36**
2ch	BL	K	CH5	8.25	9.51	16.63	15.33
	MV			8.11	9.12	16.41	14.03
	SD			8.03	9.15	15.97	15.15
	GE			6.93	8.03	13.76	9.90
	BL	R	CH1,3-6+BL	5.43	7.35	9.12	11.74
	MV		CH1,3-6+MV	4.94	7.09	8.77	10.17
	SD		CH1,3-6+SD	5.53	7.54	9.60	12.33
	GE		CH1,3-6	4.90	6.48	7.69	7.49
	GE	W	CH1-6	3.54	4.05	5.96	5.16
	COM_2ch			**3.02**	**4.04**	**5.32**	**5.27**
6ch	COM_6ch			**2.19**	**2.34**	**2.91**	**2.68**

Table 8: Breakdown of the best results by environment.

Tr.	Environment	Dev		Test	
		real	simu	real	simu
1ch	BUS	6.59	5.88	13.12	8.91
	CAF	5.90	9.88	9.84	15.11
	PED	3.45	6.14	7.57	11.95
	STR	4.60	7.71	6.63	13.45
2ch	BUS	3.91	3.44	7.52	3.68
	CAF	3.07	5.43	5.04	6.33
	PED	2.36	3.38	4.39	5.49
	STR	2.73	3.91	4.33	5.57
6ch	BUS	2.61	2.06	3.16	2.19
	CAF	2.01	2.92	2.65	2.95
	PED	2.05	2.12	2.93	2.99
	STR	2.11	2.26	2.91	2.60

4. Post evaluation results

The following results were not part of the submitted system for the 4th CHiME challenge. Table 9 shows the performance gain obtained by the sequence-discriminative training of the BLSTM acoustic model w.r.t. the sMBR criterion [30] in the 6ch track. The cross-entropy (CE) model is trained on the FC data set using the RWTH back end and the GE front end. This model is used to initialize the sMBR training. The lattices for the sMBR training were generated with a 3-gram language model. During the training we use state priors which were calculated from the output layer of the CE model as was proposed in [31]. In order to prevent overfitting we used CE-smoothing [32] with a factor of 0.1. Table 10 shows that replacing the CE model (4) by the sMBR model (4+) in the combination reduces the WER from 2.9 to 2.7% on the real test set.

Table 9: Effect of sequence training of the BLSTM acoustic model in the RWTH back end. Results with the GE front end on the 6ch track.

System		Dev		Test	
ID	Criterion	real	simu	real	simu
4	CE	3.27	3.41	4.02	3.93
4+	sMBR	2.77	3.11	3.43	3.30

Table 10: Effect of replacing the CE system (4) by the sMBR trained system (4+) in the system combination. Results on the 6ch track.

System weights						Dev		Test	
5	4	4+	3	2	1	real	simu	real	simu
0.35	0.20		0.20	0.25	0.00	2.19	2.34	2.91	2.68
0.30		0.25	0.20	0.10	0.15	2.09	2.32	2.71	2.47

5. Conclusion

In this paper we presented a detailed analysis of the acoustic models developed by RWTH, UPB and FORTH for the 4th CHiME challenge. Our joint submission based on confusion network combination of multiple systems scored second in each of the three tracks of the challenge. More specifically, we compared four different front-ends and found that the BLSTM supported GEV-beamformer consistently leads to the best ASR results in 2ch and 6ch tracks. Further we found that extending the training data set by the beamformed data only works well on real test data.

We plan to further investigate the sequence training of BLSTM back ends, since the post evaluation results have shown clearly, that further performance gain can be achieved and transferred to the final system combination.

6. Acknowledgments

This research was partially supported by Ford Motor Company and the Deutsche Forschungsgemeinschaft (DFG) under Contract No. Schl2043/11-1 and No. Ha3455/11-1. The research has also received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No. 644283, project "LISTEN".

7. References

[1] H. Cox, R. Zeskind, and M. Owen, "Robust adaptive beamforming," *IEEE Trans. on Acoustics, Speech, and Signal Processing*, vol. 35, no. 10, pp. 1365–1376, Oct. 1987.

[2] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third 'CHiME' speech separation and recognition challenge: Dataset, task and baselines," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop*, Scottsdale, AZ, USA, Dec. 2015, pp. 504–511.

[3] T. Higuchi, N. Ito, T. Yoshioka, and T. Nakatani, "Robust MVDR beamforming using time-frequency masks for online/offline ASR in noise," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Shanghai, China, Mar. 2016, pp. 5210–5214.

[4] J. Heymann, L. Drude, and R. Haeb-Umbach, "Neural network based spectral mask estimation for acoustic beamforming," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Shanghai, China, Mar. 2016, pp. 196–200.

[5] S. Wiesler, A. Richard, P. Golik, R. Schlüter, and H. Ney, "RASR/NN: The RWTH neural network toolkit for speech recognition," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Florence, Italy, May 2014, pp. 3313–3317.

[6] P. Doetsch, A. Zeyer, P. Voigtlaender, I. Kulikov, R. Schlüter, and H. Ney, "RETURNN: The RWTH extensible training framework for universal recurrent neural networks," *arXiv preprint arXiv:1608.00895, submitted for publication at ICASSP 2017*, 2016.

[7] M. Sundermeyer, R. Schlüter, and H. Ney, "rwthlm - the RWTH Aachen University neural network language modeling toolkit," in *Proc. Interspeech*, Singapore, Sep. 2014, pp. 2093–2097.

[8] J. Heymann, L. Drude, and R. Haeb-Umbach, "Wide residual BLSTM network with discriminative speaker adaptation for robust speech recognition," *submitted to the CHiME 4 workshop*, 2016.

[9] E. Vincent, S. Watanabe, J. Barker, and R. Marxer, "An analysis of environment, microphone and data simulation mismatches in robust speech recognition," *submitted to Computer Speech and Language*, 2016.

[10] B. F. Cron and C. H. Sherman, "Spatial-correlation functions for various noise models," *Journal of the Acoustical Society of America*, vol. 34, no. 11, pp. 1732–1736, Nov. 1962.

[11] E. Warsitz and R. Haeb-Umbach, "Blind acoustic beamforming based on generalized eigenvalue decomposition," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 15, no. 5, pp. 1529–1539, Jul. 2007.

[12] J. Heymann, L. Drude, A. Chinaev, and R. Haeb-Umbach, "BLSTM supported GEV beamformer front-end for the 3rd CHiME challenge," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop*, Scottsdale, AZ, USA, Dec. 2015, pp. 444–451.

[13] T. Dozat, "Incorporating Nesterov momentum into Adam," Stanford University, CS 229 Machine Learning, Tech. Rep., 2015. [Online]. Available: http://cs229.stanford.edu/proj2015/054_report.pdf

[14] D. Johnson. (2004) QuickNet, Speech group at ICSI, Berkeley. [Online]. Available: http://www1.icsi.berkeley.edu/Speech/faq/nn-train.html

[15] A. Neelakantan, L. Vilnis, Q. V. Le, I. Sutskever, L. Kaiser, K. Kurach, and J. Martens, "Adding gradient noise improves learning for very deep networks," in *Proc. Int. Conf. on Learning Representations (ICLR) Workhop Track*, San Juan, Puerto Rico, May 2016.

[16] S. Zagoruyko and N. Komodakis, "Wide residual networks," in *British Machine Vision Conference*, York, UK, Sep. 2016.

[17] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," in *Proc. Int. Conf. on Learning Representations*, San Diego, CA, USA, May 2015.

[18] S. Semeniuta, A. Severyn, and E. Barth, "Recurrent dropout without memory loss," *CoRR*, vol. abs/1603.05118, 2016. [Online]. Available: http://arxiv.org/abs/1603.05118

[19] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *Proc. Int. Conf. on Machine Learning*, Lille, France, Jul. 2015, pp. 448–456.

[20] M. Sundermeyer, Z. Tüske, R. Schlüter, and H. Ney, "Lattice decoding and rescoring with long-span neural network language models," in *Proc. Interspeech*, Singapore, Sep. 2014, pp. 661–665.

[21] M. Sundermeyer, R. Schlüter, and H. Ney, "LSTM neural networks for language modeling." in *Proc. Interspeech*, Portland, OR, USA, Sep. 2012, pp. 194–197.

[22] M. Sundermeyer, H. Ney, and R. Schlüter, "From feedforward to recurrent LSTM neural networks for language modeling," *IEEE Trans. on Audio, Speech and Language Processing*, vol. 23, no. 3, pp. 517–529, Mar. 2015.

[23] S. Martin, J. Liermann, and H. Ney, "Algorithms for bigram and trigram word clustering," *Speech communication*, vol. 24, no. 1, pp. 19–37, 1998.

[24] T. Yoshioka, N. Ito, M. Delcroix, A. Ogawa, K. Kinoshita, M. Fujimoto, C. Yu, W. J. Fabian, M. Espi, T. Higuchi *et al.*, "The NTT CHiME-3 system: Advances in speech enhancement and recognition for mobile multi-microphone devices," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop*, Scottsdale, AZ, USA, Dec. 2015, pp. 436–443.

[25] A. Stolcke, "SRILM - an extensible language modeling toolkit," in *Proc. Interspeech*, Denver, CO, USA, Sep. 2002, pp. 901–904.

[26] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," *CoRR*, vol. abs/1409.2329, 2014. [Online]. Available: http://arxiv.org/abs/1409.2329

[27] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: word error minimization and other applications of confusion networks," *Computer Speech & Language*, vol. 14, no. 4, pp. 373–400, 2000.

[28] G. Evermann and P. Woodland, "Posterior probability decoding, confidence estimation and system combination," in *NIST Speech Transcription Workshop*, vol. 27, 2000, p. 78.

[29] B. Hoffmeister, "Bayes risk decoding and its application to system combination," Ph.D. dissertation, RWTH Aachen University, Computer Science Department, Aachen, Germany, Jul. 2011.

[30] D. Povey and B. Kingsbury, "Evaluation of proposed modifications to MPE for large scale discriminative training," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Honolulu, HI, USA, Apr. 2007, pp. 321–324.

[31] V. Manohar, D. Povey, and S. Khudanpur, "Semi-supervised maximum mutual information training of deep neural network acoustic models," in *Proc. Interspeech*, Dresden, Germany, Sep. 2015, pp. 2630–2634.

[32] H. Su, G. Li, D. Yu, and F. Seide, "Error back propagation for sequence training of context-dependent deep networks for conversational speech transcription," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Vancouver, Canada, May 2013, pp. 6664–6668.

Multi-Channel Speech Recognition: LSTMs All the Way Through

Hakan Erdogan[2], *Tomoki Hayashi*[1], *John R. Hershey*[1], *Takaaki Hori*[1], *Chiori Hori*[1]
Wei-Ning Hsu[1], *Suyoun Kim*[1], *Jonathan Le Roux*[1], *Zhong Meng*[1], *Shinji Watanabe*[1]

[1]Mitsubishi Electric Research Laboratories (MERL), Cambridge MA, USA
[2]Sabanci University, Istanbul, Turkey

haerdogan@sabanciuniv.edu,
{hayashi,hershey,thori,chori,whsu,skim,leroux,zmeng,watanabe}@merl.com

Abstract

Long Short-Term Memory recurrent neural networks (LSTMs) have demonstrable advantages on a variety of sequential learning tasks. In this paper we demonstrate an LSTM "triple threat" system for speech recognition, where LSTMs drive the three main subsystems: microphone array processing, acoustic modeling, and language modeling. This LSTM trifecta is applied to the CHiME-4 distant recognition challenge. Our previous state-of-the-art ASR systems for the previous CHiME challenge employed LSTM mask estimation based beamforming, noise robust features, in addition to DNN/RNNLM based back end. The proposed system refines each module of the previous system including bidirectional LSTM (BLSTM) mask estimation based beamforming, BLSTM-DNN hybrid acoustic model, and language model rescoring based on LSTM. We perform constrained re-estimation based speaker adaptation, and also prepare several complementary systems by changing the beamforming strategy and the acoustic model configurations, and combine these systems based on word-posterior based system combination. The final system achieved 2.98% WER for the real test set in the 6-channel track, which reduces the WER from the baseline by 8.5% absolute, and also outperforms our previous CHiME-3 system by 6.1% absolutely.

1. Background

The MERL-Sabanci system, as shown in Figure 1, is a multi-channel ASR system that focuses on the CHiME-4 6ch track [1]. It is an extension of our CHiME-3 system [2], and improves upon it using the following methods:

- BLSTM mask estimation for Minimum Variance Distortionless Response (MVDR) and Generalized EigenVector (GEV) beamformers.
- BLSTM-DNN hybrid acoustic model via state posterior combination.
- Expanded noisy data training using all 6 channels of official training speech data (i.e., 6 times the amount of training data).
- Unsupervised speaker adaptation based on constrained retraining of DNN.
- Language model re-scoring based on LSTM.
- System combination across multiple methods and input features.

These techniques steadily improve the performance from the baseline. Their technical details are explained in the following section.

2. Contributions

2.1. BLSTM mask estimation for beamformers

We train a unique BLSTM neural network for single-channel mask prediction using the simulated training data for all six channels. The network takes a single channel as input and predicts both speech and noise masks for that channel using sigmoid output activations and ideal binary masks as targets. The network is trained with the binary cross-entropy loss function [3]. During recognition, the network is applied separately to each channel, and the predicted masks for the six channels are combined to obtain a single mask by taking their median. The obtained speech and noise masks are then used to predict speech and noise spatial covariance matrices which are used in MVDR and GEV beamformers to perform beamforming-based enhancement on the multi-channel signal to be recognized.

2.2. Beamforming

We perform MVDR and GEV beamforming. The version of MVDR beamforming we use only uses spatial covariance estimates of speech and noise. To obtain these spatial covariances, we make use of the masks predicted by the network. The covariances are estimated as follows:

$$\hat{\mathbf{\Phi}}_x(f) = \frac{\sum_t \hat{M}_x(t,f)\boldsymbol{Y}(t,f)\boldsymbol{Y}^H(t,f)}{\sum_t \hat{M}_x(t,f)},$$

where $\hat{M}_x$ is the predicted mask for speech or noise and $\boldsymbol{Y}(t,f)$ is the received multi-channel signal's spatial vector corresponding to time-frequency bin (t,f). For GEV beamforming [3], we form the beamforming filters by maximizing the SNR for each frequency by solving the generalized eigenvalue problem for the spatial filter $\boldsymbol{h}$:

$$\hat{\mathbf{\Phi}}_{\text{speech}}\boldsymbol{h} = \lambda\hat{\mathbf{\Phi}}_{\text{noise}}\boldsymbol{h}.$$

For MVDR beamforming, we first choose a reference microphone and then find the direction of minimum noise variance while keeping the speech signal distortionless. Using one possible formulation [4], the solution can be found as:

$$\hat{\boldsymbol{h}} = \frac{1}{\text{trace}(\hat{\mathbf{\Phi}}_{\text{noise}}^{-1}\hat{\mathbf{\Phi}}_{\text{speech}})}\hat{\mathbf{\Phi}}_{\text{noise}}^{-1}\hat{\mathbf{\Phi}}_{\text{speech}}\boldsymbol{e}_{\text{ref}},$$

where $\boldsymbol{e}_{\text{ref}}$ is a standard unit vector in direction ref.

The authors are listed in the alphabetic order. Tomoki Hayashi, Wei-Ning Hsu, Suyoun Kim, and Zhong Meng performed the work during their internship programs at MERL.

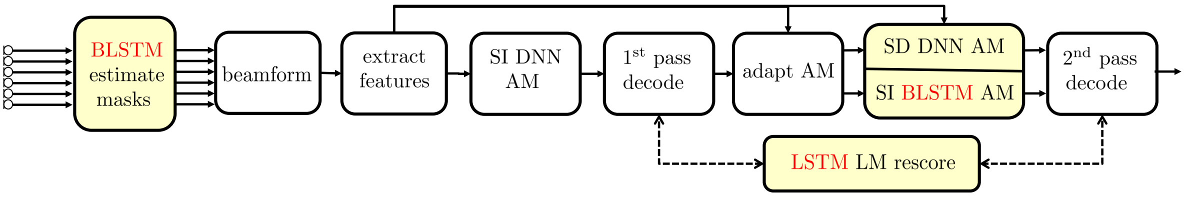


Figure 1: A flow chart of the proposed system for decoding.

2.3. Acoustic modeling and adaptation

Although RNNs (especially LSTMs) have been shown to be very effective for noise robust speech recognition [5, 6], our preliminary attempt at applying LSTMs/BLSTMs to the CHiME-4 task was not successful probably due to the limited amount of training data and the difficulty of obtaining correct state alignments from noisy speech. Instead, the acoustic models we used in our experiments are hybrid BLSTM-DNN systems. BLSTM and DNN models are separately trained with augmented training data by using the noisy speech training data from all 6 channels [7]. The DNN model configuration is the same as that of the official baseline acoustic model [1]: a 7 hidden layer sigmoid DNN with 2048 activations per layer trained by using state-level Minimum Bayes Risk (sMBR) criterion in the kaldi nnet1 module [8]. The BLSTM acoustic model has 3 layers, where each layer consists of forward and backward unidirectional LSTMs with 512 cell states and one linear bottleneck layer to combine the outputs of both unidirectional LSTMs outputting 512 activations. The BLSTM was trained based on the cross entropy criterion by using stochastic gradient descent. We used a state alignment obtained by using the DNN as a target.

On top of the training, we adapt the speaker-independent DNN to the data of each speaker in an unsupervised way. We used a constrained re-training (CRT) adaptation method where we re-estimate the DNN parameters of only a subset of layers while holding the remaining parameters fixed with the cross entropy criterion. The optimal subset of layers to be estimated is selected according to the development set performance. Since we cannot use any prior knowledge about the environment according to the CHiME-4 regulation, we train each speaker-dependent DNN with the speaker's speech from all different environments. We also use KL divergence adaptation [9] by using the speaker-independent DNN to regularize the speaker-dependent DNN. The adaptation target (1-best alignment) was obtained at the first-pass decoding, and the second-pass decoding is performed using this speaker-adapted DNN, as shown in Figure 1.

The BLSTM acoustic model and DNN model adaptation are implemented by using chainer deep learning toolkit [10].

2.4. Language model re-scoring

We train an LSTM-based RNN language model (LSTMLM) using the official training data for language modeling in CHiME-4.

RNN language models (RNNLMs) [11] robustly estimate word probability distributions by representing the contextual information in a continuous space, which are kept in the hidden layer with recurrent connections. Compared to N-gram models, RNNLMs can exploit more long-distance interword dependencies to predict the next word, and yield better performance in many tasks. However, RNNs are not able to keep very long histories because the contextual information at a certain time exponentially decays by doing recurrent propagations through time. Accordingly, we introduce LSTMLM [12, 13] to improve the system performance. The LSTM RNN has a memory cell in each hidden unit instead of a regular network unit, which can remember the contextual information for an arbitrary length of time. By expoiting the longer contextual information, LSTMLM can predict the next word more accurately than the standard RNNLMs.

In the decoding phase, word lattices are first generated using the baseline language model for CHiME-4, which is the standard 5k WSJ trigram downsized with an entropy pruning technique [14]. After that, N-best lists are generated from the lattices using a 5-gram language model with a modified Kneser-Ney smoothing [15, 16]. Finally, the N-best lists are rescored using a linear combination of the 5-gram and LSTMLM probabilities in the log domain, i.e.,

$$\log P(W) = \sum_{i=1}^{L}\{\lambda \log P_{lstm}(w_i|h_i) \\ +(1-\lambda)\log P_{5gkn}(w_i|h_i)\}, \qquad (1)$$

where $W = w_1, w_2, \ldots, w_L$ denotes each sentence hypothesis, λ the interpolation weight, and h_i the history of w_i.. The best-rescored hypothesis is selected as the result of each single system. The N-best lists are also used for system combination.

For the challenge, LSTMLM was designed as an RNN with one projection layer of 1000 units and one hidden layer of 1500 LSTM cells. We set the interpolation weight λ in Eq. (1) to 0.9 and the number of N-best hypotheses to 100, which were selected based on word error rate for the development set.

2.5. System combination

In the proposed system, multiple feature vector sequences are obtained for different pairs of beamforming and feature extraction methods, and they are separately processed by a WFST-based decoder to output word lattices. After rescoring with the LSTMLM, multiple lists of N-best hypotheses are obtained and then used for system combination.

System combination is a technique to improve recognition accuracy by combining different recognition hypotheses from different systems [17]. First, the multiple hypotheses are combined by taking their union after reweighting each hypothesis with its posterior probability. After that, minimum Bayes risk (MBR) decoding is performed on the combined hypotheses using an algorithm in [18]. With this decoding, we can find the hypothesis with the minimum expected word error rate from among all the hypotheses obtained by the multiple systems.

3. Experimental evaluation

3.1. Mask prediction network and beamforming setup

For mask prediction and beamforming, we used windows of length 1024 samples with a frame shift of 256 samples. The non-redundant FFT vector dimension was 513. Magnitude FFT was used as an input to the mask prediction network for each frame. The mask prediction network had a single BLSTM layer with 256 nodes. After the BLSTM layer, we used two feedforward layers with rectified linear unit activations with an output dimension of 513. The output layer predicted predicted 513 dimensional masks for speech and noise separately with a sigmoid activation for each output. The target ideal binary masks did not sum to one for each time-frequency bin. Ideal binary masks were chosen to be one when the corresponding source was significantly larger than the other source.

For beamforming, we pass each channel's input through the network, take the median of each channel's outputs for each time-frequency bin and use the value as a mask directly. For the MVDR beamformer, we chose microphone CH5 as the reference microphone.

3.2. Experimental Results

The first set of experiments compare the baseline script (BeamformIt [19], DNN sMBR, and 3-gram) with two beamforming techniques. Table 1 summarizes results for three types of beamforming, and both methods using the BLSTM based masks greatly improve the performance from BeamformIt. The training utilizes noisy data from channel 5 only. Also we observed that the GEV beamformer yields similar performance on simulated versus real data, both for the development and for the test sets, whereas the MVDR beamformer has systematically better performance on the simulation data. Because these properties are complementary, both beamformers are included in the final system combination.

Table 1: Average WER (%) for the front-end systems with fixed DNN sMBR, 3-gram back-end.

Track	System	Dev		Test	
		real	simu	real	simu
6ch	Baseline: BeamformIt	8.14	9.07	15.00	14.23
	BLSTM-Mask MVDR	6.66	5.55	11.39	6.39
	BLSTM-Mask GEV	7.19	7.50	10.32	9.62

The second group of experiments compares acoustic model techniques with fixed front end based on BeamformIt [19]. Table 2 shows that using all 6 channels for training is particularly effective for generalization to the test set, presumably due to the increase in speech signal variety in the training data. Although an individual BLSTM acoustic model does not outperform the DNN sMBR, the state posterior patterns of both models seem to be complementary, and the hybrid BLSTM-DNN acoustic model achieves significant improvement. Based on the result, we adopt BLSTM-DNN acoustic model as the main system, but still use DNN sMBR as a complementary system to investigate several features and training variations due to its lower computational cost.

Table 2: Average WER (%) for the back-end systems with fixed BeamformIt front-end.

Track	System	Dev		Test	
		real	simu	real	simu
6ch	Baseline: DNN sMBR 3gram	8.14	9.07	15.00	14.23
	6ch Training	7.71	8.21	12.79	12.67
	BLSTM 6ch Training	8.50	8.96	13.59	13.28
	BLSTM-DNN	7.44	7.48	11.51	11.51

Table 3: Average WER (%) for combined single systems with BLSM-mask beamformers, BLSTM-DNN, LM re-scoring using LSTM, and speaker adaptation

Track	System	Dev		Test	
		real	simu	real	simu
6ch	BLSTM-Mask MVDR	5.80	4.68	8.57	5.23
	+ LM re-scoring	2.92	2.27	4.83	2.51
	+ Adaptation	2.54	1.95	4.18	1.84
	BLSTM-Mask GEV	6.26	6.34	8.13	7.83
	+ LM re-scoring	3.12	3.11	4.23	4.06
	+ Adaptation	2.77	2.63	3.81	2.94

Table 3 reports the results on combined front-end techniques and BLSTM-DNN acoustic modeling. Here, we also report the speaker adaptation and language model re-scoring on top of the systems for both BLSTM-Mask MVDR and GEV beamformers. Note that the speaker adaptation is only performed for the DNN part of the BLSTM-DNN. The table clearly shows the improvement of the combination of beamforming and BLSTM-DNN from Tables 1 and 2, and the effectiveness of the LM re-scoring and speaker adaptation is also confirmed.

Finally we have combined our main BLSTM-DNN systems with DNN sub systems. In addition to the two beamformer results (MVDR and GEV) in Table 3, we have additionally prepared comparable systems by changing features with PNCC [20] (pncc), ETSI AFE [21] (afe), and PLP [22] with pitch features (plp+p) using DNN sMBR acoustic models (DNN), retrained DNN with beamformed features (DNN, ret), and using alternative implementation of the BLSTM-Mask GEV beamformer by [3] (GEV [3]). After that, we have combined all the lattices obtained by these systems and performed system combination using minimum Bayes risk decoding.

Table 4: Average WER (%) with final system combination.

Track	System	Dev		Test	
		real	simu	real	simu
6ch	GEV [3], DNN	2.62	2.58	3.74	3.26
	GEV, DNN, ret	2.63	2.48	3.63	2.87
	MVDR, DNN, ret	2.47	1.79	4.13	1.67
	MVDR, afe, DNN	3.20	2.89	4.99	2.57
	GEV, pncc, DNN	3.62	3.68	5.49	4.66
	GEV [3], plp+p, DNN	3.31	3.26	4.85	4.43
	Combination	2.11	1.95	2.98	1.97

Using these complementary systems, the system combination achieved 2.98%, representing an improvement of 0.6% absolute over our best single system.

Table 5: WER (%) per environment for the best system.

Track	Envir.	Dev		Test	
		real	simu	real	simu
6ch	BUS	2.73	1.53	4.30	1.59
	CAF	2.08	2.14	2.71	1.83
	PED	1.78	1.67	2.37	1.81
	STR	1.81	1.92	3.10	1.72

4. Summary

This paper describes the MERL/Sabanci submission system for the CHiME-4 speech separation and recognition challenge. Our main single system consists of BLSTM-mask-estimation based beamforming, DNN-BLSTM hybrid acoustic model, and rescoring based on LSTMLM, leading to a system that employs LSTMs all the way through. With acoustic model adaptation and system combination, we finally obtained 2.98% WER, placing third among 15 submissions. Future work will consider how to integrate these complicated modules within a deep learning framework, including beamforming network [23, 24] and end-to-end ASR [25, 26, 27].

5. References

[1] E. Vincent, S. Watanabe, A. A. Nugraha, J. Barker, and R. Marxer, "An analysis of environment, microphone and data simulation mismatches in robust speech recognition," *Computer Speech & Language*, to appear.

[2] T. Hori, Z. Chen, H. Erdogan, J. R. Hershey, J. Le Roux, V. Mitra, and S. Watanabe, "The MERL/SRI system for the 3rd CHiME challenge using beamforming, robust feature extraction, and advanced speech recognition," in *Proc. ASRU*, 2015, pp. 475–481.

[3] J. Heymann, L. Drude, and R. Haeb-Umbach, "Neural network based spectral mask estimation for acoustic beamforming," in *Proc. ICASSP*, 2016.

[4] M. Souden, J. Benesty, and S. Affes, "On optimal frequency-domain multichannel linear filtering for noise reduction," *IEEE Trans. ASLP*, vol. 18, no. 2, pp. 260–276, 2010.

[5] C. Weng, D. Yu, S. Watanabe, and B.-H. F. Juang, "Recurrent deep neural networks for robust speech recognition," in *Proc. ICASSP*, 2014, pp. 5532–5536.

[6] Z. Chen, S. Watanabe, H. Erdoğan, and J. R. Hershey, "Speech enhancement and recognition using multi-task learning of long short-term memory recurrent neural networks," in *Proc. Interspeech*, 2015.

[7] T. Yoshioka, N. Ito, M. Delcroix, A. Ogawa, K. Kinoshita, M. Fujimoto, C. Yu, W. J. Fabian, M. Espi, T. Higuchi, S. Araki, and T. Nakatani, "The NTT CHiME-3 system: Advances in speech enhancement and recognition for mobile multi-microphone devices," in *Proc. ASRU*, 2015, pp. 436–443.

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

[9] D. Yu, K. Yao, H. Su, G. Li, and F. Seide, "KL-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," in *Proc. ICASSP*, 2013, pp. 7893–7897.

[10] S. Tokui, K. Oono, S. Hido, and J. Clayton, "Chainer: a next-generation open source framework for deep learning," in *Proceedings of Workshop on Machine Learning Systems (LearningSys) in The Twenty-ninth Annual Conference on Neural Information Processing Systems (NIPS)*, 2015.

[11] T. Mikolov, M. Karafiat, L. Burget, J. Cernocky, , and S. Khudanpur, "Recurrent neural network based language model," in *Proc. Interspeech*, 2010, pp. 1045–1048.

[12] M. Sundermeyer, R. Schlüter, and H. Ney, "LSTM neural networks for language modeling." in *Proc. Interspeech*, 2012.

[13] T. Hori, C. Hori, S. Watanabe, and J. R. Hershey, "Minimum word error training of long short-term memory recurrent neural network language models for speech recognition," in *Proc. ICASSP*, 2016, pp. 5990–5994.

[14] A. Stolcke, "Entropy-based pruning of backoff language models," in *DARPA Broadcast News Transcription and Understanding Workshop*, 1998, pp. 270–274.

[15] R. Kneser and H. Ney, "Improved backing-off for M-gram language modeling," in *Proc. ICASSP*, 1995, pp. 181–184.

[16] S. F. Chen and J. Goodman, "An empirical study of smoothing techniques for language modeling," in *Proc. ACL*, 1996, pp. 310–318.

[17] G. Evermann and P. C. Woodland, "Posterior probability decoding, confidence estimation and system combination," in *Proc. NIST Speech Transcription Workshop*, 2000.

[18] H. Xu, D. Povey, L. Mangu, and J. Zhu, "Minimum Bayes risk decoding and system combination based on a recursion for edit distance," *Computer Speech & Language*, vol. 25, no. 4, pp. 802–828, 2011.

[19] X. Anguera, C. Wooters, and J. Hernando, "Acoustic beamforming for speaker diarization of meetings," *IEEE Trans. ASLP*, vol. 15, no. 7, pp. 2011–2022, 2007.

[20] C. Kim and R. M. Stern, "Power-normalized cepstral coefficients (PNCC) for robust speech recognition," in *Proc. ICASSP*. IEEE, 2012, pp. 4101–4104.

[21] "ETSI - speech recognition: Advanced front-end feature extraction algorithm," ttp://www.etsi.org/technologies-clusters/technologies/past-work/speech-recognition.

[22] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *the Journal of the Acoustical Society of America*, vol. 87, no. 4, pp. 1738–1752, 1990.

[23] X. Xiao, S. Watanabe, H. Erdogan, L. Lu, J. Hershey, M. L. Seltzer, G. Chen, Y. Zhang, M. Mandel, and D. Yu, "Deep beamforming networks for multi-channel speech recognition," in *Proc. ICASSP*, 2016, pp. 5745–5749.

[24] B. Li, T. N. Sainath, R. J. Weiss, K. W. Wilson, and M. Bacchiani, "Neural network adaptive beamforming for robust multichannel speech recognition," in *Proc. Interspeech*, 2016.

[25] A. Graves and N. Jaitly, "Towards end-to-end speech recognition with recurrent neural networks." in *ICML*, vol. 14, 2014, pp. 1764–1772.

[26] D. Bahdanau, J. Chorowski, D. Serdyuk, Y. Bengio *et al.*, "End-to-end attention-based large vocabulary speech recognition," in *Proc. ICASSP*, 2016, pp. 4945–4949.

[27] S. Kim, T. Hori, and S. Watanabe, "Joint ctc-attention based end-to-end speech recognition using multi-task learning," in *arXiv preprint*, vol. arXiv:1609.06773.

Unsupervised network adaptation and phonetically-oriented system combination for the CHiME-4 challenge

Yusuke Fujita[1], *Takeshi Homma*[2], *Masahito Togami*[1]

[1]Hitachi, Ltd. Research and Development Group, Japan
[2]Hitachi America, Ltd., USA

yusuke.fujita.su@hitachi.com

Abstract

In this paper, we describe our submitted systems for the CHiME-4 challenge and report the experimental results.

We first examine unsupervised speaker adaptation method for deep neural network (DNN) based acoustic model. The speaker-dependent DNN is constructed by re-training the speaker-independent DNN using evaluation data per speaker. Experiments show that the method provides up to 29% relative gain on the word error rate (WER).

Second, we describe a phonetically-oriented system combination method. The method utilizes phonetic similarity to construct a word alignment. It gives a better treatment of insertion and deletion errors in the word alignment. Experiments show that the method provides up to 16% relative gain.

Finally, we combine the above methods with our previous approaches for the submitted system. We utilize multi-output signals from local Gaussian modeling (LGM) based source separation as augmented training data. We also used the LGM as a preprocessing of beamforming at frontend. The submitted system achieved 4.68% of WER for the real evaluation set.

1. Background

We participate in the CHiME-4 challenge [1] and we submit all (1, 2, 6ch) tracks. We explain how the speaker-dependent deep neural network (DNN) is constructed and a new development of system combination method for this challenge. The local Gaussian model (LGM) is also emphasized because it is successfully applied to the speech enhancement for the past CHiME-3 challenge [2].

2. Contributions

2.1. Unsupervised network adaptation

Speaker adaptation is successfully applied in a lot of tasks. The CHiME-4 baseline system employs feature-space maximum likelihood linear regression (fMLLR) transform for speaker adaptation. In the CHiME-3 best paper [3] used re-training of convolution neural network (CNN) for speaker adaptation and reported significant gain of word error rate (WER). While unsupervised re-training of DNN has been shown no improvement in [4], we observed an improvement on the CHiME-4 data set.

Figure 1 shows the decoding process with unsupervised network adaptation. In this work, the baseline DNN trained with state-level minimum Bayes risk criterion (DNN+sMBR) is used as an initial acoustic model for speaker adaptation. Labels for re-training are generated from 1-best decoding results of test data. The decoding for re-training is performed using the initial acoustic model and 3-gram language model, followed by rescoring using the 5-gram and recurrent neural network language model (RNNLM). Alignments of the 1-best decoding results are generated using the initial model. Then, re-training is performed using mini-batch stochastic gradient descent (SGD) algorithm with a cross entropy criterion. The parameters of mini-batch SGD are tuned using the development set.

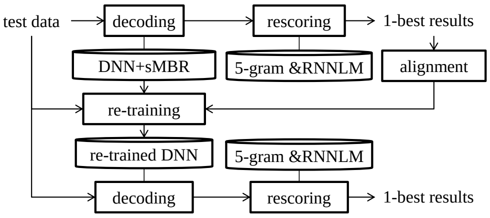


Figure 1: decoding with unsupervised network adaptation

2.2. Phonetically-oriented system combination

The ROVER [5] is a well-known technique to reduce word errors using multiple sentences obtained from multiple systems. In the approach, word alignments among multiple sentences are constructed by word-based DP matching. The word alignment makes a word set which contains words obtained from different systems in the same second. Based on the word alignment, the most trustable word within a word set is chosen. However, the word alignment often generates irrelevant word sets.

The left of Fig. 2 shows such an example: the word "their" from recognizer 1 is put into a word set containing "are" from recognizer 2 and 3. The ideal alignment in this case is that "their" from recognizer 1 is be associated with "there are" from recognizer 2 and "they are" from recognizer 3.

In this study, we employ the phonetically-oriented word alignment (POWA) proposed in [6]. A word alignment example with POWA is shown in the bottom right of Fig. 2.

Based on the POWA-based word set, we perform word selection utilizing machine learning [2].

The feature vector x used for the correct word estimators is formed as:

$$\begin{aligned}
\mathbf{x} &= (\mathbf{x}_{oc}^{\top}, \mathbf{x}_{cf}^{\top}, \mathbf{x}_{nl}^{\top})^{\top} \in \mathbb{R}^{\binom{N}{2}+2N} && (1)\\
\mathbf{x}_{oc} &= (\delta(w_i, w_j); 1 \leq i < j \leq N)^{\top} \in \mathbb{R}^{\binom{N}{2}} && (2)\\
\mathbf{x}_{cf} &= (c_i; 1 \leq i \leq N)^{\top} \in \mathbb{R}^{N} && (3)\\
\mathbf{x}_{nl} &= (\delta(w_i, \mathrm{NULL}); 1 \leq i \leq N)^{\top} \in \mathbb{R}^{N} && (4)
\end{aligned}$$

where $\delta()$ is the Kronecker delta function, N is the number of

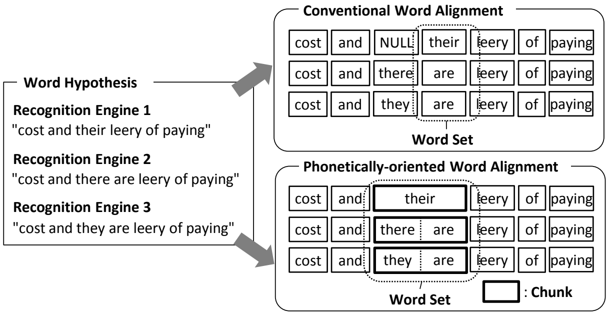

Word Hypothesis
Recognition Engine 1
"cost and their leery of paying"
Recognition Engine 2
"cost and there are leery of paying"
Recognition Engine 3
"cost and they are leery of paying"
Conventional Word Alignment
cost and NULL their leery of paying
cost and there are leery of paying
cost and they are leery of paying
Word Set
Phonetically-oriented Word Alignment
cost and their leery of paying
cost and there are leery of paying
cost and they are leery of paying
Word Set
: Chunk

Figure 2: Phonetically-oriented word alignment

recognizers, each element of $\mathbf{x}_{oc}$ is an indicator showing the chunk from a recognizer i is the same with the chunk from another recognizer j , c_i is a confidence of a chunk from a recognizer i, which is calculated as a geometric mean of words' confidences within a chunk, and NULL means the word is empty. The label vector $\mathbf{y}$ is formed as following:

$$\mathbf{y} = (\delta(w_i, w_{true}); 1 \leq i \leq N)^\top \in \mathbb{R}^N \tag{5}$$

where w_{true} means a chunk which consists of correct words. Given feature vector x and label vector y, the correct word estimator is trained by logistic regression model. The correct word estimator was trained from the development set.

2.3. LGM based source separation

2.3.1. Data augmentation using LGM

In this work, we use the data augmentation method using multiple output signals from LGM based source separation [7]. In the LGM based source separation [8], the multi-microphone signal in the time-frequency domain $\mathbf{x}(f,t)$ is expressed as

$$\mathbf{x}(f,t) = \sum_{j=1}^{J} \mathbf{c}_j(f,t), \tag{6}$$

where $\mathbf{c}_j(f,t) = [c_{1j}(f,t), \cdots, c_{Ij}(f,t)]^\top$ is the contribution of the jth source to the mixture signals, J is the number of sources, and I is the number of microphones. The source separation problem is to estimate $\mathbf{c}_j(t)$ from $\mathbf{x}(t)$.

In the LGM approach, the multichannel covariance matrix of each speech source is assumed to be a multiplication of a time-variant scalar $v_j(f,t)$ and a time-invariant multichannel matrix $\mathbf{V}_j(f)$ for jth source.

$$\mathbf{c}_j(f,t) \sim \mathcal{N}_{\mathbb{C}}(0, v_j(f,t)\mathbf{V}_j(f)) \tag{7}$$

The LGM estimates the maximum likelihood value of $v_j(f,t)$ and $\mathbf{V}_j(f)$ by using expectation-maximization algorithm. Then, the separated signal can be obtained by multichannel Wiener filtering:

$$\mathbf{c}_j(f,t) = v_j(f,t)\mathbf{V}_j(f)\mathbf{R}_x^{-1}(f,t)\mathbf{x}(f,t), \tag{8}$$

where $\mathbf{R}_x(f,t)$ is the covariance matrix of the input signal $\mathbf{x}(f,t)$ which is the sum of covariance matrix of every sources.

In this study, the number of sources is set to 3. All channels of the target source signals are used as augmented training data for acoustic modeling.

2.3.2. Semi-stationary noise separation using LGM

In the original LGM framework, all of source signals are assumed to be time-varying signals. However, in the real environments, there are a lot of semi-stationary noises. To deal with these noises, we introduce moving average smoothing of activities for the non-target noise sources in addition to the original LGM. The modification to the original LGM for non-target noise sources ($j > 0$) is following:

$$\mathbf{c}_j(f,t) \sim \mathbf{N}_{\mathbb{C}}(0, \hat{v}_j(f,t)\mathbf{V}_j(f)) \; ; j > 0 \tag{9}$$

, where $\hat{v}_j(f,t)$ is a smoothed activity:

$$\hat{v}_j(f,t) = \sum_{\tau=0}^{T_j} v_n(f, t-\tau) \tag{10}$$

T_j is the number of smoothing frames.

That modification works as a kind of regularization for avoiding over-fitting problem especially in semi-stationary noise environments. Applying the moving average filter in the each EM iteration, the target source, i.e. the most active source is extracted onto $\mathbf{c}_0$. So we no longer select the target source from separated signals using SRP-PHAT.

In this work, we use this modification of LGM for the frontend speech enhancement. For non-target two sources, the numbers of smoothing frames are set to 3 and 6. The test utterance is processed by the modified LGM before the baseline beamforming is applied.

3. Experimental evaluation

3.1. Tuning adaptation parameters

We first evaluated the sensitivity to hyper-parameters for unsupervised network adaptation. The system for this evaluation used the LGM based source separation. The structure of acoustic model and language models are the same as the baseline DNN+RNNLM system except the acoustic feature, which was 40 dimensional log mel filterbank with an energy term, followed by per utterance mean variance normalization and delta and acceleration feature augmentation.

The evaluation results for the development set are shown in Table 1. We observed the results of unsupervised network adaptation with any set of hyper-parameters always better than the non-adapted result. The small learning rate and the small number of iteration gives good result. Through this evaluation, the mini-batch size was set to 12000, the learning rate was set to 0.0004, and the number of iteration was set to 2 for further evaluations.

Table 1: Average WER (%) for various adaptation parameters

iteration	learn rate	mini-batch	WER (dev avg)
No adaptation			4.85
1	0.01	256	4.115
1	0.008	512	4.08
1	0.001	256	3.7
1	0.0004	256	3.745
1	0.0004	512	3.735
1	0.0004	12000	3.7
1	0.0001	256	3.865
2	0.0004	12000	**3.695**
10	0.0004	256	4.305

3.2. Evaluation on submitted system

The evaluation results are shown in Table 2, and the WERs per environment for the submitted systems are shown in Table 3.

The **adapted** system used unsupervised network adaptation. The initial model was from the baseline DNN+sMBR system. The baseline system used fMLLR transformed MFCC feature. The adapted system was tested only on "6ch track".

The **combined** system used phonetically-oriented system combination. The system combined four baseline systems (GMM, DNN+sMBR, DNN+5gram, DNN+RNNLM). The combined system was tested only on "6ch track".

The **LGM** system used the LGM based source separation. For the frontend of 1ch track, we applied no speech enhancement. The structure of acoustic model and language models are the same as the baseline system except the acoustic feature, which was 40 dimensional log mel filterbank with an energy term, followed by per utterance mean variance normalization and delta and acceleration feature augmentation.

The **LGM+adapted** system used unsupervised network adaptation. The initial model was from the LGM system.

The **submitted** system used phonetically-oriented system combination. The system combined 24 recognizers (12 backend models and 2 frontend methods). The backend models are comprised of 4 baselines (GMM, DNN+sMBR, DNN+5gram, DNN-RNNLM), 4 LGM-based data augmented models, 2 adapted DNN models (DNN+5gram, DNN+RNNLM) and 2 LGM-based data augmented and adapted DNN models. The frontend methods are baseline beamforming (beamformit) and LGM based beamforming as described in Section 2.3.2.

The results of the real test set on the 6ch track show the effectiveness of the unsupervised network adaptation. The relative gain was 6% of WER from the baseline and 29% from the LGM system. While the phonetically-oriented system combination was not effective for baseline systems, combination with the LGM and LGM+adapt systems achieved 16% relative gain. The LGM constantly reduced the WER and boosted the effectiveness of unsupervised network adaptation and phonetically-oriented system combination.

Table 2: Average WER (%) for the tested systems.

Track	System	Dev		Test	
		real	simu	real	simu
1ch	baseline	11.56	12.99	23.59	20.72
	LGM	9.27	11.97	16.88	17.76
	LGM+adapted	7.29	9.56	13.57	13.96
	submitted	5.89	7.36	11.42	9.23
2ch	baseline	8.21	9.50	16.55	15.40
	LGM	6.51	8.37	12.08	10.98
	LGM+adapted	5.13	6.36	9.09	7.79
	submitted	4.22	5.88	8.61	7.32
6ch	baseline	5.76	6.77	11.46	10.91
	adapted	5.37	6.36	10.77	9.18
	combined	5.77	6.80	11.48	10.72
	LGM	4.49	5.20	7.78	6.35
	LGM+adapted	3.58	3.81	5.56	4.47
	submitted	2.68	3.33	4.68	4.15

Table 3: WER (%) per environment for the best system.

Track	Envir.	Dev		Test	
		real	simu	real	simu
1ch	BUS	7.85	6.31	15.93	6.69
	CAF	6.02	9.87	11.86	9.86
	PED	4.01	5.78	9.81	9.69
	STR	5.68	7.48	8.09	10.68
2ch	BUS	5.24	4.78	12.26	4.78
	CAF	4.38	7.79	8.98	8.24
	PED	3.05	5.04	7.03	7.56
	STR	4.20	5.91	6.16	8.69
6ch	BUS	3.38	3.01	6.13	3.19
	CAF	2.20	3.92	4.50	4.17
	PED	2.33	2.88	3.87	4.20
	STR	2.80	3.53	4.24	5.02

4. Conclusion

We wrote our development for the CHiME-4 challenge and reported the experimental results. We examined unsupervised speaker adaptation for DNN based acoustic model and shown that the adaptation gives up to 29% relative gain on the 6ch track. Second, we evaluated a phonetically-oriented system combination method. Experiments showed that the system combination results up to 16% relative gain. Finally, we evaluated the combination of the above methods with LGM based source separation. The experimental results of the submitted system show that 4.68% of WER for the real evaluation set.

5. References

[1] E. Vincent, S. Watanabe, A. Nugraha, J. Barker, and R. Marxer, "An analysis of environment, microphone and data simulation mismatches in robust speech recognition," *Computer Speech and Language*, to appear.

[2] Y. Fujita, R. Takashima, T. Homma, R. Ikeshita, Y. Kawaguchi, T. Sumiyoshi, T. Endo, and M. Togami, "Unified ASR system using LGM-based source separation, noise-robust feature extraction, and word hypothesis selection," in *Proc. IEEE ASRU*, 2015, pp. 416–422.

[3] T. Yoshioka *et al.*, "The NTT CHiME-3 system: Advances in speech enhancement and recognition for mobile multi-microphone devices," in *Proc. IEEE ASRU*, 2015, pp. 436–443.

[4] H. Liao, "Speaker adaptation of context dependent deep neural networks," in *Proc. ICASSP*, May 2013, pp. 7947–7951.

[5] J. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER)," in *Proc. IEEE ASRU*, 1997, pp. 347–354.

[6] N. Ruiz and M. Federico, "Phonetically-oriented word error alignment for speech recognition error analysis in speech translation," in *Proc. IEEE ASRU*, Dec 2015, pp. 296–302.

[7] Y. Fujita, R. Takashima, T. Homma, and M. Togami, "Data augmentation using multi-input multi-output source separation for deep neural network based acoustic modeling," in *Proc. Interspeech*, 2016, pp. 3818–3822.

[8] N. Duong, E. Vincent, and R. Gribonval, "Under-determined reverberant audio source separation using a full-rank spatial covariance model," *IEEE Trans. Speech Audio Process.*, vol. 18, pp. 1830–1840, Sep. 2010.

THE I2R SYSTEM FOR CHIME-4 CHALLENGE

Tran Huy Dat, Ng Wen Zheng Terence, Sunil Sivadas, Luong Trung Tuan, Tran Anh Dung,

Institute for Infocomm Research, A*STAR, 1 Fusionopolis Way, Singapore 138632

ABSTRACT

This paper reports developments and evaluation results of I2R system for CHiME-4 challenge which addresses distant speech recognition on tablet device in challenging noisy environments. It features three tracks of 6-channel; 2-channel; and 1-channel data, respectively. Our developments are more focused on the algorithms with potentials in real-time implementation. In front-end processing, time-domain weighted delay-and-sum beamforming (WDAS) was implemented with following specific processing compared to the provided baseline processing [1]: (1) channel SNR and coherence measurements were used to calculate the beamforming weighted coefficients; (2) slow updating of the beamforming weights with 2-second windows; (3) a modified single channel speech enhancement was applied on top of output beamforming enabling further reduction of the background noise while keep controlling the introduced distortion. In the back-end processing, two new components were applied compared to the provided baseline: (1) LSTM language model for re-scoring; and (2) Semi-supervised DNN adaptation for each individual speaker in test. In evaluations, we stay with unique acoustic models for all the task and apply the processing on test data only. Consistent improvements were obtained across all three tasks. The submitted results for the real test set were 5.00%, 8.32%, and 11.19% for the 6-channel, 2-channel, and 1-channel tasks, respectively.

1. BACKGROUND

The industrial applications of speech recognition has been moving from closed talk microphones to daily real life scenarios thanks to booming developments in robotic and artificial intelligence (AI) areas. The task, however, is remained challenging due to the problems of attenuation, noise, distortion, and reverberation. Following the success of the CHiME-3 challenge which attracted many international teams to participate, CHiME-4 revisits the CHiME-3 data, i.e., utterances recorded via a 6-microphone tablet device in challenging noisy environments. The difficulty is increased by reducing the number of microphones. CHiME-4 features three tracks depending on the number of microphones available for testing: 6-channel track; 2-channel track; and 1-channel track. Excepting the 6-channel task, the channels are randomly chosen from the pool so that no specific geometrical prior information is given to the samples. The audio was recorded under real acoustic mixing conditions, i.e. talkers speaking in challenging noisy environments, including four varied noise settings: caf, street junction, public transport and pedestrian area. We participated in both three tasks and our focus is the approaches which are suitable for real-time implementations. In the front-end, the weighted delay-and-sum beamforming (WDAS) was implemented with a specific way to determine the weighted coefficients, using both coherence [1] and SNR estimations [2]. A post-processing filter is applied on top of WDAS output and that was modified from a previous speech enhancement development [3]. The modification is made to reduce the distortion level from speech enhancement and was found useful for ASR task. The same enhancement filter is applied on noisy speech in the 1-channel task. The back-end acoustic modelling follows a typical Kaldi recipe [4] and unique DNN acoustic model is applied for all the tasks [5]. In the decoding stages, LSTM LM [6] for rescoring is applied and semisupervied DNN adaptation [7] is applied on individual speaker data. Consistent improvements from baseline were obtained cross all three tasks. The major contributions come from beamforming, LSTM LM re-scoring and semisupervied DNN adaptation and additional improvements were provided by post-processing enhancement and its two-stage implementation. The submitted results for the real test set were 5.00%, 8.32%, and 11.19% for the 6-channel, 2-channel, and 1-channel tasks, respectively. These results significantly outperformed the baseline results of 11.51%, 16.58%, and 23.70% on the same datasets. The advantages of our system is that it is applicable for universal situations of environments and can be translated into real-time. We also evaluated the data-driven BLSTM trained masking GEV beamforming [8], proposed by Paderborn University (Germany), with our back-end processing on the 6-channel data. Although the masking GEV outperformed our front-end it requires extra matching data to train putting a question on its performance in an totally unknown and mismatch conditions. Further studies are necessary to prove its practical value.

2. SYSTEM DESCRIPTIONS

The block diagram of our system is illustrated in Figure 1. The highlighted yellow are the important modules which are different from the baseline method.

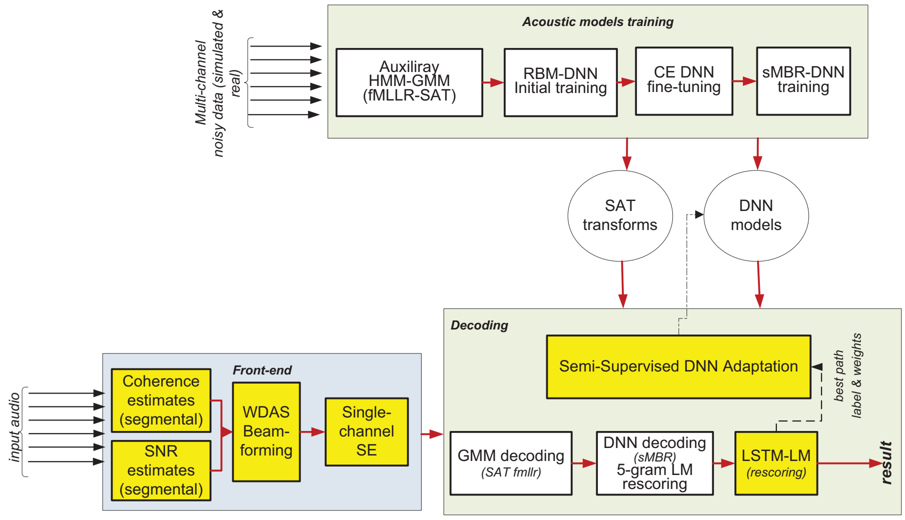

Multi-channel noisy data (simulated & real)
Acoustic models training
Auxiliray HMM-GMM (fMLLR-SAT)
RBM-DNN Initial training
CE DNN fine-tuning
sMBR-DNN training
SAT transforms
DNN models
input audio
Front-end
Coherence estimates (segmental)
SNR estimates (segmental)
WDAS Beam-forming
Single-channel SE
Decoding
Semi-Supervised DNN Adaptation
best path label & weights
GMM decoding (SAT fmllr)
DNN decoding (sMBR) 5-gram LM rescoring
LSTM-LM (rescoring)
result

Fig. 1. Overview of our CHiME-4 ASR system.

2.1. Front-end processing

Our front-end processing includes two stages of weighted-delayed-and-sum beamforming (WDAS) and a parametrized single channel speech enhancement enabling optimization of performances on the test data.

2.1.1. Beamforming

Time-domain weighted delay-and-sum (WDAS) method is applied in the beamforming step.

1. The microphone signals are first alighted using time difference of arrival (TDOA) which are estimated through GCC-PHAT.

2. The reference channel is initialized as the channel with the highest estimated SNR from channels and then iteratively tracking to the lowest negative TDOAs until they turn positive. Note that since the SNR estimated from channel number 2 is consistently bad, we have excluded this channel from our beamforming processing.

3. The weighted coefficients are calculated in two different ways before getting averaged: (1) using channel coherence measurements; (2) using SNR estimation.

$$w_i = \alpha_C \frac{CHR_i}{\sum_{j=1}^{N} CHR_j} + \alpha_S \frac{SNR_i}{\sum_{j=1}^{N} SNR_j}, \tag{1}$$

where the coherence measurements are calculated from pair-wise cross-correlation coefficients [1], noted as

$$CHR_i = \sum_{j \neq i}^{N} c_{ij}. \tag{2}$$

The SNR in each channel is estimated and updated by 2 second segments using the algorithm in [2]. α_C and α_S denote the weighting regularization coefficients between coherence and SNR measurements. Particularly, we set both of them equal to 0.5.

4. Slower updating of WDAS weights, compared to provided baseline BeamformIt front-end [1] is implemented using longer segments of 2 seconds

2.1.2. Post-processing filter

The advanatge of time-domain WDAS beamforming is that it produces very low distortions in the output signal. However, the method is less effective in removing background noise, particularly under low SNR conditions. Hence, post-processing filter is introduced to partially solve the problem. In this work, we applied the spectral estimation speech enhancement method introduced in a previous work [3]. This method estimates the speech spectral amplitude using Maximum A Posterior (MAP) criteria using generalized gamma distribution modelling of speech. While the method is effective in removing the background noise, it introduces distortions which is harmful to ASR systems. To control the distortion level, a simple modification has been applied and found to be effective in applying this method for ASR under severe noise conditions. It is done by introducing a rational power order to the original gain filter

$$\mathbf{G} \rightarrow \mathbf{G}^{\alpha}, \tag{3}$$

where the original gain filter is

$$\mathbf{G} = \frac{\hat{\mathbf{S}}}{\mathbf{X}}, \tag{4}$$

with the MAP spectral amplitude estimation noted by [3]

$$\hat{\mathbf{S}}\text{=}\mathbf{argmax_S}\left[\mathbf{p}\left(\mathbf{S},\mathbf{X}\right)\right] \tag{5}$$

The distortion controlling parameter α is chosen between $0 < \alpha < 1$. As closer to original $\alpha = 1$, the post-processing filter provides more noise removal but also more distortions. A trade-off in middle way near to $\alpha = 0.5$ seems always able to boost the ASR performances. Particularly, $\alpha = 0.5$ is used in our experiments for CHiME-4 data.

2.2. Back-end processing

2.2.1. Data augmentation

The 6-channel official training data, including both simulated and real noisy recordings provided by the challenge organizers, was used in the training [9].

2.2.2. Acoustic modelling

The acoustic modelling is carried out using standard Kaldi recipe [4]. The processing includes MFCC feature extraction followed by auxiliary HMM-GMM which provides speaker adaptive transforms (SAT) and the initial alignments. The DNN training is started with RBM initialization followed by two rounds of 4-iterations cross-entropy fine-tuning runs. The DNN training is finally carried out to deliver the acoustic models using the sMBR optimization [5].

2.2.3. Language modelling

Default 3-grams LM was used in the decoding followed by a re-scoring by provided 5-grams. Additional LSTM LM [6] was trained with provided text extracted from WSJ corpus and being used in the final re-scoring stage.

2.2.4. Decoding with semi-supervised DNN adaptation

In the decoding stages, the enhanced signals from front-end processing were used to input to the ASR system. It first passes to the HMM-GMM decoder to get the SAT-fMLLR transforms. Then the transformed features are used in the first pass of speaker independent DNN decoding using the default 3-gram LM followed by a 5-gram LM rescoring. From here, two important modifications were made, compared to the baseline method. First, instead of using RNN-LM re-scoring, we adopt more advanced LSTM LM described above. Secondly, semi-supervised adaptation is utilised, on each individual speaker data [8] using the best path state sequence and confidence measures, decoded from testing data, as the label and weightings, respectively for additional iterations of DNN fine-tuning. Five rounds of adaptations has been applied to maximize the WER reduction though it normally converges after just two rounds of adaptations.

Table 1. Average WER (%) for the tested single systems.

Track	System	Dev		Test	
		real	simu	real	simu
1ch	I2R-fb-2	6.08	7.33	**11.19**	10.87
	I2R-fb	6.14	7.42	11.25	11.34
	Noisy-I2Rb	6.15	7.60	13.05	12.89
	Baseline	11.57	12.98	23.70	20.84
2ch	I2R-fb-2	4.32	5.10	**8.32**	7.57
	I2R-fb	4.35	5.33	8.43	7.70
	BeamformIt-I2Rb	4.76	6.62	9.37	8.48
	Baseline	8.23	9.50	16.58	15.33
6ch	MaskBF-I2Rb	2.70	2.16	3.94	2.90
	I2R-fb-2	3.18	3.39	**5.00**	4.97
	I2R-fb	3.25	3.48	5.08	5.00
	BeamformIt-I2Rb	6.35	6.14	6.44	6.06
	Baseline	5.76	6.77	11.51	10.90

3. EXPERIMENTAL EVALUATIONS

This section reports the results achieved by your system. Following methods have been evaluated and compared for both 1-channel, 2-channel and 6-channel tasks, respectively.

1. **Baseline** refers to the use of provided BeamformIt front-end and also provided decoding script.

2. **Noisy-I2Rb** refers to the use of original noisy audio and our developed decoding script. This is applied for single channel task only.

3. **I2R-fb** refers to single system using our proposed front-end and back-end processing, illustrated in Fig. 1.

4. **I2R-fb-2** refers to our improved version combined two different enhancement setting ($\alpha = 0.5$ and $\alpha = 0.25$).

5. **MaskBF-I2Rb** refers to the BLSTM trained masking GEV beamforming front-end provided by Paderborn University (Germany) [9] with our back-end processing

3.1. Overall results

Table 1 reports the experimental evaluation results on both four data sets from development and testing phases. We can see that consistent and significant improvements were obtained across all the datasets and tracks, from both back-end and front-end components. Our best system (I2R-fb-2)

Table 2. WER (%) per environment for the best system.

Track	Envir.	Dev		Test	
		real	simu	real	simu
1ch	BUS	8.26	5.56	17.20	7.51
	CAF	6.46	9.99	11.82	13.69
	PED	3.64	5.58	7.70	10.27
	STR	5.94	8.22	8.05	12.03
2ch	BUS	5.65	4.14	12.60	5.64
	CAF	4.59	6.71	8.21	9.02
	PED	2.73	3.91	4.07	4.89
	STR	2.85	3.82	4.78	6.24
6ch	BUS	4.82	2.74	6.56	3.46
	CAF	3.01	4.16	4.58	5.30
	PED	2.04	2.85	4.07	4.89
	STR	2.85	3.82	4.78	6.24

achieved approximately 12%, 8%, and 7% absolute WER reductions for the real test sets in 1-channel, 2-channel and 6-channel tracks, respectively. The improvements were seen consistently over datasets. The real test set is the most challenging set but the results are closing up on the 6-channel data.

3.2. Back-end contributions

It can be seen that, our system achieved consistent improvements cross all the datasets. Most significant improvements come from our back-end processing which approximately 10%, 7% and 5% absolute accuracy gains when moving from baseline to BeamformIt-I2Rb system. Among the back-end processing components, LSTM LM re-scoring and Semi-supervised DNN adaptation contributes the most.

3.2.1. *Data augmentation*

The multi-condition training using data augmentation has proven to be very effective for the noisy ASR tasks. In our experiments, we noticed nearly 2% additional improvement compared to baseline training script just by using both 6-channel noisy data instead of single noisy in original script. While it seems redundant in speech content, it may add some more noise variation into the training which helps in delivering better models. Another explanation is adding more data may help in DNN convergence which naturally requires sufficient training data. This may have happened in this case because the size of data is significantly enlarged using 6-channel data. But our effort to further improve the training by adding more simulated data to the training was not successfully.

3.2.2. *LSTM language model re-scoring*

LSTM seems exclusively suitable for language modelling, as it could extract temporal dependency from text data while overcome fundamental vanish gradient problem in RNN training hence deliver better prediction of text contents. Consistent improvements of $2-3\%$ WER reductions compared to 5-grams LM and $1 - 2\%$ of the same compared to RNN LM were seen in our experiments, respectively.

3.2.3. *Semi-supervised DNN adaptations*

Semi-supervised DNN adaptation has repeated its great contributions in our experiments with consistent improvements from $2 - 4\%$ absolute WER reductions in both 1-channel, 2-channel and 6-channel tracks, respectively. Although the default 5-round adaptation was applied, in most of cases, the best results were converged after 1-2 steps.

3.3. Front-end contributions

Compared to provided BeamformIt baseline which stands as a very good baseline method, our front-end processing achieved consistent $1-2\%$ absolute WER reductions for both tracks of 1-channel, 2-channel, and 6-channel, respectively.

3.3.1. *Speech enhancement*

For the 1-channel tracks, the contribution of improvements was fully made by the introduced speech enhancement. Nearly 2% gain in WER reduction was obtained. Note that as the original speech enhancement did not improve the WER, the idea of gain modification to control the distortion has shown to be a practical solution enabling applications of speech enhancement methods in ASR. Although, a simplest way of introducing a rational power order is applied in this work, more sophisticated algorithms to address the introduced idea could be more useful.

For the 2-channel and 6-channel tracks, as the beamforming already enhances the input signals, effect is post-processing speech enhancement is less significant. Nevertheless, consistent improvements of $0.3-0.4\%$ were seen on top of beamforming method.

The post-processing speech enhancement module also provides possibility for system combination in front-end level while keeping acoustic models unchanged. That is more practical than fusion of totally different front-end and back-end systems, often seen in the literature. In our experiments, simply combining two enhancement in lattice improved the performances of the ASR system. Further studies in this direction are suggested.

3.3.2. *Beamforming*

Our beamforming method which had been developed and applied in our previous works [5] is similar to the BeamformIt as the time-domain WDAS is applied in both cases. However, the way to calculate beamforming weights are different: BeamfromIt uses only cross-correlation coefficients while we use estimated SNR measurements on top of coherence measurements. The SNR estimation is also used in our approach for the channel selection. Our method uses slower updating windows. Finally, our algorithm is totally real-time while the BeamformIt requires batch processing. In CHiME-4 datasets, our beamforming achieved about $0.6 - 1\%$ improvement in absolute WER reduction for 2-channel, and 6-channel tracks, respectively.

We also compared our front-end method to the BLSTM trained masking GEV beamforming provided by Paderborn University (Germany)[8]. This method uses a parallel noisy/clean training data to train a BLSTM network to get the time-frequency mask before applying it into GEV beamforming which is a spatial filter in frequency domain. The masking-GEV BF achieved great results by other participants and also got the best result in our experiments when combining with our back-end processing. It achieved amazing 3.75% WER on real test set with our back-end and is superior to our front-end. However, this method requires training data which is matching to testing in CHiME-4 and this is unknown how it would perform in totally unknown environments. Further investigations are required to confirm its practical value.

3.4. Performances over noise conditions

Breakdown of the best performed system on real test set, per each environment condition is shown in Table 2. We can see that, excepting the bus conditions, the results from each track are quite clustered over four datasets. That means that the simulation could be used to predict and improve the developments for the real conditions. That is a very good finding for the industrial developments of far-field noisy ASR applications. For the bus condition, our system underperformed in the real test set compared to the rest of conditions. Note that the same things were not observed on the masking GEV method which deliver similar results for all the conditions. Further analyses should be carried out to find out the reasons of that.

4. CONCLUSIONS

This paper reports developments and evaluation results of I2R system for CHiME-4 challenge. We achieved consistent improvements compared to provided baseline across both tracks and datasets, in both front-end and back-end processing. More significant improvement achieved in back-end processing with LSTM language modelling for re-scoring and semi-supervised DNN adaptation. Consistent improvements were also obtained in front-end processing with coherence and SNR joint analytic based WDAS beamforming and distortion-controlled speech enhancement as a post-processing filter. The proposed front-end is a real-time processing method.

5. REFERENCES

[1] Xavier Anguera, Chuck Wooters, and Javier Hernando, Acoustic beamforming for speaker diarization of meetings, IEEE Transactions on Audio, Speech and Language Processing, vol. 15, no. 7, pp. 20112023, 2007.

[2] Tran Huy Dat, Kazuya Takeda, and Fumitada Itakura, On-line gaussian mixture modeling in the log-power domain for signal-tonoise ratio estimation and speech enhancement, Speech Communication, vol. 48-1, pp. 15151527, 2006.

[3] Tran Huy Dat, Kazuya Takeda, and Fumitada Itakura, Gamma modeling of speech power and its on-line estimation for statistical speech enhancement, IEICE Transactions on Information and Systems, vol. E89D(3), pp. 10401049, 2006.

[4] Daniel Povey at el., The kaldi speech recognition toolkit, in Proceedings of IEEE workshop on automatic speech recognition and understanding (ASRU) 2011, IEEE.

[5] Jonathan W.D. and H.D. Tran, Single and multi-channel approaches for distant speech recognition under noisy reverberant conditions: i2rs system description for the aspire challenge, in Proceedings of IEEE workshop on automatic speech recognition and understanding (ASRU) 2015, IEEE, 2015.

[6] Wojciech Zaremba, Ilya Sutskever,and Oriol Vinyals Recurrent neural network regularization, CoRR, vol. abs/1409.2329, 2014.

[7] Mirko Hannemann Karel Vesely and Lukas Burget, Semisupervised training of deep neural networks, in Proceedings of IEEE workshop on automatic speech recognition and understanding (ASRU). 2011, IEEE, 2013.

[8] Jahn Heymann, Lukas Drude, Reinhold Haeb-Umbach, Neural network based spectral mask estimation for acoustic beamforming", Proceedings of ICASSP 2016, IEEE, 2016.

[9] Emmanuel Vincent, Shinji Watanabe, Jon Barker, and Ricard Marxer, An analysis of environment, microphone and data simulation mismatches in robust speech recognition, Computer Speech and Language, 2016, 2016.

The MLLP system for the 4th CHiME Challenge

Miguel Ángel del-Agua, Adrià Martínez-Villaronga, Adrià Gimènez,
Alberto Sanchis, Jorge Civera, Alfons Juan

MLLP, DSIC, Universitat Politècnica de València (UPV), Spain.
{mdelagua,amartinez1,agimenez,josanna,jcivera,ajuan}@dsic.upv.es

Abstract

The MLLP CHiME-4 system is presented in this paper. It has been built using the transLectures-UPV toolkit (TLK) developed by the MLLP research group which makes use of state-of-the-art automatic speech recognition techniques. Our best system built for the CHiME-4 challenge consists on the combination of two different sub-systems in order to deal with the variety of acoustic conditions. Each sub-system in turn, follows a hybrid approach with different acoustic models, such as Deep Neural Networks or BLSTM Networks.

1. Introduction

The CHiME Speech Separation and Recognition Challenge [1] encourage participants to develop innovative ASR approaches capable of dealing with challenging noisy environments that rely in speech processing, signal separation or machine learning. It is based on the Wall Street Journal corpus sentences, spoken by talkers located in real noisy environments, such as in a street junction, on the bus, or in a pedestrian area. All the audios have been recorded using a common 6-channel tablet microphone array.

In previous years, the challenge consisted of obtaining the best possible transcription from the 6 channels simultaneously, but given the successful results achieved, this year the challenge proposes two more tracks: 1-channel and 2-channels tracks. Each track only differs in the number of available channels for testing. Thus, the 6-channels track is the easiest since more favorable audio enhancement techniques can be applied. In the case of the 1-channel and 2-channels tracks, the audio enhancement techniques cannot exploit channel information at all which makes this tasks harder to deal with.

The MLLP CHiME-4 system has been developed focusing on the acoustic modeling aspect. Specifically, two different acoustic models have been trained following the hybrid approach. On the one hand, a Context-Dependent Deep Neural Network Hidden Markov Model (CD-DNN-HMM) and on the other hand, a Bidirectional Long Short Term Memory Neural Network (BLSTM). Both acoustic models will be trained on the same data and their output combined. From the proposed three tracks, this global back-end system have been tested in the 1-channel and 2-channel tracks.

The rest of this work is divided as follows. Section 2 describes the ASR toolkit used for the experiments. In Section 3 the proposed system is described and the conclusions are given in section 5.

2. The TransLectures-UPV Toolkit

The MLLP CHiME-4 system has been developed using the transLectures-UPV Toolkit (TLK) [2]. TLK comprises a set of tools for audio processing, feature extraction, HMM and DNN training and decoding. The main latest features added to the toolkit are the following:

- Multilingual and Convolutional NNs.
- Different DNN speaker adaptation techniques: output-feature discriminant linear regression (oDLR) [3] or Kullback-Leibler Divergence based [4].
- DNN sequence discriminative training based on Maximum Mutual Information (MMI).
- Online decoding.
- Gammatone feature extraction.

TLK has demonstrated to provide competitive results in challenging and well-known tasks. In [5] the TLK-based system dealt with TED video lectures, and in [6] the TLK system provided good results in the LibriSpeech [7] corpus.

3. Proposed System

The system proposed by the MLLP group is based on the TLK toolkit. It is composed of two transcription sub-systems that are combined following a recognizer output voting error reduction (ROVER). Each of those sub-systems are based on the HMM-NN hybrid approach. The only difference is that for the first sub-system a classical DNN is used whereas for the second sub-system a BLSTM NN is employed.

Each of those sub-systems perform a three step recognition process as can be observed in Fig. 1. The first and second steps are shown in the upper box. Regarding the first step, it is shared between both sub-systems, cepstral mean and variance normalization (CMVN) is applied and the decoding is performed using a standard DNN which provides the best possible transcription and a better feature-space Maximum Likelihood Linear Regression (fMLLR) transform. For the second step, each sub-system makes use of their own acoustic model (DNN or BLSTM) taking as input the transformed fMLLR features. The output of this system is used to perform a final third-pass recognition (the lower box of Fig. 1). During this step, an unsupervised speaker adaptation technique is applied to both, the DNN and the BLSTM. Specifically, the technique used in this work consisted of a conservative training approach using a very small learning rate and early stopping [4]. This means that a very small learning rate is estimated for a fixed number of epochs as to minimize the Word Error Rate (WER) and then this learning rate is used in evaluation. To the best of our knowledge, it is the first time that this kind of technique is applied to BLSTM NNs for acoustic modeling.

TLK allows to perform decoding efficiently with large vocabulary language models applying pruning techniques: beam search, histogram pruning, word end pruning and look-ahead. Thus, the provided 5-gram language model has been used to obtain the recognition outputs along all the steps. Once the last step is performed, the output lattices are re-scored using also the provided RNN-based language model.

BLSTM NNs have been built using TensorFlow [8]. With this purpose, a new feature has been added to TLK for decoding using TensorFlow-based graphs.

4. Experimental evaluation

The data used for training the acoustic models belong to the multi-condition training set defined by the CHiME-4 challenge. In our case, all data from channels 1,3,4,5 and 6 have been used to train the DNN and the BLSTM sub-systems.

Regarding feature extraction, classical Mel-frequency cepstral coefficients (MFCC) were extracted with a Hamming window of 25 ms. shifted at 10 ms. intervals. This MFCC features consisted of 16 MFCCs and their first and second derivatives (48-dimensional feature vectors). The resulting feature vectors were then normalized by mean and variance at speaker level. And after that, a single fMLLR transform for each training speaker was then estimated and applied to perform speaker-adaptive training (SAT).

In order to train the DNN and BLSTM based acoustic models, we first trained a basic context dependent triphone HMM model up to 64 component Gaussian mixtures, after which a second-pass fMLLR was applied. This model yielded a total of 9079 tied states, estimated following a phonetic decision tree approach.Both models were built on top of these HMM acoustic model. On one hand, the DNN-based acoustic model took as input the fMLLR features with a window size of 11, 5 hidden layers, sigmoid activation functions and an output layer of 9079. It was applied a discriminative pre-training stage and after that, the network was trained as to obtain the best frame accuracy on a validation set. On the other hand, the BLSTM acoustic model was trained with fMLLR input features (without windowing) with 4 hidden layers of 500 units each (both forward and backward directions) and an output layer of 9079. In this case, dropout was applied at the output of each cell with a probability of 0.1, and the Newbob strategy was also applied in order to reduce the learning rate by 0.8 each time the frame accuracy improved less than 3% relative on the validation set. Both networks were trained by minimizing the cross-entropy loss function, following the classical stochastic gradient descent algorithm. This two acoustic models were used for the 1-channel and 2-channels tracks. It is worth mentioning, that in the case of the 2-channel track, the audio enhancement beamformit was applied.

In Table 1 the results after each recognition step from the 1 channel track are shown, and similarly in Table 2 the results from the 2-channels track. As can be observed, the first recognition step is common to both sub-systems and tracks. With respect to the rest of recognition passes, very similar behaviors are observed in both tracks; the DNN performs better in all recognition steps and the BLSTM obtains a huge gain after the third step. For the first statement, we argue that the DNN is far more complex in terms of number of parameters, as we have trained a 5 hidden layer neural network of 2048 units per layer, while the BLSTM consist of 4 hidden layers of 500 units each

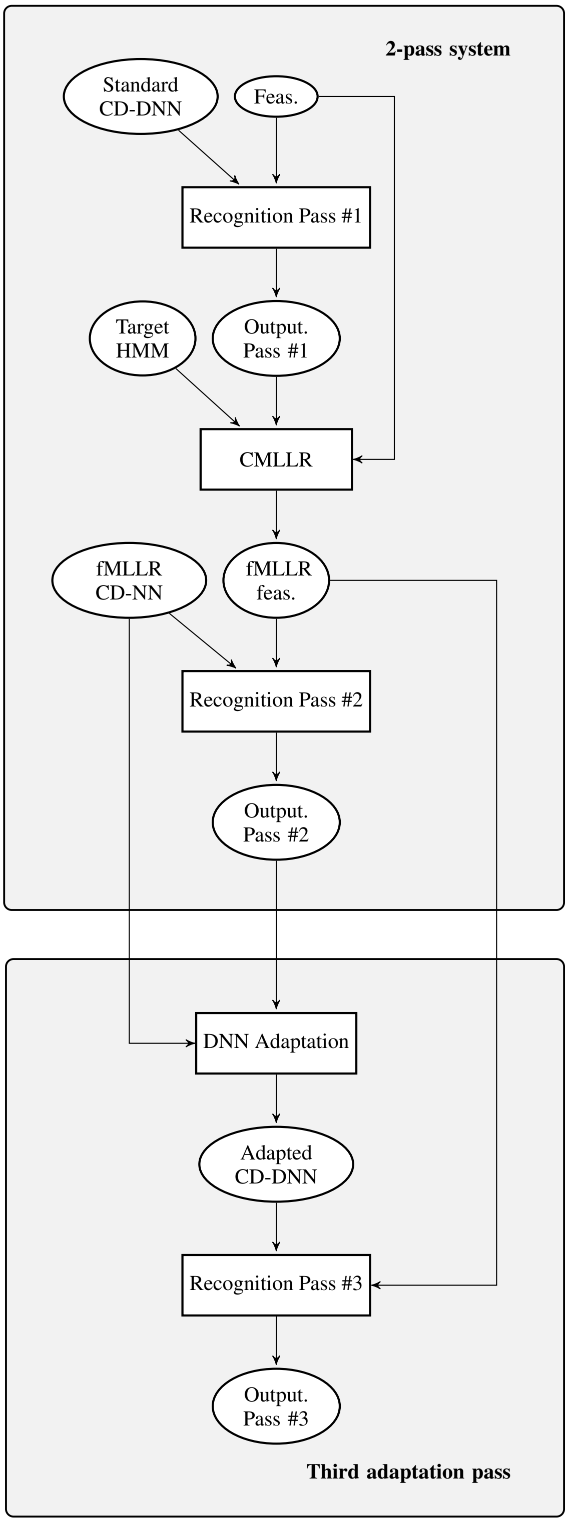


Figure 1: Multi-Pass recognition system with DNN adaptation. Top: 2-pass decoding using fMLLR features. Bottom: Third pass DNN adaptation.

Table 1: WER (%) per step for the 1-channel track.

System	Rec. Pass	Dev		Test	
		real	simu	real	simu
DNN	1	16.03	17.63	24.87	24.47
	2	12.66	14.52	19.80	19.92
	3	11.93	13.19	18.34	17.73
	+RNNLM	10.45	11.98	17.20	16.56
BLSTM	1	16.03	17.63	24.87	24.47
	2	15.10	17.18	23.09	23.56
	3	13.40	14.46	19.30	18.47
	+RNNLM	11.96	12.79	17.78	17.03

Table 2: WER (%) per step for the 2-channels track.

System	Rec. Pass	Dev		Test	
		real	simu	real	simu
DNN	1	13.83	14.35	21.14	20.80
	2	10.39	11.49	16.26	15.75
	3	9.60	10.46	14.77	13.71
	+RNNLM	8.45	9.29	13.71	12.57
BLSTM	1	13.83	14.35	21.14	20.80
	2	12.81	14.22	19.09	19.64
	3	11.63	12.67	15.50	14.93
	+RNNLM	10.12	11.36	14.31	13.46

one. Regarding the second statement, the huge WER improvement from the BLSTM at the third step comes from the fact that we are using the best transcription obtained during the previous step, i. e. the DNN, as to better perform speaker adaptation to the NN during the third step.

Once the output from both systems has been obtained, ROVER technique is applied as to combine both transcriptions. As can be seen in Table 3, the DNN system systematically outperforms the BLSTM-based. However, the combination of both systems yields the best result in both tracks. If we take a look to the real test set, the baseline provided by the organizers for the 1-channel track yielded 23.70% WER points whereas our system obtains 16.11%. This represents 32% relative reduction in WER for the 1-channel track. In the case of the 2-channels track, the baseline system achieved 16.58% average WER whereas our system achieves 12.82%. This represents a 22.7% relative reduction in WER for the 2-channel track. These improvements seems quite competitive, taking into account the simplicity of our system.

Table 4 summarizes the results obtained by the best system per environment. As shown, the most challenging has been the bus environment in all tracks for the real test set. In fact, the baseline system achieved 35.8%, while our system 21.61, which means almost 40% of relative improvement in the 1-channel track. In the case of the 2-channels track, the improvement is about 37% (from 25.37 to 16.00).

5. Conclusions

In this work we have described the MLLP ASR system developed for the CHiME-4 challenge built using TLK. The system is based on the combination of two sub-systems which make use of different acoustic models: DNNs and BLSTMs. The final system obtains 32% and 22.7% relative improvements over the 1-channel and 2-channels tracks compared to the baseline. This represents a good enough result taking into account the simplicity of our approach.

Table 3: Average WER (%) for the tested systems.

Track	System	Dev		Test	
		real	simu	real	simu
1ch	DNN	10.45	11.98	17.20	16.56
	BLSTM	11.96	12.79	17.78	17.03
	Combined	9.95	11.13	16.11	15.72
2ch	DNN	8.45	9.29	13.71	12.57
	BLSTM	10.12	11.36	14.31	13.46
	Combined	7.96	8.93	12.82	12.06

Table 4: WER (%) per environment for the best system.

Track	Envir.	Dev		Test	
		real	simu	real	simu
1ch	BUS	11.74	9.04	21.61	10.95
	CAF	11.18	14.68	18.12	19.57
	PED	7.42	9.35	13.25	15.37
	STR	9.45	11.46	11.47	16.98
2ch	BUS	8.84	7.73	16.00	8.67
	CAF	8.70	11.55	13.78	14.34
	PED	6.27	7.45	11.17	11.77
	STR	8.02	9.00	10.31	13.47

6. Acknowledgments

The work leading to this invention has received funding from the European Union Seventh Framework Programme (FP7/2007- 2013) under grant agreement no 287755 (transLectures). Also, it has received funding from the EU's ICT Policy Support Programme as part of the Competitiveness and Innovation Framework Programme under grant agreement no 621030 (EMMA). In addition, this work has been supported by the Spanish research project MORE TIN2015-68326-R (MINECO/FEDER) and the Spanish Government with the FPU scholarship FPU13/06241.

7. References

[1] E. Vincent, S. Watanabe, A. A. Nugraha, J. Barker, and R. Marxer, "An analysis of environment, microphone and data simulation mismatches in robust speech recognition," *Computer Speech and Language, to appear*, 2016.

[2] M. A. del Agua, A. Giménez, N. Serrano, J. Andrés-Ferrer, J. Civera, A. Sanchis, and A. Juan, "The translectures-upv toolkit," in *Proc. of IberSpeech*, Las Palmas de Gran Canaria (Spain), 2014.

[3] K. Yao, D. Yu, F. Seide, H. Su, L. Deng, and Y. Gong, "Adaptation of context-dependent deep neural networks for automatic speech recognition," in *Proc. of the SLT*, 2012, pp. 366–369.

[4] D. Yu, K. Yao, H. Su, G. Li, and F. Seide, "Kl-divergence regularized deep neural network adaptation for improved large vocabulary speech recognition," in *Proc. of the ICASSP*, 2013, pp. 7893–7897.

[5] M. A. del Agua, A. Martínez-Villaronga, S. Piqueras, A. Giménez, A. Sanchis, J. Civera, and A. Juan, "The mllp asr systems for iwslt 2015," in *Proc. of 12th IWSLT*, Da Nang (Vietnam), 2015.

[6] M. A. del Agua, S. Piqueras, A. Giménez, A. Sanchis, J. Civera, and A. Juan, "Asr confidence estimation with speaker-adapted recurrent neural networks," in *Proc. of InterSpeech*, San Francisco (USA), 2016, in press.

[7] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *ICASSP*. IEEE, 2015, pp. 5206–5210.

[8] M. Abadi and et al, "TensorFlow: Large-scale machine learning on heterogeneous systems," 2015, software available from tensorflow.org. [Online]. Available: http://tensorflow.org/

Multi-channel Speech Enhancement Based on Deep Stacking Network

Hui Zhang, Xueliang Zhang, Guanglai Gao

Department of Computer Science, Inner Mongolia University, Hohhot, China, 010021

alzhu.san@163.com, {cszxl,csggl}@imu.edu.cn

Abstract

Beamforming enhances sound components coming from a direction specified by a steering vector. Some beamforming methods use the time-frequency masks for the steering vector estimation. Better masks lead to better beamforming results. Meanwhile, the beamforming results carry cross-channel information which make the mask estimation easier. Therefore, the beamforming and the mask estimation can boost each other, and can be treated as a "chicken-and-egg" problem. In this work, we embed the beamforming and the mask estimation into a deep stacking network architecture as the speech separation front-end. Together with the state-of-the-art speech recognition back-end, the proposed method obtains 11.00% and 6.00% WER for the real test data in the 4th CHiME Challenge 2 channels and 6 channels tracks.

1. Background

This paper introduces the speech separation and recognition system designed for the 4th CHiME Challenge [1] 2 channels and 6 channels tracks.

From the review of the last CHiME Challenge, we find that the success is mostly relative to the time-varying minimum variance distortionless response (MVDR) beamforming [2].

A beamformer enhances the sound components coming from a direction which specified by a steering vector. The accurate steering vector estimation is the key to effective beamforming. Recently, a beamforming method was proposed that uses the time-frequency masks to estimate the steering vector [3], where the masks represent the probabilities of background noise dominating the corresponding time-frequency points. In this method, the accurate mask estimation is the key to effective steering vector estimation. Better mask estimations lead to better steering vector estimations and better beamforming results. Mask estimation is helpful for the beamforming. Beamforming is also helpful for the mask estimation. The beamforming results are built from multi-channel microphone array, so that they contain cross-channel information which is useful for the mask estimation of a certain single channel.

Because the beamforming and the mask estimation can boost each other, they can be treated as a "chicken-and-egg" problem. In [4], the authors proposed using the deep stacking network (DSN) architecture to solve the "chicken-and-egg" problem. In DSN, each basic module is used to process a "chicken-and-egg" step. DSN stacks these basic processing modules to build forward deep architectures. With the increasing of the number of stacked modules, the system's performance is improving. We consider the mask estimation and beamforming as a "chicken-and-egg" step, process them with a basic module, and embed them into a DSN to form the speech separation front-end. Specifically, we first obtain the estimated masks from a basic module. Then these estimated masks are used to perform the beamforming. Next these beamforming results are used to obtain new estimated masks by another basic module. Then these new estimated masks are used for beamforming, estimating new masks, and so on.

2. Contributions

2.1. Mask Estimation

Before getting any beamforming results, we need a initial mask estimation. We use deep neural network (DNN) as a basic module to estimate the ideal ratio mask (IRM):

$$IRM = \sqrt{\frac{|STFT^{\{speech\}}|^2}{|STFT^{\{speech\}}|^2 + |STFT^{\{noise\}}|^2}} \qquad (1)$$

where $|STFT^{\{speech\}}|$ and $|STFT^{\{noise\}}|$ is the short time Fourier transform (STFT) features of the premixed speech and noise. We obtain the STFT features by applying 320-point Fourier transform on each hamming window frame which length of 20-ms and shift with 10-ms, and using the absolute value of the first 161-D Fourier coefficients.

The DNN contains three 1024-node ReLU hidden layers, and the output transform is sigmoid. The inputs of the DNN is the STFT features of the mixtures. Before feeding into the DNN, the STFT features are compressed by a cubic root operation. The input features also contain a context window of previous 2 and subsequent 2 frames. Therefore, the input is a $161 \times 5 = 805$ dimensional vector.

The DNN is trained with all of the simulated training data with early stop controlled by a 10% left out develop set.

2.2. Beamforming

After obtaining the estimated mask, we get the beamforming results using the the time-frequency mask based MVDR beamforming method [3], where the masks represent the probabilities of background noise dominating the corresponding time-frequency points. We obtain this mask base on the estimated IRM:

$$mask = 1 - max\{IRM_1, \ldots, IRM_N\} \qquad (2)$$

where IRM_i is estimated IRM in channel i. N is number of channels. For 2 channels track, $N = 2$, and for 6 channels track, $N \leq 5$, where we drop the backward channel 2, and remove failed channels with the scripts offered by the official baseline.

2.3. Mask Estimation with Beamforming

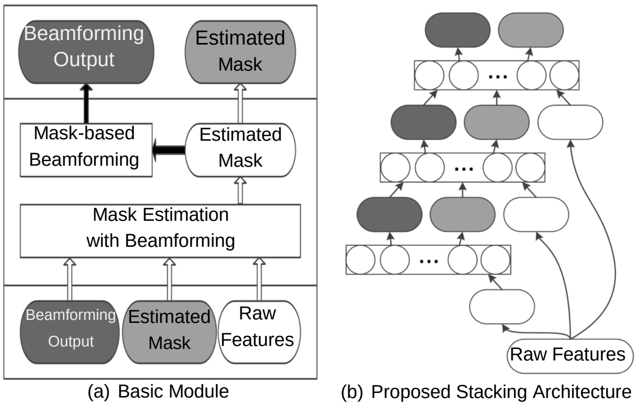
After getting the beamforming results, we can use them to improve the mask estimation. In this step we use another DNN basic module to estimate the IRM. The DNN's structure is same as the one in Sect. 2.1 except the inputs. The inputs of the DNN contain three parts: the estimated IRM from the last DNN module, the STFT features of the mixtures, and the STFT features of the corresponding beamforming results. These beamforming results may contribute to the improvement of the mask estimation. Before feeding into the DNN, all of the STFT features are compressed by a cubic root operation. All of the STFT features are extended with its previous and subsequent 1 frames as context. Therefore, the input is a $161 + 161 \times 3 + 161 \times 3 = 1127$ dimensional vector.

We use the same DNN for the 2 channels and 6 channels tracks. The beamforming results used for training are generated as follows. We first divide the simulated training utterances randomly into two sets whose size are almost the same. One part for the 6 channels track, and another for the 2 channels track. In the one for 2 channels track, we further pick 2 channels randomly for each utterance, and remove others. Then the beamforming results are generated from these two sets.

The DNN is trained with all of the simulated training data with early stop controlled by a 10% left out develop set.

2.4. Combining Mask Estimation and Beamforming

We perform the mask estimation and beamforming alternantly and iteratively by embedding them into a DSN, where we stack basic modules one by one, and as illustrated in Fig. 1. We first obtain the initial estimated IRM by the module described in Sec. 2.1. Then we get the beamforming results as described in Sec. 2.2. Next the beamforming results are used for updating the estimated IRM by the module described in Sec. 2.3. Then these updated estimated masks are used for beamforming, estimating new masks, and so on.



(a) Basic Module (b) Proposed Stacking Architecture

Figure 1: Schematic diagram of the proposed system.

2.5. ASR Back-end

We can further improve the performance of ASR systems by increasing the amount of training data, so that we use scripts offered by the official baseline to train a new ASR back-end with all of the 6 channels training data.

3. Experimental evaluation

In the official baseline, four types of ASR back-ends are involved, which are GMM-based (denoted as "GMM"), DNN-based (denoted as "DNN"), DNN-based with a larger language model (denoted as "5kng") and DNN-based with RNN-based language model (denoted as "RNNML"). We report the results using all of these four ASR back-ends, and compare the proposed system with the official baseline front-end "BeamformIt" system. The proposed front-end is named as "model-N", where N indicates the number of the stacked modules. The average WER of all systems with the baseline ASR back-end and with the new ASR back-end in Tab. 1 and Tab. 3. And the detail WER of the best system are given for each noisy environment in Tab. 2 and Tab. 4.

Compared the Tab. 3 with Tab. 1, we can see that the new ASR back-end can generate better ASR results than the baseline ASR back-end. From Tab. 1, compared with the 6 channel the model-1 system with RNNML ASR back-end and the one reported in [3], the WER in the real test data is 7.44 and 8.86, respectively. It indicates that the DNN is powerful than the complex Gaussian mixture model (CGMM) used in [3] for mask estimation. And compared among the proposed system with different numbers of the stacked modules, we find the performance of the system is improving with the increasing of the number of stacked modules. In addition, the single channel signal and the corresponding beamforming result are often mismatch in the time axis. The experimental results show that the mask estimation can benefit from the beamforming results although the inputs do not match strictly.

4. Conclusion

Because mask estimation and beamforming can boost each others, we treat them as a "chicken-and-egg" problem, and iterate them alternatingly in a DSN. The experimental results show that the proposed method can improve the ASR performance in noisy environment, and the performance of the system is improving with the increasing of the number of stacked modules. The proposed method obtains a comparable performance without any advanced language model or speaker adaptation which are the primary weapons of other participators.

5. Acknowledgments

This research was supported in part by the China national nature science foundation (No. 61365006, No. 61263037).

6. References

[1] E. Vincent, S. Watanabe, A. A. Nugraha, J. Barker, and R. Marxer, "An analysis of environment, microphone and data simulation mismatches in robust speech recognition," *Computer Speech and Language*, to appear.

[2] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third CHiME speech separation and recognition challenge: Dataset, task and baselines," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, Dec 2015, pp. 504–511.

[3] T. Higuchi, N. Ito, T. Yoshioka, and T. Nakatani, "Robust MVDR beamforming using time-frequency masks for online/offline ASR in noise," in *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2016, pp. 5210–5214.

[4] X. Zhang, H. Zhang, S. Nie, G. Gao, and W. Liu, "A pairwise algorithm using the deep stacking network for speech separation and pitch estimation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 6, pp. 1066–1078, June 2016.

Table 1: Average WER (%) for the tested systems with baseline ASR back-end.

Track	System		Dev		Test	
			real	simu	real	simu
2ch	GMM	BeamformIt	16.23	19.14	29.05	27.56
		model-1	14.53	16.25	24.49	19.50
		model-2	14.47	15.97	23.84	19.10
		model-3	14.61	15.83	24.47	19.73
	DNN	BeamformIt	10.90	12.36	20.44	19.03
		model-1	9.29	10.03	17.39	12.86
		model-2	9.08	9.89	16.58	12.91
		model-3	9.04	9.91	16.80	12.72
	5kng	BeamformIt	9.63	10.72	18.08	16.88
		model-1	7.77	8.65	15.07	10.68
		model-2	7.71	8.53	14.27	10.62
		model-3	7.74	8.63	14.38	10.71
	RNNML	BeamformIt	8.23	9.49	16.58	15.34
		model-1	6.74	7.66	13.54	9.46
		model-2	6.57	7.57	12.92	9.55
		model-3	6.57	7.57	12.75	9.37
6ch	GMM	BeamformIt	13.04	14.30	21.83	21.29
		model-1	9.64	10.10	15.08	11.81
		model-2	9.55	10.12	14.53	11.99
		model-3	9.48	10.17	14.48	11.87
	DNN	BeamformIt	8.14	9.07	15.04	14.19
		model-1	6.25	5.96	10.22	7.62
		model-2	6.10	6.08	10.07	7.87
		model-3	6.01	6.20	10.10	8.02
	5kng	BeamformIt	6.85	7.74	13.18	12.33
		model-1	4.91	5.09	8.69	6.17
		model-2	4.82	5.07	8.52	6.29
		model-3	4.91	5.03	8.47	6.60
	RNNML	BeamformIt	5.75	6.77	11.47	10.91
		model-1	4.12	4.20	7.44	5.44
		model-2	3.99	4.41	7.17	5.32
		model-3	4.03	4.42	7.15	5.58

Table 2: WER (%) per environment for the best system with baseline ASR back-end.

Track	Envir.	Dev		Test	
		real	simu	real	simu
2ch	BUS	8.17	6.06	20.15	7.08
	CAF	6.30	10.16	12.07	10.38
	PED	4.60	6.39	9.38	9.54
	STR	7.20	7.67	9.39	10.46
6ch	BUS	5.21	3.89	11.57	4.54
	CAF	3.55	5.06	5.42	5.36
	PED	3.38	3.91	5.64	5.32
	STR	4.00	4.84	5.96	7.10

Table 3: Average WER (%) for the tested systems with new ASR back-end.

Track	System		Dev		Test	
			real	simu	real	simu
2ch	GMM	BeamformIt	15.21	16.86	26.23	25.80
		model-1	13.17	14.76	22.06	18.14
		model-2	12.92	14.46	21.43	17.78
		model-3	12.94	14.45	21.50	17.78
	DNN	BeamformIt	9.52	10.58	17.59	16.94
		model-1	7.99	8.37	14.79	11.02
		model-2	7.79	8.36	14.32	10.84
		model-3	7.83	8.26	14.51	11.09
	5kng	BeamformIt	7.97	8.95	15.31	14.57
		model-1	6.65	6.99	12.86	9.17
		model-2	6.47	7.09	12.37	8.95
		model-3	6.37	7.13	12.36	8.89
	RNNML	BeamformIt	7.01	8.02	13.70	13.28
		model-1	5.58	6.25	11.47	7.99
		model-2	5.48	6.26	11.02	7.80
		model-3	5.56	6.32	11.00	7.80
6ch	GMM	BeamformIt	12.25	12.97	19.99	19.53
		model-1	9.13	9.42	14.13	10.91
		model-2	9.01	9.51	13.47	11.33
		model-3	8.97	9.52	13.62	11.29
	DNN	BeamformIt	7.30	8.27	13.08	12.79
		model-1	5.53	5.30	8.90	6.76
		model-2	5.45	5.21	8.65	7.17
		model-3	5.44	5.27	8.66	7.09
	5kng	BeamformIt	6.04	6.71	11.23	10.95
		model-1	4.44	4.17	7.38	5.24
		model-2	4.25	4.28	7.08	5.39
		model-3	4.28	4.33	6.92	5.59
	RNNML	BeamformIt	5.07	6.08	9.88	9.47
		model-1	3.74	3.56	6.23	4.40
		model-2	3.62	3.65	6.05	4.58
		model-3	3.62	3.66	6.00	4.83

Table 4: WER (%) per environment for the best system with new ASR back-end..

Track	Envir.	Dev		Test	
		real	simu	real	simu
2ch	BUS	7.11	5.31	17.22	5.85
	CAF	5.34	8.48	10.14	9.19
	PED	4.07	5.13	8.05	7.92
	STR	5.71	6.37	8.61	8.24
6ch	BUS	4.59	3.32	9.46	3.88
	CAF	3.08	4.23	4.15	4.87
	PED	3.11	3.17	4.93	4.74
	STR	3.70	3.92	5.47	5.81

CRIM's Speech Recognition System for the 4th CHiME Challenge

Md Jahangir Alam, Vishwa Gupta, Patrick Kenny

[1] Computer Research Institute of Montreal (CRIM), Montreal, Canada

`{jahangir.alam, vishwa.gupta, patrick.kenny}@crim.ca`

Abstract

This paper describes CRIM's contribution to the 4-th CHiME speech separation and recognition challenge. We took part in all the three tracks of the CHiME-4 challenge. Since the focus of this challenge was to address the more difficult 1 channel and 2 channel tasks, we focussed on algorithms that will have the largest impact on these two tasks. We focussed on increasing the training data and on using proven robust features from previous challenges so that they can favorably impact the word error rates (WER) for 1 channel and 2 channel tasks. We enhanced the training data by using the audio from all the microphones (i.e., microphones 1-6) instead of just microphone 5. We also added beamformed data from mic 1, 3-6. We band-limited the above training data to 4 kHz bandwidth and added these to the original training set, thereby doubling the training data. We tried many different robust feature parameters to see which ones actually gave lower WER than the Mel-frequency cepstral coefficients. In all our sub-systems we used the baseline language model and the backend provided by the organizers. Three different robust features actually gave lower WER for the 1 channel task. Combining the recognition outputs of 6 or 7 different features gave the optimal reduction in WER for the 1 channel, 2 channel and 6 channel tasks. Among all the features used in this task the **R**egularized **M**VDR **C**epstral **C**oefficients (RMCC) features performed the best.

Index Terms: 4th CHiME challenge, speech recognition, robust features, RMCC, ROVER, DNN.

1. Introduction

Automatic speech recognition is a key component in hands-free man-machine interaction. State-of-the-art speech recognition systems are based on statistical acoustic models which are trained in a clean and controlled environment. In recent years the use of deep neural network acoustic model and large amount of training data has helped to improve the performance of automatic speech recognition significantly. In many applications, speech recognition systems are deployed in real world scenarios (e.g. cafe, bus station, street, and pedestrian area) where the speech signal is severely distorted by background noise and reverberation. Consequently, the performance of speech recognition systems trained on clean data degrades severely in noisy and reverberant environments because of the mismatch between the training and the test conditions. Therefore, robust speech recognition in real world scenarios has attracted increasing attention in ASR research and development. This attention is due to the widespread use of mobile devices with speech enabled personal assistants.

The fourth edition of CHiME (CHiME-4) challenge, designed to be close to a real world application, provides a common framework for the evaluation and comparison of various approaches for the noise robustness of speech recognition system. Although CHiME-4 challenge revisits the corpora originally collected for CHiME-3, the level of difficulty has been increased by imposing constraint on the number of microphones available for testing. Depending on the number of microphones available for testing CHiME-4 offers three tracks: 1 channel, 2 channel and 6 channel tracks. CHiME-4 corpus is comprised of Wall Street Journal corpus sentences spoken by speakers situated in challenging noisy environments (such as bus, street junction, cafe, and pedestrian area) recorded using a 6-channel tablet based microphone array [1]. A Kaldi-based [2] baseline speech recognizer is provided by the organizers which uses sequence trained deep neural network (DNN) acoustic models and language model (LM) rescoring based on a linear combination of 5-gram LM and RNNLM [3].

In this work we present CRIM's system designed for CHiME-4 challenge tasks and report evaluation results. We took part in all the three tracks of the 4-th CHiME challenge: 1 channel (1ch), 2 channel (2ch), and 6 channel (6ch) tracks. In our contribution we mainly focussed on the robust features extraction and combination of systems based on different frontends using ROVER. In order to reduce the word error rate (WER), we tried many robust features that have performed better in other evaluations of noisy corpus such as the REVERB challenge [4] / AURORA-4 corpus [5], and also features that showed good performance in a speaker recognition task. In addition to the conventional Mel-frequency cepstral coefficients (MFCC) features, we tried the following robust features for speech recognition for CHiME-4 challenge tasks:

- ✓ The regularized MVDR spectrum-based cepstral coefficients (RMCC) [6, 7].
- ✓ Gabor filter-bank feature (GBFB) [8].
- ✓ The ETSI - advanced front end (ETSI-AFE) [9].
- ✓ Infinite impulse response – constant Q transform (IIR-CQT) [10] - based cepstral coefficients (ICQC).
- ✓ The IIR-CQT–based log filterbank (ICQF) features [11].

For the 2ch and 6ch tasks, all our systems employ beamformed speech signals supplied by a weighted delay-and-sum beamforming technique. In two systems we apply beamforming after enhancing the signals using weighted prediction error (WPE)-based dereverberation [12] and Consistent Wiener filtering (CWF)-based audio source separation [13] techniques. We denote those two systems as the WPE-MFCC, CWF-MFCC, respectively. The only difference between the CWF-MFCC and CWF2-MFCC systems is in the noise spectrum estimation while performing audio source separation using CWF. CWF-MFCC uses a MMSE-based noise spectrum estimator whereas CWF2-

MFCC utilizes regional statistics-based noise spectrum estimator. The motivation behind using the ICQC and ICQF features is that these features provide good performance in speaker verification and spoofing detection tasks [11]. As mentioned in the abstract, using all the training data (channels 1-6) gave significantly lower WER than using just the 5 channels (1, 3-6). Also, band-limiting the training data and adding it to the training data [14] had only a small effect on the WER of the development set. Among all the frontends considered for the CHiME-4 tasks, the **R**egularized **M**VDR **C**epstral **C**oefficients (RMCC) features yielded the lowest WER. Combining results of 6 or 7 different feature-based systems with ROVER (Recognizer Output Voting Error Reduction) [15] gave the lowest WER for all the tasks.

2. CHiME-4 Tasks

The CHiME-4 challenge revisits the CHiME-3 corpora with increased level of difficulty by imposing a constraint on the number of microphones available for testing. CHiME-4 tasks consist of three tracks: 1 channel (1ch), 2 channel (2ch) and 6 channel (6ch) tracks. The 6ch track is based on a subset of the channels of CHiME-3 data. CHiME-4 challenge is designed to be close to the real world applications having real acoustic mix, i.e., speakers speaking in challenging noisy environments such as bus, street junction, cafe, and pedestrian area.

3. Overview of CRIM System

In this section we provide an overview of the CRIM system as presented in fig. 1, for the 1ch, 2ch and 6ch tasks of CHiME-4 challenge. Our main contributions include:

i. We band-limit the training data to 4 kHz bandwidth and include these to the original training set, thereby doubling the training data.
ii. For multi-channel tasks, as a pre-processing step, we apply beamforming to enhance the target speech. This step is same as the baseline system provided by the organizer. In two of our systems we additionally enhance the signals using weighted prediction error (WPE)-based dereverberation [12] and Consistent Wiener filtering (CWF)-based audio source separation [13] techniques and then apply beamforming.
iii. We extract robust features by employing RMCC feature extractor.
iv. We combine different robust-feature-based systems using ROVER.

3.1. Pre-processing

As a pre-processing for 2ch and 6ch tasks we enhance the target speech by using a weighted delay and sum beamforming technique. After selecting a reference signal based on the pair-wise cross-correlation, the time delay between a microphone and the reference is estimated using the GCC-PHAT algorithm. Weights for the *m*-th microphone are estimated from the cross-correlations of the *m*-th microphone with other microphones. Finally beamformed signal $\hat{y}(t)$ is obtained using the estimated delays and microphone weights as

$$\hat{y}(t) = \sum_{m=1}^{M} w_m y_m (t - \tau_m), \tag{1}$$

where *m* is the microphone index, *M* is the total number of microphones, w_m and τ_m are the estimated weights and time delays, respectively and $y_m(t)$ is the *m*-th microphone signal.

Among our systems, one system utilizes weighted prediction error (**WPE**)-based dereverberation to enhance the 1ch, 2ch and 6ch signals. The WPE does dereverberation using a linear time invariant filter and produces *M*-channel outputs from *M*-channel inputs. From the M-channel dereverberated signals (*M* > 1) beamformed signal is obtained using a weighted delay and sum beamforming technique.

Another one of our systems employs a consistent Wiener filtering (CWF)-based audio source separation to enhance the signals. The CWF refers to a time-frequency masking which takes into account the consistency of spectrograms for the computation of true optimal solution to the Wiener filtering problem. In this framework, to estimate noise spectrum we used either a MMSE-based noise spectrum estimator or a regional statistics-based noise spectrum estimator.

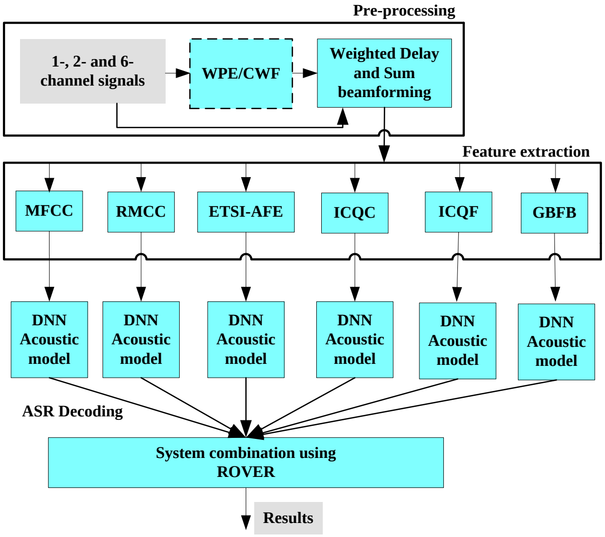


Fig. 1. Schematic diagram of CRIM's system for the 4-th CHiME challenge. Beamforming is applied to the multi-channel signals only. Only two of our systems use weighted prediction error (WPE)-based dereverberation and consistent Wiener filtering (CWF)-based audio source separation (shown with dotted rectangle).

3.2. Extraction of robust features

In this section we describe the robust features used for CHiME-4 challenge tasks.

3.2.1. The ETSI-advanced front-end (ETSI-AFE)

The ETSI-advanced frontend (**ETSI-AFE**) [9] employs a two-stage Wiener filter and blind equalization technique, which is based on the comparison to a flat spectrum and the application of the LMS (Least Mean Squares) algorithm, for improving robustness of ASR systems against additive noise distortions and channel effects.

3.2.2. Gabor filterbank features (GBFB)

The Gabor filterbank (**GBFB)** features [8] are extracted from the log Mel-filterbank spectrum using auditory motivated spectral-temporal 2D filters. These filters were tuned to specific spectro-temporal modulation patterns that occur in speech signals and motivated by the fact that some neurons in the primary auditory cortex of mammals were found to be tuned to very similar spectro-temporal modulation patterns.

3.2.3. IIR-Constant Q transform-based features

The ICQC and ICQF feature representations are derived from the infinite impulse response - constant Q transform by recursively filtering the multi-resolution fast Fourier transform of the signal. We refer to these features by the acronym ICQC for Infinite impulse response Constant Q transform Cepstrum and ICQF for Infinite impulse response Constant Q transform log filterbank features. In order to compute ICQC features we first estimate the IIR-CQT spectra by designing an infinite impulse response (IIR) filterbank that has constant Q behavior. The location of the poles of the IIR filterbank vary for each frequency bin along the real axis in order to make wider window width for lower frequency and narrower for higher frequency. Then a linear time variant (LTV) IIR filter is devised based on the poles of the filterbank. The filter is applied in the forward direction followed by reverse filtering to obtain the IIR-CQT spectrum [10]. The ICQC features, as shown in fig. 3, are obtained by applying discrete cosine transform to the estimated spectrum following logarithmic compression [11].

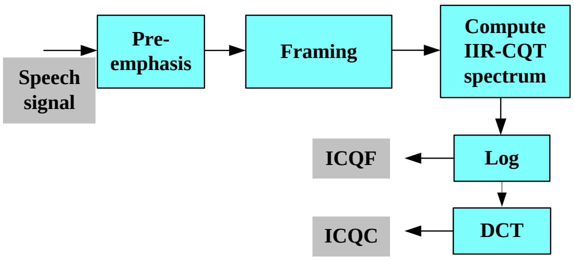


Fig. 2. The ICQC and ICQF feature extraction from the IIR-CQT spectra. Here Q = 13 was chosen.

3.2.4. Regularized MVDR cepstral coefficients

The conventional Mel-frequency cepstral coefficients (MFCC) are usually computed from a DFT-based spectral estimate. When regularized MVDR (RMVDR) spectrum estimator is used to compute the cepstral features instead of the DFT-based spectrum estimator we denote the features as the regularized MVDR cepstral coefficients (RMCC). RMCC was introduced in [6, 7] and evaluated on the AURORA-4 corpus under both clean and multistyle training modes. Here we use RMCC to extract robust features for the CHiMe-4 challenge tasks.

The first step in computing RMCC is to estimate RMVDR spectra. Similar to the MVDR spectrum estimator, the *p*-th order regularized MVDR spectral estimate can be parametrically written as

$$Y_{rmvdr}\left(f\right)=\frac{1}{\sum_{k=-p}^{k=p}\mu_r\left(k\right)e^{-i2\pi fk}}, \qquad (2)$$

where the parameter $\mu_r\left(k\right)$ of the regularized MVDR method can be obtained from a non-iterative computation using the regularized LP (RLP) coefficients a_q^r and the prediction error variance σ_e^r as:

$$\mu_r\left(k\right)=\begin{cases}\frac{1}{\sigma_e^r}\sum_{q=0}^{p-k}\left(p+1-k-2q\right)a_q^r a_{q+k}^{r*}, & \text{for } k\ge 0\\ \mu_r^*\left(-k\right), & \text{for } k<0.\end{cases} \qquad (3)$$

The regularized predictor coefficients a_q^r are computed by adding a penalty measure $\psi\left(a^u\right)$, which is a function of the unknown predictor coefficients a^u, to the objective function of the LP method and therefore, minimizing the modified objective function of the following form [1, 2]

$$\sum_n\left(y\left(n\right)+\sum_{q=1}^{p}a_q y\left(n-q\right)\right)^2+\lambda\psi\left(a^u\right), \qquad (4)$$

Where $s\left(n\right)$ is the current speech sample, regularization constant $\lambda>0$ controls the smoothness of the all-pole spectral envelope. RLP method helps to penalize the rapid changes in all-pole spectral envelope and therefore, produces a smooth spectral estimate keeping the formant positions unaffected [6]. The optimal values chosen for the model order p and regularization constant λ are 100 & 10^{-7}, respectively [6, 7].

After estimating RMVDR spectrum, RMCC features are obtained by integrating Mel-scale filterbank and taking discrete cosine transform following logarithmic compression. Mean and variance normalization is used for feature normalization.

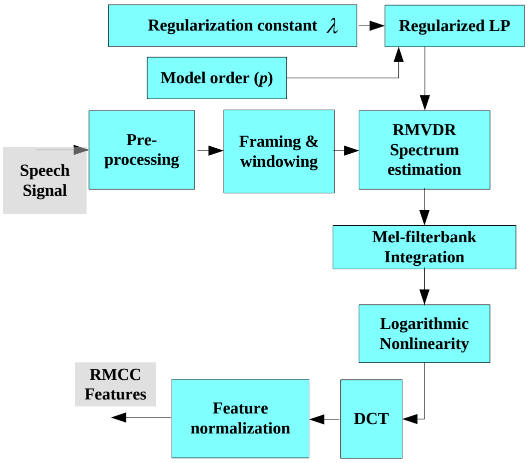


Fig. 3. Regularized MVDR cepstral coefficients (RMCC) feature extraction.

3.3. Backend

The backend of our system is very similar to the default system provided by the challenge organizers. The language models (LM) are the same: the search language model, the 5-gram rescoring LM and the RNNLM are the same. The training process is the same for the features with small dimension. For features with large dimension (like GBFB and ICQF features), the output states are the same as for the MFCC features, but the input to the DNN corresponds to the feature dimension (with +/- 5 frames context). For features with smaller dimension, the initial alignment of the training set with MFCC features is used to train the GMM-HMM sat models for the new features. As mentioned before, the training data consists of all the training data from channels 1-6 and also includes the beamformed training data from channels 1, 3-6. The data is doubled by band-limiting each training audio file to 4 kHz. The training process is the same as provided by the organizers. We discriminatively train one DNN for each feature. For each track, we generate one ctm file for each feature and each set (i.e., development and evaluation). These ctm files are generated after rescoring with 5-gram LM followed by RNNLM rescoring.

3.4. Combining systems using ROVER

In this step we combine the ctm files of 6 or 7 systems, obtained in the previous step, using ROVER. As mentioned before, some of the features gave significantly lower WER for the evaluation set for some of the tracks. Combining the results from six or seven different features-based systems reduced the WER even further.

ROVER [15] reduces word error rates for automatic speech recognition systems by exploiting differences in the nature of the errors made by multiple speech recognition systems. It works in two steps:

- ✓ The outputs of several speech recognition systems are first aligned and a single word transcription network (WTN) is built.
- ✓ The best scoring word (with the highest number of votes) at each node is selected. The decision can also incorporate word confidence scores if these are available for all systems [15].

4. Experiments and Evaluation Results

Word error rates (WER) for each feature parameter and for each task are shown in Table 1. As mentioned before, for each feature parameter, we discriminatively train one DNN as provided by the default scripts. The same DNN is used to compute WER for all the tasks. For 1 channel task, there is no beamforming. For 2 channel and 6 channel tasks, the dev and eval sets go through appropriate beamforming using the beamforming software supplied by the organizers. In Table 1, the first row in each task corresponds to the default setup provided by the organizers. We ran the provided scripts and the results correspond to those scripts. The first row only uses channel 5 training data. The 2nd row for each task uses training data from channels 1 through 6 (channel 0 is not used). We also use the training data after beamforming using channels 1, 3, 4, 5, 6. Channel 2 was not used in this beamforming.

From Table 1 we can see that for 1ch task, the RMCC, GBFB and ETS-AFE features (rows 3-5) gave lower WER for the real test set than using the MFCC features (row 2). For 2 channel and 6 channel cases, only RMCC feature gave better results than the MFCC features. We combined results from different features using ROVER. We combined them in the WER order.

Table 1 : Average WER for the tested systems.

Track	System	Dev		Test	
		real	simu	real	simu
1ch	MFCC (5ch)	11.46	13.10	23.08	20.88
	MFCC	9.46	10.65	18.87	16.43
	RMCC	8.46	11.24	15.16	15.83
	GBFB	9.33	12.74	17.61	18.03
	ETSI-AFE	10.02	12.54	17.65	17.01
	ICQF	11.03	15.93	22.12	22.28
	WPE-MFCC	14.02	15.78	28.44	22.87
	ICQC	13.62	19.03	26.06	27.62
	CWF-MFCC	16.39	18.39	31.09	23.70
	CWF2-MFCC	17.40	19.65	32.47	25.46
	ROVER	**6.79**	**9.27**	**12.70**	**13.72**
2ch	MFCC (5ch)	8.39	9.44	16.70	15.16
	MFCC	6.72	7.75	13.77	12.00
	RMCC	6.22	8.29	11.54	11.74
	GBFB	7.29	9.63	13.91	14.52
	ETSI-AFE	8.96	10.95	16.14	14.52
	ICQF	8.48	12.28	18.10	18.13
	WPE-MFCC	10.11	11.11	20.18	17.47
	ICQC	10.39	14.16	21.17	22.39
	CWF-MFCC	13.40	13.82	23.67	19.89
	CWF2-MFCC	12.79	14.31	25.81	21.23
	ROVER	**5.13**	**6.69**	**9.97**	**10.34**
6ch	MFCC (5ch)	6.08	6.82	11.50	10.73
	MFCC	4.86	5.49	9.97	8.75
	RMCC	4.86	5.98	8.65	8.71
	GBFB	5.96	7.40	10.40	10.70
	ETSI-AFE	7.09	8.67	12.42	11.30
	ICQF	6.74	9.41	13.31	13.72
	WPE-MFCC	6.75	7.82	13.54	13.27
	ICQC	8.19	10.16	14.16	16.00
	CWF-MFCC	8.13	9.94	17.09	15.56
	CWF2-MFCC	9.20	11.78	18.71	16.47
	ROVER	**4.00**	**5.07**	**7.23**	**7.53**

Table 2 : WER per environment for the best system.

Track	Envir.	Dev		Test	
		Real	simu	real	simu
1ch	BUS	8.54	7.95	18.75	9.73
	CAF	7.51	12.37	13.80	16.59
	PED	4.68	7.04	9.55	13.73
	STR	6.43	9.71	8.70	14.85
2ch	BUS	6.40	5.66	14.21	7.28
	CAF	5.24	8.63	9.90	12.05
	PED	3.78	5.03	8.20	10.80
	STR	5.10	7.42	7.58	11.23
6ch	BUS	5.24	4.48	9.44	4.97
	CAF	3.95	6.28	6.50	8.11
	PED	2.74	3.86	6.02	7.47
	STR	4.07	5.65	6.95	9.58

For 1ch task, we achieved the best results when we combine following 6 features: RMCC, GBFB, ETSI-AFE, MFCC, ICQF, and ICQC as shown in the last row for 1 channel results. For the 2 channel task, we achieved the best results when we combine 7 different features, namely, RMCC, MFCC, GBFB, ETSI-AFE, ICQF, WPE-MFCC and ICQC. For 6 channel task also, we achieved the best results when we combine the outputs from these 7 different feature parameters in the same order. These results are shown in the last row of each track. Results for each environment after ROVER are shown in Table 2. For 1 channel task, for real test set, we have reduced the WER by 45% (from 23.08% to 12.7%). For 2 channel task, WER has been reduced by 40% (from 16.7% to 9.97%), and for the 6 channel task the WER has been reduced by 37% (from 11.5% to 7.23%).

In table 3 we compared the WER of CRIM's system with the USTC-iFlytek system for CHiME-4 challenge with the lowest WER on the real portion of evaluation set [16]. Since we only used the default LMs, this comparison is with the default LMs for both the systems. Note that in [16], DNN-based single channel speech enhancement was used to enhance the signals, and, besides DNN-based acoustic model, deep convolutional neural networks (DCNN)-based upgraded acoustic models were also used. As we can see from table 3, CRIM's WER for 1ch system is close to the WER for the best CHiME-4 system. The primary reason for this is the noise robust RMCC features.

Table 3: WER comparison of CRIM's system with the best CHiMe-4 system [16] using the baseline (or default) language models on the evaluation set (real only).

Track	Real	
	CRIM	Best system [16]
1ch	12.7	11.15
2ch	9.97	5.41
6ch	7.23	3.24

5. Conclusion and Future Works

We presented automatic speech recognition systems developed at CRIM for the all three tracks (1ch, 2ch and 6ch) of CHiME-4 challenge. We used the same backend and baseline language models provided by the organizer. Therefore, to reduce word error rates (WER) we mainly focussed on the extraction of robust features and on system combination of various robust features-based sub-systems. Compared to the other features the RMCC features provided lowest WERs in all three tracks. By combining multiple hypotheses from different robust features-based systems we were able to reduce WER significantly from the baseline system. For 1ch track, for real test set, the WER was reduced by 45% (from 23.08% to 12.7%). For 2ch track, WER was reduced by 40% (from 16.7% to 9.97%), and for the 6 channel task the WER was reduced by 37% (from 11.5% to 7.23%).

In our future works we intend to keep RMCC features extractor fixed and focus on modifying the acoustic model and language models.

6. Acknowledgements

This work has been made possible by investments from the Ministere de l'economic et exportation (MEIE) of Government du Quebec.

7. References

[1] Emmanuel Vincent, Shinji Watanabe, Jon Barker, and Ricard Marxer, "An analysis of environment, microphone and data simulation mismatches in robust speech recognition," Submitted to Computer Speech and Language, 2016.

[2] Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G. and Vesely, K. "The Kaldi Speech Recognition Toolkit" in proc. of ASRU, pp. 4. Hawaïï, USA, December 2011.

[3] The 4th CHiME speech separation and recognition challenge: http://spandh.dcs.shef.ac.uk/chime_challenge/index.html.

[4] The REVERB challenge: http://reverb2014.dereverberation.com.

[5] N. Parihar, J. Picone, D. Pearce, H.G. Hirsch, "Performance analysis of the Aurora large vocabulary baseline system," Proceedings of the European Signal Processing Conference, Vienna, Austria, 2004.

[6] M. J. Alam, P. Kenny, D. O'Shaughnessy, "Regularized Minimum Variance Distortionless Response-Based Cepstral Features for Robust Continuous Speech Recognition", Speech Communication (2015), vol. 73, pp. 28-46.

[7] M. J. Alam, P. Kenny, P, Dumouchel, D. O'Shaughnessy, "Robust Feature Extractors for Continuous Speech Recognition", Proc. EUSIPCO (2014), Lisbon, Portugal.

[8] Schädler, M. R., Meyer, B. T., and Kollmeier, B., "Spectro-temporal modulation subspace-spanning filter bank features for robust automatic speech recognition", Journal of the Acoustical Society of America (2012), Volume 131 (5), pp. 4134-4151.

[9] ETSI ES 202 050, Speech Processing, Transmission and Quality aspects (STQ), "Distributed speech recognition; advanced front-end feature extraction algorithm; Compression algorithms;" (2003).

[10] P. Cancela, M. Rocamora, E. Lopez, "An efficient multi-resolution spectral transform for music analysis," in proc. of the ISMIR, 2009.

[11] M. J. Alam, P. Kenny, "Low level and high level features for spoofing detection," submitted to IEEE journal of selected topics on Signal Processing (2016), August.

[12] Marc Delcroix, Takuya Yoshioka, Atsunori Ogawa, Yotaro Kubo, Masakiyo Fujimoto, Ito Nobutaka, Keisuke Kinoshita, Miquel Espi, Takaaki Hori, Tomohiro Nakatani, and Atsushi Nakamura, "Linear prediction-based dereverberation with advanced speech enhancement and recognition technologies for the REVERB challenge," in Proc. of the REVERB Workshop (2014).

[13] J. Le Roux and E. Vincent, "Consistent Wiener filtering for audio source separation," IEEE Signal Processing Letters (2013), vol. 20, no. 3, pp. 217–220.

[14] L. Deng, J. Li, J. Huang, K. Yao, D. Yu, F. Seide, M. Seltzer, G. Zweig, X. He, J. Williams, Y. Gong, and A. Acero. "Recent advances of deep learning for speech research at Microsoft," ICASSP, 2013.

[15] J. G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer Output Voting Error Reduction (ROVER)", Proc. ASRU, pp. 347-354, 1997.

[16] Jun Du, Yan-Hui, Lei Sun, Feng Ma, Hai-Kun Wang, Jia Pan, Cong Liu, Chin-Hui Lee, "The USTC – iFlytek System for CHiME-4 Challenge", http://spandh.dcs.shef.ac.uk/chime_workshop/papers/CHiME_2016_paper_21.pdf

Robust Automatic Speech Recognition for the 4th CHiME Challenge Using Copula-based Feature Enhancement

Alireza Bayestehtashk [1], *Izhak Shafran*[2]

[1]Oregon Health & Science University
[2]Google Inc

bayesteh@ohsu.edu, izhak@google.com

Abstract

In this paper, we investigate the application of the copula model for enhancing features in automatic speech recognition task. We compute a set of utterance-specific nonlinear transformations based on the copula model and use these transformations to obtain the enhanced features for every utterance in the dataset. These features improve the performance of the baseline system by about 4.3%, 1.4%, and 0.5% (absolute) respectively for 1-channel, 2-channel and, 6-channel. Further gains were obtained when our system was combined with the baseline system using minimum Bayes risk decoding to achieve 4.3%, 2.4%, and 1.2% absolute WER improvements for the respective channels.

1. Background

Generally, the mismatch between the training and testing conditions degrades the performance of machine learning tasks including automatic speech recognition (ASR). In real-world ASR applications, it is impractical to obtain training data that is representative of wide range of background noise and reverberations under which utterances are spoken, even when training data is modified using additive noise and simulated reverberations such as in multi-style training (MTR). These variations are currently modeled implicitly by the ASR acoustic models, such as deep neural networks (DNNs), recurrent neural networks (RNNs) and Gaussian mixture models (GMMs). The typical input features presented to the acoustic models are the logarithm of the mel-warped frequencies after passing it through a filter bank or mel-warped cepstral coefficient (MFCC).

The strategies to compensate the mismatch between the training and testing can be categorized into model based and feature based methods. The model-based methods attempt to model the variations associated with speech and neglect other variations such as background noise or channel distortion.

Feature mismatch reduction: In this approach features are extracted in a manner that minimizes the effect of additive and convolutional noise. The simplest version of such a normalization is the well-known cepstral mean-variance normalization (CMVN) that removes the convolutional channel noise in the homomorphic cepstral domain. The method assumes that the channel noise varies slowly, a mild assumption that is often true. The key advantage of this feature-based method is that it generalizes remarkably well to test utterances with channels distortions that have never been seen before. Many other feature-based transformations have been developed and investigated, but with limited success. One such previously developed approach shares the same motivation as our work [1]. They learn a coarse transformation so that the histogram of their test features matches those of their training features.

These approaches are *ad hoc* in that they treat each feature component independently and do not take into account the joint distribution of the feature vector. Moreover, they do not consider the influence of the transformation in computing the likelihood of the input signal. Copula models provide a principled approach for decoupling the marginal distributions from the component that models the interaction between the random variables. As such, they are well-suited to address the effect of the mismatch between the train and test set. In our previous study [2], we showed that the CMVN and histogram equalization are two special cases of copula-based models.

In state-of-the-art ASR systems, CMVN is the only feature processing used to address mismatch between the training and testing condition. This assumes that components of input feature vectors are statistically independent, which is typically a poor assumption. In the section below, we propose a method to avoid this assumption and address the mismatch using a very flexible multivariate distribution – the multivariate copula model.

2. The Multivariate Copula Model

The standard multivariate distribution estimation methods such as GMM entirely focus on choosing a parametric form for the joint distribution of the variables. The choice of joint distribution automatically dictates a specific form for marginal distributions,which may not be appropriate for a given application or data. It would be convenient if the choice of suitable marginal distribution is decoupled from that of the joint distribution. Sklar's theorem provides the necessary theoretical foundation to decouple these choices. The theory formally states that any joint distribution can be uniquely factorized into its univariate marginal distributions and a Copula distribution. The Copula distribution is a joint distribution with uniform marginal distributions on the interval $[0, 1]$:

$$f(X) = c(F_1(x_1), F_2(x_2), \dots, F_n(n))\Pi_{i=1}^{n} f_i(x_i) \qquad (1)$$

where $\{f_i(x_i)\}_{i=1}^{n}$ are the marginal density functions of f, $\{F_i(x_i)\}_{i=1}^{n}$ their corresponding marginal cumulative distribution functions, and $c(\cdot)$ is the Copula density function.

Equation (1) shows that any continuous density function can be constructed by combining a Copula density function and a set of marginal density functions.

Gaussian Copula model: Gaussian Copula density function is the most common multivariate Copula function:

$$c_{gaus}(U; R) = \frac{1}{|R|^{\frac{1}{2}}} \exp\{-\frac{1}{2}U^T(R^{-1} - I)U\} \qquad (2)$$

where R is the correlation matrix.

The Gaussian Copula model can be constructed by substituting the Gaussian Copula density function into Equation (1):

$$f(X; R, \Lambda) = c_{gaus}(U; R) \prod_{i=1}^{n} f_i(x_i; \lambda_i) \tag{3}$$

where $u_i = \Phi^{-1}(F_i(x_i))$ and Φ^{-1} is the quantile function of standard univariate normal distribution.

The main difference between the Gaussian Copula model in Equation (3), and standard Gaussian distribution is that the marginal density functions in the Gaussian distribution are necessarily Gaussian while the marginal density functions of the Gaussian Copula model can by any continuous density function and this capability makes the Gaussian Copula model more flexible than the Gaussian distribution.

In our previous work, we have shown how to compute the optimal feature transformation to minimize the KL distance between two multivariate Gaussian copula distributions [2].

3. Experimental Setup & Results

Akin to speaker adapted training, we estimate the acoustic models in 3 stages: (a) estimate a canonical multivariate copula distribution of the 13-dim MFCC features using all the utterances in the single channel noisy training data; (b) transform each utterance in the training data to reduce the KL distance between the multivariate distribution of the given utterance and the canonical distribution; and (c) train a standard acoustic model in the transformed feature space. At test time, we transform the features of each utterance to the canonical multivariate copula distribution space before decoding.

Compared to the performance of the baseline system [3], tabulated in Table 1 for different conditions, our copula-based system, in Table 2 shows significant improvement in several conditions, but not all. Note, 5gkn stands for 5-gram Knesser-Ney smoothed LM provided with the baseline system. The gains are particularly remarkable in single channel input for which it is well-suited. Note, we haven't applied any special processing for multi-channel case and hence didn't expect gains there. The gains are highest in bus background noise and our hypothesis is that there is more structure and correlation in the noise in this case for which the multivariate copula is an apt representation. We expect applying copula-based feature enhancement to give further improvements when it is applied to frequency spectrum before the filterbank and MFCC where the noise components can be modeled in a fine grained manner. Finally, our copula-based system is sufficiently different from the baseline system that we are able to obtain additional gain through system combination using MBR, as reported in Table 3.

4. References

[1] F. Hilger and H. Ney, "Quantile based histogram equalization for noise robust large vocabulary speech recognition," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 14, no. 3, pp. 845–854, 2006.

[2] A. Bayestehtashk, I. Shafran, and A. Babaeian, "Robust speech recognition using multivariate copula models," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 5890–5894.

[3] E. Vincent, S. Watanabe, A. A. Nugraha, J. Barker, and R. Marxer, "An analysis of environment, microphone and data simulation mismatches in robust speech recognition," *Computer Speech and Language*, to appear.

Table 1: Average WERs of the baseline systems trained on single channel data.

Track	System	Dev		Test	
		simu	real	simu	real
1ch	DNN	17.4	16.5	26.0	30.0
	smbr	15.8	14.6	24.0	27.1
	smbr+5gkn	13.9	12.3	22.1	24.3
	smbr+rnn	12.8	11.5	20.8	22.9
2ch	GMM	18.7	16.3	27.3	28.7
	DNN	13.5	12.2	20.4	22.4
	smbr	12.1	10.8	18.8	20.0
	smbr+5gkn	10.7	9.6	16.4	17.6
	smbr+rnn	9.3	8.4	15.2	16.2
6ch	GMM	14.2	12.7	21.1	21.7
	DNN	10.1	9.5	15.9	16.6
	smbr	9.0	8.2	14.2	14.7
	smbr+5gkn	7.8	7.0	12.1	12.8
	smbr+rnn	6.7	6.0	10.9	11.3

Table 2: Average WERs of the baseline systems trained on single channel features after copula-based transformation.

Track	System	Dev		Test	
		simu	real	simu	real
1ch	GMM	23.0	19.8	30.0	29.4
	DNN	17.6	15.4	24.9	24.4
	smbr	16.5	13.9	23.5	23.1
	smbr+5gkn	14.7	12.1	21.7	20.1
	smbr+rnn	13.2	10.7	20.4	18.6
	copula+baseline	12.1	9.8	19.2	18.6
2ch	GMM	18.1	15.2	24.9	24.4
	DNN	13.9	12.1	20.4	19.8
	smbr	12.7	10.7	19.1	18.2
	smbr+5gkn	10.9	9.1	17.2	16.4
	smbr+rnn	9.6	8.0	15.6	14.8
	copula+baseline	8.8	7.3	13.9	13.8
6ch	GMM	14.4	12.5	19.7	19.3
	DNN	10.8	9.6	16.0	15.4
	smbr	9.8	8.2	15.2	14.5
	smbr+5gkn	8.2	7.1	13.0	12.2
	smbr+rnn	7.1	6.1	11.7	10.8
	copula+baseline	6.3	5.4	10.1	10.1

Table 3: Average WERs after combining the baseline and copula-based system using MBR decoding.

Track	Envir.	Dev		Test	
		simu	real	simu	real
1ch	BUS	10.3	12.6	13.8	26.0
	CAF	15.7	10.5	23.5	20.8
	PED	9.3	6.6	18.8	15.7
	STR	12.9	9.6	20.6	11.9
2ch	bus	7.2	9.2	10.0	19.4
	CAF	11.8	7.5	16.2	14.1
	PED	6.9	4.9	14.2	12.0
	STR	9.1	7.7	15.2	9.7
6ch	bus	5.3	6.8	6.7	13.3
	CAF	7.7	5.1	11.2	9.5
	PED	5.1	3.9	10.0	8.5
	STR	7.2	5.7	12.5	9.1

The SJTU CHiME-4 system: Acoustic Noise Robustness for Real Single or Multiple Microphone Scenarios

Yanmin Qian *Tian Tan*

Key Lab. of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering
SpeechLab, Department of Computer Science and Engineering, Shanghai Jiao Tong University, China
{yanminqian,tantian}@sjtu.edu.cn

Abstract

Noise robust speech recognition is one of the most challenging problems. This paper described the most important technical designs in the SJTU CHiME-4 Challenge system covering data usage, feature normalization, advanced acoustic model, auxiliary feature joint training, multi-model joint decoding and multi-pass decoding pipeline. The impacts on the final recognition accuracy from each technology are explored and compared. With the proposed technologies, our final system obtains a very large improvement compared to the formal released baseline system. The final average WERs of the real test set are 6.41%, 9.14%, 13.91% for 6-channel, 2-channel, and 1-channel, respectively.

1. Background

This paper describes the key points and contributions of the SJTU system (Shanghai Jiao Tong University) for the 4th CHiME Challenge [1]. We participate in all the evaluations for the challenge, including 6-ch / 2-ch / 1-ch tracks. Our works mainly focus on the acoustic modeling, so the front-end we used is the released baseline BeamformIt, the language model is the baseline RNNLM. In comparison to CHiME-3 challenge, our new progress mainly includes:

- Data augmentation using all channels with the beamformed data
- Feature normalization
- Advanced acoustic model including very deep CNN [2] and auxiliary feature joint training [3]
- System combination using the multi-model joint decoding and multi-pass decoding pipeline.

In the next section, we will describe these key technologies in detail.

2. Contributions

2.1. Data usage

Compared to the released baseline only using 18 hours noisy training data from channel 5, the training set is augmented with data from all channels (excluding the channel 2 located at the back of the device), and moreover the beamformed audio stream on these channels is also pooled together, which totally results 6×18=108 hours for training.

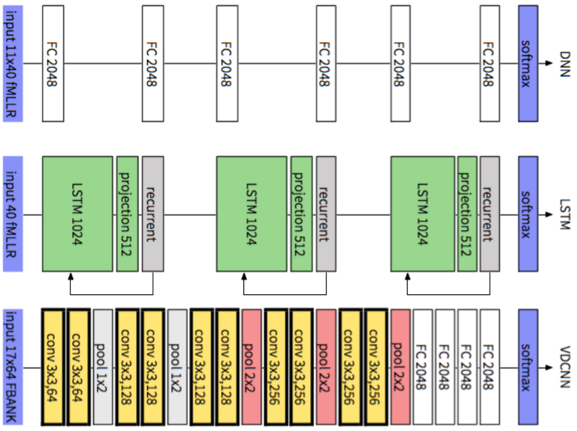


Figure 1: Model structures and configs used in our systems

2.2. Feature normalization

The appropriate feature normalization is very important for speech recognition in noisy scenarios. It can make the system more robust to the changes in environments and channels. CMN, CVN and CMVN are compared with FBANK, and the FBANK with CMN on per speaker shows the best performance.

2.3. Advanced acoustic models

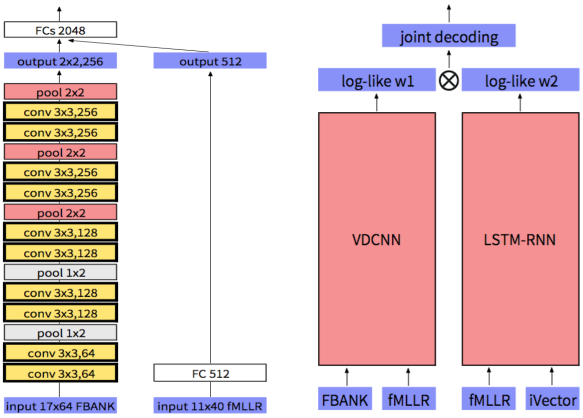
In addition to the basic DNN model, which is used in the released baseline, other advanced models are applied. One is named very deep CNN (VDCNN), which is proposed in our recent work [2, 4, 5], and particularly it shows the powerful potentiality on noise robustness [2]. Another is LSTM-RNN, and it has been verified effective on several tasks [6]. The model structures and configurations used in this work are illustrated in Figure 1, and more details can be referred to the work in [2].

2.4. Joint training with auxiliary features

The use of auxiliary features in factor-aware training is one type of adaptation popular for robust ASR [3, 7, 8, 9]. We use the same framework as our previous work for LSTM-RNN based speaker-aware training using i-vector [8], which concatenating the auxiliary feature with the original feature at the input layer.

In contrast, for the VDCNN usage, [5] proposed another auxiliary feature joint training architecture shown as the left part of Figure 2. Considering the auxiliary features, such as fMLLR and i-vector, are the non-topographical, they are separately

Thank Mr. Mengxiao Bi for the support on the computational cluster platform. This work was supported by Shanghai Sailing Program No. 16YF1405300, Program for Professor of Special Appointment (Eastern Scholar) at Shanghai Institutions of Higher Learning, China NSFC projects (No. 61573241 and No. 61603252) and Interdisciplinary Program (14JCZ03) of Shanghai Jiao Tong University in China.



(a) Joint training of VDCNNs with auxiliary features

(b) Joint decoding of VDCNN & RNN

Figure 2: The architectures of VDCNN with auxiliary features joint training, and VDCNN & RNN joint decoding

transformed with a normal fully-connected layer first, and then the outputs are concatenated with those of the VDCNN block to be fed into the following shared MLP layers. Both fMLLR and i-vector can be used as auxiliary features for VDCNNs here.

2.5. Joint decoding with VDCNN and RNN

To explore the huge complementarity within VDCNN and LSTM-RNN, a joint decoding scheme shown as the right part of Figure 2 is implemented [5, 10]. It uses a weighted sum combination of acoustic log likelihoods from VDCNN and LSTM-RNN systems. Moreover, the DNN system also can be added into this framework to perform the multi-model (three) joint decoding.

2.6. Final multi-pass decoding system

Embedded with these above key features, our final submitted system is based on a multi-pass decoding framework, which is illustrated as Figure 3. It consists of 5 stages, shown as P1∼P5.

- **P1:** The front-end audio processing, including beamforming for multi-channel condition and feature extraction. In the 1-ch track, the single channel audio is used to extract all types of features directly.
- **P2:** Speaker-independent acoustic models are built individually, including DNN, VDCNN & LSTM-RNN. and auxiliary features based modeling are also constructed.
- **P3:** The DNN-SI system is adapted by the 2-pass mode, which uses 1-best from the first pass SI model. Then the 1-best from the adapted DNN-SA model is used to do the cross-adaptation for VDCNN and LSTM-RNN, named VDCNN-SA and LSTM-RNN-SA respectively.
- **P4:** Three speaker-adaptation models, including DNN-SA, VDCNN-SA and LSTM-RNN-SA are integrated to perform the proposed multi-model joint-decoding.
- **P5:** The RNNLM rescoring is applied on the lattices from the P4 stage to get the final results of the fusion system. If only considering the best single system, the lattices from VDCNN-SA in P3 are applied with RNNLM rescoring to generate the best single system results.

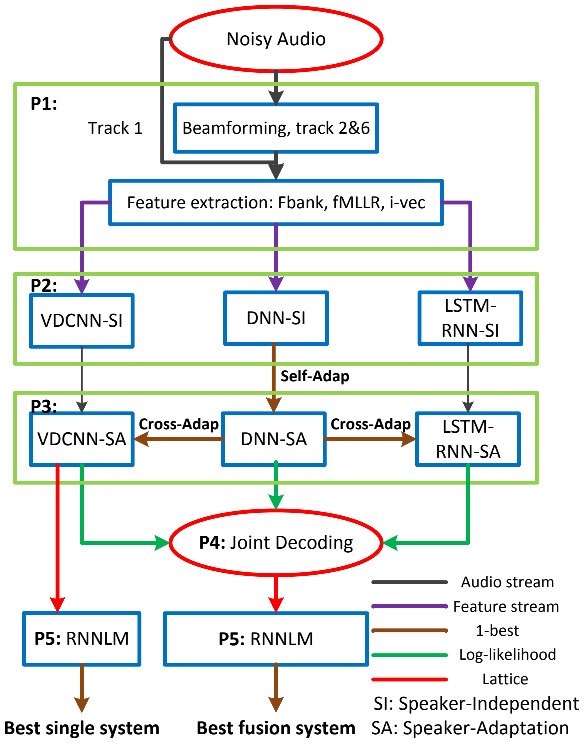


Figure 3: The multi-pass decoding for the CHiME4-Challenge

3. Experimental evaluation

The detailed results comparison in our system will be described in this section. The GMM-HMM system was trained using the released standard Kaldi [11] recipe. It is a MFCC-LDA-MLLT-FMLLR GMM-HMMs system. After that, a forced-alignment is performed to get the state level labels for NN training. In this work, all the DNN models are constructed using Kaldi [11], and other models are built using CNTK [12]. It is noted that except the results in Table 4 which used SMBR training and RNNLM rescoring, all the results in other tables used the CE criterion in training and the released trigram in decoding.

3.1. Data augmentation

Data augmentation was first evaluated, different amount of data, described in Section 2.1, were compared. In this experiment, DNN systems with fMLLR feature were used. As shown in Table 1, using more data always get better performance. For the fast investigation on the other system configuration, only the beamformed audio stream was used in training first (18 hours) in the following experiments, and the final submitted system will be retrained using all 108 hours data.

Table 1: WER (%) comparison of different training data usages for the 6ch-track, using fMLLR features in DNN models. The beamformed data on ch1-ch6 is used for testing in all setups

System	fMLLR	
	dev-real	dev-sim
Chan5	9.39	10.46
BF	9.30	10.51
Chan1-6	8.49	9.29
Chan1-6+BF	8.20	8.90

3.2. Acoustic models

Different acoustic models were then constructed, including DNN, LSTM, CNN and very deep CNN. As shown in Table 2, VDCNN get a 10% relative improvement on the real data over the DNN with the speaker dependent feature.

Table 2: WER (%) comparison of different acoustic models for the 6ch-track. Beamformed data on ch1-ch6 is used for both training and testing in all setups. **Feats** indicates the model input feature

System	Feats	dev-real	dev-sim
DNN	fMLLR	9.30	10.51
LSTM	fMLLR	10.26	11.69
CNN	FBANK	10.14	12.22
VDCNN		8.66	10.52

3.3. Auxiliary feature joint training

The auxiliary feature joint trainings in the VDCNN model and LSTM-RNN model are implemented. The different types of auxiliary features are explored and the related results are shown in Table 3. For the i-vector, a GMM with 2048 Gaussians is used to extract a 10-dimensional i-vector for each utterance, and these i-vectors were obtained using MFCC features. We can see that joint training with auxiliary features obtain consistent gains on both VDCNN and LSTM-RNN, and the improvement in VDCNN is especially large which demonstrats the superiority of the proposed new architecture.

Table 3: WER (%) comparison of the very deep CNNs and LSTM-RNNs with auxiliary features joint training for the 6ch-track. Beamformed data on ch1-ch6 is used for both training and testing in all setups. **Aux** indicates the auxiliary feature

System	Feats	Aux	dev-real	dev-sim
VDCNN	FBANK	—	8.66	10.52
		fMLLR	7.92	8.90
		fMLLR+ivec	7.69	8.83
LSTM	fMLLR	—	10.26	11.69
		ivec	10.23	11.52

3.4. Submitted system

At last, we give the final submitted results in Table 4. As stated above, the augmented 108 hours data was used for all model trainings, and the multi-pass decoding shown as the Figure 3 was performed to obtain the 1-best results. Considering we only want to focus on the acoustic modeling, so the released RNNLM was applied for the rescoring.

Due to the limited evaluation time, we can not finish the testing using the best fusion system on time. Accordingly the results from the best single system (applying RNNLM on VDCNN-SA in P3) are submitted as our final results for the challenge. All the results covering three tracks, including both dev and eval under different environments.

Table 4: WER (%) for the best submitted system.

Track	Envir.	Dev		Eval	
		real	sim	real	sim
1ch	BUS	8.32	6.87	22.25	9.64
	CAF	6.21	10.00	14.46	14.74
	PED	3.91	6.06	10.05	12.55
	STR	6.70	8.78	8.91	14.87
	AVG	6.28	7.93	**13.91**	12.95
2ch	BUS	5.92	4.71	14.06	6.33
	CAF	4.53	7.24	8.16	10.31
	PED	3.54	4.45	7.44	8.85
	STR	5.19	6.50	6.89	9.47
	AVG	4.79	5.73	**9.14**	8.74
6ch	BUS	4.31	3.88	8.45	4.59
	CAF	3.72	5.25	5.60	6.31
	PED	2.67	3.47	5.01	5.96
	STR	4.22	4.90	6.57	8.31
	AVG	3.73	4.37	**6.41**	6.29

4. References

[1] E. Vincent, S. Watanabe, A. A. Nugraha, J. Barker, and R. Marxer, "An analysis of environment, microphone and data simulation mismatches in robust speech recognition," *Computer Speech and Language*, to appear.

[2] Y. Qian, M. Bi, T. Tan, and K. Yu, "Very deep convolutional neural networks for noise robust speech recognition," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 24, no. 12, pp. 2263–2276, 2016.

[3] Y. Qian, T. Tan, and D. Yu, "Neural network based multi-factor aware joint training for robust speech recognition," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 24, no. 12, pp. 2231–2240, 2016.

[4] M. Bi, Y. Qian, and K. Yu, "Very deep convolutional neural networks for lvcsr," in *Proceedings of Interspeech*, 2015, pp. 3259–3263.

[5] Y. Qian and P. Woodland, "Very deep convolutional neural networks for robust speech recognition," in *Proceedings of SLT*, 2016.

[6] H. Sak, A. W. Senior, and F. Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling." in *Proceedings of Interspeech*, 2014, pp. 338–342.

[7] M. L. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *Proceedings of ICASSP*, 2013, pp. 7398–7402.

[8] T. Tan, Y. Qian, D. Yu, S. Kundu, L. Lu, K. C. SIM, X. Xiao, and Y. Zhang, "Speaker-aware training of lstm-rnns for acoustic modelling," in *Proceedings of ICASSP*, 2016, pp. 5280–5284.

[9] Y. Qian, T. Tan, D. Yu, and Y. Zhang, "Integrated adaptation with multi-factor joint-learning for far-field speech recognition," in *Proceedings of ICASSP*, 2016, pp. 5770–5775.

[10] P. Woodland, X. Liu, Y. Qian, C. Zhang, M. Gales, P. Karanasou, P. Lanchantin, and L. Wang, "Cambridge university transcription systems for the multi-genre broadcast challenge," in *Proceedings of ASRU*, 2015, pp. 639–646.

[11] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The kaldi speech recognition toolkit," in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, dec 2011, iEEE Catalog No.: CFP11SRW-USB.

[12] A. Agarwal, E. Akchurin, G. Basoglu, G. Chen, S. Cyphers, J. Droppo, A. Eversole, B. Guenter, M. Hillebrand, R. Hoens *et al.*, "An introduction to computational networks and the computational network toolkit," Tech. Rep. MSR-TR-2014-112, August 2014.[Online]. Available: http://research. microsoft. com/apps/pubs/default. aspx, Tech. Rep., 2014.

CHiME4: Multichannel Enhancement Using Beamforming Driven by DNN-based Voice Activity Detection

Zbyněk Koldovský, Jiri Malek, Marek Boháč, and Jakub Janský

Faculty of Mechatronics, Informatics, and Interdisciplinary Studies,
Technical University of Liberec, Studentská 2, 461 17 Liberec, Czech Republic.
zbynek.koldovsky@tul.cz

Abstract

In this work, we focus on methods for enhancing the six-channel CHiME4 data using beamforming that is driven by voice activity detectors (VAD). We propose two beamformers and two VADs that are based on trained deep neural networks (DNN). Their combinations are compared when used as front-ends whose outputs are forwarded to the baseline automatic speech recognition system. Results in term of Word-Error-Rate (WER) achieved when the acoustic model of the baseline is or is not adapted for the given front-end (re-trained on enhanced training sets) are reported.

1. Introduction

Many multichannel speech enhancement systems apply beamforming methods such as the conventional Delay-and-Sum Beamformer (DSB), various implementations of minimum variance distortionless (MVDR) beamformer, or a generalization of the latter one, the linearly constrained minimum variance (LCMV) beamformer [1]. In order to achieve optimum performance, parameters have to be estimated and tracked with a sufficient accuracy. If not, the target signal in the system output can be distorted, which often deteriorates the final performance achieved by back-end processors (e.g., automatic speech recognition systems) even if the Signal-to-Noise Ratio (SNR) in a beamformer's output is improved.

In the conventional beamforming, the free-field sound propagation is assumed, and the DSB relies purely on the Time-Difference-Of-Arrival (TDOA) estimation. By contrast, the MVDR and LCMV can regard reverberation and multiple sources when using relative transfer functions (RTFs); see [2]. Such systems tend to be less robust as compared to the conventional approach. In particular, they are more sensitive to possible nonlinearities in the signal path as well as to various measurement (sensor) failures. On the other hand, their performance is potentially higher than that of the DSB, especially, in multi-source and reverberant conditions. The goal of this work is to compare the methods within CHiME4.

The baseline system of CHiME4 utilizes a state-of-the-art DSB technique named BeamformIt, proposed in [3]. The method estimates TDOAs using generalized cross-correlations (GCC-PHAT) and performs a robust multichannel TDOA tracking, which significantly helps to avoid sudden changes and estimation errors in TDOA. This and other straightforward modifications such as a mechanism that helps to avoid microphone failures make BeamformIt robust and useful for CHiME4. A practical drawback is that BeamformIt is passing through the signals several times before the output is computed, which hampers its direct applicability in continuous (on-line) processing.

Multichannel enhancement systems applying MVDR or LCMV with the aid of Deep Neural Networks (DNN) were applied to CHiME3 data; see, e.g., [4, 5]. The beamformers rely on the estimation of the noise covariance and of the source steering vector from masked signals, where the masks are obtained as outputs of DNNs.

In this work, approximate Minimum Mean-Squared Error beamformer (MMSE), recently proposed in [6], is modified in order to be applied within CHiME4. Similarly to [4, 5], the beamformer exploits DNNs, however, the DNNs are used to control the estimation of RTFs, not the estimation of noise/speech covariances. This is done through applying the RTF estimator from [7] where speech presence probabilities are obtained as the outputs of Voice-Activity Detectors (VAD) that are realized using the DNNs.

The performance of the MMSE depends purely on the accuracy of the estimated RTFs. As such, the beamformer strongly relies on the linearity of the observed signals. However, this appears to be often violated in the CHiME4 data, e.g., because of microphone failures and nonlinear gain fluctuations. The results of this work thus provide a comparison of the advanced beamforming with BeamformIt. We compare also a Filter-and-Sum Beamformer (FSB) based on the estimated RTFs, which could be seen as a solution on the half way between the MMSE and BeamformIt.

The paper is organized as follows. Section 2 describes the problem and basic beamforming approaches. Section 3 provides details of the proposed multichannel enhancement systems. Section 4 defines the back-end solutions that we use for CHiME4. Section 5 reports the results and Section 6 concludes the paper.

2. Problem Description

2.1. Model

A noisy recording of a directional source observed through m microphones can be described, in the short-term frequency domain, as

$$\mathbf{x}(k,\ell) = \mathbf{g}(k,\ell)s(k,\ell) + \mathbf{y}(k,\ell), \tag{1}$$

where $\mathbf{x}(k,\ell)$ is the $m \times 1$ vector of the signals on microphones, $s(k,\ell)$ is the target speech as observed on a reference microphone, and $\mathbf{y}(k,\ell)$ involves all other interfering sources and noise components that are uncorrelated with $s(k,\ell)$; k is the frequency index and ℓ is the frame index.

The vector $\mathbf{g}(k,\ell)$ determines the position of the target speaker. Its elements contain relative transfer functions (RTFs) related to the reference microphone [2]. Since the speaker can perform movements during utterances, $\mathbf{g}(k,\ell)$ is varying in time. Nevertheless, we assume that the changes are slow, so $\mathbf{g}(k,\ell)$ is approximately constant during blocks of frames.

From now on we will omit the arguments k and ℓ from the notation. They will be used only when the more precise notation is needed.

2.2. MVDR and MMSE beamforming

The MVDR beamformer is a popular multichannel processor that extracts s from $\mathbf{x}$, thereby reduces noise, enhances or even dereverberates the target signal [8]. Its output is $u = \mathbf{w}_{\mathrm{MVDR}}^H \mathbf{x}$ where

$$\mathbf{w}_{\mathrm{MVDR}} = \frac{\mathbf{C}_{\mathbf{y}}^{-1}\mathbf{g}}{\mathbf{g}^H \mathbf{C}_{\mathbf{y}}^{-1}\mathbf{g}}. \tag{2}$$

Here, $\mathbf{C}_{\mathbf{y}} = \mathrm{E}[\mathbf{y}\mathbf{y}^H]$ is the covariance matrix of the noise signal $\mathbf{y}$, $\mathrm{E}[\cdot]$ stands for the expectation operator, and $\cdot^*$ and $\cdot^H$ denote the conjugate value and the conjugate transpose, respectively.

The beamformer can be followed by a Wiener postfilter that attenuates the residual noise $y_{\mathrm{res}} = \mathbf{w}_{\mathrm{MVDR}}^H \mathbf{y}$ in the output of MVDR. The whole operation is equivalent to the Minimum Mean Square Error (MMSE) beamforming [1] and is given by

$$\mathbf{w}_{\mathrm{MMSE}} = \mathbf{w}_{\mathrm{MVDR}} \underbrace{\frac{\mathrm{E}[|u|^2] - \mathrm{E}[|y_{\mathrm{res}}|^2]}{\mathrm{E}[|u|^2]}}_{\text{Wiener postfilter}}. \tag{3}$$

To apply MMSE and MVDR efficiently in practice, it is crucial to estimate $\mathbf{C}_{\mathbf{y}}$, $\mathbf{g}$ and y_{res} with a sufficient accuracy.

2.3. Previous MVDR implementations for CHiME3

In [4, 5], $\mathbf{C}_{\mathbf{y}}$ is estimated with the aid of trained DNNs that compute frequency-dependent speech presence probabilities. The probabilities are used to control the noise covariance update so that the update is suspended during the speaker activity and vice versa. Then, the steering vector is estimated as the principal vector of the target covariance, which is estimated as the difference between the covariance of input signals $\mathbf{C} = \mathrm{E}[\mathbf{x}\mathbf{x}^H]$ and that of noise $\mathbf{C}_{\mathbf{y}}$.

The principal vector can be significantly biased in low SNR conditions. In the frequency bands where the target signal is not active, a vector steered towards another directional (interfering) source can be obtained instead. The above noise covariance estimation is not effective in two aspects. First, the computation of masks requires to pass data through a large DNN with as many outputs as is the number of frequency bins, which is computationally expensive. Second, the noise covariance should be updated continuously, also during the speaker activity, when the noise is nonstationary. The methods we propose here aims to overcome these drawbacks.

2.4. Filter-and-sum beamforming

The computation of the inversion matrix in (2) increases the computational burden and makes the MVDR (MMSE) beamformer sensitive to estimation errors. Once the steering vector $\mathbf{g}$ is estimated, a method that is less sensitive to possible errors and does not require the knowledge (estimation) of $\mathbf{C}_{\mathbf{y}}$ is represented by

$$\mathbf{w}_{\mathrm{FSB}} = \frac{1}{m}(\mathbf{g}^{-1})^*, \tag{4}$$

where $\mathbf{g}^{-1}$ contains the reciprocal values of the elements of $\mathbf{g}$. This method, in fact, performs a filter-and-sum beamforming (FSB) that is a generalization of the DSB for reverberant environments. Indeed, in the free-field conditions, the FSB coincides with the DSB, because the elements of $\mathbf{g}$ correspond to pure delay filters, and $\mathbf{g}^{-1}$ are their respective inverse delays.

The FSB can be followed by the Wiener postfilter similarly to (3) if any estimate of the residual noise in the FSB output (i.e., an estimate of $\mathbf{w}_{\mathrm{FSB}}^H \mathbf{y}$) is available.

3. Front-End

In this section, details of four different systems for multichannel speech enhancement are described. Each system is a combinations of a VAD and of a beamformer.

Two VADs are considered where both are designed through trained DNNs. One VAD performs a detailed speech presence detection, that is, within each frequency bin. The other VAD performs only the per-frame detection. The VADs are used to estimate $\mathbf{g}^{-1}$ using the method from [7].

Then, two beamformers are considered: A variant of the approximate MMSE beamformer described in [6], and the simpler FSB, which was described above.

The processing of signals proceeds in the short-time Fourier (STFT) domain where the window length is 512 samples and the frame shift is 128 samples. The systems operate in a batch-online processing regime. Each batch of 100 STFT frames is processed independently in the following steps.

1. The input channels are selected based on their time domain correlation coefficients. Specifically, for the ith channel, the maximal correlation coefficient with the other channels is computed; let us denote the value μ_i. If this value is smaller than a threshold, the channel is not used. However, at least two channels are kept for further processing (the channels with maximum μ_i).
2. The reference channel is CH5 unless it has been withdrawn in the previous step. If yes, the channel with the maximum μ_i is selected.
3. VAD is applied to the selected channels.
4. The steering vector $\mathbf{g}$ as well as $\mathbf{g}^{-1}$ are assumed to be approximately constant within the batch of frames. The elements of $\mathbf{g}^{-1}$, that is, the respective RTFs related to the reference channel, are estimated using the estimator from [7] where speech presence probabilities are replaced by the outputs of VAD.
5. A given beamformer is applied. Its output is transformed back to the time domain using the inverse Fourier transform and overlap-add.

3.1. VAD using DNN

We consider two VADs: The first detector, referred to as sVAD, yields the speech activity over every frame of the processed signal. The second one detector, referred to as dVAD, estimates the speech activity for every frequency bin and every signal frame. Both VADs are implemented as DNNs trained using the Torch framework[1]. Training as well as testing sets were created from the CHiME4 training data.

sVAD is trained to estimate Wiener gains (values between 0 and 1). Each STFT frame is represented by raw magnitude of the 40 mel filter bank features (which are not decorrelated). The input feature vector concatenates the analyzed frame, 10 frames

[1] http://torch.ch

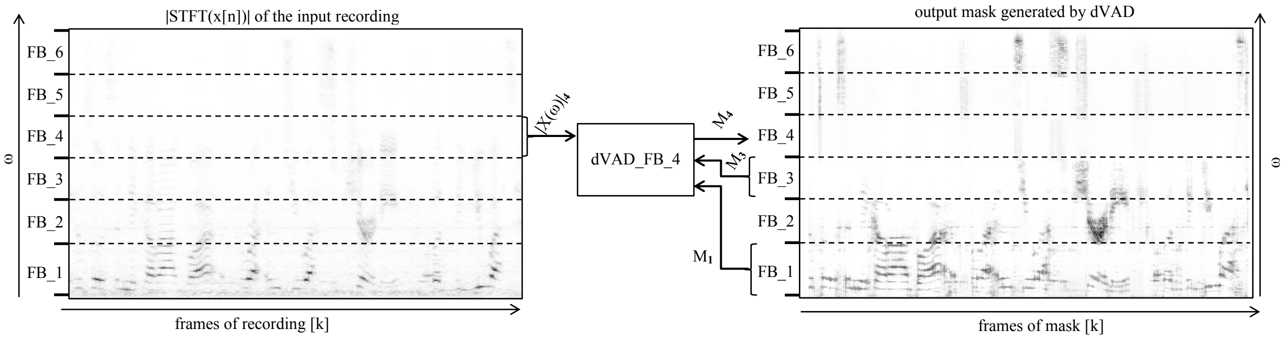


Figure 1: Illustration of the data flow of dVAD_FB_4 (white–black color scale refers to 0–maximum values)

before and 2 after it. Global zero-mean and unit-variance normalization is applied (computed from the training data).

sVAD consists of 5 hidden layers (3x256 and 2x128 neurons, respectively) all with sigmoid activation function. Binary Cross Entropy criterion was optimized using 1024 minibatches and finished within 50 epochs. No pre-training or dropout was used, data order was randomized every epoch.

dVAD consists of 6 smaller DNNs, referred to as dVAD_FB_1,...,dVAD_FB_6. Each DNN has one of six frequency bands (FB_1,...,FB_6) on its input together with reduced outputs of the previous DNNs. For example, the input of dVAD_FB_4 is illustrated in Figure 1.

The output of each DNN is a vector of values from the interval $[0; 1]$ containing the speech presence probabilities for the respective frequency band and frame. The reduced outputs (used on the inputs of the other DNNs) contain averages over 10 neighboring bins. For a given frequency bin, the training output label is zero if the SNR for the frequency is smaller than 5 dB. Otherwise, the label is set to one.

The structure of dVAD is computationally cheaper by about 50% as compared to a VAD that resides in a big DNN that computes the speech probabilities in all frequency bins simultaneously. dVAD_FB_1,...,dVAD_FB_6 were trained subsequently. Therefore, zero mean and unit variance normalization of the input data was applied between the training of each DNN.

Each dVAD_FB_x consists of 5 hidden layers (2x350, 256 and 2x128 neurons, respectively) all with ReLU activation function. For the kth frame, the context of frames $k-8$, $k-6$, $k-4$, $k-2$, $k+2$ and $k+4$ is used. Mean Square Error criterion is optimized within 1024 minibatches. No pre-training was applied; training data order was randomized. The training was finished between epochs 54 and 60.

3.2. Approximate MMSE beamformer

We implement the MMSE beamformer as an approximate MVDR followed by the Wiener post-filter. The MVDR part exploits a blocking matrix to obtain noise reference signals. The blocking matrix is defined as (without any loss on generality, assume that the reference channel is CH1)

$$\mathbf{B} = \begin{pmatrix} -1 & g_2^{-1} & 0 & \dots & 0 \\ -1 & 0 & g_3^{-1} & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ -1 & 0 & 0 & \dots & g_m^{-1} \end{pmatrix}, \tag{5}$$

where g_i^{-1} denotes the ith element of $\mathbf{g}^{-1}$. The noise reference signal is obtained by passing the input through the blocking matrix, that is,

$$\mathbf{u} = \mathbf{B}\mathbf{x}, \tag{6}$$

however, this signal is different from the noise term $\mathbf{y}$ in (1). Since the beamformer operates with a batch of frames, the least-square estimate of $\mathbf{y}$ using $\mathbf{u}$ can be computed as

$$\widehat{\mathbf{y}} = \mathbf{C}\mathbf{B}^H(\mathbf{B}\mathbf{C}\mathbf{B}^H)^{-1}\mathbf{B}\mathbf{x}, \tag{7}$$

where $\mathbf{C} = \mathrm{E}[\mathbf{x}\mathbf{x}^H]$ is replaced by its sample mean estimate. The estimator (7) is scale-invariant in the sense that any scaling substitution $\mathbf{B} \leftarrow \mathbf{\Lambda}\mathbf{B}$ where $\mathbf{\Lambda}$ is regular does not have any influence on $\widehat{\mathbf{y}}$. In particular, this property is useful when $\mathbf{B}$ is derived using blind methods such as Independent Component Analysis (ICA) that can estimate $\mathbf{B}$ only up to the unknown scaling factor $\mathbf{\Lambda}$; see, e.g., [9].

The covariance of $\widehat{\mathbf{y}}$ is equal to

$$\mathbf{C}_{\widehat{\mathbf{y}}} = \mathrm{E}[\widehat{\mathbf{y}}\widehat{\mathbf{y}}^H] = \mathbf{C}\mathbf{B}^H(\mathbf{B}\mathbf{C}\mathbf{B}^H)^{-1}\mathbf{B}\mathbf{C}. \tag{8}$$

In the approximate MVDR, the strategy is to replace $\mathbf{C}_{\mathbf{y}}$ in (2) by $\mathbf{C}_{\widehat{\mathbf{y}}}$. The steering vector $\mathbf{g}$ can be computed directly from $\mathbf{g}^{-1}$; an alternative approach is to compute $\mathbf{g}$ as a vector from the null space of $\mathbf{B}$.

Since the rank of $\mathbf{C}_{\widehat{\mathbf{y}}}$ is $m-1$, its inversion matrix does not exist. We therefore replace $\mathbf{C}_{\widehat{\mathbf{y}}}^{-1}$ by the Moore-Penrose pseudoinverse denoted as $\mathbf{C}_{\widehat{\mathbf{y}}}^{\dagger}$. Then, the approximate MVDR beamformer is represented by

$$\widehat{\mathbf{w}}_{\mathrm{MVDR}} = \frac{\mathbf{C}_{\widehat{\mathbf{y}}}^{\dagger}\mathbf{g}}{\mathbf{g}^H\mathbf{C}_{\widehat{\mathbf{y}}}^{\dagger}\mathbf{g}}. \tag{9}$$

In case that the target channel is different from the reference channel, the scale-invariant least-squares can be applied as in (7). Then, all enhanced channels can be obtained as $\widehat{\mathbf{W}}_{\mathrm{MVDR}}\mathbf{x}$ where

$$\widehat{\mathbf{W}}_{\mathrm{MVDR}} = \frac{\mathbf{C}\widehat{\mathbf{w}}_{\mathrm{MVDR}}(\widehat{\mathbf{w}}_{\mathrm{MVDR}})^H}{(\widehat{\mathbf{w}}_{\mathrm{MVDR}})^H\mathbf{C}\widehat{\mathbf{w}}_{\mathrm{MVDR}}}. \tag{10}$$

From now on, let $\widehat{\mathbf{w}}_{\mathrm{MVDR}}$ denote the approximate MVDR for the selected target channel. Let the output be denoted as $v = \widehat{\mathbf{w}}_{\mathrm{MVDR}}^H\mathbf{x}$.

Using (7), the residual noise in the output can be estimated as

$$r = \widehat{\mathbf{w}}_{\mathrm{MVDR}}^H\widehat{\mathbf{y}}. \tag{11}$$

According to (3), the gain of the Wiener postfilter can be approximated as

$$G(k,\ell) = \frac{\max\{|v(k,\ell)|^2 - |r(k,\ell)|^2, \epsilon\}}{|v(k,\ell)|^2 + \epsilon}, \tag{12}$$

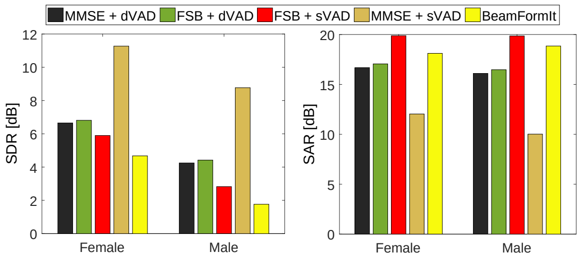


Figure 2: Results of the objective evaluation experiment in terms of SDR and SIR. The results were averaged over the four noisy environments BUS, CAF, STR and PED.

where ϵ is a small positive constant that prevents from division by zero. The final output of the approximate MMSE is

$$\widehat{s}(k,\ell) = G(k,\ell)v(k,\ell). \qquad (13)$$

It is worth noting that $G(k,\ell)$ can be modified in various heuristic ways before it is applied in (13). In CHiME-4, we set $G(k,\ell) = 1$ for k corresponding to frequencies higher than 3 kHz. By contrast, for the frequencies below 100 Hz, $G(k,\ell) = 0.01$. The gain could be also modified according to the output of the VAD. For example, if for given k the VAD yields speech probability higher than 0.5, we set $G(k,\ell) = 1$ to avoid the distortion of the speech in the system output.

4. Back-End Solutions

For the experimental evaluation, we consider two automatic speech recognition back-ends:

1. the baseline DNN+RNNLM back-end [10] provided by CHiME4 organizers, and
2. the same back-end with a re-trained acoustic model.

The front-end processing usually introduces additional artifacts into the processed speech signals, which are unknown to the acoustic model trained on the unprocessed signals. This may lead to a deterioration of the performance of the ASR system and motivates us to adapt the acoustic model for the given front-end. This is done as follows. The training set is enhanced by the front-end processor, by which a new training set is obtained. Then, this set is used by the training procedure of the baseline DNN models, which results in an adapted acoustic model.

5. Experiments and Results

5.1. Objective evaluation

Here, we describe an experiment where the proposed multichannel enhancement (front-end) systems are compared with BeamformIt in terms of signal separation criteria from BSS_Eval [11][2]. Two utterances were selected from the development set: `F01_421C0201` (a female speaker) and `M04_052C0112` (a male speaker). Four simulated (SIMU) noisy variants of each utterance (BUS, CAF, STR and PED) were processed by the enhancement systems. The outputs were evaluated in terms of Signal-to-Distortion Ratio (SDR) and Signal-to-Artefact Ratio (SAR). The results in terms of Signal-to-Interference Ratio (SIR) were similar to SDR, but we do not show them to save space. Averaged SDR and SAR over the environments are shown in Figure 2.

The proposed systems outperform BeamformIt in terms of SDR and SIR, which confirms their advanced ability to enhance the signal. The best SDR was achieved by MMSE+sVAD. On the other hand, the results in terms of SAR show that the proposed systems tend to introduce more artifacts into the enhanced signal. Only FSB+sVAD yields higher SAR than BeamformIt. The worst SAR yields MMSE+sVAD, which is the compromise for the high SDR and SIR.

5.2. CHiME4

Now we present the speech recognition results achieved by 10 systems. Each proposed ASR system is denoted by $A(B)$ where A denotes the front-end system, e.g., MMSE:sVAD, and B denotes the acoustic model used within the baseline ASR system, which is either ”Base” (original model) or ”Adapt” (the model adapted to the front-end). The case when the CHiME4 data are sent directly to the baseline ASR without any processing is denoted as ”Unprocessed”.

The resulting absolute Word Error Rates (WER) are shown in Table 1. Detailed results of FSB:sVAD(Base) and of the baselines for different noisy environments are presented in Table 2.

Comparing the proposed front-end systems, those using the FSB beamformer yield superior results compared to those with MMSE. The difference in simulated sets is about 2-3% WER. In case of the real-worlds recordings, the difference is up to 9%.

The choice of the VAD does not appear to have much influence on the final WER, especially in the combination with FSB. Considering the MMSE beamforming, the dVAD improves the WER compared to sVAD by 0-6%.

The adaptation of the acoustic models appears to be beneficial for the systems with MMSE, where it improves the performance by 0-2%. On the other hand, the re-training did not bring any significant improvement for the FSB technique.

Table 1: Absolute WER (%) averaged over four environments for the 6-channel track. The best achieved results are written in bold.

System	Dev		Test	
	real	simu	real	simu
Unprocessed (Base)	9.83	8.86	19.90	10.79
BeamformIt (Base)	**5.77**	**6.76**	**11.52**	10.91
MMSE:sVAD (Base)	10.91	9.31	22.39	9.72
MMSE:sVAD (Adapt)	10.56	9.21	20.61	9.11
MMSE:dVAD (Base)	7.78	9.84	16.27	9.68
MMSE:dVAD (Adapt)	7.89	9.28	16.09	9.40
FSB:sVAD (Base)	7.26	7.23	13.48	**7.70**
FSB:sVAD (Adapt)	7.23	7.68	13.46	7.95
FSB:dVAD (Base)	7.09	8.00	13.48	7.85
FSB:dVAD (Adapt)	7.43	8.24	14.40	8.16

6. Conclusions

From the results of our experiments we conclude that, among the proposed systems, FSB:sVAD(Base) appears to be the most effective for CHiME4. It achieves WER between 7%-13%, which improves the WER achieved on unprocessed data by about 1.5%-6.5%. The system is computationally simple, because the FSB does not use the matrix pseudo-inversion in (9), and the sVAD performs the computationally save per-frame de-

[2] We use version 2.3 of BSS_Eval, which contains `bss_decomp_tvfilt.m`, a function that enables us to evaluate time-variant mixtures.

Table 2: Absolute WER (%) per environment. The best achievements are written in bold.

(a) FSB:sVAD (Base)

Envir.	Dev		Test	
	real	simu	real	simu
BUS	10.27	6.21	22.21	**5.68**
CAF	6.55	9.73	12.59	**8.91**
PED	4.57	5.52	10.71	**6.85**
STR	7.54	7.46	**8.40**	**9.34**

(b) BeamformIt (Base)

Envir.	Dev		Test	
	real	simu	real	simu
BUS	**7.43**	**5.97**	**16.88**	7.66
CAF	**5.77**	**8.13**	**10.20**	11.52
PED	**3.73**	**5.47**	**9.87**	10.35
STR	**6.15**	**7.45**	9.13	14.12

(c) Unprocessed (Base)

Envir.	Dev		Test	
	real	simu	real	simu
BUS	16.06	10.07	33.17	9.58
CAF	8.44	10.59	19.22	11.95
PED	5.44	6.34	14.63	9.64
STR	9.38	8.44	12.61	11.97

tection. The method achieves the best WER over the compared systems in the simulated test set.

For the other sets, in particular in the real-world sets, the best WER was achieved with BeamformIt. The experiment of Section 5.1 has demonstrated on typical simulated recordings that BeamformIt achieves lower SDR as well as lower SAR compared to FSB:sVAD. While the simulated recordings are sufficiently linear and do not contain microphones failures, the real-world recordings of CHiME4 do. We therefore attribute the better WER achieved by BeamformIt in real-world sets to its robustness against nonlinear effects rather than to its ability to enhance the target signal.

The improvement of the proposed methods in terms of the robustness against microphone failures and other nonlinearities is the subject of our future progress.

7. Acknowledgments

We thank Francesco Nesta and Trausti Thormundsson for helpful discussions. This work was supported by The Czech Science Foundation through Project No. 14-11898S and by California Community Foundation through Project No. DA-15-114599.

8. References

[1] H. L. Van Trees, *Detection, Estimation, and Modulation Theory, Optimum Array Processing*, ser. Detection, Estimation, and Modulation Theory. Wiley, 2004.

[2] S. Gannot, D. Burshtein, and E. Weinstein, “Signal enhancement using beamforming and nonstationarity with applications to speech,” *IEEE Transactions on Signal Processing*, vol. 49, no. 8, pp. 1614–1626, Aug 2001.

[3] X. Anguera, C. Wooters, and J. Hernando, “Acoustic beamforming for speaker diarization of meetings,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 2011–2022, Sept 2007.

[4] J. Heymann, L. Drude, and R. Haeb-Umbach, “Neural network based spectral mask estimation for acoustic beamforming,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 196–200.

[5] S. Araki, M. Okada, T. Higuchi, A. Ogawa, and T. Nakatani, “Spatial correlation model based observation vector clustering and mvdr beamforming for meeting recognition,” in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2016, pp. 385–389.

[6] Z. Koldovský and F. Nesta, “Approximate mvdr and mmse beamformers exploiting scale-invariant reconstruction of signals on microphones,” in *Acoustic Signal Enhancement (IWAENC), 2016 15th International Workshop on*, September 2016.

[7] I. Cohen, “Relative transfer function identification using speech signals,” *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 5, pp. 451–459, Sept 2004.

[8] O. Schwartz, S. Gannot, and E. A. P. Habets, “Multimicrophone speech dereverberation and noise reduction using relative early transfer functions,” *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 2, pp. 240–251, Feb 2015.

[9] Y. Takahashi, T. Takatani, K. Osako, H. Saruwatari, and K. Shikano, “Blind spatial subtraction array for speech enhancement in noisy environment,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 4, pp. 650–664, May 2009.

[10] E. Vincent, S. Watanabe, A. A. Nugraha, J. Barker, and R. Marxer, “An analysis of environment, microphone and data simulation mismatches in robust speech recognition,” *Computer Speech and Language*, to appear.

[11] E. Vincent, R. Gribonval, and C. Fevotte, “Performance measurement in blind audio source separation,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462–1469, July 2006.

Proc. of the 4th Intl. Workshop on Speech Processing in Everyday Environments (CHiME 2016), San Francisco, CA, USA, Sep. 13, 2016

Wrapper-Based Acoustic Group Feature Selection for Noise-Robust Automatic Sleepiness Classification

Dara Pir[1], *Theodore Brown*[1,2], *Jarek Krajewski*[3,4]

[1]Dept. of Computer Science, The Graduate Center, City University of New York, New York, USA
[2]Dept. of Computer Science, Queens College, City University of New York, New York, USA
[3]Institute for Safety Technology, University of Wuppertal, Wuppertal, Germany
[4]Engineering Psychology, Rhenish University of Applied Science, Cologne, Germany

dpir@gradcenter.cuny.edu, tbrown@gc.cuny.edu, krajewsk@uni-wuppertal.de

Abstract

This paper presents a noise-robust Wrapper-based acoustic Group Feature Selection (W-GFS) method and its large noise Optimized (OW-GFS) version for automatic sleepiness classification and compares their performances with Correlation-based Feature Selection (C-FS) and Pearson Correlation Coefficient Feature Selection (CC-FS) filters. We use Interspeech 2011 Speaker State Challenge's "Sleepy Language Corpus" and baseline feature set. Group Feature Selection (GFS) considers the feature space in Low Level Descriptor groups rather than individually. Reduced time-complexity and potential generalization power of GFS are discussed. A model to predict on test data with changing Signal-to-Noise Ratio (SNR) is presented based on results from artificially corrupted development data with 10 dB SNR white-noise. Using Support Vector Machine, W-GFS achieves 2.6%, 4.2%, and 1.9% relative Unweighted Average Recall (UAR) improvement over the C-FS, CC-FS, and baseline feature set systems, respectively, on white-noise corrupted test data with randomly changing SNR within a broad range. The corresponding improvements for OW-GFS, using Voted Perception, are 4.8%, 9.8%, and 2.2% relative UAR on strongly white-noise corrupted test data with randomly changing SNR between -5 and +5 dB. Finally, we discuss consistent results obtained using everyday environment noises.

Index Terms: robust paralinguistics, computational paralinguistics, noise-robust feature selection, wrapper method, filter method

1. Introduction

The prevalence of sleep related accidents [1, 2, 3] and the imperative to prevent them highlights the importance of sleepiness detection systems. In situations where the use of certain types of detection methods, e.g., a spontaneous eye-blink detection system [4] requiring the use of intrusive sensors, is not optimal, speech can offer a unique advantage [5, 6, 7]. Moreover, the widespread nature of the sleep phenomenon is indicative of the abundance of applications concerned with its detection.

Computational paralinguistics tasks like sleepiness classification deal with the manner in which something is said rather than the content of what is said [8]. The binary task of Sleepiness Sub-Challenge was presented as part of the Interspeech 2011 Speaker State Challenge and employed the "Sleepy Language Corpus" (SLC) [9]. The 4368 acoustic baseline features generated using the openSMILE software [10] include those deemed relevant to sleepiness state [11] and result in a Sub-Challenge baseline score of 70.3% Unweighted Average Recall (UAR). The findings of the Sub-Challenge demonstrate that using larger feature sets result in superior performances. Furthermore, in the presence of various types and levels of noise, larger feature sets provide a larger pool for subsequent feature selection operations to choose from, in a data-driven fashion [12]. Using domain knowledge to design relevant features for classification in noisy environments is an alternative feature-based approach [13].

The two main types of feature selection methods are filters and wrappers [14]. The filter evaluates feature subsets based on statistical properties of data whereas the wrapper uses a classifier's performance score for the evaluation. The wrapper searches the feature space and evaluates feature subsets for selection. Wrapper-based Group Feature Selection (W-GFS) [15] uses a linear method, a fast variant of Best Incremental Ranked Subset (BIRS) [16], for feature space search and WEKA toolkit's [17] Support Vector Machine (SVM) [18] implementation, Sequential Minimal Optimization (SMO) [19] with linear Kernel, for feature subset evaluation. W-GFS modifies the basic wrapper by considering features in groups defined by Low Level Descriptor (LLD) partitions [20] rather than individually. Group Feature Selection (GFS) approach is motivated by two factors. First, GFS improves the tractability of the computationally intensive wrapper method by reducing the time complexity of the subset search component [15]. Second, an LLD-based GFS could potentially improve the generalization power of the classification algorithm by avoiding overfitting that may result from using a detailed individual feature search. Optimized Wrapper-based Group Feature Selection (OW-GFS) operates identically to W-GFS but its more restrictive selection criteria does not consider groups with evaluation scores of less than 55% UAR for selection.

The novel aspects of this work, to the best of our knowledge, are the following. First, although W-GFS has been used for another paralinguistics classification task [15], a specialized selection mechanism was employed that removed less than 1% of the features in the best performance. In this work, our two GFS methods remove about 80% and 90% of the features. In this mode, which achieves meaningful dimensionality reduction, the use of W-GFS is novel. Second, implementation of W-GFS in the context of noise-robust paralinguistics is novel. Finally, OW-GFS is a novel method that provides further noise-robustness under high noise conditions.

This paper is organized as follows. Section 2 describes the LLD-based partitioning and the BIRS algorithm for feature space search. Section 3 provides details about the corpus. Noise-robust feature selection and performance evaluation

Table 1: *Results in % UAR of SMO and VP classifications using the four feature selection methods and the baseline (BL) represented by columns of the table on high noise level test data. The best performances for each column are depicted in bold.*

CLS	W-GFS	OW-GFS	C-FS	CC-FS	BL
SMO1	61.5	62.3	59.6	59.9	**65.0**
SMO2	62.5	62.7	61.1	60.4	64.2
SMO3	**64.2**	63.5	61.6	**60.5**	63.2
SMO4	**64.2**	64.2	62.5	60.4	60.8
SMO5	63.4	64.0	62.9	60.2	58.5
SMO6	63.5	64.3	62.7	60.0	57.5
SMO7	62.6	64.1	62.9	60.4	55.9
VP	63.1	**66.4**	**63.4**	59.1	58.9

Table 2: *Results on medium noise level test data.*

CLS	W-GFS	OW-GFS	C-FS	CC-FS	BL
SMO1	64.1	64.3	64.1	61.9	65.8
SMO2	66.1	64.7	64.9	62.9	**66.4**
SMO3	67.2	65.4	65.3	64.6	66.1
SMO4	**67.5**	65.7	65.9	64.7	64.7
SMO5	**67.5**	65.9	65.9	64.9	61.7
SMO6	66.7	65.6	66.0	64.7	61.0
SMO7	65.4	66.1	66.1	**65.9**	59.5
VP	65.1	**67.4**	**66.2**	61.9	62.6

Table 3: *Results on low noise level test data. An additional complexity parameter = 0.01 (used by classifier SMO8) is needed to cover the range of interest for CC-FS.*

CLS	W-GFS	OW-GFS	C-FS	CC-FS	BL
SMO1	66.6	66.6	65.9	63.1	66.8
SMO2	68.2	67.3	66.3	64.8	67.1
SMO3	69.0	68.1	67.2	65.6	67.1
SMO4	**69.6**	**68.2**	**67.8**	66.1	**67.2**
SMO5	69.3	68.0	67.4	66.2	66.9
SMO6	68.6	68.0	67.1	66.4	67.0
SMO7	67.8	67.0	66.4	66.8	64.6
SMO8	...	...	...	**67.2**	...
VP	63.7	65.1	63.9	63.5	64.5

Table 4: *Results on unknown noise level test data.*

CLS	W-GFS	OW-GFS	C-FS	CC-FS	BL
SMO1	64.1	64.4	63.2	61.6	**65.9**
SMO2	65.6	64.9	64.1	62.7	**65.9**
SMO3	66.8	65.7	64.7	63.6	65.5
SMO4	**67.1**	66.1	**65.4**	63.7	64.2
SMO5	66.7	66.0	**65.4**	63.8	62.4
SMO6	66.2	66.0	65.3	63.7	61.8
SMO7	65.3	65.7	65.1	**64.4**	60.0
VP	64.0	**66.3**	64.5	61.5	62.2

methods are explained in section 4. The experimental results are discussed in section 5 and the paper's conclusions and suggested future work are covered in the last section.

2. Background

2.1. LLD-Based Groups

Acoustic features are generated by chunk level application of functionals like arithmetic mean to LLD contours like RMS energy [21, 9]. The Sleepiness Sub-Challenge uses three sets of LLDs, each having a corresponding set of functionals listed in [9]. Using LLD-partitioned groups is acoustically motivated. If application of a statistical functional to an LLD contour generates a feature relevant to a classification task, it is likely that application of other functionals to the same LLD could be useful for the task as well and vice versa [15].

2.2. BIRS Search

BIRS is a linear forward search algorithm performed in two steps: ranking and feature subset selection. In the ranking step, the features are ranked from highest to lowest based on their evaluation score. In the feature subset selection step, the entire ranked feature set is traversed starting with an empty subset which selects features whose addition results in a subset that is evaluated to a higher UAR value, by a threshold level. Our fast variant of the algorithm used here does not employ cross-validation and t-test in the subset selection step. Wrapper evaluation cycles are used as the time complexity measure. The algorithm performs $2 * N$ evaluations, where N is the number of individual features in the search space. Our LLD-based GFS reduces the algorithm's $N = 4368$ evaluation cycles, in each step, to 118 cycles, i.e., the number of LLDs in the baseline feature set.

3. Corpus

The SLC used in our classification contains speech recordings of 99 subjects made in realistic car and lecture-room settings and has a duration of 21 hours. The original 44.1 kHz recordings made with a microphone-to-mouth distance of 0.3 m are down-sampled to 16 kHz and use 16 bit quantization [9]. The levels of sleepiness 1 through 10 are reported according to the Karolinska Sleepiness Scale (KSS) [22] which is shown to be valid in certain studies [23]. A level equal or below 7.5 is classified as non-sleepy and one above 7.5 as sleepy.

4. Method

We first explain how our two W-GFS and OW-GFS methods and the two filters, Correlation-based Feature Selection (C-FS) [24] and Pearson Correlation Coefficient Feature Selection (CC-FS) [25] (implemented by WEKA's CfsSubsetEval and CorrelationAttributeEval, respectively), are used in the development phase to obtain the four noise-robust feature sets which are subsequently used in the evaluation phase. For the GFS methods, in the development phase, we train on the training set and predict on the development set. For the two filters, we train on the combined training set (training plus development sets combined). Next, we describe our evaluation of the four noise-robust selected feature sets using test data sets with changing noise levels. For evaluation, we train on the combined training set and report predictions on the test set. Features are standardized to standard normal and WEKA's Synthetic Minority Oversampling Technique (SMOTE) implementation [26] is used to balance the number of the classes in the development sets.

4.1. Noise-Robust Feature Selection on Development Data

In the absence of knowledge about the nature of the background noise, we model our feature selection systems using additive white Gaussian noise. First, in matched manner, we use devel-

Table 5: *Best performance results (bold entries) from Tables 1, 2, 3, and 4. The highest value of W-GFS and OW-GFS methods is displayed under "Best GFS" column.*

Noise	Best GFS	C-FS	CC-FS	BL
High	66.4	63.4	60.5	65.0
Med	67.5	66.5	65.9	66.4
Low	69.6	67.8	67.2	67.2
Unknown	67.1	65.4	64.4	65.9

Table 6: *% Improvement in relative UAR of the best performing model (Best Pair) over the best C-FS, CC-FS, and baseline models on each noise level test data.*

Noise	Best Pair	↑ C-FS	↑ CC-FS	↑ BL
High	OW-GFS, VP	4.8	9.8	2.2
Med	W-GFS, SMO4	1.6	2.5	1.8
Low	W-GFS, SMO4	2.6	3.5	3.6
Unknown	W-GFS, SMO4	2.6	4.2	1.9

opment data with additive white-noise of 10 dB Signal-to-Noise Ratio (SNR) level (generated by MATLAB's Communications System Toolbox function *awgn* [27]) to find the optimum linear kernel SMO complexity parameter. We model our feature selection methods for noise-robustness based on results from this mid-range SNR level. Second, using the obtained complexity, we perform W-GFS to obtain the noise-robust feature set which we will use to evaluate W-GFS on test data. Third, to add more robustness under high noise levels, the selected feature set obtained by W-GFS is reduced by removing groups with less than 55% UAR scores. The resultant feature set will be used to evaluate OW-GFS on test data. Finally, we obtain the two other feature sets using the C-FS and CC-FS filters. For CC-FS, we use the top 400 features as in [28]. The four noise-robust selected feature sets, in the mentioned order, are of sizes 935, 407, 138, and 400, respectively.

4.2. Evaluation System on Test Data

We use WEKA's linear kernel SMO and VotedPerceptron (VP) [29] implementations in the evaluation phase. If the type of noise is known, evaluation can be performed in a matched manner. In the absence of knowledge about the nature of the everyday environment noise, our four prediction models are trained in a partially matched fashion, i.e., using the clean combined training set reduced by the four noise-robust feature sets obtained using additive white-noise in the development phase. Predictions are made on noisy test data. Since the degree of similarity between white-noise and the particular everyday environment noise is unknown, prediction in a fully matched manner could produce unpredictable outcome. Noisy test data is produced as described below.

We generate three test sets with high, medium, and low levels of additive white-noise, respectively. To generate the high level noise test data, following a uniform distribution, we randomly add white-noise to the test data using an SNR level between -5 and +5 dB. This generation process allows for evaluation under changing noise levels. The medium noise level test data is generated in a similar manner except that the SNR range is between +5 and +15 dB. In order to include clean data as part of our test sets, the low noise level test data is generated similarly to the other levels using the +15 to +25 dB range but only with a 50% chance following a uniform distribution. The remaining 50% of data is clean.

Table 7: *Results on everyday environment noise test data (counterpart of Table 6's last row). The "Best Pair" obtains 63.1 % UAR.*

Noise	Best Pair	↑ C-FS	↑ CC-FS	↑ BL
Unknown	OW-GFS, SMO7	2.1	3.1	1.4

In practice, hyperparameters tuned in the development phase are used for prediction in the test and evaluation phases. However, using W-GFS for tuning the SMO complexity parameter gives the model an unfair advantage over others. To fairly compare our four feature selection and baseline models using the SMO classifier, therefore, we need to evaluate their performances using several SMO complexity parameters spanning the range of interest. The seven values of interest range from 0.00005 to 0.005 in approximately double increments, i.e., 0.00005, 0.0001, 0.0002, ..., 0.005. The corresponding classifiers are named SMO1, SMO2, SMO3, ..., SMO7, respectively. For the VP classifier, WEKA's default settings are used.

5. Experimental Results

Table 1 depicts results obtained on high noise level test data. The highest value in this table represents the model (feature selection method and classifier pair) that achieves best performance on high noise level test data. Tables 2 and 3 are generated similarly for the medium and low level noise test data. Table 4 is the average of the high, medium, and low noise level test data tables and represents the unknown noise level. The highest value in this table represents the model that achieves best performance under changing and unknown noise levels.

To facilitate comparison of results obtained by the four feature selection methods and the baseline we generate Tables 5 and 6. Table 5 displays the best performance results (bold entries) from Tables 1, 2, 3, and 4. Results from these tables demonstrate that our two GFS methods obtain the top two performances for each noise level. The highest value obtained by the W-GFS and OW-GFS methods is displayed under the common "Best GFS" column. Table 6 is constructed in the following manner. Column 1 displays the noise level. Column 2 identifies the model (method and classifier pair) that attains best performance on each noise level test data. Column 3 (↑ C-FS) depicts, for each level, the percent improvement in relative UAR of the best model over the best C-FS model. Similarly, columns 4 (↑ CC-FS) and 5 (↑ BL) show improvements of the best model over the best CC-FS and best baseline models. These results demonstrate that the best GFS method consistently outperforms the C-FS, CC-FS, and baseline models on all four noise level test data. Specifically, for high noise, the OW-GFS and VP pair outperforms the best C-FS, CC-FS, and baseline models by 4.8%, 9.8%, and 2.2% relative UAR, respectively. The overall best performing model under changing and unknown noise level, the W-GFS and SMO4 (SMO with complexity = 0.0005) pair, outperforms the best C-FS, CC-FS, and baseline models by 2.6%, 4.2%, and 1.9% relative UAR, respectively.

Finally, we evaluated the four feature selection methods and the baseline on test data with additive everyday environment noises. Recording of nature plus driving car sounds was undergone SNR level changes according to the same distributions that was used in generating the unknown noise level test data for additive white-noise. The resultant test data was generated directly rather than through the averaging process used for the white-noise case. The results are displayed in Table 7. The performance improvement pattern is similar to that of the white-noise case (last row of Table 6) although the best performance value of 63.1% UAR using everyday environment noise (not shown in the table) is expectedly lower than the 67.1% obtained by the white-noise counterpart.

6. Conclusions and Future Work

In the absence of specific knowledge about the type and number of noise sources, we used additive Gaussian white-noise to model the background noise. This noise model was employed by four feature selection methods to obtain four reduced feature sets. Systems based on these reduced feature sets performed sleepiness classification on the SLC test data with additive white and everyday environment noises whose SNR levels are changed dynamically following a uniform distribution. In a partially matched design, our best GFS systems showed performance improvement over the two alternative filter systems and the baseline. For further real-world noise-robustness, our GFS systems could be trained on models that incorporate actual everyday environment noises and subsequent predictions could be made in a matched manner.

7. References

[1] A. I. Pack, A. M. Pack, E. Rodgman, A. Cucchiara, D. F. Dinges, and C. W. Schwab, "Characteristics of crashes attributed to the driver having fallen asleep," *Accident Analysis & Prevention*, vol. 27, no. 6, pp. 769–775, 1995.

[2] A. T. McCartt, S. A. Ribner, A. I. Pack, and M. C. Hammer, "The scope and nature of the drowsy driving problem in new york state," *Accident Analysis & Prevention*, vol. 28, no. 4, pp. 511–517, 1996.

[3] W. Vanlaar, H. Simpson, D. Mayhew, and R. Robertson, "Fatigued and drowsy driving: A survey of attitudes, opinions and behaviors," *Journal of Safety Research*, vol. 39, no. 3, pp. 303–309, 2008.

[4] P. P. Caffier, U. Erdmann, and P. Ullsperger, "Experimental evaluation of eye-blink parameters as a drowsiness measure," *European Journal of Applied Physiology*, vol. 89, no. 3-4, pp. 319–325, 2003.

[5] J. Krajewski and B. Kröger, "Using Prosodic and Spectral Characteristics for Sleepiness Detection," in *INTERSPEECH 2007 – 8th Annual Conference of the International Speech Communication Association, August 27-31, Antwerp, Belgium, Proceedings*, 2007, pp. 1841–1844.

[6] F. Hönig, A. Batliner, T. Bocklet, G. Stemmer, E. Nöth, S. Schnieder, and J. Krajewski, "Are men more sleepy than women or does it only look like – automatic analysis of sleepy speech," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 995–999.

[7] F. Hönig, A. Batliner, E. Nöth, S. Schnieder, and J. Krajewski, "Acoustic-Prosodic Characteristics of Sleepy Speech – Between Performance and Interpretation," in *Speech Prosody 2014*, pp. 864–868.

[8] B. Schuller and A. Batliner, *Computational Paralinguistics: Emotion, Affect and Personality in Speech and Language Processing*. John Wiley & Sons, 2014.

[9] B. Schuller, S. Steidl, A. Batliner, F. Schiel, and J. Krajewski, "The INTERSPEECH 2011 Speaker State Challenge," in *INTERSPEECH 2011 – 12th Annual Conference of the International Speech Communication Association, August 28–31, Florence, Italy, Proceedings*, 2011, pp. 3201–3204.

[10] F. Eyben, M. Wöllmer, and B. Schuller, "openSMILE — The Munich Versatile and Fast Open-Source Audio Feature Extractor," in *Proc. ACM Multimedia (MM), ACM, Florence, Italy*. ACM, 2010, pp. 1459–1462.

[11] L. S. Dhupati, S. Kar, A. Rajaguru, and A. Routray, "A novel drowsiness detection scheme based on speech analysis with validation using simultaneous eeg recordings," in *Automation Science and Engineering (CASE), 2010 IEEE Conference on*. IEEE, 2010, pp. 917–921.

[12] B. Schuller, D. Arsic, F. Wallhoff, and G. Rigoll, "Emotion Recognition in the Noise Applying Large Acoustic Feature Sets," in *Speech Prosody 2006*.

[13] V. Mitra, A. Tsiartas, and E. Shriberg, "Noise and reverberation effects on depression detection from speech," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5795–5799.

[14] R. Kohavi and G. H. John, "Wrappers for feature subset selection," *Artificial intelligence*, vol. 97, no. 1–2, pp. 273–324, 1997.

[15] D. Pir and T. Brown, "Acoustic Group Feature Selection Using Wrapper Method for Automatic Eating Condition Recognition," in *INTERSPEECH 2015 – 16th Annual Conference of the International Speech Communication Association, Dresden, Germany, September 6-10, 2015, Proceedings*, 2015, pp. 894–898.

[16] R. Ruiz, J. C. Riquelme, and J. S. Aguilar-Ruiz, "Incremental wrapper-based gene selection from microarray data for cancer classification," *Pattern Recognition*, vol. 39, no. 12, pp. 2383–2392, 2006.

[17] M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten, "The weka data mining software: an update," *ACM SIGKDD explorations newsletter*, vol. 11, no. 1, pp. 10–18, 2009.

[18] C. Cortes and V. Vapnik, "Support-Vector Networks," *Machine Learning*, vol. 20, no. 3, pp. 273–297, 1995.

[19] J. Platt, "Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines," Technical Report MSR-TR-98-14, Microsoft Research, April 1998.

[20] F. Weninger, F. Eyben, B. Schuller, M. Mortillaro, and K. Scherer, "On the acoustics of emotion in audio: What speech, music, and sound have in common," *frontiers in Psychology*, vol. 4, pp. 227–239, 2013.

[21] B. Schuller, S. Steidl, and A. Batliner, "The INTERSPEECH 2009 Emotion Challenge," in *INTERSPEECH 2009 – 10th Annual Conference of the International Speech Communication Association, September 6–10, 2009, Brighton, UK, Proceedings*, 2009, pp. 312–315.

[22] A. Shahid, K. Wilkinson, S. Marcu, and C. M. Shapiro, "Karolinska sleepiness scale (kss)," in *STOP, THAT and One Hundred Other Sleep Scales*. Springer, 2012, pp. 209–210.

[23] J. Krajewski, A. Batliner, and M. Golz, "Acoustic sleepiness detection: Framework and validation of a speech-adapted pattern recognition approach," *Behavior Research Methods*, vol. 41, no. 3, pp. 795–804, 2009.

[24] M. A. Hall and L. A. Smith, "Feature Selection for Machine Learning: Comparing a Correlation-based Filter Approach to the Wrapper," in *FLAIRS Conference*, 1999, pp. 235–239.

[25] J. Lee Rodgers and W. A. Nicewander, "Thirteen Ways to Look at the Correlation Coefficient," *The American Statistician*, vol. 42, no. 1, pp. 59–66, 1988.

[26] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, "Smote: synthetic minority over-sampling technique," *Journal of artificial intelligence research*, pp. 321–357, 2002.

[27] MATLAB and Communications System Toolbox Release 2014b, The Mathworks, Inc., Natick, Massachusetts, United States.

[28] F. Eyben, F. Weninger, and B. Schuller, "Affect Recognition in Real-Life Acoustic Conditions A New Perspective on Feature Selection," in *INTERSPEECH 2013 – 14th Annual Conference of the International Speech Communication Association, August 25–29, Lyon, France, Proceedings*, 2013, pp. 2044–2048.

[29] Y. Freund and R. E. Schapire, "Large Margin Classification Using the Perceptron Algorithm," *Machine Learning*, vol. 37, no. 3, pp. 277–296, 1999.

Index of Authors