The RWTH/UPB System Combination for the CHiME 2018 Workshop

Markus Kitza[1], *Wilfried Michel*[1], *Christoph Boeddeker*[2], *Jens Heitkaemper*[2], *Tobias Menne*[1], *Ralf Schlüter*[1], *Hermann Ney*[1],*Joerg Schmalenstroeer*[2], *Lukas Drude*[2], *Jahn Heymann*[2], *Reinhold Haeb-Umbach*[2]

[1]RWTH Aachen University
[2]Paderborn University

[1]{kitza, michel, menne, schluter, ney}@i6.informatik.rwth-aachen.de,
[2]{boeddeker, heitkaemper, schmalen, drude, heymann, haeb}@nt.uni-paderborn.de

Abstract

This paper describes the systems for the single-array track and the multiple-array track of the 5th CHiME Challenge. The final system is a combination of multiple systems, using Confusion Network Combination (CNC). The different systems presented here are utilizing different front-ends and training sets for a Bidirectional Long Short-Term Memory (BLSTM) Acoustic Model (AM). The front-end was replaced by enhancements provided by Paderborn University [1]. The back-end has been implemented using RASR [2] and RETURNN [3]. Additionally, a system combination including the hypothesis word graphs from the system of the submission [1] has been performed, which results in the final best system.

1. Background

This contribution presents a system combination approach for the single-array track and the multiple-array track of the 5th CHiME Challenge. In contrast to the provided baseline system [4] the back-end has been replaced completely and is described in Section 2.2. Furthermore, additional systems using different front-ends have been developed. The front-ends are described in Section 2.1. All results presented here were achieved using the official training set and along the rules of the challenge [4].

2. Contributions

Our contributions build on the single-channel and multichannel enhancement front-end provided by Paderborn University [1]. Only the acoustic model is modified and extended with system combination. No rescoring techniques are used and all systems are trained and evaluated using the baseline lexicon and 3-gram language model.

2.1. Front-ends

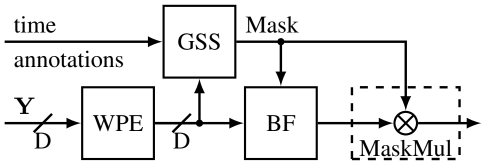

Figure 1: *Front-end with all components*

In addition to the baseline front-end (BL), another front-end has been provided by the team from Paderborn University [1]. The front-end system has a modular layout with several parts which are described in detail in [1].

2.1.1. Dereverbaration

For dereverbaration the weighted prediction error (WPE) [5] method was employed.

2.1.2. Guided source separation (GSS)

To separate the different sources we applied the complex Angular Central Gaussian Mixture Model (cACGMM) [6]. To avoid the permutation problem and to ease the estimation of the model parameters, we exploited the time annotations provided by the challenge organizers, which indicates when a particular speaker is active. These source activity patterns guide the estimation of the mixture model parameters and avoid the need to solve the frequency permutation and the global speaker permutation problem. Furthermore, it renders the estimation of the number of active sources unnecessary.

To cleanup the in-ear data we applied the GSS twice, global and utterance-wise GSS. In the first run a complete session is considered as a whole, i.e., one set of coefficients is estimated using the Expectation Maximization (EM) algorithm. This, of course, cannot account for speaker movements during the session. Therefore we ran a second utterance-wise GSS, using the parameters obtained in the first run as initialization. For array data no initialization on the session is possible therefore we used utterance-wise enhancement. When using an additional noise mask we prevented the permutation problem by adding left and right temporal context to the utterance. Thereby, introducing time slots with silence for the target speaker, which allows the algorithm to distinguish between target speaker and noise.

2.1.3. Beamforming and masking

Masks estimated from the GSS output, are used for beamforming and/or mask-based source extraction. As beamformer we employed the Minimum Variance Distortionless Response (MVDR) beamformer with Blind Analytic Normalization (BAN) [7, 8, 9]. The same setup is used for both the single array and the multiple array track, the only difference being that the number of input channels is increased from 4 to 24 in the multiple array track. An overview can be seen in Table 1.

2.2. Back-end

The Kaldi based back-end has been replaced with our own back-end using RASR [2] and RETURNN [3].

Table 1: *Overview on available front-ends.*

Name	Track	BF	Context	WPE	GSS	Mask
F0	in-ear	no	± infs	no	4 class	yes
F1	single	yes	±15s	yes	5 class	no
F2						yes
F3					4 class	
F4	multi				5 class	no
F5						yes

2.2.1. Silence normalization

To reduce the amount of silence in the training and testing data, utterance wise Bi-Gaussian modeling for speech frame selection is employed [10, 11]. In this method, first, the log-energies of each frame of an utterance are computed. Then the distribution of log-energy coefficients is estimated using a Gaussian Mixture Model (GMM) of two mixtures. The cluster corresponding to smaller value of center is treated as noise or non-speech class, and the cluster corresponding to larger value of center is considered as speech. A threshold is computed to determine the decision making boundary between speech and non-speech class. Usually, it is chosen as the point between the two centers where the probabilities are equal. Then larger blocks of non-speech audio frames are discarded such that the speech to silence ration is approximately 0.9.

2.2.2. Data filtering

In contrast to the baseline, the transcriptions are not filtered, but utterances are algorithmically selected. Two methods for filtering out noisy utterances are evaluated. First, a GMM is trained on the in-ear data. Then a search word graph is computed based on the GMM and word based confidence scores are computed using forward-backward algorithm. Only utterances where the words of the reference transcription have a confidence score above a certain threshold are selected. In the second method, framewise state alignments are calculated for each utterance and then sorted by their average maximum likelihood score. Then a percentile of segments with a lower score are discarded. On the one hand, both methods are similar in the sense, that they tend to favor shorter utterances. On the other hand they vary in their selection of utterances. If they are tuned to discard 50% of the array dataset, their selection only overlaps in 50% of the remaining utterances.

2.2.3. Acoustic model

The acoustic model is trained in the established hybrid model fashion. In a first step a linear alignment is used to start a mono-phone Hidden Markov Model (HMM) training on 16-dimensional Mel Frequency Cepstral Coefficients (MFCC) with first and second order derivatives as well as energy features. The mixture components are split twice before the training data is re-aligned using the mono-phone model. This procedure is iterated 20 times. In contrast to Kaldi, the transition probabilities are not learned from training data, but one set is selected by hand and applied to all HMMs. Another difference is the choice of covariance modeling. The HMM-GMM systems in RASR are trained with dense pooled covariance matrices.

In a second step, a state-tied triphone HMM-GMM model is trained. The states are tied using a phonetic decision tree, which optimizes the log-likelihood, where the log-likelihood for a class is modeled by a single Gaussian with a diagonal covariance matrix. The mixtures are again split two times between 16 iterations of realignments. Based on the triphone system the data is filtered as described in section 2.2.2. Based on the filtered data, the state-tying is repeated and a second HMM-GMM model is trained.

In the third step, hybrid acoustic models based on BLSTM neural networks are trained. The topology consists of six fully connected BLSTM layers with 600 cells each for the forward and the backward direction which are combined after every layer. For regularization several methods are employed: dropout [12], L_2 regularization, gradient noise [13], focal loss [14]. Adam [15] with Nesterov momentum is used for optimization. In addition learning rate scheduling is done as described in [16]. The Deep Neural Network (DNN) models are trained with 40-dimensional Mel frequency cepstral coefficients as input features and 4500 clustered triphone states as targets. In contrast to the baseline neither fMLLR, i-vectors nor sequence discriminative training is used.

3. Experimental evaluation

In this chapter we will first describe the progression of our system throughout the challenge, discussing the importance of different parameters. Afterward in sections 3.2 and 3.3 we evaluate our best single systems and go into detail about the system combination.

3.1. Finding a baseline

During the process of building a baseline system, it became apparent that several parts of our pipeline had to be reevaluated. The first intuition was replicating the Kaldi HMM-GMM baseline. We used the same segment lists as the baseline but our models did not converge in any meaningful way, staying at a word error rate around 100%. The convergence behavior improved when we switched from to original in-ear data to enhanced in-ear data F0 resulting in about 75% word error rate on the in-ear data which was still about 10% absolute worse than the Kaldi baseline GMM evaluated on the same set of in-ear data. At this point, we looked into optimizing the dimension of the state-tying tree (CART) as well as the linear discriminant analysis (LDA) dimension for constrained Maximum Likelihood Linear Regression (CMLLR). Their influence can be seen in tables 3 and 4, respectively.

Table 3: *WER (%) for different state-tying tree dimensions of HMM-GMM systems trained and evaluated on enhanced in-ear data.*

CART Dimension	2000	3000	4000	5000
WER	73.8	72.0	71.3	71.6

Table 4: *WER (%) for different LDA dimensions of HMM-GMM systems trained and evaluated on enhanced in-ear data.*

LDA Dimension	50	60	70	80
WER	73.2	73.0	71.8	74.6

Besides CART and LDA dimension we also had to double the number of splits performed during training to reach con-

Table 2: *Overview of the acoustic models used for system combination. The duration given is without any augmentation applied. The speed pertubation method inflates the data threefold.*

Name	Implementation	Topology	Training data		Augmentation	Filtering
			in-ear [hours]	array [hours]		
BL	Kaldi	TDNN	unprocessed (80)	unprocessed (32)	speed	transcription
B1	RASR/	BLSTM	enhanced (76)	unprocessed (35)	—	alginment
B2	RETURNN		enhanced (76)	enhanced (40)		confidence

vergence of the GMM-HMM. We have also observed that the model classified about 40% of all training frames as silence. This lead to a high prevalence of silence during recognition. We tried tuning the time distortion penalties to suppress silence. But this had no effect on the word error rate. Taking these considerations into account, we introduced silence normalization and utterance filtering into the system, reducing the GMM-HMM to 65% word error rate on the enhanced in-ear dataset F0.

Table 5: *WER (%) for CMLLR adaptation applied on GMM and BLSTM acoustic models on the F0 dataset.*

Model	Adaptation	WER (%)
GMM	None	73.4
GMM	CMLLR	65.6
BLSTM	None	48.3
BLSTM	CMLLR	49.5

During training of the first neural acoustic models, we observed that gains obtained by constrained maximum likelihood transformation did not carry over to the neural networks. As can be seen in Table 5, while CMLLR improves performance considerably for the GMM it even had a slightly negative impact on BLSTM performance. Therefore it was omitted for the BLSTM models. To validate the topology of the BLSTM model, the layer width and depth were optimized independently. As can be seen in Table 6, small variations in dimension of the layers has no impact on the performance of the BLSTM models. For our hyperparameter configuration, a depth of 6 layers seems to be optimal (Table 7).

Table 6: *WER (%) for different layer dimensions of the BLSTM systems trained on a mixture of enhanced in-ear and array data and evaluated on enhanced in-ear data.*

Layer width	400	500	600	800
WER	52.6	52.4	52.4	52.4

Table 7: *WER (%) for different layer dimensions of the BLSTM systems trained on a mixture of enhanced in-ear and array data and evaluated on enhanced in-ear data.*

Layer depth	5	6	7	8
WER	51.3	50.4	51.4	51.5

3.2. Single Systems

We compare three systems for system combination. The combination of training data and filtering can be seen in Table 2. Beside the provided baseline system BL, we evaluate BLSTM systems, which are trained on partly different data. All systems are evaluated on all front-ends introduced in Table 1. The results of each system evaluated on the different front-ends can be seen in Table 8. While system $B2$ performs worse on the baseline front-end it is able to outperform the baseline time delay neural networks (TDNN) on the enhanced front-ends. System $B1$ is consistently worse than the baseline system, but it provides improvements in system combination.

Table 8: *Comparison of word error rate (WER) in % between the investigated back-ends and front-ends.*

Track	Front-end	Back-end		
		BL	B1	B2
Single	BL	81.7	86.8	87.7
	F1	74.0	79.2	73.4
	F3	70.8	75.4	**68.4**
Multiple	F4	67.9	69.2	**60.7**
	F5	62.5	65.6	61.6

3.3. System combination

The system combinations are done by confusion network combination (CNC) as described in [17]. For system combination, all combinations of front-end and back-end were decoded separately and a hypothesis word graph was created for each combination. Each word graph was converted to a confusion network (CN) and aligned to the CN of the best performing system. To determine the set of systems which should be combined for optimal performance a greedy search procedure was employed.

Table 9: *Overall WER (%) for the single track system combination tested on the development set. Each row represents a (n+1)-way combination of the previous best combination and all remaining systems.*

System	2-way	3-way	4-way	5-way
+ F1–B2	66.7	**64.2**	-	-
+ F2–B2	**64.9**	-	-	-
+ F1–B1	66.5	64.5	**63.8**	-
+ F2–B1	66.5	65.6	64.8	64.8
+ F1–BL	65.6	65.0	64.5	64.8
+ F2–BL	66.2	65.3	65.0	64.7
+ F3–BL	66.6	65.7	64.9	64.9

First the best single system is combined with every other system individually and combination weights are tuned. Then the best combination is chosen and again combined with all other systems to find the best 3-way combination. This procedure is iterated until no further gains could be achieved.

Table 10: *Overall WER (%) for the multi track system combination tested on the development set. Each row represents a (n+1)-way combination of the previous best combination and all remaining systems.*

System	2-way	3-way	4-way
+F5–B2	**56.1**	-	-
+F4–B1	58.0	56.5	56.2
+F5–B1	57.9	56.4	56.2
+F4–BL	57.7	**54.6**	-
+F5–BL	56.9	55.8	55.9

Table 11: *Results of the best system combination. WER (%) per session and location together with the overall WER.*

Track	Session		Kitchen	Dining	Living	Overall
Single	Dev	S02	74.88	63.13	56.17	63.76
		S09	63.52	66.09	59.92	
	Eval	S01	79.03	56.55	71.97	62.72
		S21	69.80	49.78	56.14	
Multiple	Dev	S02	61.95	58.80	49.30	54.56
		S09	51.74	53.76	52.29	
	Eval	S01	70.31	46.89	61.07	55.26
		S21	65.51	46.57	49.36	

The results for system combination for single and multi array track can be seen in Table 9 and Table 10 respectively. We can observe reductions in word error rate of 7% and 10% for the single and multiple array tracks. Detailed results of the best system combination are listed in Table 11.

4. Conclusion

The 5th iteration of the CHiME Challenge proves to be a challenging task which necessitates many customized steps in addition to our usual speech recognition pipeline. Great improvements could be achieved by replacing and extending the acoustic preprocessing pipeline. Noise reduction in the front-end was essential for convergence of the acoustic model training. Improvements from noise reduction far outweigh any system mismatch, as the training and evaluation data sets have been processed differently. But even then, bootstrapping an initial system is a nontrivial task and requires elaborate filtering and normalization of the available training data.

System combination using confusion networks helped to further boost the performance. Here combinations from different front-ends led to higher improvements than varying only the back-end. We still need to investigate if earlier fusion of the feature streams provide similar benefits. In the end a relative reduction of word error rate of 33% compared to the baseline could be achieved.

5. Acknowledgments

The research has received funding from the European Union's Horizon 2020 research and innovation program under the Marie Skłodowska-Curie grant agreement No. 644283, project "LISTEN" as well as from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme grant agreement No. 694537. The publication reflects only the author's view and the Research Executive Agency (REA), under the power delegated by the European Commission, is not responsible for any use that may be made of the information it contains.

6. References

[1] C. Boeddeker, J. Heitkaemper, J. Schmalenstroeer, L. Drude, J. Heymann, and R. Haeb-Umbach, "Front-end processing for the CHiME-5 dinner party scenario," in *CHiME5 Workshop*, Hyderabad, India, 2018.

[2] A. R. Simon Wiesler, P. Golik, R. Schlüter, and H. Ney, "RASR/NN: The RWTH neural network toolkit for speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2014*. Florence, Italy: IEEE, 2014, pp. 3313–3317.

[3] P. Doetsch, A. Zeyer, P. Voigtlaender, I. Kulikov, R. Schlüter, and H. Ney, "RETURNN: The RWTH extensible training framework for universal recurrent neural networks," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2017*. New Orleans, USA: IEEE, Mar. 2017, pp. 5345–5349.

[4] J. Barker, S. Watanabe, E. Vincent, and J. Trmal, "The fifth CHiME Speech Separation and Recognition Challenge: Dataset, task and baselines," in *Proceedings of the 19th Annual Conference of the International Speech Communication Association (INTERSPEECH 2018)*, Hyderabad, India, Sep. 2018.

[5] T. Yoshioka and T. Nakatani, "Generalization of multi-channel linear prediction methods for blind MIMO impulse response shortening," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 20, no. 10, pp. 2707–2720, 2012.

[6] N. Ito, S. Araki, and T. Nakatani, "Complex angular central gaussian mixture model for directional statistics in mask-based microphone array signal processing," in *European Signal Processing Conference (EUSIPCO),*. IEEE, 2016, pp. 1153–1157.

[7] 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.

[8] H. Erdogan, J. R. Hershey, S. Watanabe, M. I. Mandel, and J. Le Roux, "Improved MVDR beamforming using single-channel mask prediction networks." in *Interspeech*, 2016, pp. 1981–1985.

[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] I. Magrin-Chagnolleau, G. Gravier, and R. Blouet, "Overview of the 2000-2001 ELISA consortium research activities," in *2001: A Speaker Odyssey-The Speaker Recognition Workshop*, 2001.

[11] B. Kingsbury, J. Cui, X. Cui, M. Gales, K. Knill, J. Mamou, L. Mangu, D. Nolden, M. Picheny, B. Ramabhadran, R. Schlüter, A. Sethy, and P. Woodland, "A high-performance cantonese keyword search system," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vancouver, Canada, May 2013, pp. 8277–8281.

[12] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: a simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[13] 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," *arXiv preprint arXiv:1511.06807*, 2015.

[14] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," *arXiv preprint arXiv:1708.02002*, 2017.

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

[16] A. Zeyer, P. Doetsch, P. Voigtlaender, R. Schlter, and H. Ney, "A comprehensive study of deep bidirectional LSTM RNNs for acoustic modeling in speech recognition," in *IEEE International Conference on Acoustics, Speech, and Signal Processing*, New Orleans, LA, USA, Mar. 2017, pp. 2462–2466.

[17] 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.