The JHU Multi-Microphone Multi-Speaker ASR System for the CHiME-6 Challenge

*Ashish Arora, *Desh Raj, *Aswin Shanmugam Subramanian, *Ke Li, Bar Ben-Yair, Matthew Maciejewski, Piotr Żelasko, Paola García, Shinji Watanabe, Sanjeev Khudanpur

Center for Language and Speech Processing & Human Language Technology Center of Excellence
The Johns Hopkins University, Baltimore, MD 21218, USA.
{aarora8,aswin,kli26}@jhu.edu, draj@cs.jhu.edu

Abstract

This paper summarizes the JHU team's efforts in tracks 1 and 2 of the CHiME-6 challenge for distant multi-microphone conversational speech diarization and recognition in everyday home environments. We explore multi-array processing techniques at each stage of the pipeline, such as multi-array guided source separation (GSS) for enhancement and acoustic model training data, posterior fusion for speech activity detection, PLDA score fusion for diarization, and lattice combination for automatic speech recognition (ASR). We also report results with different acoustic model architectures, and integrate other techniques such as online multi-channel weighted prediction error (WPE) dereverberation and variational Bayes-hidden Markov model (VB-HMM) based overlap assignment to deal with reverberation and overlapping speakers, respectively. As a result of these efforts, our ASR systems achieve a word error rate of 40.5% and 67.5% on tracks 1 and 2, respectively, on the evaluation set. This is an improvement of 10.8% and 10.4% absolute, over the challenge baselines for the respective tracks.

Index Terms: CHiME-6 challenge, robust speech recognition, speaker diarization, multi-channel, multi-speaker

1. Introduction

Far-field automatic speech recognition (ASR) and speaker diarization are important areas of research and have many real-world applications such as transcribing meetings [1, 2, 3, 4] and in-home conversations [5]. Although deep learning methods (including end-to-end approaches) have achieved promising results for several tasks such as Switchboard [6, 7] and LibriSpeech [8, 9], their performance remains unsatisfactory for far-field conditions in real environments, such as the CHiME-5 dataset [5]. This can be attributed to: (i) noise and reverberation in the acoustic conditions, (ii) conversational speech and speaker overlaps, and (iii) challenge-specific restrictions such as insufficient training data.

Several advances have been made in the last decade to tackle the challenges offered by real, far-field speech. For ASR, this improvement can be attributed to improved neural network architectures [10, 11], effective data augmentation techniques [12], and advances in speech enhancement [13]. Previous work has tried tackling reverberation and noise present in the far-field recording by multi-style training with data augmentation via room impulse responses and background noises [14, 15]. Recently, spectral augmentation has been successfully used for both end-to-end [16] and hybrid ASR systems [17]. Adapting the acoustic model to the environment [18] and speaker [19] has also been studied. Another popular direction is front-end based approaches such as dereverberation [20] and denoising through beamforming [21, 22], which utilize multi-microphone data. Far-field speaker diarization [23] has also benefited from enhancement methods [24, 25] and approaches to handle overlapping speech [26]. Recently, guided source separation (GSS) [13] was proposed, which makes use of additional information such as time and speaker annotations for mask estimation. However, it requires a strong diarization system to perform good separation.

In this paper, we describe a multi-microphone multi-speaker ASR system developed using many of these methods for the CHiME-6 challenge [27]. The challenge aims to improve speech recognition and speaker diarization for far-field conversational speech in challenging environments in a multi-microphone setting. The CHiME-6 data [5] contains a total of 20 4-speaker dinner party recordings. Each dinner party is two to three hours long and is recorded simultaneously on the participants' ear-worn microphone and six microphone arrays placed in the kitchen, dining room, and the living room. The challenge consists of two tracks. Track 1 allows the use of oracle start and end times of each utterance, and speaker labels for each segment. This track focuses on core ASR techniques, and measures system performance in terms of transcription accuracy. Track 2 is a "diarization+ASR" track. It additionally requires end-pointing speech segments in the recording, and assigning them speaker labels, i.e diarization. To this end, VoxCeleb2 data [28] is permitted for training a diarization system, and concatenated minimum-permutation word error rate (cpWER) is used to measure speaker-attributed transcription accuracy.

Our system for Track 2, as shown in Figure 1, consists of three main modules: *enhancement*, *diarization* and *recognition*, described in Sections 2, 3 and 4 below. The *enhancement module* performs (i) dereverberation using online multichannel weighted prediction error (WPE), followed by (ii) denoising with a weighted delay-and-sum beamformer, and (iii) multi-array guided source separation (GSS), described below. In the *diarization module*, the beamformer outputs, one per array, are used for (i) speech activity detection and (ii) speaker diarization, both of which fuse information across arrays to improve accuracy, and (iii) overlap-aware variational Bayes hidden Markov model (VB-HMM) resegmentation to assign multiple speakers to overlapped speech regions. Speaker-marks from this diarization module are used in the multi-array GSS (part of the enhancement module) to produce enhanced, speaker-separated waveforms. The *recognition module* processes the GSS output using (i) an acoustic and n-gram language model for ASR decoding, (ii) an RNN language model for lattice rescoring, and (iii) sMBR lattice combination. We augment the clean acous-

* equal contribution.

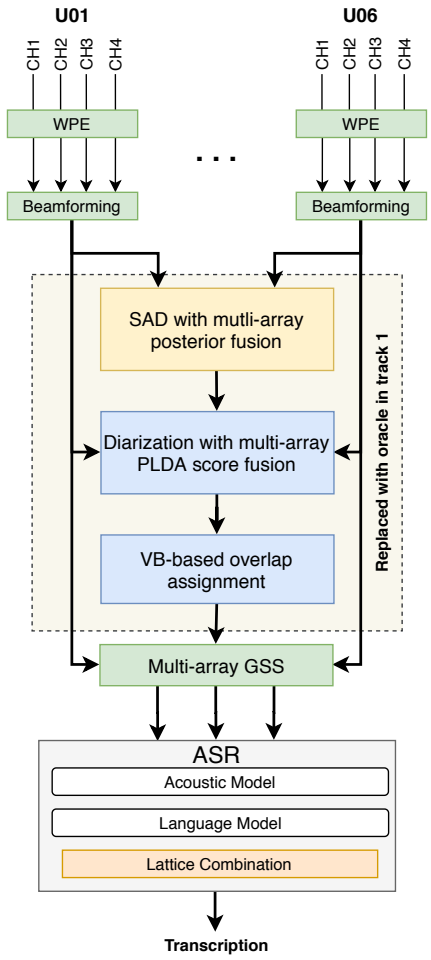

Figure 1: *Overview of the decoding pipeline for track 2. For track 1, we use a similar system, with the exception that the diarization module (shown in the dotted box in the figure) is replaced with oracle speech segments and speaker labels.*

tic training data with dereverberated, beamformed and GSS-enhanced far-field data to match the test conditions.

The diarization module is replaced with oracle speech segments and speaker labels in our system for Track 1.

2. Speech Enhancement

2.1. Dereverberation and Denoising

We used an online version of the publicly available NARA-WPE [20] implementation of weighted prediction error (WPE) based dereverberation for multi-channel signals [29] for all the channels in each array. This was followed by array-level weighted delay-and-sum beamforming using the BeamformIt tool [21]. All further processing was done on the dereverberated and beamformed signals.

2.2. Guided Source Separation (GSS)

Multi-array GSS [13, 30] was applied to enhance target speaker speech signals. For track 1, we used oracle speech segmentations and speaker labels, while for track 2, we used the segmentation estimated by the speaker diarization module described in Section 3. In GSS, the source activity pattern for each speaker derived from the segmentation aids in resolving the permutation ambiguity. A context window is used to obtain an extended segment which can bring sufficient sparsity in the activity pattern of the target speaker to further reduce speaker permutation issues. We found a 20-second context to be ideal from our experiments. GSS gave a relative WER improvement of 9.3% and 6.5% on Dev and Eval respectively for track 2 over the baseline delay-and-sum beamformer.

Table 1: *Neural network architecture for speech activity detection (SAD). T is the input length, $\mathcal{C}$ the set of output classes.*

Layer	Layer context	Total context	Input×Output
tdnn1	$[t-2, t+2]$	5	150×256
tdnn2	$[t-1, t+2]$	8	1024×256
tdnn3	$\{t-3, t, t+3, t+6\}$	17	1024×256
stats1	$[0, T)$	T	
tdnn4	$\{t-6, t, t+6, t+12\}$	T	1024×256
stats2	$[0, T)$	T	
tdnn5	$\{t-12, t, t+12, t+24\}$	T	$256 \times \lvert\mathcal{C}\rvert$

Table 2: *Reducing SAD errors by fusing posterior probabilities from multiple arrays. SAD errors are comprised of missed speech (MS) and false alarms (FA).*

System	Dev			Eval		
	MS	FA	Total	MS	FA	Total
Baseline (U06)	2.7	**0.6**	3.3	4.4	**1.5**	5.9
Posterior Mean	1.7	0.7	2.4	3.2	1.9	**5.1**
Posterior Max	**1.1**	0.8	**1.9**	**2.4**	2.8	5.2

3. Speaker Diarization

3.1. Speech Activity Detection

For speech activity detection (SAD), we first trained a neural network classifier to assign each frame in an utterance a label from $\mathcal{C} = \{silence, speech, garbage\}$. We used the architecture shown in Table 1, consisting of time-delay neural network (TDNN) layers to capture long temporal contexts [31], interleaved with stats-pooling layers to aggregate utterance-level statistics. To obtain classifier training targets, we used a speaker-independent GMM-HMM ASR system to align whole recordings with the training transcriptions. To leverage multiple channels, we carried out posterior fusion of the classifier outputs across all the arrays at test time. i.e. if $p_t^k(i)$ denotes the classifier's probability for class $i \in C$ at frame t based on array k, then $p_t(i) = f(p_t^k(i))$, where f is the fusion criterion.

We then post-processed the per-frame classifier outputs to enforce minimum and maximum speech/silence durations, by constructing a simple HMM whose state transition diagram encodes these constraints, treating the per-frame SAD posteriors $p_t(\cdot)$ like emission probabilities, and performing Viterbi decoding to obtain the most likely SAD label-sequence.

For the fusion criterion, we experimented with simple element-wise mean and max, and found them to be effective. From Table 2, we can see that applying posterior fusion across arrays improved the SAD error by approximately 34% relative. Since most of the gain comes from reduction in missed speech,

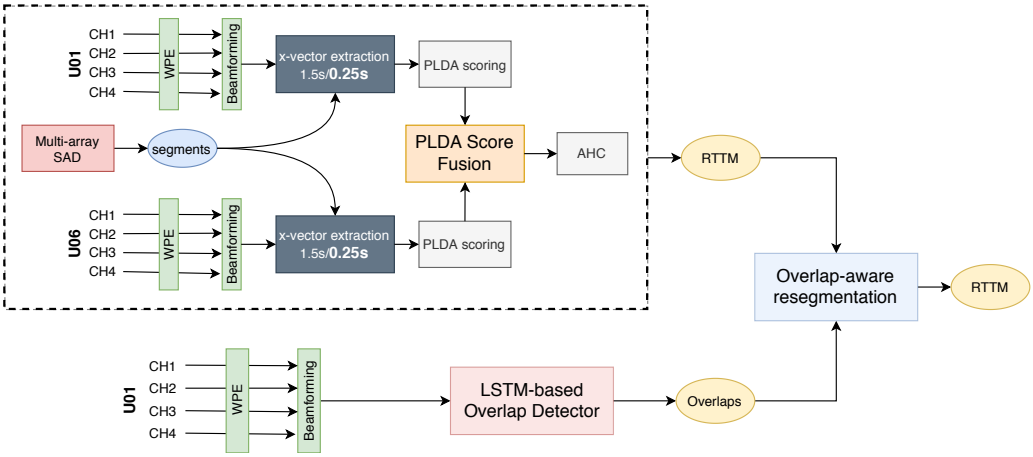

Figure 2: *The two-pass speaker diarization module. Synchronous SAD marks across microphone arrays enables PLDA score fusion before AHC in first-pass diarization (dotted rectangle). Overlap detection (bottom) enables the second-pass resegmentation, initialized by the first-pass output (upper RTTM), to assign more than one speaker to overlapped-speech regions in the final output (right RTTM).*

this also positively impacts downstream recognition rate.

We also tried microphone-level posterior fusion jointly across all the arrays, but it did not yield any improvements over using the beamformed signals as described above.

3.2. First Pass Speaker Diarization

Our first-pass diarization followed the method described in [32]. The test recordings were cut into into overlapping 1.5 second *segments* with a 0.25 second[1] stride [33], an x-vector was extracted from each segment, and agglomerative hierarchical clustering (AHC) was performed on the x-vectors using the probabilistic linear discriminant analysis (PLDA) score of each pair of x-vectors as their pairwise similarity.

The x-vector extractor we used is similar to that of [34]: it is comprised of several TDNN layers and stats-pooling, and was trained, per Challenge stipulations, on only the VoxCeleb data [28]. Data augmentation was performed by convolving the audio with simulated room impulse responses [14] and by adding background noises from the CHiME-6 training data. PLDA parameters were trained on (x-vectors of) 100k speech segments from the ca 40 speakers in the CHiME-6 training data.

Similar to the SAD posterior probability fusion described in Section 3.1, we investigated improving diarization by leveraging multiple microphone(array)s at test time. To compute the pairwise similarity of two 1.5 second segments during clustering, we fused the PLDA scores for their x-vectors extracted from different arrays. We found that multi-array PLDA score fusion, specifically the element-wise maximum across arrays, provided noticeable gains.

3.3. Overlap-Aware Resegmentation

Since AHC is not designed to handle overlapping speakers, we resegmented the audio using an *overlap-aware* version of the VB-HMM of [35]. Speaker labels from the first-stage diarization of Section 3.2 were used to initialize the per-frame speaker posterior matrix, also known as the $\boldsymbol{Q}$-matrix, and one iteration of VB-HMM inference was performed to convert this (binary) $\boldsymbol{Q}$-matrix into per-frame speaker-probabilities. Separately, we trained an *overlap detector*—a 2-layer bidirectional LSTM with SincNet input features [36] and binary (non-overlapped/overlapped speech) output labels—using the CHiME-6 training data. The per-frame decisions of the overlap detector on the test data were then used to assign each frame to either one or two most likely speakers according to $\boldsymbol{Q}$, as described in [26]. An unintended attribute of VB resegmentation was a significant number of very short segments. The computational complexity of the GSS module (cf Section 2.2) is severely impacted by this growth in the number of segments. We therefore removed all segments shorter than 200ms.

The overall diarization process is shown in Figure 2.

3.4. Diarization Performance

The CHiME Challenge provided two "ground truths" for diarization, i.e. two NIST-style rich transcription time marks (RTTM): one based on utterance-level time marks by human annotators (named Annotation RTTM), and another on forced-alignment of the acoustics to the transcripts (resp Alignment RTTM). The diarization output was scored against each RTTM using the DiHARD `dscore` toolkit[2], and diarization error rate (DER) as well as Jaccard error rate (JER) were computed.

Table 3 shows improvements in diarization performance, relative to the AHC baseline, due to PLDA score fusion, using a 0.25s stride (v/s 0.75s), and overlap-aware resegmentation. The gains from PLDA fusion across arrays appears modest and somewhat inconsistent relative to the single array (U06) DER, but was consistently better than the *average* single-array DER across the six arrays. The shorter (0.25s) x-vector stride yielded a robust improvement, and the most significant improvement came from overlap detection and multiple-speaker assignment (denoted overlap assign.)

Finally, note that diarization performance seems to degrade (particularly for the Eval data) when scored against the Alignment RTTM, but shows significant improvement with the Annotation RTTM. The former stipulates tighter speech boundaries by, for instance, marking short pauses between words as non-speech. This increases the (measured) false alarm errors of our diarization module. However, retaining such pauses is beneficial for the downstream ASR task, by inducing more appropri-

[1]The CHiME baseline system used a 0.75 second stride.

[2]https://github.com/nryant/dscore

Table 3: *Diarization performance on track 2, showing the impact of the modifications of Sections 3.2 and 3.3 to the baseline x-vector/AHC system. Some of the improvement derives from the improved SAD of Section 3.1.*

System	Dev		Eval	
	DER	JER	DER	JER
Alignment RTTM				
Baseline (U06)	63.42	70.83	**68.20**	72.54
PLDA Fusion	63.97	71.65	71.56	71.32
+ 0.25s stride	61.00	66.23	69.64	69.81
+ overlap assign.	**58.18**	**59.92**	69.92	**65.64**
Annotation RTTM				
Baseline (U06)	61.62	69.84	62.01	71.43
PLDA Fusion	60.09	70.31	62.97	70.09
+ 0.25s stride	57.85	65.36	61.60	69.35
+ overlap assign.	**50.43**	**57.81**	**58.26**	**64.38**

ate utterance segmentation.

4. Automatic Speech Recognition

We built a hybrid DNN-HMM system using algorithms and tools available in the Kaldi ASR toolkit.

4.1. Acoustic Modeling

Our baseline acoustic model (AM) was a 15-layer factorized time-delay neural network or TDNN-F [37]. Training data for this model comprised of 80h clean CHiME-6 audio from the 2 worn-microphones of the speaker of each transcribed utterance, 320h from a 4x distortion of this clean audio using synthetic room impulse responses and CHiME-6 background noises, and 200h raw far-field audio derived by randomly sampling utterances from the many arrays. These 600h were subject to 0.9x and 1.1x speed-perturbation to yield 1800h of AM training data. The baseline TDNN-F was trained with the lattice-free maximum mutual information (LF-MMI) objective. A context-dependent triphone HMM-GMM system was first trained using standard procedures in Kaldi, and frame-level forced-alignments were generated between the speech and reference transcripts to *guide* LF-MMI training [38].

Our CHiME-6 Challenge baseline system used this acoustic model and achieved a word error rate (WER) of 51.8% and 51.3% respectively on the CHiME-6 Dev and Eval test sets. We will designate this AM by (a) in this paper.

We then experimented with several other model architectures and data selection and augmentation methods.

4.1.1. Neural Network Architecture

We first trained an AM with 6 CNN layers comprising 3×3 convolutional kernels, and 16 TDNN-F layers using the 1800h training set, designated model (b) in this paper, and lowered the WERs to 49.6% and 49.3% on the Dev and Eval sets, respectively. But we discovered that these models take an inordinate amount of time to train, limiting our ability for exploration.

To expedite the turnaround time of AM training experiments, and informed by experiments in data selection and augmentation described in Section 4.1.2, we created a 500h training set comprised of the 80h of clean ear-worn microphone audio and 320h of 4x distortions of the clean audio using synthetic room impulse responses and CHiME-6 background noises, as above, supplemented with 60h raw far-field audio derived by randomly sampling utterances from the many arrays and 40h from multi-array GSS enhancement, as described in Section 2.2.

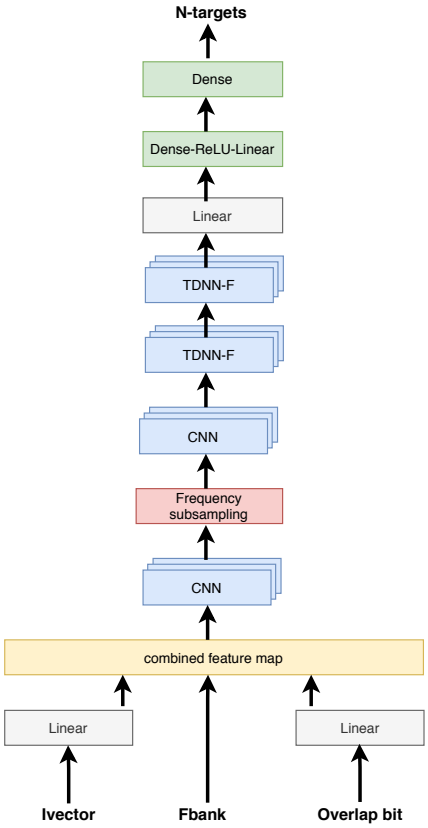

Figure 3: *Illustration of the acoustic model architecture, inputs and outputs. N-targets is the number of leaves (context dependent HMM states) in the bi-phone clustering trees.*

With this 500h training set, we experimented with the following AM architectures.

(c) **6 CNN + 12 TDNN-F**, comprised of 6 CNN layers with 3x3 convolutional kernels, followed by 12 TDNN-F layers. Frequency subsampling by a factor of two was applied at the third and fifth convolutional layers. The TDNN-F layers were 1536-dimensional with a factorized (lower) rank of 160.

(d) **6 CNN + 10 TDNN-F + 2 Self-attention**, in which the last two TDNN-F layers were replaced by self-attention layers. Each self-attention layer had 15 attention heads with key and value dimensions of 40 and 80, respectively.

(e) **CNN + TDNN + LSTM**, comprised of 6 CNN layers, followed by 9 TDNN and 3 LSTM layers, interleaved (3+1)×3. The CNN layers the are same as above, TDNN layers have 1024-dimensional hidden units, and the LSTM has a 1024-dim layer and 256-dim output and recurrent projections.

The input to these AMs were 40-dim log-Mel filterbank coefficients and the outputs were context-dependent HMM states (senones) derived from a left-biphone tree. We determined empirically that a tree with 4500 leaves, an L_2-regularization value of 0.03 for the CNN and TDNN-F layers, and 0.04 for the dense

Table 4: *Track 1 ASR WERs for AM architecture described in Section 4.1.1. The CNN + TDNN-F configuration works best.*

(Model) Architecture	Dev	Eval
(a) 15 TDNN-F	51.8%	51.3%
(b) 6 CNN + 16 TDNN-F	49.6%	49.3%
(c) 6 CNN + 12 TDNN-F	**48.3%**	**48.5%**
(d) 6 CNN + 10 TDNN-F + 2 SA	49.9%	49.4%
(e) 6 CNN + 3 $\times$ (3 TDNN + 1 LSTM)	50.1%	49.8%

Table 5: *Track 1 WERs when training the 6 CNN + 16 TDNN-F AM on only enhanced (and not reverberated, noisy) speech.*

(Model) Training Data	Dev	Eval
(b) 1800h incl. reverb'ed & raw far-field	49.6%	49.3%
(f) 675h incl. only clean & enhanced	**44.5%**	**44.9%**
(g) 675h of model (f) + sp/sil probs	45.0%	45.3%
(h) 3 $\times$ 500h from models (c)-(e)	44.6%	45.4%

layers performed well. Using 6-epochs instead of 4-epochs (in the baseline) further improved the AMs.

As shown in Table 4, the CNN + TDNN-F architecture performed better than the others, so we selected it for subsequent experiments.

4.1.2. Training Data Selection and Augmentation

Inspired by the findings of [12], where they reported good performance on CHiME data not by using the matched far-field speech, or synthetically reverberated speech, for AM training, but instead using speech enhancement during both AM training and test time, we created a new AM training data set as follows. We applied beam-forming to the 4 microphone arrays to obtain 4 $\times$ 40h of speech, obtained another 40h from multi-array GSS described in Section 2.2, and combined them with the 80h of ear-worn microphone data described above. Data clean-up was applied to this 280h data set, to remove utterances that failed forced-alignment under a narrow beam, followed by speed perturbation, resulting in a 675h AM training data set.

Note from Table 5 that careful data selection indeed confirmed the findings of [12], reducing the WER to 44.5% and 44.9% respectively on the Dev and Eval sets. The model that attained this is designated (f) in this paper.

We also trained two additional AMs, one with the same 675h of data as model (f), but with improved estimates of the inter-word silence and pronunciation probabilities (see [39]), and another with speed perturbation of the 500h data set of models (c)-(e). These models are designated (g) and (h) respectively in Table 5. While they do not perform better than model (g), they were used for system combination in track 1, as described in Table 7 below.

4.1.3. Overlap-Aware Training

Since the CHiME data have a significant proportion of overlapped speech, we looked into providing the AM a 1-bit input indicating the presence/absence of overlap. This *overlap bit* was determined during training from the time alignments, and used alongside the 40-dim filter-bank features and 100-dim i-vectors. We first projected the overlap bit to 40-dim, and the i-vector to 200-dim, and applied batch normalization and L_2 regularization. We then combined the two with the filter-bank features to create a single-channel input to the first CNN layer of model (f) described above.

Table 6: *Track 1 WERs when the presence of overlapped speech is known to the acoustic model.*

(Model) Input features	Dev	Eval
(f) log-Mel filter-bank and i-vector	44.5%	44.9%
(i) + overlapped speech indicator bit	**44.4%**	**44.5%**

The resulting model is designated (i) in Table 6, which illustrates that knowledge of the presence of overlapped speech yields a modest WER improvement in track 1 conditions. While we could have used model (i) in track 2 by using the output of the overlap detector of Section 3.3 as the overlap bit, we did not have sufficient time to carefully conduct these experiments.

4.2. Language Modeling and Rescoring

We used the training transcriptions to build our language models (LMs). We used a 3-gram LM trained with the SRILM toolkit [40] in the first pass decoding. For neural LMs, we used Kaldi to train recurrent neural network LMs (RNNLMs) [41]. We performed a pruned lattice rescoring [42] with a forward and a backward (reversing the text at the sentence level) LSTM. We first rescored with the forward LSTM and then performed another rescoring on top of the rescored lattices using the backward LSTM. Both LSTMs are 2-layer projected LSTMs with hidden and projection layer dimensions 512 and 128, respectively. We also used L_2-regularization on the embedding, hidden, and output layers.

4.3. Lattice Combination

For track 1, we used the lattice combination method to combine four CNN + TDNN-F acoustic models, as described in the data selection and augmentation section 4.1.2. All four acoustic models had similar performance in terms of WER. While three of the models were trained with close-talk and enhanced far-field data, we additionally used far-field, and RIR and noise augmented data to train the fourth model. For track 2, we performed GSS with different input signals (multi-array beamforming and individual array beamforming), followed by array level lattice combination. Diarization output was shared for all input array signals and Minimum Bayes risk decoding [43] was applied on top of the combined lattice.

5. CHiME Challenge Results

We show improvement in WER for tracks 1 and 2, obtained using the different modifications described in the previous sections, in Table 7 and Table 8, respectively.

From Table 7, we see that using CNN + TDNN-F acoustic model trained with LF-MMI objective function improved the performance over the baseline system by approx. 2% absolute. For further improvement, we trained the setup similar to test conditions by augmenting the training data with test enhancements [12]. This provided almost 5% absolute WER gain. Adding the overlap bit provided a small improvement in WER. Lattice combination and LM rescoring were individually effective, but their combination provided a significant WER improvement of 4%. Finally, we were able to obtain more than 10% absolute WER improvement over the baseline results in

Table 7: *Stepwise improvement in WERs on Track 1.*

System (Acoustic Model)	Dev (%)	Eval (%)
Baseline (a)	51.75	51.29
CNN-TDNN-F (b)	49.59	49.37
+Augmentation (f)	44.51	44.92
+Overlap Feature (i)	44.37	44.54
+LM Rescoring (i)	42.82	42.94
+Lattice combination (f)+(g)+(h)+(i)	41.75	42.07
+Lattice combination and Rescoring	40.27	40.47

Table 8: *Stepwise improvement in WERs on Track 2.*

System (Acoustic Model)	Dev (%)	Eval (%)
Baseline (a)	84.25	77.94
CNN-TDNN-F (f)	82.53	75.83
+Multi-array SAD & PLDA Fusion (f)	78.24	73.55
+GSS (f)	70.97	68.75
+VB Overlap Assignment (f)	69.28	68.78
+LM Rescoring (f)	68.69	67.94
+Lattice combination (f)	68.29	68.26
+Lattice combination and Rescoring (f)	67.76	67.51

track 1.

Table 8 shows a similar step-wise WER improvement for track 2. Again, we obtained 2% improvement using the CNN + TDNN-F architecture. Using multi-array fusion techniques in SAD and first-pass diarization reduced missed speech and speaker confusion, resulting in additional 4% and 2% improvement in the dev and eval sets, respectively. Using GSS and overlap-aware VB-HMM resegmentation provided significant improvements of 5–8%, since they allow the ASR to handle overlapping segments. Finally, LM rescoring and lattice combination provided additional gains similar to those observed in track 1. As a result of these techniques, we were able to obtain absolute WER improvement of 17% and 10% on the dev and eval sets, respectively.

6. Conclusion

We described our system for the sixth CHiME challenge for distant multi-microphone conversational speaker diarization and speech recognition in everyday home environments. We explored several methods to incorporate multi-microphone and multi-array information for enhancement, diarization, and ASR. For track 1, significant improvements in WER were obtained through data augmentation and language model rescoring. Through careful training data selection, we reduced the training time of the system 3-fold while also improving its performance. In track 2, array fusion and overlap handling in the diarization module provided better speaker segments, resulting in improved multi-array GSS enhancement. Additionaly gains were obtained through acoustic modeling and RNNLM rescoring, similar to track 1.

7. Acknowledgement

We thank Daniel Povey for the tremendous support for the system development, Jing Shi for help with data simulation in our trials with neural beamforming and Yiming Wang for help with the lattice combination setup. This work was partially supported by grants from the JHU Applied Physics Laboratory, Nanyang Technological University, and the Government of Israel, and an unrestricted gift from Applications Technology (AppTek) Inc.

8. References

[1] T. Hain, L. Burget, J. Dines, P. N. Garner, F. Grézl, A. E. Hannani, M. Huijbregts, M. Karafiát, M. Lincoln, and V. Wan, "Transcribing meetings with the AMIDA systems," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, pp. 486–498, 2012.

[2] T. Hori, S. Araki, T. Yoshioka, M. Fujimoto, S. Watanabe, T. Oba, A. Ogawa, K. Otsuka, D. Mikami, K. Kinoshita, T. Nakatani, A. Nakamura, and J. Yamato, "Low-latency real-time meeting recognition and understanding using distant microphones and omni-directional camera," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, pp. 499–513, 2012.

[3] S. Renals and P. Swietojanski, "Distant speech recognition experiments using the AMI corpus," in *New Era for Robust Speech Recognition, Exploiting Deep Learning*, 2017.

[4] T. Yoshioka, H. Erdogan, Z. Chen, X. Xiao, and F. Alleva, "Recognizing overlapped speech in meetings: A multichannel separation approach using neural networks," in *INTERSPEECH*, 2018.

[5] J. Barker, S. Watanabe, E. Vincent, and J. Trmal, "The fifth 'CHiME' speech separation and recognition challenge: dataset, task and baselines," *arXiv preprint arXiv:1803.10609*, 2018.

[6] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, and G. Zweig, "Achieving human parity in conversational speech recognition," *ArXiv*, vol. abs/1610.05256, 2016.

[7] Y. Wang, T. Chen, H. Xu, S. Ding, H. Lv, Y. Shao, N. Peng, L. Xie, S. Watanabe, and S. Khudanpur, "Espresso: A fast end-to-end neural speech recognition toolkit," *arXiv preprint arXiv:1909.08723*, 2019.

[8] C. Lüscher, E. Beck, K. Irie, M. Kitza, W. Michel, A. Zeyer, R. Schlüter, and H. Ney, "RWTH ASR systems for LibriSpeech: Hybrid vs attention - w/o data augmentation," in *INTERSPEECH*, 2019.

[9] G. Synnaeve, Q. Xu, J. Kahn, E. Grave, T. Likhomanenko, V. Pratap, A. Sriram, V. Liptchinsky, and R. Collobert, "End-to-end ASR: from supervised to semi-supervised learning with modern architectures," *ArXiv*, vol. abs/1911.08460, 2019.

[10] C. Zorila, M. Li, D. Hayakawa, M. Liu, N. Ding, and R. Doddipatla, "Toshiba's speech recognition system for the CHiME 2020 challenge."

[11] N. Kanda, R. Ikeshita, S. Horiguchi, Y. Fujita, K. Nagamatsu, X. Wang, V. Manohar, N. E. Y. Soplin, M. Maciejewski, S.-J. Chen *et al.*, "The Hitachi/JHU CHiME-5 system: Advances in speech recognition for everyday home environments using multiple microphone arrays," in *The 5th International Workshop on Speech Processing in Everyday Environments (CHiME 2018), Interspeech*, 2018.

[12] C. Zorila, C. Boeddeker, R. Doddipatla, and R. Haeb-Umbach, "An investigation into the effectiveness of enhancement in ASR training and test for CHiME-5 dinner party transcription," *arXiv preprint arXiv:1909.12208*, 2019.

[13] 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.

[14] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5220–5224.

[15] Y. Wang, D. Snyder, H. Xu, V. Manohar, P. S. Nidadavolu, D. Povey, and S. Khudanpur, "The JHU ASR system for VOiCES from a Distance challenge 2019," *Proc. Interspeech 2019*, pp. 2488–2492, 2019.

[16] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "Specaugment: A simple data augmentation method for automatic speech recognition," *arXiv preprint arXiv:1904.08779*, 2019.

[17] W. Zhou, W. Michel, K. Irie, M. Kitza, R. Schlüter, and H. Ney, "The RWTH ASR system for TED-LIUM release 2: Improving hybrid HMM with SpecAugment," *ArXiv*, vol. abs/2004.00960, 2020.

[18] M. L. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *2013 IEEE international conference on acoustics, speech and signal processing*. IEEE, 2013, pp. 7398–7402.

[19] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*. IEEE, 2013, pp. 55–59.

[20] L. Drude, J. Heymann, C. Boeddeker, and R. Haeb-Umbach, "NARA-WPE: A python package for weighted prediction error dereverberation in Numpy and Tensorflow for online and offline processing," in *Speech Communication; 13th ITG-Symposium*. VDE, 2018, pp. 1–5.

[21] 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.

[22] T. Nakatani and K. Kinoshita, "A unified convolutional beamformer for simultaneous denoising and dereverberation," *IEEE Signal Processing Letters*, vol. 26, pp. 903–907, 2019.

[23] P. García, J. Villalba, H. Bredin, J. Du, D. Castán, A. Cristia, L. Bullock, L. Guo, K. Okabe, P. S. Nidadavolu, S. Kataria, S. Chen, L. Galmant, M. Lavechin, L. Sun, M.-P. Gill, B. Ben-Yair, S. Abdoli, X. Wang, W. Bouaziz, H. Titeux, E. Dupoux, K. A. Lee, and N. Dehak, "Speaker detection in the wild: Lessons learned from JSALT 2019," *ArXiv*, vol. abs/1912.00938, 2019.

[24] L. Sun, J. Du, X. Zhang, T. Gao, X. Fang, and C.-H. Lee, "Progressive multi-target network based speech enhancement with SNR-preselection for robust speaker diarization," in *ICASSP 2020*, 2020.

[25] S. Kataria, P. S. Nidadavolu, J. Villalba, N. Chen, P. García, and N. Dehak, "Feature enhancement with deep feature losses for speaker verification," *ArXiv*, vol. abs/1910.11905, 2020.

[26] L. Bullock, H. Bredin, and L. P. Garcia-Perera, "Overlap-aware diarization: resegmentation using neural end-to-end overlapped speech detection," *arXiv preprint arXiv:1910.11646*, 2019.

[27] S. Watanabe, M. Mandel, J. Barker, and E. Vincent, "CHiME-6 challenge: Tackling multispeaker speech recognition for unsegmented recordings," *arXiv preprint arXiv:2004.09249*, 2020.

[28] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612*, 2017.

[29] T. Nakatani, T. Yoshioka, K. Kinoshita, M. Miyoshi, and B.-H. Juang, "Speech dereverberation based on variance-normalized delayed linear prediction," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1717–1731, 2010.

[30] N. Kanda, C. Boeddeker, J. Heitkaemper, Y. Fujita, S. Horiguchi, K. Nagamatsu, and R. Haeb-Umbach, "Guided source separation meets a strong ASR backend: Hitachi/Paderborn university joint investigation for dinner party ASR," *arXiv preprint arXiv:1905.12230*, 2019.

[31] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *INTERSPEECH*, 2015.

[32] G. Sell, D. Snyder, A. McCree, D. Garcia-Romero, J. Villalba, M. Maciejewski, V. Manohar, N. Dehak, D. Povey, S. Watanabe *et al.*, "Diarization is hard: Some experiences and lessons learned for the JHU team in the inaugural DIHARD challenge." in *Interspeech*, vol. 2018, 2018, pp. 2808–2812.

[33] F. Landini, S. Wang, M. Diez, L. Burget, P. Matějka, K. Žmolíková, L. Mošner, O. Plchot, O. Novotnỳ, H. Zeinali *et al.*, "BUT system description for DIHARD speech diarization challenge 2019," *arXiv preprint arXiv:1910.08847*, 2019.

[34] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.

[35] M. Díez, L. Burget, and P. Matejka, "Speaker diarization based on Bayesian HMM with eigenvoice priors," in *Odyssey*, 2018.

[36] M. Ravanelli and Y. Bengio, "Speaker recognition from raw waveform with SincNet," *2018 IEEE Spoken Language Technology Workshop (SLT)*, pp. 1021–1028, 2018.

[37] D. Povey, G. Cheng, Y. Wang, K. Li, H. Xu, M. Yarmohammadi, and S. Khudanpur, "Semi-orthogonal low-rank matrix factorization for deep neural networks," in *Proc. Interspeech 2018*, 2018, pp. 3743–3747. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2018-1417

[38] D. Povey, V. Peddinti, D. Galvez, P. Ghahremani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for asr based on lattice-free mmi," in *Interspeech 2016*, 2016, pp. 2751–2755. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2016-595

[39] G. Chen, H. Xu, M. Wu, D. Povey, and S. Khudanpur, "Pronunciation and silence probability modeling for asr," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[40] A. Stolcke, "SRILM-an extensible language modeling toolkit," in *Seventh International Conference on Spoken Language Processing*, 2002.

[41] H. Xu, K. Li, Y. Wang, J. Wang, S. Kang, X. Chen, D. Povey, and S. Khudanpur, "Neural network language modeling with letter-based features and importance sampling," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2018, pp. 6109–6113.

[42] H. Xu, T. Chen, D. Gao, Y. Wang, K. Li, N. Goel, Y. Carmiel, D. Povey, and S. Khudanpur, "A pruned RNNLM lattice-rescoring algorithm for automatic speech recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5929–5933.

[43] 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.