CHiME-6 Challenge: Tackling multispeaker speech recognition for unsegmented recordings

Shinji Watanabe, Michael Mandel, Jon Barker, Emmanuel Vincent, Ashish Arora, Xuankai Chang, Sanjeev Khudanpur, Vimal Manohar, Daniel Povey, Desh Raj, et al.

CHiME-6 Challenge: Tackling Multispeaker Speech Recognition for Unsegmented Recordings

[1]*Shinji Watanabe,* [2]*Michael Mandel,* [3]*Jon Barker,* [4]*Emmanuel Vincent*
[1]*Ashish Arora,* [1]*Xuankai Chang,* [1]*Sanjeev Khudanpur,* [1]*Vimal Manohar,* [1]*Daniel Povey,* [1]*Desh Raj,*
[1]*David Snyder,* [1]*Aswin Shanmugam Subramanian,* [1]*Jan Trmal,* [1]*Bar Ben Yair,* [5]*Christoph Boeddeker,*
[2]*Zhaoheng Ni,* [6]*Yusuke Fujita,* [6]*Shota Horiguchi,* [7]*Naoyuki Kanda,* [7]*Takuya Yoshioka,* [8]*Neville Ryant*

[1]Johns Hopkins University, USA, [2]The City University of New York, USA, [3]University of Sheffield, UK, [4]Inria, France, [5]Paderborn University, Germany, [6]Hitachi, Ltd., Japan, [7]Microsoft, USA, [8]Linguistic Data Consortium, USA

shinjiw@ieee.org

Abstract

Following the success of the 1st, 2nd, 3rd, 4th and 5th CHiME challenges we organize the 6th CHiME Speech Separation and Recognition Challenge (CHiME-6). The new challenge revisits the previous CHiME-5 challenge and further considers the problem of distant multi-microphone conversational speech diarization and recognition in everyday home environments. Speech material is the same as the previous CHiME-5 recordings except for accurate array synchronization. The material was elicited using a dinner party scenario with efforts taken to capture data that is representative of natural conversational speech. This paper provides a baseline description of the CHiME-6 challenge for both segmented multispeaker speech recognition (Track 1) and unsegmented multispeaker speech recognition (Track 2). Of note, Track 2 is the first challenge activity in the community to tackle an unsegmented multispeaker speech recognition scenario with a complete set of reproducible open source baselines providing speech enhancement, speaker diarization, and speech recognition modules.

Index Terms: CHiME challenge, speech recognition, speech enhancement, speech separation, speaker diarization, computational paralinguistics

1. Introduction

Automatic speech recognition (ASR) performance in difficult reverberant and noisy conditions has improved tremendously in the last decade [1–6]. This can be attributed to advances in speech processing, audio enhancement, and machine learning, but also to the availability of real speech corpora recorded in cars [7, 8], quiet indoor environments [9, 10], noisy indoor and outdoor environments [11, 12], and challenging broadcast media [13, 14]. Among the applications of robust ASR, voice command in domestic environments has attracted a great deal of interest recently, due in particular to the release of the Amazon Echo, Google Home and other devices targeting home automation and multimedia systems. The CHiME-1 [15] and CHiME-2 [16] challenges and corpora have contributed to popularizing research on this topic, together with the DICIT [17], Sweet-Home [18], and DIRHA [19] corpora. These corpora feature single-speaker reverberant and/or noisy speech recorded or simulated in a single home, which precludes the use of modern speech enhancement techniques based on machine learning. The two voiceHome corpora [20, 21] address this issue, but they are fairly small.

In parallel to research on acoustic robustness, research on conversational speech recognition has also made great progress, as illustrated by the recent announcements of super-human performance [22, 23] achieved on the Switchboard telephone conversation task [24] and by the ASpIRE challenge [25]. Distant-microphone recognition of noisy, overlapping, conversational speech is now widely believed to be the next frontier. Early attempts in this direction can be traced back to the ICSI [26], CHIL [27], and AMI [28] meeting corpora, the LLSEC [29] and COSINE [30] face-to-face interaction corpora, and the Sheffield Wargames corpus [31]. These corpora were recorded using advanced microphone array prototypes which are not commercially available, and as result could only be installed in a few laboratory rooms. The VOiCES corpus [32] utilizes an ad-hoc array of commercial microphones, with pre-recorded speech and noise played over speakers. The DIPCO corpus [33], inspired by the CHiME-5 challenge [34] but of shorter duration, provides recordings of dinner table interactions between four participants recorded simultaneously on several commercially available microphone arrays. The Santa Barbara Corpus of Spoken American English [30] stands out as the only large-scale corpus of naturally occurring spoken interactions between a wide variety of people recorded in real everyday situations including face-to-face or telephone conversations, card games, food preparation, on-the-job talk, story-telling, and more. Unfortunately, it was recorded via a single microphone.

The CHiME-6 challenge, which builds upon CHiME-5 [34], targets the problem of distant microphone conversational speech recognition in everyday home environments. The speech material has been collected from twenty real dinner parties that have taken place in real homes. The recordings have been made using multiple commercially available 4-channel microphone arrays and have been fully transcribed. The challenge features:

- simultaneous recordings from multiple microphone arrays;
- real conversation, i.e. talkers speaking in a relaxed and unscripted fashion;
- a range of room acoustics from 20 different homes each with two or three separate recording areas;
- real domestic background noises, e.g., kitchen appliances, air conditioning, movement, etc.

In the following, we introduce the recording scenario and the two proposed challenge tracks in Section 2. We describe the software baselines and the challenge instructions for Tracks 1 and 2 in Sections 3 and 4, respectively. We report the corresponding results in Section 5 and conclude in Section 6. More details can be found on the challenge website:

https://chimechallenge.github.io/chime6/

2. Scenario and tracks

2.1. The scenario

The dataset is made up of the recording of twenty separate dinner parties taking place in real homes. Each dinner party has four participants — two acting as hosts and two as guests. The party members are all friends who know each other well and who are instructed to behave naturally. Efforts have been taken to make the parties as natural as possible. The only constraints are that each party should last a minimum of 2 hours and should be composed of three phases, each corresponding to a different location:

- kitchen: preparing the meal in the kitchen area
- dining: eating the meal in the dining area
- living: a post-dinner period in a separate living room area

Participants have been allowed to move naturally from one location to another but with the instruction that each phase should last at least 30 minutes. Participants are free to converse on topics of their choosing — there is no artificial scenario. Some personally identifying material has been redacted post-recording as part of the consent process. Background television and commercial music has been disallowed in order to avoid capturing copyrighted content.

2.2. The recording set up

Each party has been recorded with a set of six Microsoft Kinect devices. The devices have been strategically placed such that there are always at least two capturing the activity in each location. Each Kinect device has a linear array of 4 sample-synchronised microphones and a camera. The raw microphone signals and video have been recorded. Each Kinect is recorded onto a separate laptop computer.

In addition to the Kinects, to facilitate transcription, each participant is wearing a set of Soundman OKM II Classic Studio binaural microphones. The audio from these is recorded via a Soundman A3 adapter onto Tascam DR-05 stereo recorders being worn by the participants. The recordings have been divided into training, development test, and evaluation test sets. Each set features non-overlapping homes and speakers. For more details about these datasets, see [34].

2.3. Tracks

For the first time, the challenge moves beyond automatic speech recognition (ASR) and also considers the task of diarization, i.e., estimating the start and end times and the speaker label of each utterance. The challenge features two tracks:

1. ASR only: recognise a given evaluation utterance given ground truth diarization information,
2. diarization+ASR: perform both diarization and ASR

Both tracks are multi-array, i.e., all microphones of all arrays can be used. Track 1 is a rerun of the CHiME-5 challenge [34] and Track 2 is similar to the "Diarization from multichannel audio using system SAD" track of the DIHARD II challenge [35], with the following key differences:

- an **accurate array synchronization** script is provided,
- the **impact of diarization error on speech recognition error** will be measured,
- **upgraded, state-of-the-art baselines** are provided for diarization, enhancement, and recognition.

These baselines and related implementations are integrated in the Kaldi speech recognition toolkit [36] as a recipe.

For each track, we will produce two separate ASR rankings:

A Systems based on conventional acoustic modeling and official language modeling: the outputs of the acoustic model must remain frame-level tied phonetic (senone) targets and the lexicon and language model must not be changed compared to the conventional ASR baseline,

B All other systems, including systems based on the end-to-end ASR baseline or systems whose lexicon and/or language model have been modified.

Ranking **A** focuses on acoustic robustness only, while ranking **B** addresses all aspects of the scenario.

3. Track 1

Concerning Track 1, we provide baseline systems for array synchronization, speech enhancement, and speech recognition. All systems are integrated in the Kaldi CHiME-6 recipe[1].

3.1. Overview

The main script (`run.sh`) executes array synchronization, data preparation, data augmentation, feature extraction, Gaussian mixture model - hidden Markov model (GMM-HMM) training, data cleaning, and chain model training. After training, `run.sh` calls the inference script (`local/decode.sh`), which includes speech enhancement and recognition given the trained model. Participants can also execute `local/decode.sh` independently with their own ASR models or pre-trained models downloaded from the Kaldi model storage site[2]. Detailed technical descriptions of system components can be found in [37]. We outline the process below.

1. Array synchronization (stage 0)
   This stage first downloads the array synchronization tool, and generates the synchronized audio files across arrays along with their corresponding JSON files. Note that this requires sox v14.4.2, which is installed via miniconda in `./local/check_tools.sh`. Details of the array synchronization procedure are presented in Section 3.2.
2. Data, dictionary, and language model (stages 1–3)
   These stages prepare data directories, the lexicon, and language models in the format expected by Kaldi. The lexicon has a 127,712 word vocabulary. We use a maximum entropy-based 3-gram language model, which achieves the best perplexity on the development set.
3. Data augmentation (stages 4–7)
   In these stages, we augment and fix the training data. Point source noises are extracted from the noise regions in the CHiME-6 corpus. Here, we use a subset of 400 k utterances from the array microphones, their augmentations, and all worn microphone utterances during training. We did not include enhanced speech data for training to maintain the simplicity of the system.
4. Feature extraction (stage 8)
   We extract 13-dimensional Mel-frequency cepstral coefficient (MFCC) features for GMM-HMM systems.

[1] https://github.com/kaldi-asr/kaldi/tree/master/egs/chime6/s5_track1

[2] http://kaldi-asr.org/models/m12

5. GMM training (stages 9–13)
   These stages train monophone and triphone GMM-HMM models. These models are used for cleaning training data and generating lattices for training the chain model.
6. Data cleaning (stage 14)
   This stage performs cleanup of the training data using the GMM model.
7. Chain model training (stage 15)
   We use a factorized time delay neural network (TDNN-F) adapted from the Switchboard recipe 7q model [38].
8. Decoding (stage 16)
   This stage performs speech enhancement and recognition for the test set. This stage calls `local/decode.sh`, which includes speech enhancement (described in Section 3.3) and decoding and scoring (described in Section 3.4).

3.2. Array synchronization

The new array synchronisation baseline is available on GitHub[3]. It compensates for two separate issues: *audio frame-dropping* (which affects the Kinect devices only) and *clock-drift* (which affects all devices). It operates in the following two stages:

1. *Frame-dropping* is compensated by inserting 0's into the signals where samples have been dropped. These locations have been detected by comparing the Kinect audio with an uncorrupted stereo audio signal recovered from the video AVI files that were recorded (but not made publicly available). The frame-drop locations have been precomputed and stored in the file `chime6_audio_edits.json`, which is then used to drive the synchronisation software.
2. *Clock-drift* is computed by comparing each device's signal to the session's 'reference' binaural recordings (the binaural mic of the speaker with the lowest ID number). Specifically, cross-correlation is used to estimate delays between the device and the reference at regular intervals throughout the recording session. A relative speed-up or slow-down can then be approximated using a linear fit through these estimates. The signal is then synchronised to the reference using a `sox` command to adjust the speed of the signal appropriately. This adjustment is typically very subtle, i.e., less than 100 ms over a 2.5 h recording session. Note, the approach failed for devices *S01_U02* and *S01_U05* which appear to have temporarily changed speeds during the recording session and have required a piece-wise linear fit. The adjustments for clock-drift compensation have been precomputed and the parameters to drive the `sox` commands are stored in `chime6_audio_edits.json`.

Note, after frame-drop and clock-drift compensation, the WAV files that are generated for each device will have slightly different durations. For each session, device signals can be safely truncated to the duration of the shortest signal across devices, but this step is not performed by the synchronisation tool.

Finally, the CHiME-5 transcript JSON files are processed to fit the new alignment. In the new version, utterances will have the *same* start and end time on every device.

[3] https://github.com/chimechallenge/chime6-synchronisation

3.3. Speech enhancement

We provide two baseline speech enhancement front-ends based on open-source implementations of guided source separation (GSS) [39] and BeamformIt [40], respectively. Both of them are combined with an open source version [41] of weighted prediction error (WPE) dereverberation [42], and integrated into our Kaldi recipe. They can be installed in the Kaldi tool installation directory.

The first front-end consists of WPE, a spatial mixture model that uses time annotations (GSS), beamforming. They are applied to multiple arrays. GSS is performed with the setup of `multiarray=outer_array_mics` meaning that only the first and last microphones of each array are used. This is the default speech enhancement front-end for the CHiME-6 Track 1 recipe.

The alternative front-end applies WPE based dereverberation and weighted delay-and-sum beamforming (BeamformIt) to the reference array. Users can easily switch from GSS to BeamformIt by specifying the enhancement option (e.g., `--enhancement beamformit`).

3.4. Decoding and scoring

We perform two-stage decoding, which refines i-vector extraction based on the first pass decoding result to achieve robust decoding of noisy speech [37]. We also provide a scoring script for both development and evaluation: `local/score_for_submit.sh`. The language model weight and insertion penalty are optimized based on the development set.

Note that, during scoring, we filter the tags ([noise], [inaudible], [laughs], and [redacted]), and normalize ambiguous filler words[4].

4. Track 2

Concerning Track 2, we provide baseline systems for array synchronization, speech enhancement, speech activity detection (SAD), speaker diarization, and speech recognition. All systems are integrated in the Kaldi CHiME-6 recipe[5].

4.1. Overview

The main script (`run.sh`) is similar to `run.sh` in Track 1 as described in Section 3, which performs array synchronization, data preparation, data augmentation, feature extraction, GMM-HMM training, data cleaning, and chain model training. `run.sh` in Track 2 additionally includes SAD model training on the CHiME-6 dataset, and diarization model training on the VoxCeleb dataset [43]. We allow the participants to use VoxCeleb in addition to CHiME-6 data, since it is necessary to build a good diarization system.

After training, `run.sh` finally calls the inference script (`local/decode.sh`), which performs speech enhancement, SAD, speaker diarization, and speech recognition based on the trained models. Participants can also execute `local/decode.sh` independently with their own SAD, diarization, and ASR models or pre-trained models.

[4] For example, we perform the following replacements to filter out variants of the filler word 'hmm': `sed -e 's/\<mhm\>/hmm/g; s/\<mm\>/hmm/g; s/\<mmm\>/hmm/g;'`. The actual filtering rules can be found in `local/wer_output_filter`.

[5] https://github.com/kaldi-asr/kaldi/tree/master/egs/chime6/s5_track2

Stages 1–7 are the same as those of Track 1, as described in Section 3.1. But these are followed by new subsequent stages:

1. SAD training
   We use a TDNN+LSTM (long short-term memory) model trained on the CHiME-6 dataset with ground truth alignments obtained by a GMM-HMM. Participants can also download a pretrained SAD model[6].
2. Diarization training
   An x-vector neural diarization model [44] is trained with the VoxCeleb data [43]. This script is adapted from the Kaldi VoxCeleb v2 recipe. A probabilistic linear discriminant analysis (PLDA) model [45] is trained on the CHiME-6 dataset. Participants can also download a pretrained diarization model[7].
3. Decoding and scoring (stage 16)
   In Track 2, only raw recordings are given without segment or speaker information; i.e., `local/decode.sh` has to perform the whole pipeline consisting of speech enhancement, SAD, speaker diarization, and ASR decoding and scoring. These steps are detailed below.

4.2. Array synchronization

Track 2 uses the exact same array synchronization technique as described in Section 3.2.

4.3. Speech enhancement

Unlike Track 1, Track 2 only provides the BeamformIt-based speech enhancement front-end (see Section 3.3) due to the risk of degradation in GSS performance using estimated diarization information instead of ground truth speech segment information (which is unavailable).

4.4. Speech activity detection

The SAD baseline relies on the neural architecture in [46]. It was trained using data (`train_worn_u400k`) from 1) the CHiME-6 worn microphone utterances and 2) a randomly selected subset of 400 k array microphone utterances. We generate speech activity labels using an HMM-GMM system trained with the `train_worn_simu_u400k` data from 1) the CHiME-6 worn microphone utterances perturbed with various room impulse responses generated from a room simulator and 2) a randomly selected subset of 400 k array microphone utterances.

As a neural network architecture, we use 40-dimensional MFCC features as input, 5 TDNN layers, and 2 layers of statistics pooling [46]. The overall context of the network is set to be around 1 s, with around 0.8 s of left context and 0.2 s of right context. The network is trained with a cross-entropy objective to predict speech/non-speech labels.

During inference, SAD labels for the test recordings are obtained by Viterbi decoding using an HMM with minimum duration constraints of 0.3 s for speech and 0.1 s for silence. We also prepared an SAD decoding script to evaluate the SAD performance on the CHiME-6 data. Note that the baseline system only performs SAD (and all other post-processing steps including speaker diarization and ASR) for the U06 array for simplicity. Exploring multi-array fusion techniques for SAD, diarization, and ASR is an integral part of the challenge.

[6] http://kaldi-asr.org/models/12/0012_sad_v1.tar.gz

[7] http://kaldi-asr.org/models/12/0012_diarization_v1.tar.gz

4.5. Speaker diarization

The speaker diarization baseline relies on the segment files obtained by SAD. It is an x-vector system [47] with a 5-layer TDNN trained on the VoxCeleb dataset [43]. PLDA is trained on CHiME-6 data (`train_worn_simu_u400k`). Agglomerative hierarchical clustering (AHC) [48] is performed. Since the number of speakers in CHiME-6 is four in every session, this prior information is used by AHC.

Our speaker diarization system consistently uses the reference RTTM converted from the original JSON file via data preparation (`run.sh --stage 1`) by using the Kaldi RTTM conversion script[8]. The diarization result is also obtained as an RTTM file, and the diarization error rate (DER) and Jaccard error rate (JER) are computed using dscore[9] (used in the DIHARD II challenge).

Similar to the SAD system, this baseline system only performs diarization for the U06 array for simplicity.

4.6. Decoding and scoring

The RTTM files obtained by speaker diarization in Section 4.5 are converted to the Kaldi data format. We perform two-stage decoding, which refines the i-vector extraction based on the first pass decoding result to achieve robust decoding for noisy speech [37]. Again, the baseline system only performs ASR for the U06 array for simplicity.

We provide a scoring script for both development and evaluation. The language model weight and insertion penalty are optimized based on the development set. Multispeaker scoring is performed to obtain the concatenated minimum-permutation word error rate (cpWER).

The cpWER is computed as follows:

1. Concatenate all utterances of each speaker for both reference and hypothesis files.
2. Compute the WER between the reference and all possible speaker permutations of the hypothesis. There are 24 such permutations.
3. Pick the lowest WER among them (this is assumed to be the best permutation).

cpWER is directly affected by the speaker diarization results. In addition to the cpWER, which shows the error rate of entire recordings, we also report detailed errors per utterance by recovering the utterance information from the reference.

4.7. RTTM refinement

In the original CHiME-5 annotations, utterance boundaries are marked by human annotators, among other information, in an RTTM file. While these utterance boundaries are sufficient for training and testing ASR, there are utterances that include long pauses between words, making them an imperfect reference for diarization. To obtain a more precise diarization reference, we apply forced alignment between the transcripts and the cleaner binaural recordings.

The acoustic model we use for forced alignment is the triphone GMM-HMM model trained in the baseline system (see Section 3.1). We use `steps/align_si.sh` to align the worn (binaural) microphone recordings for both the development set

[8] https://github.com/kaldi-asr/kaldi/blob/master/egs/wsj/s5/steps/segmentation/convert_utt2spk_and_segments_to_rttm.py

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

	Dev. WER	Eval. WER
CHiME-6 baseline	51.8%	51.3%
CHiME-5 baseline [34]	81.1%	73.3%
CHiME-5 top system [49]	45.6%	46.6%

Table 1: *CHiME-6 Track 1 baseline ASR results, compared to the baseline and top systems for the (equivalent) CHiME-5 multiple device track.*

and the evaluation set. The decoding beam size is 20 by default, but if this search fails we perform a second alignment with a beam size of 150. These alignment results are at the word level. In order to merge them into utterances, we identify all contiguous words separated by gaps of at most 300 ms of silence. Instances of [noise] also separate utterances. These alignments are then saved in the RTTM format as the refined reference for SAD and diarization.

5. Baseline results

5.1. Track 1

Table 1 presents the Track 1 baseline ASR results with the official 3-gram language model (corresponding to category A). The CHiME-6 baseline is significantly better than the baseline and close to the best system [49] for the CHiME-5 multiple device track, which is equivalent to CHiME-6 Track 1. Note that the CHiME-6 baseline is designed to be a compromise between performance and simplicity, e.g., we purposefully did not use system combination but the result is still close to [49], which is based on complex multi-path enhancement processing with system combination.

5.2. Track 2

Table 2 shows the SAD performance obtained by the CHiME-6 baseline SAD system, as described in Section 4.4. It lists the SAD results computed against both human-annotated (old) and force-aligned (new) RTTMs, as described in Section 4.7. This result shows that the evaluation set is more difficult than the development set in terms of SAD performance. Also, the new and old RTTMs have significant differences in the development set while they have marginal difference in the evaluation set.

Table 3 shows the DER and JER obtained by the CHiME-6 baseline speaker diarization system, as described in Section 4.5. In spite of using a state-of-the-art diarization technique [44], both metrics show over 60% error rates and improving the diarization performance is one of the main challenges in Track 2.

Finally, Table 4 shows the performance gap between the Track 1 and Track 2 baselines. The main differences between the Track 1 and 2 baselines come from the use of advanced speech enhancement (GSS [39], as described in Section 3.3) and the use of speech segmentation from manual annotations or automatic speaker diarization. Note that both tracks use the same acoustic and language models. Therefore, by comparing Tracks 1 and 2 with BeamformIt, we can observe that the main degradation (around 15% absolute) comes from speaker diarization.

6. Summary

This paper describes the CHiME-6 challenge outline, baselines, and experimental results. Newly introduced audio synchronization and a state-of-the-art Kaldi baseline simplify challenge entry for Track 1, while Track 2 significantly increases the difficulty due to the need for speaker diarization. To help the challenge participants tackle these difficulties, we provide a complete set of open source Kaldi recipes for both Track 1 and Track 2 which combine speech enhancement, speaker diarization, and speech recognition. This is the first trial in the community to provide open source recipes for unsegmented multispeaker ASR, and a lot of effort has been provided through volunteer activities by speech separation and recognition researchers in addition to the challenge organizers. Our future work is to provide the analysis of this challenge including the investigation of the effectiveness of the techniques proposed during the challenge, a review of evaluation metrics, the relationship between diarization and speech recognition errors, and so on. Through this analysis, we would like to make significant progress toward this challenging, realistic, and unsolved multispeaker speech processing problem.

7. References

[1] T. Virtanen, B. Raj, and R. Singh, Eds., *Techniques for Noise Robustness in Automatic Speech Recognition*. Wiley, 2012.

[2] J. Li, L. Deng, R. Haeb-Umbach, and Y. Gong, *Robust Automatic Speech Recognition — A Bridge to Practical Applications*. Elsevier, 2015.

[3] S. Watanabe, M. Delcroix, F. Metze, and J. R. Hershey, Eds., *New Era for Robust Speech Recognition — Exploiting Deep Learning*. Springer, 2017.

[4] E. Vincent, T. Virtanen, and S. Gannot, Eds., *Audio Source Separation and Speech Enhancement*. Wiley, 2018.

[5] S. Makino, Ed., *Audio Source Separation*. Springer, 2018.

[6] R. Haeb-Umbach, S. Watanabe, T. Nakatani, M. Bacchiani, B. Hoffmeister, M. L. Seltzer, H. Zen, and M. Souden, "Speech processing for digital home assistants: Combining signal processing with deep-learning techniques," *IEEE Signal Processing Magazine*, vol. 36, no. 6, pp. 111–124, 2019.

[7] A. Moreno, B. Lindberg, C. Draxler, G. Richard, K. Choukri, S. Euler, and J. Allen, "SPEECHDAT-CAR. a large speech database for automotive environments," in *Proc. 2nd Int. Conf. on Language Resources and Evaluation (LREC)*, 2000.

[8] J. H. L. Hansen, P. Angkititrakul, J. Plucienkowski, S. Gallant, U. Yapanel, B. Pellom, W. Ward, and R. Cole, ""CU-Move": Analysis & corpus development for interactive in-vehicle speech systems," in *Proc. Eurospeech*, 2001, pp. 2023–2026.

[9] L. Lamel, F. Schiel, A. Fourcin, J. Mariani, and H. Tillman, "The translingual English database (TED)," in *Proc. 3rd Int. Conf. on Spoken Language Processing (ICSLP)*, 1994.

[10] E. Zwyssig, F. Faubel, S. Renals, and M. Lincoln, "Recognition of overlapping speech using digital MEMS microphone arrays," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 7068–7072.

[11] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third 'CHiME' speech separation and recognition challenge: Analysis and outcomes," *Computer Speech and Language*, vol. 46, pp. 605–626, 2017.

[12] 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*, vol. 46, pp. 535–557, 2017.

[13] G. Gravier, G. Adda, N. Paulsson, M. Carré, A. Giraudel, and O. Galibert, "The ETAPE corpus for the evaluation of speech-based TV content processing in the French language," in *Proc. 8th Int. Conf. on Language Resources and Evaluation (LREC)*, 2012, pp. 114–118.

	Dev.			Eval.		
	Missed speech	False alarm	Total error	Missed speech	False alarm	Total error
Annotation RTTM	2.5%	0.8%	3.3%	4.1%	1.8%	5.9%
Alignment RTTM	1.9%	0.7%	2.6%	4.3%	1.5%	5.8%

Table 2: *CHiME-6 Track 2 baseline SAD results. Annotation RTTM is the original RTTM obtained by human annotation while alignment RTTM is based on forced alignment and is considered as the official RTTM file for the challenge.*

	Dev.		Eval.	
	DER	JER	DER	JER
Annotation RTTM	61.6%	69.8%	62.0%	71.4%
Alignment RTTM	63.4%	70.8%	68.2%	72.5%

Table 3: *CHiME-6 Track 2 baseline diarization results. Annotation RTTM is the original RTTM obtained by human annotation while alignment RTTM is based on forced alignment and is considered as the official RTTM file for the challenge.*

	Enhancement	Segmentation	Dev. WER	Eval. WER
Track 1	BeamformIt	Oracle	69.8%	61.2%
Track 1	GSS	Oracle	51.8%	51.3%
Track 2	BeamformIt	Diarization	84.3%	77.9%

Table 4: *CHiME-6 Track 1 and 2 baseline ASR results with BeamformIt-based [40] and GSS-based [39] speech enhancement. We used the same acoustic and language models for both tracks.*

[14] P. Bell, M. J. F. Gales, T. Hain, J. Kilgour, P. Lanchantin, X. Liu, A. McParland, S. Renals, O. Saz, M. Wester, and P. C. Woodland, "The MGB challenge: Evaluating multi-genre broadcast media recognition," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2015, pp. 687–693.

[15] J. Barker, E. Vincent, N. Ma, H. Christensen, and P. Green, "The PASCAL CHiME speech separation and recognition challenge," *Computer Speech and Language*, vol. 27, no. 3, pp. 621–633, 2013.

[16] E. Vincent, J. Barker, S. Watanabe, J. Le Roux, F. Nesta, and M. Matassoni, "The second CHiME speech separation and recognition challenge: Datasets, tasks and baselines," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2013, pp. 126–130.

[17] A. Brutti, L. Cristoforetti, W. Kellermann, L. Marquardt, and M. Omologo, "WOZ acoustic data collection for interactive TV," in *Proc. 6th Int. Conf. on Language Resources and Evaluation (LREC)*, 2008, pp. 2330–2334.

[18] M. Vacher, B. Lecouteux, P. Chahuara, F. Portet, B. Meillon, and N. Bonnefond, "The Sweet-Home speech and multimodal corpus for home automation interaction," in *Proc. 9th Int. Conf. on Language Resources and Evaluation (LREC)*, 2014, pp. 4499–4509.

[19] M. Ravanelli, L. Cristoforetti, R. Gretter, M. Pellin, A. Sosi, and M. Omologo, "The DIRHA-English corpus and related tasks for distant-speech recognition in domestic environments," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2015, pp. 275–282.

[20] N. Bertin, E. Camberlein, E. Vincent, R. Lebarbenchon, S. Peillon, É. Lamandé, S. Sivasankaran, F. Bimbot, I. Illina, A. Tom, S. Fleury, and E. Jamet, "A French corpus for distant-microphone speech processing in real homes," in *Proc. Interspeech*, 2016, pp. 2781–2785.

[21] N. Bertin, E. Camberlein, R. Lebarbenchon, E. Vincent, S. Sivasankaran, I. Illina, and F. Bimbot, "VoiceHome-2, an extended corpus for multichannel speech processing in real homes," *Speech Communication*, vol. 106, pp. 68–78, 2019.

[22] W. Xiong, J. Droppo, X. Huang, F. Seide, M. L. Seltzer, A. Stolcke, D. Yu, and G. Zweig, "Toward human parity in conversational speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 12, pp. 2410–2423, 2017.

[23] G. Saon, G. Kurata, T. Sercu, K. Audhkhasi, S. Thomas, D. Dimitriadis, X. Cui, B. Ramabhadran, M. Picheny, L.-L. Lim, B. Roomi, and P. Hall, "English conversational telephone speech recognition by humans and machines," in *Proc. Interspeech*, 2017, pp. 132–136.

[24] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "SWITCHBOARD: Telephone speech corpus for research and development," in *Proc. IEEE International Conf. on Acoustics, Speech, and Signal Proc. (ICASSP)*, vol. 1, 1992, pp. 517–520.

[25] M. Harper, "The automatic speech recognition in reverberant environments (ASpIRE) challenge," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2015, pp. 547–554.

[26] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters, "The ICSI meeting corpus," in *Proc. IEEE Int. Conf. on Acoustics, Speech, and Signal Processing (ICASSP)*, 2003, pp. 364–367.

[27] D. Mostefa, N. Moreau, K. Choukri, G. Potamianos, S. Chu, A. Tyagi, J. Casas, J. Turmo, L. Cristoforetti, F. Tobia, A. Pnevmatikakis, V. Mylonakis, F. Talantzis, S. Burger, R. Stiefelhagen, K. Bernardin, and C. Rochet, "The CHIL audiovisual corpus for lecture and meeting analysis inside smart rooms," *Language Resources and Evaluation*, vol. 41, no. 3–4, pp. 389–407, 2007.

[28] S. Renals, T. Hain, and H. Bourlard, "Interpretation of multiparty meetings: The AMI and AMIDA projects," in *Proc. 2nd Joint Workshop on Hands-free Speech Communication and Microphone Arrays (HSCMA)*, 2008, pp. 115–118.

[29] LLSEC, "Lincoln laboratory speech enhancement corpus," https://www.ll.mit.edu/mission/cybersec/HLT/corpora/SpeechCorpora.html, 1996.

[30] A. Stupakov, E. Hanusa, D. Vijaywargi, D. Fox, and J. Bilmes, "The design and collection of COSINE, a multi-microphone in situ speech corpus recorded in noisy environments," *Computer Speech and Language*, vol. 26, no. 1, pp. 52–66, 2011.

[31] C. Fox, Y. Liu, E. Zwyssig, and T. Hain, "The Sheffield wargames corpus," in *Proc. Interspeech*, 2013, pp. 1116–1120.

[32] C. Richey, M. A. Barrios, Z. Armstrong, C. Bartels, H. Franco, M. Graciarena, A. Lawson, M. K. Nandwana, A. Stauffer, J. van Hout, P. Gamble, J. Hetherly, C. Stephenson, and K. Ni, "Voices obscured in complex environmental settings (VOiCES) corpus," in *Proc. Interspeech*, 2018, pp. 1566–1570.

[33] M. Van Segbroeck, A. Zaid, K. Kutsenko, C. Huerta, T. Nguyen, X. Luo, B. Hoffmeister, J. Trmal, M. Omologo, and R. Maas, "DiPCo–dinner party corpus," *arXiv preprint arXiv:1909.13447*, 2019.

[34] J. Barker, S. Watanabe, E. Vincent, and J. Trmal, "The fifth 'CHiME' speech separation and recognition challenge: Dataset, task and baselines," in *Proc. Interspeech*, 2018, pp. 1561–1565.

[35] N. Ryant, K. Church, C. Cieri, A. Cristia, J. Du, S. Ganapathy, and M. Liberman, "The second DIHARD diarization challenge: Dataset, task, and baselines," *Proc. Interspeech*, pp. 978–982, 2019.

[36] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz, J. Silovský, G. Stemmer, and K. Veselý, "The Kaldi speech recognition toolkit," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2011.

[37] V. Manohar, S. J. Chen, Z. Wang, Y. Fujita, S. Watanabe, and S. Khudanpur, "Acoustic modeling for overlapping speech recognition: JHU CHiME-5 challenge system," in *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6665–6669.

[38] 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, pp. 3743–3747.

[39] 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 *Proc. 5th Int. Workshop on Speech Processing in Everyday Environments (CHiME)*, 2018, pp. 35–40.

[40] 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. 2011–2021, 2007.

[41] 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 *ITG Fachtagung Sprachkommunikation (ITG)*, 2018.

[42] 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.

[43] A. Nagrani, J. S. Chung, W. Xie, and A. Zisserman, "Voxceleb: Large-scale speaker verification in the wild," *Computer Speech and Language*, vol. 60, p. 101027, 2020.

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

[45] P. Kenny, "Bayesian speaker verification with heavy tailed priors," in *Proc. Odyssey*, 2010.

[46] P. Ghahremani, V. Manohar, D. Povey, and S. Khudanpur, "Acoustic modelling from the signal domain using CNNs," in *Proc. Interspeech*, 2016, pp. 3434–3438.

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

[48] X. Anguera, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, "Speaker diarization: A review of recent research," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 2, pp. 356–370, 2012.

[49] J. Du, T. Gao, L. Sun, F. Ma, Y. Fang, D.-Y. Liu, Q. Zhang, X. Zhang, H.-K. Wang, J. Pan, J.-Q. Gao, C.-H. Lee, and J.-D. Chen, "The USTC-iFlytek systems for CHiME-5 challenge," in *Proc. 5th Int. Workshop on Speech Processing in Everyday Environments (CHiME)*, 2018, pp. 11–15.