# BUT System for CHiME-6 Challenge

*Kateřina Žmolíková, Martin Kocour*, Federico Landini*, Karel Beneš*, Martin Karafiát*, Hari Krishna Vydana, Alicia Lozano-Diez, Oldřich Plchot, Murali Karthick Baskar, Ján Švec, Ladislav Mošner, Vladimír Malenovský, Lukáš Burget, Bolaji Yusuf, Ondřej Novotný, František Grézl, Igor Szöke, and Jan "Honza" Černocký*

Brno University of Technology, BUT Speech@FIT and IT4I Centre of Excellence, Czechia

izmolikova@fit.vutbr.cz

## Abstract

This paper describes BUT's efforts in the development of the system for the CHiME-6 challenge with far-field dinner party recordings [1]. Our experiments are on both diarization and speech recognition parts of the system. For diarization, we employ the VBx framework which uses Bayesian hidden Markov model with eigenvoice priors on x-vectors. For acoustic modeling, we explore using different subsets of data for training, different neural network architectures, discriminative training, more robust i-vectors, and semi-supervised training on VoxCeleb data. Besides, we perform experiments with a neural network-based language model, exploring how to overcome the small size of the text corpus and incorporate across-segment context. When fusing our best systems, we achieve 41.21% / 42.55% WER on Track 1, for development and evaluation respectively, and 55.15% / 69.04% on Track 2, for development and evaluation respectively. Aside from techniques used in our final submitted systems, we also describe our efforts in end-to-end diarization and end-to-end speech recognition.

## 1. Introduction

CHiME-6 challenge [1] is a continuation of a series of CHiME challenges in automatic speech recognition (ASR) in difficult environments. It uses the dataset recorded for CHiME-5 [2] with new array synchronization and updated task definitions. The dataset consists of recordings of dinner parties in real homes. The parties were recorded with 6 distant, 4-channel microphone arrays. The main difficulties of the dataset are spontaneity of speech, high overlap ratio, and the presence of background noise.

For CHiME-6 challenge, two tasks were defined. In Track 1, the task is speech recognition of the distant microphone recordings. For this task, the oracle segmentation was provided and no external data were allowed to be used. In Track 2, the task is analogous, but no segmentation is provided, thus diarization needs to be performed. To make the diarization task feasible, Track 2 allowed the use of VoxCeleb [3, 4] data. Besides, both tracks were split into Category A and Category B, where Category A is restricted to hybrid ASR systems with baseline language model, while Category B covers everything else.

In this paper, we summarize the contributions of Brno University of Technology for the challenge. We focused mainly on diarization, acoustic modeling and language modeling.

## 2. Contributions

In this section, we describe both the methods used for the final submitted systems (in 2.1, 2.2, 2.3, 2.4) and additional experiments with end-to-end approaches (2.5, 2.6).

### 2.1. Diarization

For Track 2, we based our diarization on agglomerative hierarchical clustering (AHC) of x-vectors, followed by another x-vector clustering based on Bayesian hidden Markov model and variational Bayes inference (VBx[1]). This approach was successfully applied to the Second DIHARD Challenge [5, 6, 7] and we adapted it to comply with the rules of the CHiME-6 challenge in terms of allowed augmentations for training the x-vector extractor. We used the speech activity detection (SAD), x-vector extractor and probabilistic linear discriminant analysis (PLDA) modules from the baseline recipe [8, 9]. Moreover, we used only the enhanced recording from kinect U06 as in the baseline recipe. However, we extracted the x-vectors every 0.25 s instead of 0.75 s for having seen improvements previously [7].

Due to the high number of x-vectors to cluster, we used a two-step AHC: we divided the x-vectors from the whole recording into smaller groups accounting for several minutes, clustered each one individually and finally performed clustering with all the clusters to obtain four final clusters.

With a similar configuration but using a final threshold that allows for underclustering, we obtained the AHC-based initialization for VBx which was in turn run until convergence [7].

We further refined the diarization by a three-step procedure: First, we provided the speaker labels from the first VBx diarization to guided source separation (GSS) [10] to obtain four recordings where each speaker is enhanced in one of them. Then we ran the VBx diarization on each of the enhanced recordings and finally, we pooled the labels corresponding to the respective enhanced speakers to produce the final diarization. Note that it is possible for some of the segments that are speech according to the SAD labels not to have any speaker assigned, we assigned to them the speaker from the original VBx output as depicted in Figure 1.

The performance of the methods is summarized in Table 1. The baseline approach used x-vectors extracted every 0.75 s and a single step of AHC to produce the diarization output.

*Equal contribution.

[1] https://github.com/BUTSpeechFIT/VBx

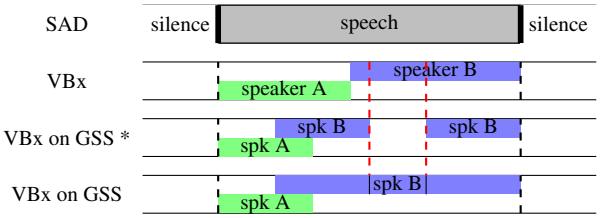


Figure 1: *Label handling with GSS outputs.*

Table 1: *Comparison of different diarization methods in terms of Diarization Error Rate (DER) and Jaccard error rate (JER).*

| | Development | | Evaluation | |
|---|---|---|---|---|
| | DER | JER | DER | JER |
| Baseline | 63.42 | 70.83 | 68.20 | 72.54 |
| 2-step AHC | 60.21 | 65.21 | 71.84 | 71.80 |
| VBx | 51.67 | 53.20 | 75.11 | 71.77 |
| VBx on GSS[2] | 51.44 | 48.45 | 80.57 | 66.33 |

### 2.2. Enhancement

For the speech enhancement module, we have used the GSS method provided by the baseline. We applied the enhancement also on training data, using the oracle segmentation, and added the data to our training set for both Track 1 and Track 2 (referred to as *enhanced* in Section 2.3). In Track 2, we used the segmentation estimated by VBx diarization as guidance for GSS. For estimation of the masks, we used 40 seconds context on each side and the beamforming filters were estimated and applied every 5 seconds. Table 2 summarizes how the diarization and enhancement affect the automatic speech recognition (ASR) performance in Track 2. Results are obtained with the system corresponding to the first row in Table 4. Note that there is a slight inconsistency between these results caused by a different setting of decoding.

Table 2: *Impact of diarization and enhancement on the ASR performance on development set in Track 2. Column Diarization refers to the diarization used for ASR. We always used VBx diarization for guidance in GSS.*

| Diarization | Enhancement | WER [%] |
|---|---|---|
| VBx | BeamformIt | 73.4 |
| VBx | GSS | 62.9 |
| VBx on GSS | GSS | 59.6 |

### 2.3. Acoustic model

We explored the impact of different subsets of training data on the performance of the acoustic model. The first combination consisted of the left microphone from worn data with all far-field data enhanced by GSS (*Worn (L) + enhanced*). Secondly, we used both worn microphones with enhanced data as in [11] (*Worn (S) + enhanced*). We further enlarged this dataset by adding worn data augmented with artificial room impulse responses [12] (*+ WornRVB*). The final combination used worn data from both microphones, enhanced data by GSS and a subset of 250k non-overlapped parts of far-field data (*+250k non-overlapped*). With all subsets, we applied speed perturbation and data cleaning as in the baseline.

Table 3 compares the results of these combinations. Increasing the amount of data by adding more microphones was beneficial. Reverberation and adding far-field data still improves the performance. However, we decided to use *Worn (S) + enhanced* combination in other experiments, since the size of this combination was relatively small, while the improvement was similar compared to the larger combinations.

In all experiments in this section, the acoustic model was based on a convolutional time-delay neural network with semi-orthogonal factorization (CNN-TDNNf) [13]. The baseline enhancement is used for Track 1 and GSS enhancement and VBx on GSS diarization is used for Track 2.

Table 3: *Comparison of WER [%] on the development set when training the acoustic model with different data.*

| | | | Track 1 | Track 2 |
|---|---|---|---|---|
| 1 | Worn (L) + enhanced | (201 h) | 48.94 | - |
| | (1) + w/o cleaning | (265 h) | 49.14 | - |
| 2 | Worn (S) + enhanced | (308 h) | 47.85 | 59.29 |
| 3 | (2) + WornRVB | (1047 h) | 47.57 | 59.22 |
| 4 | (3) + 250k non-overlapped | (1329 h) | 47.31 | 59.02 |

We also explored other approaches for improving the performance of the acoustic model which we present in Table 4. Firstly, we re-transcribed the training data by basic CNN-TDNNf system and system re-trained on pruned (beam 3) output lattices (*full-lattices*). Second, we extended the lattice-free maximum mutual information with state-level minimum Bayes risk (sMBR). We also considered incorporating segments from VoxCeleb [4, 3] data[3] shorter than 5 seconds for semi-supervised training. Finally, in order to capture the speaker movement, we replaced the baseline offline i-vector extraction by the online version. In doing so, we dropped the baseline pseudo-speakers and treated each person as a single speaker. Finally, we estimated a secondary offline i-vectors stream only on non-overlapped parts to help the system to do speaker separation.

Table 4: *Improvements of acoustic model using discriminative training, semi-supervised training on VoxCeleb and 2-stream i-vectors on the development set in terms of WER [%].*

| | | Track 1 | Track 2 |
|---|---|---|---|
| 5 | CNN-TDNNf | 47.85 | 59.29 |
| 6 | (5) + full-lattices | 47.54 | 59.23 |
| 7 | (5) + sMBR | 47.32 | 58.82 |
| 8 | (7) + VoxCeleb | 46.80 | 57.92 |
| 9 | (7) + speaker + online i-vector | 46.63 | 58.46 |
| 10 | (7) + non-overlapped + online i-vector | 46.47 | - |

[2]In S01 we obtained five speakers so we slightly decreased VBx's $F_A$ for that recording since we know that would allow for less speakers.

[3]This data was allowed only for Track 2 but we analyzed this approach on Track 1 too.

### 2.4. Language model

To improve over the baseline count-based language model, we have trained an LSTM language model (LSTM-LM) using BrnoLM toolkit[4].

In order to cope with the tiny size of the available training data, we have combined two regularization techniques during training: the standard dropout along non-recurrent connections and random replacement of input tokens. We have obtained the best results with dropout 0.5 and input corruption rate 0.3.

We rescored 3000-best hypotheses from the ASR lattices, carrying over the hidden state between segments of each speaker [14]. Table 5 shows the improvements achieved with model *Worn (S) + enhanced + WornRVB + 250k non-overlapped* from Table 3[5]. The gains were similar with other acoustic models.

Table 5: *Results of rescoring the development set of Track 1 with LSTM-LM consisting of two 650-units layers. Perplexity is for the each LM separately, WER is in interpolation of the respective LSTM-LM with the baseline.*

| | Perplexity | WER [%] |
|---|---|---|
| baseline | 157.7 | 48.24 |
| + LSTM | 152.1 | 46.94 |
| + across-segment | 136.5 | 46.61 |
| + input corruption | 131.1 | 46.08 |

### 2.5. Towards end-to-end diarization

We developed some end-to-end diarization systems based on [15]. The neural network architecture is based on the encoder from a transformer trained to directly perform diarization. The network is trained via permutational invariant training (PIT) to minimize the binary cross entropy. The output of the network allows overlap and a threshold over the outputs is used to obtain the final diarization decisions.

In Table 6, we show a summary of the results for different end-to-end diarization systems on the development set. For the systems in the first three rows, We used just the channel 1 (CH1) of the training data, which gave us better performance than using all of them. We obtained improvements when we deleted the first minute in the training data, since the introductions from the speakers were not properly labeled. We also observed that pretraining the system with simulated data of two speakers created from VoxCeleb, and then fine-tuning it to the challenge training data removing the last layer and increasing the number of speakers to four, provided further improvements. The last two rows show the results when fine-tuning with one audio file per kinect (instead of just channel 1), obtaining the mixed audio either from the original audio files or after applying the weighted prediction error (WPE) [16, 17]. Finally, we would like to point out that results degraded notably when scoring our systems with the new reference labels (RTTM) provided by the organizers during the evaluation period[6]. This might be due to a mismatch between training annotations and new annotations provided only for the development set.

[4]https://github.com/BUTSpeechFIT/BrnoLM

[5]The corresponding WER there is further improved by MBR decoding.

[6]The new references were obtained for the development and evaluation sets after using forced alignment in contrast with the original labels as released in CHiME-5.

Table 6: *Results of end-to-end diarization DNN on the development set of Track 2. Results deleting first minute of recordings (Yes/No), pre-training the network with VoxCeleb (Yes/No) and with old and new RTTMs are presented in terms of DER (%).*

| Data | Del 1min | VoxCeleb pretrain | DER Old RTTMs | DER New RTTMs |
|---|---|---|---|---|
| CH1 | N | N | 70.3 | 80.6 |
| CH1 | N | Y | 64.6 | 73.9 |
| CH1 | Y | Y | 63.6 | 71.7 |
| mix | Y | Y | 63.5 | 71.7 |
| WPE+mix | Y | Y | **62.4** | **70.9** |

### 2.6. Towards end-to-end speech recognition

We have explored end-to-end networks for developing ASR systems. Our end-to-end ASR systems were LSTM-based encoder-decoder networks and transformer networks as in [18, 19]. All the ASR systems were trained with characters as target units and no external language models were used. 80-dimensional Mel-filter bank features with deltas and double deltas were used as features. The models were early stopped using the dev-worn set, and a beam size of 10 was used when decoding to obtain the hypotheses.

Table 7: *Performances of various end-to-end ASR systems in terms of WER [%] on the dev-worn data and on the dev-far-field data.*

| Model | Dev-worn | Dev-far-field |
|---|---|---|
| LSTM-Attention | 60.19 | 66.51 |
| Transformer 1 | 66.06 | 73.39 |
| Transformer 2 | 64.66 | 68.70 |
| Transformer 3 | 61.60 | 66.70 |

Table 7 presents the results for end-to-end models. LSTM-Attention refers to an LSTM-based encoder-decoder network with 5 encoder layers and 1 decoder layer with a hidden-layer of size 320. Each encoder layer comprises a bi-directional LSTM followed by a linear projection layer. A dropout of 0.3 was used after each B-LSTM in the encoder layer. The decoder consists of a unidirectional-LSTM with a dropout of 0.1 and it uses an additive attention mechanism as in [20]. The model was initially trained on the worn-data and further trained on far-field data.

All transformer models had the same architecture as described in [19] but differ on the training mechanisms. The models have 12 encoder layers and 6 decoder layers with a hidden-layer of size 256 and a feed-forward layer of size 2048. They use 4 attention heads.

Transformer 1 refers to a model trained on the worn data. This model is further trained on far-field data and we refer to it as Transformer 2 in the table. Finally, Transformer 3 refers to a model whose encoder is initially pre-trained with auto-regressive predictive coding (APC) loss on VoxCeleb data as described in [21]. The model is further trained on both worn and far-field.

From the results in Table. 7, the LSTM-Attention model performs better than the transformer models. APC pre-training

on VoxCeleb data improves the performance with respect to only using CHiME 6 data. APC pre-training was tried on the LSTM-Attention network but the model did not converge. The reasons for this behavior need to be further investigated. The performance of our end-to-end ASR systems is not on par with the hybrid-ASR systems presented in Table 4 and for this reason we present the results separately.

## 3. Final systems

For the final systems for both Track 1 and Track 2, we used ROVER [22] fusion over different acoustic models. In all systems, we used GSS enhancement of test data, and for Track 2, VBx on GSS diarization.

For Track 1, we fused 8 systems: (3) + (4) + (5) + (6) + (7) + (9) + (10) (in Tables 3 and 4) + CNN-TDNNf system trained on *Worn (S) + enhanced + 250k non-overlapped* with full-lattices.

For Track 2, we fused 7 systems: (3) + (4) + (5) + (6) + (7) + (8) + (9) (in Tables 3 and 4). The final results in the submission format prescribed by the challenge are in Tables 8 and 9.

Table 8: *Final results on Track 1, Categories A and B.*

| | Development WER | Evaluation WER |
|---|---|---|
| Category A | 42.75 | 44.34 |
| Category B | 41.21 | 42.55 |

Table 9: *Final results on Track 2, Categories A and B.*

| | Development | | | Evaluation | | |
|---|---|---|---|---|---|---|
| | DER | JER | WER | DER | JER | WER |
| Category A | 51.44 | 48.45 | 55.60 | 80.57 | 66.33 | 69.17 |
| Category B | | | 55.15 | | | 69.04 |

## 4. References


[1] S. Watanabe, M. Mandel, J. Barker, E. Vincent, A. Arora, X. Chang, S. Khudanpur, V. Manohar, D. Povey, D. Raj, D. Snyder, A. S. Subramanian, J. Trmal, B. B. Yair, C. Boeddeker, Z. Ni, Y. Fujita, S. Horiguchi, N. Kanda, and T. Yoshioka, "CHiME-6 Challenge: Tackling multispeaker speech recognition for unsegmented recordings," in *6th International Workshop on Speech Processing in Everyday Environments (CHiME 2020)*, 2020.

[2] 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.

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

[4] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," *arXiv preprint arXiv:1806.05622*, 2018.

[5] N. Ryant, K. Church, C. Cieri, A. Cristia, J. Du, S. Ganapathy, and M. Liberman, "The second DIHARD diarization challenge: Dataset, task, and baselines," *arXiv preprint arXiv:1906.07839*, 2019.

[6] F. Landini, S. Wang, M. Diez, L. Burget, P. Matějka, K. Žmolíková, L. Mošner, A. Silnova, O. Plchot, O. Novotný, H. Zeinali, and J. Rohdin, "BUT System for DIHARD Speech Diarization Challenge 2019," in *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

[7] M. Diez, L. Burget, F. Landini, S. Wang, and H. Černocký, "Optimizing Bayesian HMM based x-vector clustering for the second DIHARD speech diarization challenge," in *Proceedings of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020.

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

[9] 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 *Interspeech*. ISCA, 2018, pp. 2808–2812.

[10] 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.

[11] 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," 2019.

[12] 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)*, 2017, pp. 5220–5224.

[13] 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 *Proceedings of Interspeech*, 09 2018, pp. 3743–3747.

[14] K. Irie, A. Zeyer, R. Schlüter, and H. Ney, "Training Language Models for Long-Span Cross-Sentence Evaluation," in *IEEE Automatic Speech Recognition and Understanding Workshop*, Sentosa, Singapore, Dec. 2019, pp. 419–426.

[15] Y. Fujita, N. Kanda, S. Horiguchi, K. Nagamatsu, and S. Watanabe, "End-to-end neural speaker diarization with permutation-free objectives," 2019.

[16] 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, Sept 2010.

[17] 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 *13. ITG Fachtagung Sprachkommunikation*, Oct 2018.

[18] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, "Hybrid ctc/attention architecture for end-to-end speech recognition," *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.

[19] S. Karita, N. Chen, T. Hayashi, T. Hori, H. Inaguma, Z. Jiang, M. Someki, N. E. Y. Soplin, R. Yamamoto, X. Wang, S. Watanabe, T. Yoshimura, and W. Zhang, "A comparative study on transformer vs rnn in speech applications," 2019.

[20] J. K. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Advances in neural information processing systems*, 2015, pp. 577–585.

[21] S. Ling, Y. Liu, J. Salazar, and K. Kirchhoff, "Deep contextualized acoustic representations for semi-supervised speech recognition," *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020. [Online]. Available: http://dx.doi.org/10.1109/ICASSP40776.2020.9053176

[22] J. G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (ROVER)," in *1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings*. IEEE, 1997, pp. 347–354.