STCON System for the CHiME-7 Challenge

Tatiana Prisyach, Yuri Khokhlov, Maxim Korenevsky, Anton Mitrofanov, Tatiana Timofeeva, Ilia Odegov, Rauf Nasretdinov, Iurii Lezhenin, Dmitriy Miroshnichenko, Arsenii Karelin, Mariya Mitrofanova, Roman Svechnikov, Sergei Novoselov, Aleksei Romanenko

STCON LLC., Kingdom of Saudi Arabia

{prisyach, khokhlov, korenevsky, mitrofanov-aa, timofeeva, odegov, nasretdinov, lezhenin, miroshnichenko, karelin, korenevskaya, svechnikov, novoselov, romanenko}@speechpro.com

Abstract

This paper describes the STCON system for the CHiME-7 challenge Task 1 (DASR) aimed at distant automatic speech transcription and segmentation with multiple recording devices. The system is generally similar to the Speech Technology Center (STC) system for the CHiME-6 challenge but uses more sophisticated and advanced models for diarization and ASR. Carefully designed pipeline provides significant improvements compared to the baseline system.

Index Terms: speech recognition, speaker diarization, WPE, GSS, TS-VAD, WavLM, E-branchformer, Pruned Stateless Transducer, CHiME-7.

1. Introduction

The CHiME-7 challenge continues the series of challenges on multichannel speech processing in everyday environments [1, 2, 3]. Compared to the previous challenge, several changes were introduced. The Task1 (Distant ASR, DASR) subsumed the whole CHiME-6 task, namely diarization and recognition of highly overlapped multi-talker speech recorded on multiple microphones, but training data was extended with partially annotated Mixer6 Speech corpus [4] and development/test data was extended by both Mixer6 and Amazon Dinner Party Corpus (DiPCo) [5] data. Additionally, it was allowed to utilize several external datasets as well as pre-trained ASR/Speaker recognition models. In the Acoustic Robustness (AR) sub-track participants were provided with oracle speaker-wise segmentation of speech while in the Main track the segmentation should have been obtained automatically via diarization process. Accordingly, in AR sub-track the Speaker Attributed WER (SA-WER) was used as a main metric, while in the Main Track the Diarization/Jaccard Error Rate (DER/JER) and more strict Diarization-Aware WER (DA-WER) were used instead. Importantly, the single system should have been used for all datasets and any prior classification of dataset was prohibited. The Task2 devoted to the Unsupervised Domain Adaptation for conversational Speech Enhancement (UDASE) is out of scope of this paper.

The speaker diarization task has a long history. The well established approach consists of a Voice Activity Detection (VAD) followed by the extraction of speaker embeddings from a well-trained Speaker Recognition models on short speech segments, which are then clustered in order to assign cluster label to each segment. While working excellent for non-overlapping speech, this approach degrades significantly when two or more people talk simultaneously. Several approaches were proposed to tackle this problem. One direction uses End-to-End Neural Diarization (EEND) [6, 7, 8] models which predict activity for all speakers talking in a given segment. While the same speaker's labels may differ in different segments, this permutation ambiguity can be overcome with the processing of overlapping segments and subsequent "stitching" of speaker labels. The another branch of approaches is based on Target Speaker VAD (TS-VAD) [9, 10] models which predict speakers' activity based on their embeddings. The initial embeddings can be extracted from clustering-based diarization segments and then gradually improved using iterative refinement. There are also approaches based on preliminary speech sources separation or target speaker's speech extraction (TSE) [11, 12].

ASR also degrades severely on overlapping speech segments. Diarization itself does not help to improve ASR quality but can be utilized in order to extract speech of each separate speaker especially in multi-microphone scenarios. Some of the most viable approaches for this are the already mentioned TSE and a beamforming. Many beamforming algorithms (like MVDR [13], GEV [14], etc.) needs to know spatial covariance matrices in each frequency bin whose estimation relies on speaker masks. Guided Source Separation (GSS) [15] performs a speaker masks estimation based on the information about utterance boundaries, so it can benefit much from an accurate speaker diarization. TSE in turn needs good speaker embeddings which can be estimated from diarization information or obtained as a by-product of TS-VAD refinement procedure.

ASR systems have shown a great progress last years. Rapid development of end-to-end approaches as well as new advanced architectures like Conformers [18] provided significant improvements on many known benchmarks. Another promising direction is Self-Supervised Learning [19, 20, 21] that provides models which are trained on a huge amount of unsupervised data and may be fine-tuned to different downstream tasks including ASR. Even very limited amount of supervised data is usually enough to obtain reasonable ASR performance which is of great importance for low-resource scenarios. Some approaches [16, 17] try to solve multiple tasks simultaneously to benefit from E2E approach on ASR, dereverberation and beamforming with self-supervised learning.

This paper describes the STCON system for the CHiME-7 DASR task which is similar to the STC system for the CHiME-6 challenge and consists of the following modules. Speech pre-processing includes block-wise WPE [22] dereverberation. The diarization module includes basic clustering-based diarization followed by TS-VAD refinement and some kind of post-processing. The speech enhancement module consists of GSS, MVDR beamforming and audio normalization (to avoid clipping). ASR module includes several models, both end-to-end and hybrid, whose decoding results are re-scored with several external LMs and fused together.

The main contributions of this paper include: 1) basic diarization pipeline with estimation of actual number of speakers and DOVER-Lap based fusion of channel-wise diarization results, 2) adjusting TS-VAD model training for heterogeneous data, 3) careful tuning of diarization post-processing and speech separation pipelines, and 4) fine-tuning of SSL model for the challenge data and using its embeddings to train complementary models.

The rest of the paper is organized as follows. In Section 2, we describe the speech pre-processing and separation pipeline as well as the ASR system trained for the AR sub-track. Then in Section 3 the diarization pipeline and TS-VAD training is described in details. Section 4 presents the overall results and conclusions.

2. DASR: Acoustic Robustness subtask

The baseline recipe uses a Transformer-based model with pre-trained and freezed WavLM [21] feature extractor (frontend), trained using ESPNet [23] with CTC+attention loss on the highly augmented dataset. It includes full CHiME-6 and Mixer6 training sets (Multiple Distant Microphones, MDM) plus synthetic (reverberated and noisy) data from close-talking microphones as well as GSS-enhanced data.

2.1. Data pre-processing

Baseline system includes components for speech dereverberation using Weighted Prediction Error (WPE) [22] and speech separation using Guided Source Separation (GSS) [15, 24] performed on selected 80% of all channels with maximum values of MicRank criterion based on Envelope Variance [25]. We tuned both components to improve separated speech quality. For dereverberation we used all-channel block WPE applied with 2-minute long blocks. The blockwise variant is much faster than one applied to the entire signal and provides comparable quality.

In GPU-GSS toolkit (unlike the original implementation) the speech of target speaker found in utterance contexts is considered as target speech. This provides better results when using wide contexts. We found that expanding both contexts up to 45 seconds improves WER notably (-2.1% on CHiME-6 dev).

The baseline version of GSS utilizes information about speakers' activity for both masked updates of complex Angular-Central GMM (cACGMM) during EM iterations, and hard initialization of its weights. Following [9] we used the soft initialization of cACGMM weigths from TS-VAD predictions (Subsection 3.3). The possibility to initialize cACGMM differently than from activity mask can be also beneficial in case when oracle speaker boundaries are provided. Indeed, after applying baseline GSS and recognizing its outputs one can use ASR alignment to initialize cACGMM. This two-pass approach brings substantial WER improvements (-0.9% on CHiME-6 dev) due to skipping intra-word pauses during initialization (unlike the hard initialization which assigns equal non-zero values to all frames of speaker's utterance). We used this approach in all AR subtrack experiments unless otherwise stated. Besides we found that with such initialization a number of GSS training iterations can be reduced to 5.

In order to account for the speakers movements around the room during the long utterance, beamforming was performed on chunks of 4.8 seconds. Finally, we applied the signal level normalization to avoid high clipping rate in GSS results. The progress from each change is presented in Table 1.

Table 1: *WERs(%) for DEV sets from GSS tuning.*

mod	chime	dipco	mixer6
baseline	34.1	34.0	23.3
+ context 45s	32.0	32.0	23.3
+ Alignment initialization	31.1	31.5	23.3
+ 4.8s beamforming chunk	31.0	31.5	23.3
+ 5 iter of cACGMM training	29.9	31.5	23.3
+ Audio normalization	**29.8**	**31.1**	**23.0**

2.2. Train dataset selection

Multiple experiments on large baseline training dataset are both very time-consuming and computational resources demanding. After first training attempts we decided to reduce training dataset and found that limiting it with WPE+GSS data only does not harm recognition accuracy on CHiME devset. So the most of subsequent trainings used this reduced dataset with 3-fold SP which speeded up experiments significantly.

2.3. Acoustic model improvement

Prior to experimenting with different architectures we tuned some training hyper-parameters, namely way of batching and learning rate schedule, that provided significant WER reduction on the baseline Transformer architecture (this corresponds to the line "Transformer (ours)" in Table 3.

In addition to the Transformer architecture used in the baseline recipe, we also trained ESPNet end-to-end models based on UConv-Conformer [26], E-Branchformer [27].

Although freezed WavLM embeddings themselves provide impressive results, one can benefit from unfreezing its weights and fine-tune them on domain specific data. We did this and observed notable improvements in WER on all devsets.

We also tried to replace WavLM with several other pre-trained open-source models permitted by the challenge rules, namely Wav2Vec2.0 [19] and HuBERT [20]. The comparison of different SSL frontends in terms of devset WER is shown in Table 2. Since WavLM provided the best results in terms of WER, it was chosen as a main embedding extractor to use. We also found that switching off SpecAugment [28] improves recognition accuracy.

Table 2: *WERs(%) for DEV sets on different SSL frontends. All models were fine-tuned as a whole, without parameters freezing*

SSL	chime	dipco	mixer6
WavLM	**27.4**	**30.1**	**20.3**
HuBERT	34.4	37.9	24.3
W2v2 XLSR53	36.0	40.9	24.6
W2v2 large LV60K	31.5	34.7	20.5

After fine-tuning WavLM parameters on training dataset within UConv-Conformer framework, this extractor was used to obtain highly informative domain-specific embeddings, which were used as features to train other types of acoustic models. First, we trained several phoneme-based Kaldi [29] multi-stream (MS) TDNN-F hybrids [30] and found that these model are comparable in accuracy to ESPNet end-to-end models. We also trained several version of Pruned Stateless Transducer (with additional CTC head) based on ZipFormer encoder using k2-IceFall implementation [31]. k2 models are also comparable to ESPNet ones with slight improvements on Mixer6 devset. WERs on several backend architectures are shown in Table 3.

Table 3: *Devsets WER(%) for different backends with and without WavLM Fine-tuning (FT)*

Model	FT	chime	dipco	mixer6
Transformer (baseline)	✗	32.6	33.5	20.3
Transformer (ours)	✗	26.0	28.0	16.9
	✓	22.4	26.3	14.9
UConv Conformer	✓	22.2	26.1	14.5
E-branchformer	✓	22.2	26.7	14.3
Hybrid MS TDNN-F	✓	22.2	24.8	13.6
k2 Zipformer	✓	21.8	25.6	13.3

In addition to ESPNet joint CTC-attention decoding for E2E models, we used Kaldi decoding with TLG graph [32]. 3-gram language model was built with SRILM [33] and VariKN [34] toolkits using CHiME, Mixer6, LibriSpeech and WSJ training texts. Transformer-LM [35] and AWD-LSTM-LM [36] trained using the same texts were utilized for rescoring ASR results.

In order to adapt ASR models to the domain peculiarities we added devsets data (several copies after processing with different GSS versions) to the ASR training. We tried both recognized and reference transcriptions of devsets for training. Recognized transcriptions worked significantly worse, so we opt to use reference ones. GSS processed devsets of CHiME-6 and DiPCo with three different GSS setting were used in training ASR models. We still used transcripts recognized with the best ASR model to add 106 hours of untranscribed Mixer6 data and used them for training one of our final models. In this experiment we randomly split the entire set of segments into 10 equal subsets and took subsets data from 10 different Mixer6 channels. The more detailed information about training subsets duration can be found in Table 4.

Table 4: *Sizes of data sets (in hours) used for training ASR models. Subsets used for not all models are marked with asterisk*

data	chime	dipco	mixer6	overall
train sup GSS + SP	85	-	165	250
train unsup MDM*	-	-	106	106
dev GSS x3*	18	10	-	28
total	**103**	**10**	**271**	**384**

We used the well-known Kaldi implementation of lattice-fusion as the combination method for various acoustic models results. The combination selection procedure is performed iteratively on dev sets. It starts from the combination of all available recognition results, and with each iteration discards recognition result from combination if this improves WER.

3. DASR: Main track

3.1. Selecting the diarization pipeline

Prior to the publication of the official main track baseline recipe we had compared several open-source frameworks and pre-trained models for the speaker diarization task (e.g. NeMo TitaNet [37] combined with Pyannote [38]) but results on CHiME devset were not satisfactory, so we resorted to use TS-VAD [9] approach which was quite successful in the previous CHiME challenge. Official recipe publication revealed superiority of the baseline diarization over the TS-VAD approach on DiPCo and Mixer6 but we decided to continue elaborating pre-selected TS-VAD direction.

3.2. Basic diarization

Basic diarization provides an initial segmentation for TS-VAD i-vectors extraction. We used the clustering-based approach for this. Clustering was performed on speaker embeddings extracted from single-channel recordings from each session dereverberated using all-channels WPE. After the diarization of each channel recordings the results were joined via DOVER-Lap [39]. In the joining procedure all channels were used except those that were not selected by MicRank on any of session segments. We compared several speaker embedding extractors, namely ResNet34 [40] and wav2vec2.0-based ones, developed in STC and trained on VoxCeleb1,2 data, as well as several open-source models, such as NeMo TitaNet/SpeakerNet and SpeechBrain [41] Ecapa TDNN [42]. The diarization results on CHiME devset for different extractions and clustering approaches are presented in Table 5. The best results on CHiME devset were obrained using the SpeechBrain model and ResNet34-based STC model. So, their fusion (via embeddings concatenation) followed by Spectral Clustering (SC) [43] was used in our pipeline.

Table 5: *CHiME Devset DER/JER for different speaker embeddings extractors and clustering algorithms. The number of clusters is set to 4 in all experiments*

Extractor	Clustering	DER	JER
TDNNF (xvectors)	SC	59.01	63.35
STC Resnet34	SC	**47.11**	**49.18**
	KMeans	54.21	54.88
	GMM	54.34	55.32
STC Wav2Vec-based	SC	53.78	57.28
	KMeans	57.57	60.36
SpeechBrain Ecapa TDNN	SC	**46.32**	**48.28**
	KMeans	49.54	51.44
Nemo TitaNet	SC	50.16	55.00
Nemo Ecapa TDNN	SC	54.76	59.50

In the original TS-VAD the number of speakers were fixed to be four. Since then several modification were proposed which relax this constraint to allow any number of speaker not much then used in training [44] or even arbitrary [10]. Since we could rely on information that no session contains more then 4 speakers, we chose the approach from [44]. But inferring an actual number of speakers is still a problem. There are some approaches to estimate it in course of clustering itself like NME-based approach from Spectral Clustering (SC) [43], dendrogram-based approach from Agglomereative Hierarchical Clustering (AHC) [45] or Silhouette index [46] but in our experiments all them made too many errors. As a result we used the following 2-stage procedure. First we extracted short-term embeddings on 1.5 seconds intervals with 50% overlap and put them into 4 clusters using Spectral Clustering. Then for each cluster we joined all segments related to it and extracted long-term embeddings on 10 seconds interval without overlap. Then we estimated number of speakers by analysing cosine distances between long-term embedding cluster centroids. After inferring the number of speakers we re-clustered short-term embeddings to obtain the final results of basic diarization.

3.3. Training TS-VAD models

We mainly followed training approach and architecture described in [9]. The main difficulty of training TS-VAD mod-

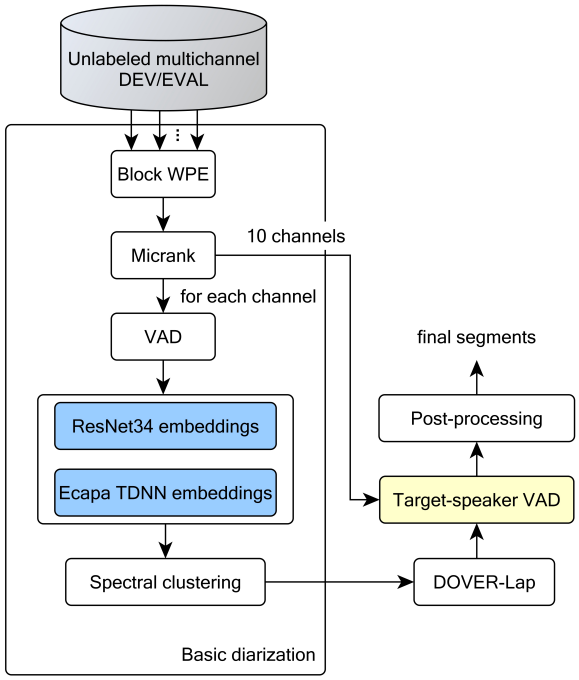

Figure 1: *The proposed diarization pipeline*

els for the CHiME-7 was in proper train dataset selection. Unlike the CHiME data which are well-labeled, the Mixer6 data are poorly and only partially labelled so cannot be used for training directly. Also there is no any training data for DiPCo. Thus no in-domain training data is available for two challenge datasets of three. Training only on CHiME data causes severe over-fitting and poor TS-VAD performance on both DiPCo and Mixer6 devsets. Adding VoxCeleb data acts as a regularization but doesn't help much. We tried to estimate DiPCo RIRs based on devset data and generate DiPCo-like recordings using LibriSpeech audios and estimated RIRs but observed no improvements on DiPCo dev/eval sets. That's why we decided to use DiPCo devset in training directly: we trained 5 different TS-VAD models each of which uses 4 of 5 DiPCo devset sessions in training and combined their predictions[1]. This led to substantial improvement of DER/JER on DiPCo evalset. In order to incorporate Mixer6 data in training too, we performed basic diarization (Subsection 3.2) of Mixer6 trainset into exactly 2 speakers. This gave us a better speaker labelling of Mixer6 data and facilitated including them in TS-VAD training.

TS-VAD inference was performed for each channel separately and then predictions were fused via averaging over selected channels. Channel selection was performed based on MicRank and exactly 10 channels were selected for all datasets.

3.4. Post-processing

We found that a simple post-processing of the diarization results can improve WER significantly. We used several post-processing strategies. Firstly, we expanded each segments to the left and right (the best value was 0.5s for TS-VAD and 1s for basic diarization). Second, we combined two speech segments separated by a pause shorter than 0.2s. We also selected the threshold for TS-VAD (the best value was 0.3). Table 6 shows the post-processing results for basic diarization and TS-VAD.

We also observed WER improvements from using TS-VAD weights for GSS initialization for segments from basic diarization (see Table 6, result with *) and even for oracle segments in AR subtrack.

Table 6: *Tuning post-processing for CHiME devset. "l" and "r" stand for expansion sizes in seconds, "sep" stands for maximum pause length to combine neigbouring speech segments, "thr" is a TS-VAD threshold. The result obtained when TS-VAD predictions were used to initialize GSS weights is marked with asterisk*

Diariz.	l/r	sep	thr	WER	DER	JER
Basic				41.3	40.2	41.4
	0.5/0.5			39.5	39.1	38.3
	0.5/0.5	2		37.6	35.6	36.6
	0.25/0.75	0.2		36.6	35.0	35.9
	0.25/0.75	2		36.3	34.0	34.8
	1/1	0.2		35.2	36.6	35.7
	1/1	0.2		**34.0***	36.6	35.7
TS-VAD	1/1	0.2	0.3	32.5	38.8	34.8
	1/1	0.2	0.4	33.8	34.4	34.2
	0.5/0.5	0.2	0.3	**32.2**	29.3	30.2
	0.5/0.5	0.2	0.4	33.6	27.9	31.0

3.5. Entire system pipeline

The entire system pipeline starts from basic diarization followed by TS-VAD, segments from which are post-processed and passed into several version on GSS+beamforming. Obtained speaker segments are recognized by several acoustic models and decoding results are rescored and fused together. The scheme of the diarization pipeline is shown on Figure 1.

4. Results and conclusions

Diarization and ASR results for the main track are shown in tables 7, 8 and 9 respectively.

Table 7: *DER/JER results for main track. BL/BD stand for baseline and basic diarization respectively*

		BL	BD	TS-VAD
chime	dev	39.9/51.1	40.6/41.4	33.5/35.7
	eval	56.3/62.5	39.6/44.5	36.6/41.3
dipco	dev	29.8/41.4	25.9/28.4	26.3/27.7
	eval	27.9/40.9	26.6/36.8	22.7/33.2
mixer6	dev	16.5/22.8	16.7/23.3	15.6/23.0
	eval	9.3/11.0	17.5/16.2	16.3/13.9

The obtained results demonstrate that the careful tuning of the well-known pipeline and using advanced architectures for ASR models can provide significant improvements compared to rather strong baseline. Nevertheless, there is a large room for improvements in both diarization and ASR. There are many new and promising ideas which can bring much improvement and this is a great field for the future work.

[1] Values of DER/JER reported on DiPCo devset are valid since they were obtained for each session with the model trained without this session

Table 8: *WER results on devsets for the best single system (BSS) and fusion for AR subtrack and main track*

	chime	dipco	mixer	macro-avg
AR subtrack				
Baseline	32.5	33.5	20.3	28.8
BSS	21.7	23.4	13.0	19.4
Fusion	**19.5**	**23.1**	**12.6**	**18.4**
Main track				
Baseline	62.4	56.6	22.5	47.1
BSS	32.1	36.3	17.3	28.6
Fusion	**31.4**	**33.7**	**15.8**	**26.9**

Table 9: *WER results on evalsets for BSS and fusion for AR subtrack and main track*

	chime	dipco	mixer	macro-avg
AR subtrack				
Baseline	35.5	36.3	30.9	34.2
BSS	27.6	22.6	17.9	22.7
Fusion	**23.0**	**19.3**	**12.8**	**18.3**
Main track				
Baseline	77.4	54.7	33.7	55.3
BSS	39.5	31.9	29.2	33.5
Fusion	**34.4**	**28.3**	**25.4**	**29.4**

5. Acknowledgements

We are grateful to STC Voice Biometrics Team for the awesome speaker embeddings extractor and valuable discussions.

6. References

[1] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third 'chime' speech separation and recognition challenge: Dataset, task and baselines," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2015, pp. 504–511.

[2] 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*, 2018, pp. 1561–1565.

[3] 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, T. Yoshioka, and N. Ryant, "Chime-6 challenge:tackling multispeaker speech recognition for unsegmented recordings," 2020.

[4] https://catalog.ldc.upenn.edu/LDC2013S03.

[5] M. V. Segbroeck, A. Zaid, K. Kutsenko, C. Huerta, T. Nguyen, X. Luo, B. Hoffmeister, J. Trmal, M. Omologo, and R. Maas, "Dipco – dinner party corpus," 2019.

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

[7] Y. Fujita, N. Kanda, S. Horiguchi, Y. Xue, K. Nagamatsu, and S. Watanabe, "End-to-end neural speaker diarization with self-attention," 2019.

[8] S. Horiguchi, Y. Fujita, S. Watanabe, Y. Xue, and P. Garcia, "Encoder-decoder based attractors for end-to-end neural diarization," *IEEE/ACM Transactions on Audio Speech and Language Processing*, vol. 30, pp. 1493–1507, 2022.

[9] I. Medennikov, M. Korenevsky, T. Prisyach, Y. Khokhlov, M. Korenevskaya, I. Sorokin, T. Timofeeva, A. Mitrofanov, A. Andrusenko, I. Podluzhny, A. Laptev, and A. Romanenko, "Target-speaker voice activity detection: a novel approach for multi-speaker diarization in a dinner party scenario," in *INTERSPEECH*, 2020, pp. 274–278.

[10] D. Wang, X. Xiao, N. Kanda, T. Yoshioka, and J. Wu, "Target speaker voice activity detection with transformers and its integration with end-to-end neural diarization," 2022.

[11] K. Žmolíková, M. Delcroix, K. Kinoshita, T. Ochiai, T. Nakatani, L. Burget, and J. Cernocky, "Speakerbeam: Speaker aware neural network for target speaker extraction in speech mixtures," *IEEE Journal of Selected Topics in Signal Processing*, vol. PP, pp. 1–1, 06 2019.

[12] M. Elminshawi, W. Mack, S. R. Chetupalli, S. Chakrabarty, and E. A. P. Habets, "New insights on target speaker extraction," 2022.

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

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

[15] C. Boeddeker, J. Heitkaemper, J. Schmalenstroeer *et al.*, "Front-end processing for the CHiME-5 dinner party scenario," in *CHiME Workshop*, 2018, pp. 35–40.

[16] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented transformer for speech recognition," *arXiv:2005.08100*, 2020.

[17] A. Baevski, H. Zhou, A. Mohamed, and M. Auli, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *arXiv:2006.11477*, 2020.

[18] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *arXiv:2106.07447*, 2021.

[19] S. Chen, C. Wang, Z. Chen, Y. Wu, S. Liu, Z. Chen, J. Li, N. Kanda, T. Yoshioka, X. Xiao, J. Wu, L. Zhou, S. Ren, Y. Qian, Y. Qian, J. Wu, M. Zeng, X. Yu, and F. Wei, "Wavlm: Large-scale self-supervised pre-training for full stack speech processing," *arXiv:2110.13900*, 2021.

[20] X. Chang, T. Maekaku, Y. Fujita, and S. Watanabe, "End-to-end integration of speech recognition, speech enhancement, and self-supervised learning representation," 2022.

[21] Y. Masuyama, X. Chang, S. Cornell, S. Watanabe, and N. Ono, "End-to-end integration of speech recognition, dereverberation, beamforming, and self-supervised learning representation," 2022.

[22] 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*, 2018, pp. 1–5.

[23] S. Watanabe, T. Hori, S. Karita, T. Hayashi, J. Nishitoba, Y. Unno, N. Enrique, Y. Soplin, J. Heymann, M. Wiesner, N. Chen, A. Renduchintala, and T. Ochiai, "Espnet: End-to-end speech processing toolkit," *arXiv:1804.00015*, 2018.

[24] D. Raj, D. Povey, and S. Khudanpur, "Gpu-accelerated guided source separation for meeting transcription," *arXiv:2212.05271*, 2022.

[25] M. Wolf and C. Nadeu, "Channel selection measures for multi-microphone speech recognition," *Speech Communication*, vol. 57, p. 170–180, 02 2014.

[26] A. Andrusenko, R. Nasretdinov, and A. Romanenko, "Uconv-conformer: High reduction of input sequence length for end-to-end speech recognition," *arXiv:2208.07657*, 2022.

[27] K. Kim, Fel, a. Wu, Y. Peng, J. Pan, P. Sridhar, K. J. Han, and S. Watanabe, "E-branchformer: Branchformer with enhanced merging for speech recognition," *arXiv:2210.00077*, 2022.

[28] 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:1904.08779*, 2019.

[29] D. Povey, A. Ghoshal, G. Boulianne *et al.*, "The Kaldi speech recognition toolkit," in *IEEE ASRU Workshop*, 2011.

[30] K. Han, R. Prieto, and T. Ma, "State-of-the-art speech recognition using multi-stream self-attention with dilated 1d convolutions," in *IEEE ASRU Workshop*, 2019, pp. 54–61.

[31] https://github.com/k2-fsa/icefall.

[32] K. An, H. Xiang, and Z. Ou, "Cat: A ctc-crf based asr toolkit bridging the hybrid and the end-to-end approaches towards data efficiency and low latency," 2020.

[33] A. Stolcke, "Srilm — an extensible language modeling toolkit," in *Proceedings of the 7th International Conference on Spoken Language Processing (ICSLP 2002)*, vol. 2, 07 2004.

[34] https://vsiivola.github.io/variKN.

[35] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[36] S. Merity, N. S. Keskar, and R. Socher, "Regularizing and optimizing LSTM language models," *CoRR*, vol. abs/1708.02182, 2017. [Online]. Available: http://arxiv.org/abs/1708.02182

[37] N. R. Koluguri, T. Park, and B. Ginsburg, "Titanet: Neural model for speaker representation with 1d depth-wise separable convolutions and global context," *arXiv:2110.04410*, 2021.

[38] H. Bredin, R. Yin, J. M. Coria, G. Gelly, P. Korshunov, M. Lavechin, D. Fustes, H. Titeux, W. Bouaziz, and M.-P. Gill, "pyannote.audio: neural building blocks for speaker diarization," *arXiv:1911.01255*, 2019.

[39] D. Raj, L. P. Garcia-Perera, Z. Huang, S. Watanabe, D. Povey, A. Stolcke, and S. Khudanpur, "DOVER-Lap: A method for combining overlap-aware diarization outputs," *2021 IEEE Spoken Language Technology Workshop (SLT)*, 2021.

[40] A. Gusev, V. Volokhov, T. Andzhukaev, , S. Novoselov, G. Lavrentyeva, M. Volkova, A. Gazizullina, A. Shulipa, A. Gorlanov, A. Avdeeva, A. Ivanov, T. Pekhovsky, and Y. Matveev, "Deep speaker embeddings for far-field speaker recognition on short utterances," *arXiv:2002.06033*, 2020.

[41] M. Ravanelli, T. Parcollet, P. Plantinga, A. Rouhe, S. Cornell, L. Lugosch, C. Subakan, N. Dawalatabad, A. Heba, J. Zhong, J.-C. Chou, S.-L. Yeh, S.-W. Fu, C.-F. Liao, E. Rastorgueva, F. Grondin, W. Aris, H. Na, Y. Gao, R. De Mori, and Y. Bengio, "Speechbrain: A general-purpose speech toolkit," *arXiv:2106.04624*, 2021.

[42] N. Dawalatabad, M. Ravanelli, F. Grondin, J. Thienpondt, B. Desplanques, and H. Na, "Ecapa-tdnn embeddings for speaker diarization," in *Interspeech*, 2021.

[43] T. Park, K. Han, M. Kumar, and S. Narayanan, "Auto-tuning spectral clustering for speaker diarization using normalized maximum eigengap," *IEEE Signal Processing Letters*, vol. 27, pp. 381–385, 2019.

[44] M. He, D. Raj, Z. Huang, J. Du, Z. Chen, and S. Watanabe, "Target-speaker voice activity detection with improved i-vector estimation for unknown number of speaker," in *INTERSPEECH*, 2021, pp. 3555–3559.

[45] D. Defays, "An efficient algorithm for a complete link method," *The Computer Journal*, vol. 20, no. 4, pp. 364–366, 01 1977.

[46] P. Rousseeuw, "Silhouettes: A graphical aid to the interpretation and validation of cluster analysis," *Journal of Computational and Applied Mathematics*, vol. 20, pp. 53–65, 11 1987.