INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

Auxiliary loss function for target speech extraction and recognition with weak supervision based on speaker characteristics

Katerina Zmolikova[1], *Marc Delcroix*[2], *Desh Raj*[3], *Shinji Watanabe*[3,4], *Jan "Honza" Černocký*[1]

[1]Brno University of Technology, Faculty of Information Technology, Speech@FIT, Czechia
[2]NTT Corporation, Japan
[3]Center for Language and Speech Processing, The Johns Hopkins University, Baltimore, USA
[4]Language Technologies Institute, Carnegie Mellon University, Pittsburgh, USA

izmolikova@fit.vutbr.cz

Abstract

Automatic speech recognition systems deteriorate in presence of overlapped speech. A popular approach to alleviate this is target speech extraction. The extraction system is usually trained with a loss function measuring the discrepancy between the estimated and the reference target speech. This often leads to distortions to the target signal which is detrimental to the recognition accuracy. Additionally, it is necessary to have the strong supervision provided by parallel data consisting of speech mixtures and single-speaker signals. We propose an auxiliary loss function for retraining the target speech extraction. It is composed of two parts: first, a speaker identity loss, forcing the estimated speech to have correct speaker characteristics, and second, a mixture consistency loss, making the extracted sources sum back to the original mixture. The only supervision required for the proposed loss is speaker characteristics obtained from several segments spoken by the target speaker. Such weak supervision makes the loss suitable for adapting the system directly on real recordings. We show that the proposed loss yields signals more suitable for speech recognition and further, we can gain additional improvements by adaptation to target data. Overall, we can reduce the word error rate on LibriCSS dataset from 27.4% to 24.0%.

Index Terms: Target speech extraction, SpeakerBeam, Weakly supervised loss, Long recordings

1. Introduction

The performance of many speech technologies, such as automatic speech recognition (ASR), gets deteriorated in presence of speaker overlap. While on clean speech recordings, ASR systems reach reasonable performance today, multi-speaker ASR is still a challenge, that attracts research interest [1]. A common approach to alleviate the problem consists of pre-processing the mixture of multiple speakers to remove the interfering speech and then applying a conventional ASR system to the resulting single-speaker speech [2].

Two important directions in this research are *speech separation* and *target speech extraction*. In speech separation [3, 4, 5], the goal is to blindly obtain the signals of all the sources present in the mixture. In target speech extraction [6, 7, 8], an enrollment utterance of the target speaker is used to extract the speech signal of this speaker from the mixture. Both directions have their advantages in different application scenarios. In this work, we focused on target speech extraction, however, the proposed method can be applied analogously to speech separation.

Most recent target speech extraction and separation methods employ neural networks [3, 4, 5, 6, 7, 8]. In these methods, the neural networks are trained with loss functions that compare the reference single-speaker signal to the estimated output (e.g. mean square error or signal-to-noise ratio). Such loss functions however do not perfectly reflect which output signals lead to good ASR performance. Namely, the systems trained with the conventional loss functions often lead to too aggressive removal of the interference causing distortion of the target speech. Previous works proposed joint training with ASR system to alleviate this problem [9, 10]. However, this requires transcriptions of the target speech which might not be always available.

In this work, we propose an auxiliary loss function for neural network-based speech separation/extraction that is based on speaker characteristics. The loss function is composed of two parts — a *speaker identity loss* and a *mixture consistency loss*. The speaker identity loss forces the output to have the characteristics of the desired speaker. To evaluate how well the speaker characteristics match, we use x-vectors [11] and Probabilistic Linear Discriminant Analysis (PLDA) [12], a popular model for speaker verification. The mixture consistency loss forces all signals extracted from the mixture to sum back to the mixture. Such a constraint naturally arises from the assumed mixing model and further restricts the network output.

The proposed loss function gives more freedom to the output of the network than the conventional strongly supervised loss, which enables less aggressive processing by keeping more noise, which induces less distortion to the extracted speech. Besides, the only supervision necessary for computing the loss is several segments of speech of the target speaker, which are used to estimate the desired speaker characteristics for the speaker identity loss. In the case of long recordings, such as meetings, these segments can be obtained by applying diarization. This weak supervision makes it possible to adapt the system directly on the target data, where strong supervision in form of parallel single-speaker signals is not available.

As an application scenario to test our method, we choose automatic speech recognition on long recordings such as meetings that have received increased interest recently [13, 14, 15]. We employed SpeakerBeam [6, 9] architecture as the method for target speech extraction. In our experiments, we show two benefits of the proposed loss: first, retraining with such loss improves ASR performance and second, it can be used to adapt to the target data without using parallel single-speaker references.

http://dx.doi.org/10.21437/Interspeech.2021-986

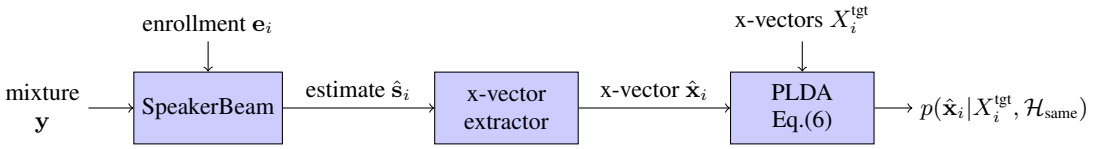

Figure 1: *Scheme of the speaker identity loss.*

2. Related works

Several works in the area of universal sound separation have explored weakly supervised objective functions [16, 17, 18]. Notably, in [16, 17] authors propose an objective function consisting of sound event classification and mixture consistency, similarly to our proposed objective function. Direct application of the loss from [16, 17] to our task would however require training a speaker classifier on the adaptation data, which we avoid by using the generative PLDA model. The PLDA model can be trained on a disjoint set of speakers and is considered state-of-the-art in speaker verification.

For speech separation, several works explored using different modalities, such as spatial [19, 20, 21] or visual [22, 23, 24] cues as weak supervision. Our proposed scheme does not require other modalities, but if they were available, the objective function could still be complemented by the speaker identity loss presented in this work.

3. Target speech extraction

3.1. Problem formulation

The problem of target speech extraction is to isolate the speech of a target speaker from an observed mixture of multiple overlapping speakers and noise. We assume a mixing model

$$\mathbf{y} = \sum_{i=0}^{I-1} \mathbf{s}_i + \mathbf{n}, \tag{1}$$

where I is the number of speakers in the mixture, $\mathbf{s}_i$ for $i = 0, \ldots, I-1$ is the speech time-domain signal of the ith speaker, $\mathbf{n}$ is the additional noise and $\mathbf{y}$ is the observed mixture. The speech of the target speaker is extracted using additional information in the form of enrollment utterance, which is spoken by the target speaker. We denote the enrollment utterance for speaker i as $\mathbf{e}_i$. Given the enrollment utterance $\mathbf{e}_i$, the goal is to obtain an estimate $\hat{\mathbf{s}}_i$ of the target speech $\mathbf{s}_i$.

3.2. SpeakerBeam

SpeakerBeam [6, 9] performs target speech extraction using a neural network. In this work, we use frequency-domain SpeakerBeam and denote the magnitude of short-time Fourier transform (STFT) of $\mathbf{y}, \mathbf{s}_i, \hat{\mathbf{s}}_i, \mathbf{e}_i$ as $|\mathbf{Y}|, |\mathbf{S}_i|, |\hat{\mathbf{S}}_i|, |\mathbf{E}_i|$, respectively. We use a frequency domain implementation to speed-up experiments, although a time-domain one could be used as well.

The SpeakerBeam network consists of two parts: the main network and the auxiliary network. The auxiliary network accepts the enrollment utterance $|\mathbf{E}_i|$ as the input and outputs a fixed-length embedding vector $\boldsymbol{\alpha}_i$ representing the speaker characteristics. The main network performs the extraction itself, i.e. it maps the mixed speech $|\mathbf{Y}|$ to a time-frequency mask $\mathbf{M}_i = f(|\mathbf{Y}|, \boldsymbol{\alpha}_i)$, where f is the function modeled by the neural network. The mask $\mathbf{M}_i$ can be used to get the estimate of the target speaker speech as $|\hat{\mathbf{S}}_i| = \mathbf{M}_i \odot |\mathbf{Y}|$, where $\odot$ denotes element-wise multiplication. The embedding vector $\boldsymbol{\alpha}_i$ is used to inform the main network about the target speaker. Several ways of informing the main network using the embedding have been explored in the past. Here, we use the approach based on scaling activations [6].

Conventionally, SpeakerBeam is trained with supervised loss $\mathcal{L}_{\text{sup}}$, which uses the target speech $\mathbf{s}_i$ as a reference. Here, we employ the phase-sensitive mean square error between the estimated and reference speech in STFT domain [25, 26]

$$\mathcal{L}_{\text{sup}}(\hat{\mathbf{s}}_i, \mathbf{s}_i) = || \, |\hat{\mathbf{S}}_i| - |\mathbf{S}_i| \max(0, \cos(\boldsymbol{\phi}_y - \boldsymbol{\phi}_{s_i})) \, ||^2, \tag{2}$$

where $\boldsymbol{\phi}_y$ and $\boldsymbol{\phi}_{s_i}$ are the phase of STFT of $\mathbf{y}$ and $\mathbf{s}_i$, respectively.

4. Proposed weakly supervised loss

We propose a loss function which uses weak supervision in the form of speaker characteristics. The speaker characteristics are obtained from a set of N_i segments of speech of the target speaker i, denoted as $S_i^{\text{tgt}} = \{\mathbf{s}_i^{(\text{tgt},1)}, \mathbf{s}_i^{(\text{tgt},2)}, \ldots, \mathbf{s}_i^{(\text{tgt},N_i)}\}$. When using a segmented speech corpus such as WSJ, the segments can be simply different utterances from the same speaker. In case of long recordings such as meetings, the segments can be obtained from other parts of the recording where the speaker is speaking, after applying diarization.

We devise a loss function $\mathcal{L}_{\text{spk}}$, which forces the output of SpeakerBeam for each speaker i to have the same speaker characteristics as S_i^{tgt} (Section 4.1). Forcing the correct speaker information however does not tie the output of SpeakerBeam to the input mixture in any way. For this reason, we add an additional loss $\mathcal{L}_{\text{mix}}$, which encourages mixture consistency, i.e. the extracted signals from the mixture sum back to the mixture (Section 4.2). The full weakly supervised loss function, $\mathcal{L}_{\text{wsup}}$, is then

$$\mathcal{L}_{\text{wsup}}(\hat{\mathbf{s}}_i, S_i^{\text{tgt}}) = \lambda_{\text{spk}} \mathcal{L}_{\text{spk}} + \lambda_{\text{mix}} \mathcal{L}_{\text{mix}} \tag{3}$$

where λ_{spk} and λ_{mix} are hyper-parameters weighting the parts of the loss.

4.1. Speaker identity loss

To encourage the SpeakerBeam estimate $\hat{\mathbf{s}}_i$ to have the same speaker characteristics as segments in S_i^{tgt}, we employ concepts from speaker identification. Namely, we use x-vectors [11] to represent the speaker characteristics and Probabilistic Linear Discriminant Analysis (PLDA) [12] to model the x-vectors. Let us denote the x-vector extracted from $\hat{\mathbf{s}}_i$ as $\hat{\mathbf{x}}_i$ and the set of x-vectors extracted from S_i^{tgt} as $X_i^{\text{tgt}} = \{\mathbf{x}_i^{(\text{tgt},1)}, \ldots, \mathbf{x}_i^{(\text{tgt},N_i)}\}$.

In PLDA, the distribution of x-vectors is modeled as

$$p(\mathbf{r}) = \mathcal{N}(\mathbf{r}; \mathbf{m}, \boldsymbol{\Sigma}_{\text{ac}}) \tag{4}$$

$$p(\mathbf{x}|\mathbf{r}) = \mathcal{N}(\mathbf{x}; \mathbf{r}, \boldsymbol{\Sigma}_{\text{wc}}), \tag{5}$$

where $\mathbf{x}$ is the x-vector, $\mathbf{r}$ is the speaker mean, $\mathbf{m}$ is the global mean and $\boldsymbol{\Sigma}_{\text{ac}}$, $\boldsymbol{\Sigma}_{\text{wc}}$ are the across-speaker and within-speaker

co-variance matrices. In the loss function, we aim to maximize the likelihood of the estimated x-vector $\hat{\mathbf{x}}_i$ given the x-vectors X_i^{tgt}, under the hypothesis $\mathcal{H}_{\text{same}}$ that both have been generated from the same speaker

$$p(\hat{\mathbf{x}}_i|X_i^{\text{tgt}}, \mathcal{H}_{\text{same}}) = \int p(\hat{\mathbf{x}}_i|\mathbf{r})p(\mathbf{r}|X_i^{\text{tgt}})\, d\mathbf{r} = \mathcal{N}(\hat{\mathbf{x}}_i; \boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i) \tag{6}$$

$$\boldsymbol{\Sigma}_i = (\boldsymbol{\Sigma}_{\text{ac}}^{-1} + N_i\boldsymbol{\Sigma}_{\text{wc}}^{-1})^{-1} \tag{7}$$

$$\boldsymbol{\mu}_i = \boldsymbol{\Sigma}_i\,(\boldsymbol{\Sigma}_{\text{ac}}^{-1}\mathbf{m} + N_i\boldsymbol{\Sigma}_{\text{wc}}^{-1}\bar{\mathbf{x}}_i), \tag{8}$$

where $\bar{\mathbf{x}}_i$ and N_i are the mean and the number of x-vectors in X_i^{tgt}. The equality follows Eq. (213) in [27]. The loss function is then the inverse log-likelihood summed over all speakers in the mixture

$$\mathcal{L}_{\text{spk}} = -\sum_{i=0}^{I-1} \log p(\hat{\mathbf{x}}_i|X_i^{\text{tgt}}, \mathcal{H}_{\text{same}}). \tag{9}$$

Note that $\boldsymbol{\mu}$ and $\boldsymbol{\Sigma}$ can be pre-computed for each speaker in advance. The evaluation of the loss function is then the evaluation of the Normal p.d.f. of the estimated x-vector $\hat{\mathbf{x}}_i$. Figure 1 shows the process for computing the speaker identification loss. The extraction of the x-vector including the feature extraction needs to be implemented in a differentiable way, which is possible using a toolkit such as PyTorch [28][1].

4.2. Mixture consistency loss

The mixture consistency loss reflects a property that should hold for the extracted sources, i.e. summing back to the original signal. This directly follows from the assumed mixing model in Eq. (1). To enforce this, we minimize the mean-square error between the sum of the extracted signals and the observed mixture in time-domain

$$\mathcal{L}_{\text{mix}} = ||\mathbf{y} - \sum_{i=0}^{I-1} \hat{\mathbf{s}}_i||^2. \tag{10}$$

Note that we neglect the noise factor $\mathbf{n}$ in the loss. This could possibly lead to the network learning to include the noise in the extracted sources. Extending the mixture consistency with a noise model is a possible future direction, which could be beneficial for very noisy conditions.

4.3. Overall steps

The proposed weakly supervised loss can be used for retraining a target speech extraction system on data, where for each mixture $\mathbf{y}$ and corresponding enrollment utterance $\mathbf{e}_i$, there is a set of segments S_i^{tgt} spoken by the target speaker. Here, we describe the steps we follow in our experiments with the proposed loss:

1. Train SpeakerBeam with supervised loss $\mathcal{L}_{\text{sup}}(\hat{\mathbf{s}}_i, \mathbf{s}_i)$. This step requires parallel single-speaker recording $\mathbf{s}_i$ for each mixture $\mathbf{y}$ and represents the baseline.
2. Re-train SpeakerBeam initialized in step 1 with weakly supervised loss $\mathcal{L}_{\text{wsup}}(\hat{\mathbf{s}}_i, S_i^{\text{tgt}})$ on the same training data as in step 1, but using only weak supervision in form of segments S_i^{tgt} spoken by the target speaker. The segments are taken from other utterances of the target speaker in the training data.
3. Re-train SpeakerBeam with weakly supervised loss $\mathcal{L}_{\text{wsup}}(\hat{\mathbf{s}}_i, S_i^{\text{tgt}})$ on the target data. As the target data consists of long recordings, we use segments of the target speaker defined by a diarization as S_i^{tgt}.

The goal of step 2 is to evaluate the effect of the loss itself compared to the supervised loss, while in step 3 we explore whether it is possible to use the loss to adapt to the target data.

5. Experiments

5.1. Dataset

We use two different sources of data, i.e. artificially mixed short utterances for training, and long meeting-like recordings for testing and adaptation. The artificially mixed data are based on LibriSpeech dataset [29]. We simulate mixtures of 2 speakers. We will denote the mixed data as LibriSpeech-mix. LibriSpeech dataset is also used to get enrollment utterances in all experiments. For testing and adaptation, we use the LibriCSS dataset [13], which contains multi-channel recordings simulating conversations. For our experiments, we use the first channel only. Each recording was created using multiple utterances from LibriSpeech [29] from multiple speakers. The utterances were played back from a loudspeaker in a room. The recordings can be grouped into six different overlap conditions from 0% to 40% of overlap. Alternatively, the recordings can be grouped into 10 different sessions, where each session contains different speakers. Each session then contains one recording for each overlap condition. We use *session0* as the development set and the remaining sessions as the evaluation set.

5.2. Configuration

SpeakerBeam The main network of SpeakerBeam consists of 3 BLSTM layers, each with 600 units and 2 fully connected layers with ReLU activation. We apply the scaling of the activations after the first BLSTM layer. The auxiliary network has 2 fully connected layers with 64 units, ReLU activation after the first layer. For supervised training, we used Adam optimizer with learning rate 1e-3 and gradient clipping 1. We trained the network for 450k iterations with batch size 36. We used STFT with window size and shift of 512 and 128 samples. When extracting the target speech, we apply SpeakerBeam by chunks of 10 seconds with 5 second shift.

x-vectors, PLDA We use x-vector extractor and PLDA model from VBx recipe[2]. Both are trained on VoxCeleb corpus [30]. The architecture and training are further described in [31].

Proposed loss For re-training the network with the proposed loss $\mathcal{L}_{\text{wsup}}$, we use Adam optimizer, with learning rate 1e-6 and gradient clipping 1. We perform 80k iterations with batch size 1. We weigh both parts of the loss equally, setting $\lambda_{\text{spk}} = \lambda_{\text{mix}} = 0.5$, unless stated otherwise.

ASR system We use the hybrid HMM-DNN model from [32]. The acoustic model is a 17-layer factored TDNN [33] trained using the lattice-free MMI objective [34]. The model was trained on the 960h Librispeech data with 3x speed perturbation, and additionally fine-tuned for 1 epoch on reverberated Librispeech data. We use the official 3-gram language model provided with Librispeech for decoding.

5.3. Results

We evaluate the target speech extraction performance with speech recognition on the LibriCSS dataset. We show re-

[1] We will provide the implementation at `https://github.com/BUTSpeechFIT/spkb_plda_loss`

[2] VBx recipe `https://github.com/BUTSpeechFIT/VBx`

Table 1: *Speech recognition performance in terms of Word error rate (WER) on single-channel LibriCSS data using oracle diarization.*

		$\mathcal{L}_{\text{wsup}}$	on target data	WER [%] all	WER [%] OV40
(1)	Mixtures	-	-	26.2	41.6
(2)	SpeakerBeam	✗	✗	24.1	33.2
(3)	SpeakerBeam	✓	✗	20.8	31.1
(4)	SpeakerBeam	✓	✓	**20.3**	**30.2**

Table 2: *Speech recognition performance in terms of Word error rate (WER) on single-channel LibriCSS data using RPN and TS-VAD diarization.*

		$\mathcal{L}_{\text{wsup}}$	on target data	RPN all	RPN OV40	TS-VAD all	TS-VAD OV40
	DER [%]	-	-	9.5	14.2	7.6	9.5
	WER [%]						
(1)	Mixtures	-	-	31.2	47.2	28.4	42.4
(2)	SpeakerBeam	✗	✗	30.1	42.3	27.4	36.3
(3)	SpeakerBeam	✓	✗	27.0	40.0	24.3	33.7
(4)	SpeakerBeam	✓	✓	**26.6**	**39.4**	**24.0**	**33.0**

sults on the evaluation set (*all*) and the condition with highest amount of overlap (*OV40*). We compare 4 different setups: (1) unprocessed data without SpeakerBeam applied, (2) baseline SpeakerBeam trained with the supervised loss $\mathcal{L}_{\text{sup}}$ on LibriSpeech-mix, (3) SpeakerBeam retrained with the weakly supervised loss $\mathcal{L}_{\text{wsup}}$ on LibriSpeech-mix, and (4) SpeakerBeam retrained on target LibriCSS data with $\mathcal{L}_{\text{wsup}}$. The setups (2)-(4) correspond to steps 1-3 as described in Sec. 4.3. Note that for (4) there are no parallel data available.

For the experiments, it is necessary to have diarization outputs — first, for ASR decoding, and second, to get speaker labels when adapting to the target data. To avoid the influence of the diarization errors, we first perform experiments using oracle diarization. The results are shown in Table 1. Comparing the results on unprocessed data (row (1)) with applying baseline SpeakerBeam trained with the supervised loss $\mathcal{L}_{\text{sup}}$ (row (2)), we can see that the target speech extraction improves the ASR performance significantly, especially when a higher amount of overlap is present. Retraining the SpeakerBeam system with the weakly supervised loss $\mathcal{L}_{\text{wsup}}$ on the original artificially mixed data (row (3)) improves the performance further. By exploring the outputs of original and retrained SpeakerBeam, we can see that after retraining with $\mathcal{L}_{\text{wsup}}$, the resulting speech contains slightly more noise, but less speech distortion. Such outputs may be more favorable for the ASR system. Note that the network in (2) is fully converged and training it longer with supervised loss does not yield better results. The improvements in (3) are thus not caused by simply longer training. Finally, retraining SpeakerBeam directly on the target evaluation set, leads to further improvement, showing that it is possible to adapt to the target conditions using the speaker labels only.

Although not directly comparable, a separation using a similar network architecture and hybrid ASR back-end achieved a WER of 35.5% on OV40 condition in [13]. The performance of the proposed system could also be improved using more sophisticated network architectures for both front-end and back-ends as in [13].

In the second set of experiments, we used outputs of the diarization system rather than the oracle ground-truth diarization, to see whether errors in the speaker labels have a detrimental effect on the adaptation. We used two different diarization systems, i.e. region proposal network (RPN) [35] and target-speaker voice activity detection (TS-VAD) [36][3]. Table 2 shows the results of the ASR, together with the diarization error rates (DER) of the diarization systems. We can see that in both cases, the trends are similar as with the oracle diarization. Including the proposed loss brings from 2.3% to 3.1% WER improvement, and adapting to the target data further improves the performance by 0.3-0.7% WER. The adaptation to the target data was thus not significantly affected by the diarization errors.

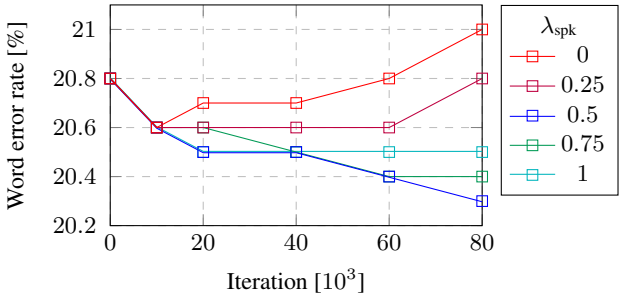

Figure 2: *Speech recognition performance as a function of number of iterations during adaptation and different values of weight λ_{spk}. The mixture consistency weight λ_{mix} is set to $1 - \lambda_{spk}$.*

To understand better how the two parts of the loss function, defined in Sec. 4, affect the training, we experimented with different weights $\lambda_{\text{spk}}, \lambda_{\text{mix}}$ during the adaptation stage. Figure 2 shows the speech recognition performance as a function of the number of iterations performed. We set the weights so that $\lambda_{\text{mix}} + \lambda_{\text{spk}} = 1$. The results show greater importance of the speaker identity loss $\mathcal{L}_{\text{spk}}$. When the mixture consistency is dominant in the loss, the performance starts to worsen after 10k iterations. The speaker consistency loss leads to improvements even by itself, however, the best results are still obtained when both parts of the loss are balanced.

6. Conclusion

In this paper, we have investigated a novel loss for target speech extraction, which makes use of speaker characteristics. We have shown that retraining with the proposed loss improves automatic speech recognition performance. Further, due to its weakly supervised nature, it is possible to adapt to target data using diarization labels, and bring further improvements. Although we have focused on target speech extraction in this study, the loss is applicable also to speech separation.

7. Acknowledgement

The work reported here was started at JSALT 2020 at JHU, supported by Microsoft, Amazon and Google. We'd like to thank Pavel Denisov, Christoph Boeddeker, Thilo von Neumann, Tobias Cord-Landwehr, Zili Huang and Maokui He for their help during the workshop. K. Zmolikova was partly supported by Czech Ministry of Education, Youth and Sports from project no. LTAIN19087 "Multi-linguality in speech technologies". Part of high-performance computation run on IT4I supercomputer and was supported by the Ministry of Education, Youth and Sports of the Czech Republic through e-INFRA CZ (ID:90140).

[3]We did not use VBx diarization as it does not deal with speaker overlaps.

8. References

[1] Y.-m. Qian, C. Weng, X.-k. Chang, S. Wang, and D. Yu, "Past review, current progress, and challenges ahead on the cocktail party problem," *Frontiers of Information Technology & Electronic Engineering*, vol. 19, no. 1, pp. 40–63, 2018.

[2] D. Wang and J. Chen, "Supervised speech separation based on deep learning: An overview," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 26, no. 10, pp. 1702–1726, 2018.

[3] Y. Luo and N. Mesgarani, "Conv-Tasnet: Surpassing ideal time–frequency magnitude masking for speech separation," *IEEE/ACM transactions on audio, speech, and language processing*, vol. 27, no. 8, pp. 1256–1266, 2019.

[4] J. R. Hershey, Z. Chen, J. Le Roux, and S. Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *ICASSP*. IEEE, 2016, pp. 31–35.

[5] D. Yu, M. Kolbæk, Z.-H. Tan, and J. Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *ICASSP*. IEEE, 2017, pp. 241–245.

[6] K. Žmolíková, M. Delcroix, K. Kinoshita, T. Ochiai, T. Nakatani, L. Burget, and J. Černocký, "SpeakerBeam: Speaker aware neural network for target speaker extraction in speech mixtures," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 4, pp. 800–814, 2019.

[7] Q. Wang, H. Muckenhirn, K. Wilson, P. Sridhar, Z. Wu, J. R. Hershey, R. A. Saurous, R. J. Weiss, Y. Jia, and I. L. Moreno, "VoiceFilter: Targeted Voice Separation by Speaker-Conditioned Spectrogram Masking," in *Interspeech*, 2019, pp. 2728–2732. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2019-1101

[8] C. Xu, W. Rao, E. S. Chng, and H. Li, "Time-domain speaker extraction network," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 327–334.

[9] M. Delcroix, K. Zmolikova, K. Kinoshita, A. Ogawa, and T. Nakatani, "Single channel target speaker extraction and recognition with SpeakerBeam," in *ICASSP*. IEEE, 2018, pp. 5554–5558.

[10] S. Settle, J. Le Roux, T. Hori, S. Watanabe, and J. R. Hershey, "End-to-end multi-speaker speech recognition," in *ICASSP*. IEEE, 2018, pp. 4819–4823.

[11] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *ICASSP*. IEEE, 2018, pp. 5329–5333.

[12] P. Kenny, "Bayesian speaker verification with heavy-tailed priors." in *Odyssey*, vol. 14, 2010.

[13] Z. Chen, T. Yoshioka, L. Lu, T. Zhou, Z. Meng, Y. Luo, J. Wu, X. Xiao, and J. Li, "Continuous speech separation: Dataset and analysis," in *ICASSP*. IEEE, 2020, pp. 7284–7288.

[14] S. Watanabe, M. Mandel, J. Barker, E. Vincent *et al.*, "CHiME-6 Challenge: Tackling multispeaker speech recognition for unsegmented recordings," in *6th International Workshop on Speech Processing in Everyday Environments (CHiME 2020)*, 2020.

[15] T. Yoshioka, I. Abramovski, C. Aksoylar, Z. Chen, M. David, D. Dimitriadis, Y. Gong, I. Gurvich, X. Huang, Y. Huang *et al.*, "Advances in online audio-visual meeting transcription," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 276–283.

[16] F. Pishdadian, G. Wichern, and J. Le Roux, "Learning to separate sounds from weakly labeled scenes," in *ICASSP*. IEEE, 2020, pp. 91–95.

[17] ——, "Finding strength in weakness: Learning to separate sounds with weak supervision," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 28, pp. 2386–2399, 2020.

[18] Q. Kong, Y. Wang, X. Song, Y. Cao, W. Wang, and M. D. Plumbley, "Source separation with weakly labelled data: An approach to computational auditory scene analysis," in *ICASSP)*. IEEE, 2020, pp. 101–105.

[19] P. Seetharaman, G. Wichern, J. Le Roux, and B. Pardo, "Bootstrapping single-channel source separation via unsupervised spatial clustering on stereo mixtures," in *ICASSP*. IEEE, 2019, pp. 356–360.

[20] L. Drude, D. Hasenklever, and R. Haeb-Umbach, "Unsupervised training of a deep clustering model for multichannel blind source separation," in *ICASSP*. IEEE, 2019, pp. 695–699.

[21] E. Tzinis, S. Venkataramani, and P. Smaragdis, "Unsupervised deep clustering for source separation: Direct learning from mixtures using spatial information," in *ICASSP*. IEEE, 2019, pp. 81–85.

[22] R. Gao and K. Grauman, "Co-separating sounds of visual objects," in *Proceedings of the IEEE International Conference on Computer Vision*, 2019, pp. 3879–3888.

[23] R. Gao, R. Feris, and K. Grauman, "Learning to separate object sounds by watching unlabeled video," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 35–53.

[24] H. Zhao, C. Gan, A. Rouditchenko, C. Vondrick, J. McDermott, and A. Torralba, "The sound of pixels," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 570–586.

[25] M. Kolbæk, D. Yu, Z. Tan, and J. Jensen, "Multitalker speech separation with utterance-level permutation invariant training of deep recurrent neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 10, pp. 1901–1913, 2017.

[26] H. Erdogan, J. R. Hershey, S. Watanabe, and J. Le Roux, "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," in *ICASSP*. IEEE, 2015, pp. 708–712.

[27] K. P. Murphy, "Conjugate bayesian analysis of the gaussian distribution," 2007. [Online]. Available: https://www.cs.ubc.ca/~murphyk/Papers/bayesGauss.pdf

[28] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in PyTorch," 2017.

[29] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *ICASSP*. IEEE, 2015, pp. 5206–5210.

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

[31] F. Landini, S. Wang, M. Diez, L. Burget *et al.*, "BUT system for the second DIHARD speech diarization challenge," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6529–6533.

[32] D. Raj, P. Denisov, Z. Chen, H. Erdogan, Z. Huang, M.-K. He, S. Watanabe *et al.*, "Integration of speech separation, diarization, and recognition for multi-speaker meetings: System description, comparison, and analysis," in *IEEE Spoken Language Technology Workshop (SLT)*, 2021.

[33] 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 *Interspeech*, 2018.

[34] D. Povey, V. Peddinti, D. Galvez, P. Ghahremani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for asr based on lattice-free MMI," in *Interspeech*, 2016.

[35] Z. Huang, S. Watanabe, Y. Fujita, P. García, Y. Shao, D. Povey, and S. Khudanpur, "Speaker diarization with region proposal network," in *ICASSP*. IEEE, 2020, pp. 6514–6518.

[36] I. Medennikov, M. Korenevsky, T. Prisyach, Y. Khokhlov, M. Korenevskaya *et al.*, "Target-Speaker Voice Activity Detection: A Novel Approach for Multi-Speaker Diarization in a Dinner Party Scenario," in *Interspeech*, 2020, pp. 274–278. [Online]. Available: http://dx.doi.org/10.21437/Interspeech.2020-1602