LEARNING SPEAKER REPRESENTATION FOR NEURAL NETWORK BASED MULTICHANNEL SPEAKER EXTRACTION

Kateřina Žmolíková[1,2], *Marc Delcroix*[1], *Keisuke Kinoshita*[1], *Takuya Higuchi*[1], *Atsunori Ogawa*[1], *Tomohiro Nakatani*[1]

[1]NTT Communication Science Laboratories, NTT Corporation, Kyoto, Japan
[2]Brno University of Technology, Speech@FIT, Czech Republic

ABSTRACT

Recently, schemes employing deep neural networks (DNNs) for extracting speech from noisy observation have demonstrated great potential for noise robust automatic speech recognition. However, these schemes are not well suited when the interfering noise is another speaker. To enable extracting a target speaker from a mixture of speakers, we have recently proposed to inform the neural network using speaker information extracted from an adaptation utterance from the same speaker. In our previous work, we explored ways how to inform the network about the speaker and found a speaker adaptive layer approach to be suitable for this task. In our experiments, we used speaker features designed for speaker recognition tasks as the additional speaker information, which may not be optimal for the speaker extraction task. In this paper, we propose a usage of a sequence summarizing scheme enabling to learn the speaker representation jointly with the network. Furthermore, we extend the previous experiments to demonstrate the potential of our proposed method as a front-end for speech recognition and explore the effect of additional noise on the performance of the method.

Index Terms— speaker extraction, speaker adaptive neural network, multi-speaker speech recognition, speaker representation learning, beamforming

1. INTRODUCTION

In recent years, recognition of speech in adverse conditions have improved significantly due to both more robust systems and advanced front-end enhancement. However, recognizing speech corrupted by interference from other speakers still remains very difficult. Conventionally, the problem of interfering speakers has been tackled by methods including Non-negative matrix factorization [1] for single-channel case and Independent Component Analysis [2] or statistical model based systems making use of spatial cues [3, 4, 5, 6] for multi-channel case.

Meanwhile, for the problem of extracting speech in presence of noise, deep learning based approaches gained much attention. Methods based on denoising auto-encoders [7] or mask-estimation networks [8, 9] have been shown to be efficient for this task. For the case of multi-channel recordings, recently proposed methods [10, 11] successfully combine the deep neural network enhancement with conventional beamforming by using the masks estimated by the neural network to compute the beamforming filters.

To mimic this scheme in the case of interfering speakers, one needs to deal with the problem of neural network based estimation of the target speaker mask, i.e. mask extracting speech of the target speaker from a mixture. Several recent works attempted to apply deep learning to recover speech corrupted by other speakers. Namely deep clustering [12, 13] and its variants [14] have made a significant progress. In other approach, permutation invariant training [15, 16] was applied to this task with promising results.

In our previous work [17], we have proposed an alternative approach, which makes use of speaker information to make the neural network follow the target speaker through an utterance and extract it from the mixture. By focusing on the target speaker we make the processing independent of the number of speakers in the mixture and avoid permutation ambiguity problem. This problem arises when the neural network aims at recovering all speakers in the mixture at once — it is then ambiguous which of the speakers is associated with which output and the association can change between different processing segments. The specific use of the speaker information overcomes this problem, because it enables to follow the speaker over different processing segments and even over different recordings or sessions.

In the previous work, we explored different methods of passing the speaker information to the network. The experiments have shown efficiency of speaker adaptive layer, a method previously proposed for speaker adaptation of acoustic models [18, 19]. In this method, one of the layers of the network is factorized into several sub-layers. The output of this layer is then obtained as a weighted combination of the outputs of the sub-layers, where the weights depend on the speaker information. For the speaker information, we extract a representation of the speaker from a short adaptation utterance containing only the target speaker's speech. Although the experiments with this approach yielded promising results, they have shown a need for closer investigation of a proper choice of the speaker representation. Especially, we noticed the performance degrades greatly for same-gender mixtures. One factor limiting the performance is that the used speaker representations were designed for speaker recognition and not optimized for the speaker extraction task.

In this work, we further investigate the choice of a speaker representation for this task. Notably, we employ a sequence sum-

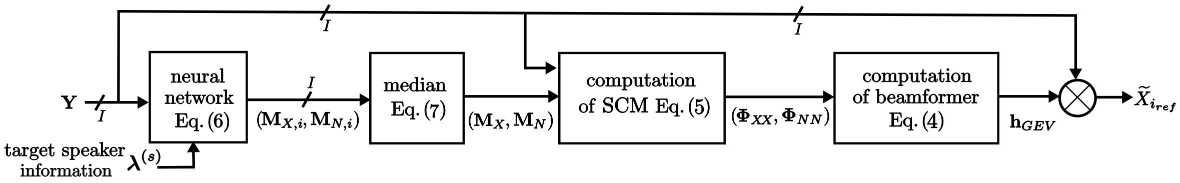


Fig. 1: The entire processing with the multichannel signal $\mathbf{Y}_i$ as input and the estimated target speaker signal $\widetilde{\mathbf{X}}_{i_{\mathrm{ref}}}$ as the output.

marizing framework [20] which enables to learn the speaker representation jointly with the neural network. Furthermore, in contrast with the previous work, we test the efficiency of the speaker extraction on a speech recognition task. We also extend the investigations to explore the effect of noise on the performance of the method.

The rest of the paper is structured as follows. In Section 2, we summarize the entire processing chain which uses neural network based beamforming. In Section 3, we then introduce the scheme of speaker adaptive layer used to adapt the network for extracting the target speaker. In Section 4, we describe the used speaker information and propose a use of sequence summarizing network. Section 5 compares this work with related works and finally, Section 6 describes the experiments and Section 7 concludes the paper.

2. NEURAL NETWORK BASED BEAMFORMING

In this section, we describe the overall processing chain which estimates a speech signal of a target speaker from an input mixture. It consists of a mask estimation part performed by a neural network and a beamforming part which is based on the estimated masks. This scheme is based on the work in [10] where the neural network based beamformer was used for speech-noise separation. Our work focuses on the mask estimation part, but we briefly review the entire scheme for completeness. The entire processing is depicted in Fig. 1.

2.1. Mask based beamforming

In this work, we model the signal received at i-th microphone in the Short time Fourier transform (STFT) domain as

$$Y_i(t,f) = \underbrace{S_i^{(0)}(t,f)}_{} + \underbrace{\sum_{j=1}^{J-1} S_i^{(j)}(t,f) + V_i(t,f)}_{} \tag{1}$$

$$= \quad X_i(t,f) + \qquad N_i(t,f), \tag{2}$$

where $i = 1 \dots I$ is the microphone index, $t = 1 \dots T$ is the time frame index, $f = 1 \dots F$ is the frequency-bin index, $Y_i(t,f)$ is the observed signal at the i-th microphone, $S_i^{(0)}(t,f)$ is the image of the speech signal of the target speaker, $S_i^{(j)}(t,f)$ is the image of the speech signal of j-th interfering speaker and $V_i(t,f)$ is the noise signal. We will denote the desired signal as $X_i(t,f)$ and the undesired signal collectively as $N_i(t,f)$.

To obtain the estimated image of the target signal at the reference microphone i_{ref}, the beamforming process proceeds as follows

$$\widetilde{X}_{i_{\mathrm{ref}}}(t,f) = \mathbf{h}^{\mathrm{H}}(f)\mathbf{Y}(t,f), \tag{3}$$

where $\mathbf{h}^{\mathrm{H}}(f)$ is a vector of beamforming coefficients and $\mathbf{Y}(t,f) = [Y_1(t,f) \dots Y_I(t,f)]^{\mathrm{T}}$. For the computation of the beamforming filters we followed [10] and used the Generalized Eigenvector beamformer (GEV) with filters computed as

$$\mathbf{h}_{\mathrm{GEV}}(f) = \underset{\mathbf{h}(f)}{\arg\max} \frac{\mathbf{h}^{\mathrm{H}}(f)\mathbf{\Phi}_{XX}(f)\mathbf{h}(f)}{\mathbf{h}^{\mathrm{H}}(f)\mathbf{\Phi}_{NN}(f)\mathbf{h}(f)}, \tag{4}$$

where $\mathbf{\Phi}_{XX}(f)$ and $\mathbf{\Phi}_{NN}(f)$ are the spatial covariance matrices (SCM) of the desired and undesired signal, respectively. They can be obtained as

$$\mathbf{\Phi}_{rr}(f) = \sum_{t=1}^{T} M_r(t,f)\mathbf{Y}(t,f)\mathbf{Y}^{\mathrm{H}}(t,f), \tag{5}$$

where $r \in \{X, N\}$ and $M_r(t,f)$ denotes a time-frequency mask for the desired or undesired signal.

2.2. Neural network based mask estimation

We use a neural network to obtain the time-frequency masks $M_r(t,f)$ needed to recover the desired signal. The input of the network consists of a mixture of speech signals and it is trained to output mask for the target speaker. To be able to extract the target speaker, we inform the network using additional speaker information. The processing by the neural network then takes place as follows

$$(\mathbf{M}_{X,i}(t), \mathbf{M}_{N,i}(t)) = g(\mathbf{y}_i(t), \boldsymbol{\lambda}^{(s)}), \tag{6}$$

where g is the transformation computed by the neural network, $\mathbf{y}_i(t) = [|Y_i(t,1)| \dots |Y_i(t,F)|]$ are the input features, $\mathbf{M}_{r,i}(t) = [M_{r,i}(t,1) \dots M_{r,i}(t,F)]$ and $\boldsymbol{\lambda}^{(s)}$ is an additional speaker information about the target speaker s. Note that the neural network processes data from each channel separately and final masks are then obtained from the channel specific masks as

$$\mathbf{M}_r(t) = \underset{i}{\mathrm{median}}(\mathbf{M}_{r,i}(t)). \tag{7}$$

The network is trained to predict ideal binary masks which are computed using the ratio of the target and the interference signal power in each time-frequency point obtained from parallel noisy and clean data.

The speaker information $\boldsymbol{\lambda}^{(s)}$ is obtained from an adaptation utterance — an utterance spoken by the target speaker without other interfering speakers. The way of conditioning the neural network processing g on the speaker information will be precised in Sections 3 and 4.

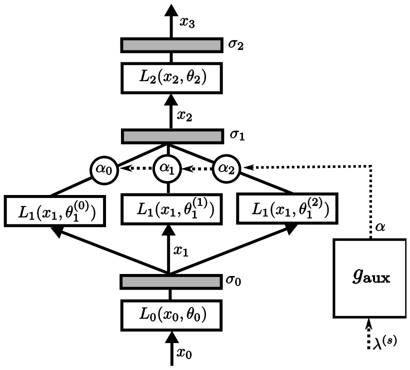


Fig. 2: Scheme of the speaker adaptive layer configuration.

3. SPEAKER ADAPTIVE LAYER SCHEME

In our previous work [17], we explored different ways of changing the network's behavior depending on the speaker information $\boldsymbol{\lambda}^{(s)}$. We sought for a method amenable to speakers unseen during the training and at the same time powerful enough to modify the network processing substantially depending on the speaker information. Our investigations included adding the speaker information to the input layer as an additional feature, training speaker specific networks or layers. The approach best complying with the requirements turned out to be the speaker adaptive layer scheme inspired by a method developed for speaker adaptation of acoustic models [18, 19].

In this approach, one of the layers of the network is factorized into several sub-layers. The output of this factorized layer is obtained as a weighted combination of the outputs of the sub-layers. The weights of this combination are inferred by an auxiliary network which has speaker information as its input. This way, the network parameters can be adapted to extract the target speaker. Both the main and auxiliary network are trained jointly.

We can express the computation of the neural network as

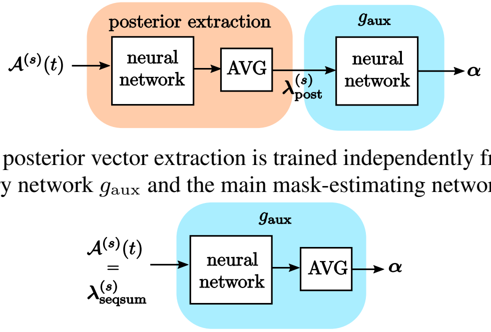
$$\mathbf{x}_{n+1} = \begin{cases} \sigma_n(L_n(\mathbf{x}_n;\theta_n)) & \text{for } n \neq k, \\ \sigma_n(\sum_{m=0}^{M-1} \alpha_m^{(s)} L_n(\mathbf{x}_n;\theta_n^{(m)})) & \text{for } n = k, \end{cases} \tag{8}$$

where k is the index of the factorized layer, $\mathbf{x}_n$ denotes the input to the nth layer, $L_n(\mathbf{x},\theta)$ is the transformation computed by the nth layer parametrized by θ and σ_n is an activation function. For fully connected layers $\theta = \{\mathbf{W}, \mathbf{b}\}$ and $L(\mathbf{x},\theta) = \mathbf{W}\mathbf{x} + \mathbf{b}$, where $\mathbf{W}$ is a weight matrix and $\mathbf{b}$ is a bias vector. The weights $\boldsymbol{\alpha}$ are computed from the speaker information $\boldsymbol{\lambda}^{(s)}$ by an auxiliary network which is trained jointly with the rest of the network as

$$\boldsymbol{\alpha} = g_{\text{aux}}(\boldsymbol{\lambda}^{(s)}). \tag{9}$$

Figure 2 shows an example of this configuration.

The effect of the speaker information is a modification of an entire layer in the network and thus is more powerful than simply appending the information as an additional feature to the input of the network. Furthermore, this method is easily usable for unseen speakers as the weights $\boldsymbol{\alpha}$ can be inferred from any kind of speaker information. The experiments in [17] indeed confirmed the ability of this architecture to extract different speakers from the mixture depending on the speaker information.



(a) The posterior vector extraction is trained independently from the auxiliary network g_{aux} and the main mask-estimating network.

(b) In the sequence-summarizing network scheme, the auxiliary network g_{aux} works on the adaptation utterance directly.

Fig. 3: Scheme of two options of using speaker information in the auxiliary network g_{aux}. In both cases, the network g_{aux} is trained jointly with the main mask-estimating network.

4. SPEAKER INFORMATION

In our first experiments [17], we followed the way speaker adaptive layer was used for speaker adaptation [19], i.e. first, a fixed-length speaker representation is extracted from the adaptation utterance and this is then used as the speaker information on the input of the auxiliary network. This *separate scheme* is detailed below. However, the experiments have shown a need for a closer investigation of the form in which the speaker information is presented to the network. Here, we propose a *joint scheme* using the sequence summarizing network [20], which enables to learn the speaker representation jointly with learning to extract the speaker. This way, we omit the separate step of extracting the speaker representation that could possibly lead to losing information which is important for identifying the speaker in the mixture.

4.1. Separate speaker representation extraction

In the separate scheme, the speaker information on the input of the auxiliary network is a fixed-length vector extracted from an adaptation utterance. The auxiliary network is then a simple feed-forward network mapping this vector to the weights $\boldsymbol{\alpha}$. As the vectors representing speakers, we can use for example i-vectors or a representation extracted by a neural network such as speaker posteriors. We refer to [21, 22] for an exhaustive description of i-vectors.

For obtaining the speaker posteriors, a separate neural network is trained which classifies a frame of speech features into classes formed by the speakers from the training set. Using this network on an adaptation utterance from an unseen speaker results into a sequence of posterior vectors encoding the 'similarity' of the unseen speaker and the training speakers. The final speaker representation is then given by averaging the posterior vector over the entire adaptation utterance as

$$\boldsymbol{\lambda}^{(s)}_{\text{post}} = \frac{1}{T_{\mathcal{A}}} \sum_{t=0}^{T_{\mathcal{A}}-1} \text{post}(\mathcal{A}^{(s)}(t)), \tag{10}$$

where $\mathcal{A}^{(s)}(t)$ are features extracted from t-th frame of an adaptation utterance of a speaker s, $\mathrm{post}(\mathcal{A}^{(s)}(t))$ are the posteriors predicted by the speaker classifier network from these features and $T_{\mathcal{A}}$ is the number of frames in the adaptation utterance. The transformation g_{aux} is then performed by a simple feed-forward neural network that maps the speaker posteriors to the weights $\boldsymbol{\alpha}$. This scheme is depicted in Figure 3a.

4.2. Joint speaker representation learning

In the separate scheme, it is unclear what speaker representation to extract so that it would best capture the information useful for the main network to extract a speaker from a mixture. To avoid the middle step of extracting the speaker representation we propose to use the adaptation utterance directly at the input of the auxiliary network.

In this case, the auxiliary network obtains frame-level features which should be mapped to utterance-level weights $\boldsymbol{\alpha}$. This can be simply achieved with the sequence summarizing method [20] which employs an averaging operation directly into the network. The sequence summarization was previously proposed for learning auxiliary features for speaker adaptation. Here, we combine this method with the speaker adaptive layer scheme.

The speaker information is in this case time-dependent and matches the adaptation utterance features

$$\boldsymbol{\lambda}^{(s)}_{\mathrm{seqsum}}(t) = \mathcal{A}^{(s)}(t) \tag{11}$$

and the auxiliary network includes the averaging operation at its output resulting in utterance-level weights

$$\boldsymbol{\alpha} = g_{\mathrm{aux}}(\boldsymbol{\lambda}^{(s)}_{\mathrm{seqsum}}(t)) = \frac{1}{T_{\mathcal{A}}} \sum_{t=0}^{T_{\mathcal{A}}-1} z(\boldsymbol{\lambda}^{(s)}_{\mathrm{seqsum}}(t)), \tag{12}$$

where $z(\boldsymbol{\lambda}^{(s)}_{\mathrm{seqsum}}(t))$ are activations of the last layer of the auxiliary network with input $\boldsymbol{\lambda}^{(s)}_{\mathrm{seqsum}}(t)$ and $T_{\mathcal{A}}$ is the number of frames in the adaptation utterance. The auxiliary network including the average operation is trained jointly with the main network. The network thus itself extracts the information useful for the extraction task. Figure 3b depicts this scheme.

5. RELATED WORK

The enhancement of multi-speaker recordings using neural network based approach have been explored in several previous works [12, 14, 15, 23]. In contrast with our work, these works aim to recover all the speakers present in the recording at once regardless of their identity. This kind of approach gives rise to two problems. The first problem is the dependency of the number of speakers, i.e. the number of speakers needs to be either known a priori or estimated at some point of the processing. The second problem is the permutation ambiguity, i.e. when the method processes the input segment by segment, the order of the recovered speakers can change from one segment to another and additional tracking is required to associate the corresponding speakers. By introducing the speaker information into the processing, these problems are avoided, since the network learns to simultaneously track and extract the target speaker only.

This work is also related to the area of speaker adaptation of neural networks for acoustic modeling, which also aims to modify the processing of the neural network by introducing the speaker information. The speaker adaptive layer approach used in this work was previously used for speaker adaptation in [18, 19]. Strong difference between these two use cases is that while for acoustic modeling, the output of the network should be just slightly altered by the speaker information, in the speaker extraction task, the speaker information should completely change the outcome.

The sequence summarizing scheme proposed in this work, was also previously used for speaker adaptation in [20]. In this work, the sequence summarizing network was used to create auxiliary features for the network. In our work, we combine this scheme with the speaker adaptive layer method and apply it to the speaker extraction task.

6. EXPERIMENTS

In this section, we evaluate the performance of the proposed scheme. We use simulated data with two active speakers. We first report results with Signal to distortion ratio (SDR) measure [24] to compare the newly proposed sequence summarizing method with the results in the previous work [17]. Then, we extend the experiments to evaluate automatic speech recognition performance and finally, test the effect of additional noise.

6.1. Data

The created data are based on recordings from Wall Street Journal dataset [25]. The lists of utterances for the training, development and evaluation sets were taken from CHiME3 challenge [26], this means 7138 utterances from 83 speakers in the training set, 410 utterances from 10 speakers in the development set and 330 utterances from 10 speakers in the evaluation set. For each utterance, we mixed an interference utterance from a different speaker within the same set with signal-to-interference ratio of $0\,\mathrm{dB}$ on average.

To simulate the multichannel signals, we used room impulse responses created with the image method [27, 28] with a circular microphone array with 8 microphones, $20\,\mathrm{cm}$ diameter and $\mathrm{RT60}{=}0.2\,\mathrm{s}$. The speakers are located at 1 or $1.5\,\mathrm{m}$ meter distance from the microphone array, in angles from range 0 to 180°.

For each mixture, we randomly chose an adaptation utterance from the target speaker (different than the utterance in the mixture) and used the image of this utterance at one of the microphones in the array. The length of the utterance is about $10\,\mathrm{s}$ on average. Note that in our experiment, the adaptation utterance may be uttered from a different location than the test utterance and may contain different level of background noise.

For the data with additional noise, we used noise recorded for REVERB challenge [29]. For the training data, we used noise recorded in SmallRoom1, while for the development and evaluation sets, noise recorded in SmallRoom2. For each mixture, two random segments from the noise recordings were chosen and two noise sources were placed in the room by convolving them with generated room impulse responses. We controlled the SNR of the mixture with respect to noise to be 5,10,15 or $20\,\mathrm{dB}$. The noise of the same SNR levels was also used in the adaptation utterances (however the level of noise in an utterance and its correponding adaptation utterance may differ).

6.2. Settings

6.2.1. Mask estimation network settings

The architecture of the mask estimation network was similar to that used in [10] with the size of the layers changed to suit better the speaker extraction task. The network consisted of 4 layers, i.e. one BLSTM layer, two fully connected layers with ReLU activation and one fully connected layer with a sigmoid activation. The number of neurons in the four layers is 512-1024-1024-512, respectively. As the speaker adaptive layer, we chose the second layer in the network and factorized it into 30 sub-layers. The auxiliary network predicting the weights $\boldsymbol{\alpha}$ was composed of two fully connected layers with 50 neurons and a ReLU activation and the output fully connected layer with a linear activation (and the averaging operation in the case of the sequence summarizing network). The main and the auxiliary networks were jointly trained to optimize cross-entropy between ideal binary masks and the estimated masks. For the optimization we used Adam optimizer.

6.2.2. Speaker representation settings

The different speaker representations were extracted from an adaptation utterance, i.e. a different utterance spoken by the target speaker, including reverberation and noise for the noise experiments.

We used a feed-forward network with three layers and a sigmoid activation with 512 neurons per layer for extracting the posterior features. The input of the network were filterbank features and the network was trained to classify each frame into classes formed by speakers in the training data or an additional class for silence (the silence class is ignored when extracting the posterior vectors for the test data). This corresponds to $83 + 1$ classes, as there are 83 speakers in the training set.

For getting the i-vectors, we trained an i-vector extractor using Kaldi [31]. The extractor was trained on single-speaker reverberant training data, using MFCC features and cepstral mean normalization. The Universal Background Model (UBM) consisted of 2048 gaussians and the final i-vectors had dimensionality of 100. We used utterance-level i-vectors, i.e. we computed an i-vector for each adaptation utterance.

The settings of the sequence summarizing network follow the settings of the auxiliary network described above. The input features of the network are coefficients of the magnitude spectra of each frame of the adaptation utterance.

6.2.3. Beamforming settings

For beamforming, we used GEV beamformer as described in Section 2.1. We post-processed the masks obtained by the neural network by thresholding values less than 0.3 to 0 to discard the most uncertain regions. The noise spatial covariance matrix was regularized by adding $\epsilon = 1e^{-3}$ to the diagonal. The output signal was additionally processed by a single-channel postfilter [10, 30] to reduce the speech distortions. The window size used to compute STFT was $32\,\mathrm{ms}$ with a $8\,\mathrm{ms}$ shift.

6.2.4. ASR settings

We used a simple DNN acoustic model, which consisted of 5 fully connected hidden layers with 2048 nodes and ReLU activation functions for the ASR evaluation. The output layer had 2048 nodes corresponding to the HMM states.

The input features of the acoustic model consisted of 40 log mel filterbank coefficients with their delta and delta-delta coefficients, and a context extension window of 19 frames. The features were mean normalized per utterance.

For training the acoustic model, we used HMM state alignments obtained from single channel noise-free training data using a GMM-HMM system. We used discriminative pre-training to initialize the DNNs. The training data consisted of the same utterances as for the training of the mask estimation DNN.

We prepared 2 types of acoustic model, one trained with the single channel noise-free training data and one trained on data processed by the same front-end as the data used for evaluation.

All results were obtained using a trigram language model. The ASR results are presented in terms of word error rate (WER).

6.3. Speaker representation experiments

In the first experiments, we aimed to compare the efficiency of using different speaker information as described in Section 4. As a follow-up to our previous work [17], we first tested the performance with SDR [24]. The SDR measures the distortion compared to the clean, single-speaker recordings. Table 1 shows the SDR for the original recordings before enhancement (mixtures), recordings enhanced using ideal binary masks for deriving the beamformer (*enh oracle*) and recordings enhanced using different types of speaker information. The results are further divided into same-gender and different-gender mixtures as this is a factor strongly influencing the difficulty of the extraction task.

As we can see from the results, using posterior features (*enh post*) or i-vectors (*enh ivec*) lead to comparable performance. Both cases lead to much larger improvement in the different-gender case than in the same-gender case. The usage of the sequence summarizing network (*enh seqsum*) improves the performance further, more significantly for same-gender mixtures. The direct learning of the weights $\boldsymbol{\alpha}$ from the adaptation utterance thus succeeds to extract useful information about the speaker better than the separately extracted speaker representations.

The same extraction methods are then compared in terms of WER in Table 2. The two sets of results in Table 2 differ by the data used for training the ASR systems. In the first case, the system is trained on clean single-speaker recordings and tested on the enhanced data. In the second case, we enhance also the training data and train the ASR systems on the corresponding enhanced data.

The first observation in these results is the large difference between the systems trained on clean and enhanced data. Even when using the oracle masks for beamforming, the system trained on clean data has difficulties decoding the enhanced recordings. This is caused mainly by a large number of insertions which occur when there is residual speech of the interfering speaker. Even when the speech signal of the second speaker is strongly suppressed, the system trained on purely clean data tries

Table 1: Results of the experiments with different speaker representations (posteriors, i-vectors) and sequence summarizing network compared to the original mixtures and oracle enhancement with ideal binary masks. All results are in terms of SDR in dB (the higher the better).

	Development set			**Evaluation set**		
	same	**diff**	**all**	**same**	**diff**	**all**
mixtures	-0.18	0.52	0.17	-0.54	0.38	-0.02
enh oracle	8.85	8.64	8.75	8.96	9.55	9.29
enh post	1.29	8.08	4.75	2.64	8.19	5.73
enh ivec	3.20	7.78	5.54	2.38	8.01	5.52
enh seqsum	**5.03**	**8.14**	**6.61**	**3.81**	**8.72**	**6.55**

Table 2: Results of the ASR experiments with different speaker representations (posteriors, i-vectors) and sequence summarizing network compared to the clean single-speaker recordings, the original mixtures and oracle enhancement with ideal binary masks. All results are in terms of WER[%].

ASR train data→	***single-speaker***		***matched***	
test data	**Dev**	**Eval**	**Dev**	**Eval**
single speaker	6.40	6.15	6.40	6.15
mixtures	89.59	92.7	77.87	80.35
enh oracle	23.25	25.44	8.39	7.57
enh post	45.63	46.09	27.48	24.62
enh ivec	43.28	47.73	22.92	27.46
enh seqsum	37.8	41.88	**17.75**	**20.81**

to transcribe the weak residual signal. This effect vanishes when the system is retrained with the enhanced data, in this case the system is robust towards the small interferences and the number of insertions is greatly reduced.

The comparison between the different methods follows the same trend as the speech enhancement results. The posterior features and i-vectors lead to very comparable accuracy while the sequence summary network outperforms both significantly in all cases. In both the enhancement and speech recognition results, we can still see a gap compared to the beamforming with oracle masks, especially in the same-gender case. We did not include gender-specific results of ASR due to limited space, however, the difference between same and different gender mixtures remains (e.g. for sequence summarizing network and matched ASR train data, the WER on development set is 10.22/25.3 for same and different gender mixtures, respectively). This indicates a further possibility for improvement by e.g. further optimizing the neural network architecture or changing the ratio of same and different gender mixtures in the training data.

6.4. Effect of noise

To keep the investigations simple, the previous experiments were conducted with speaker mixtures without any additional noise. Here, we extend the experiments to data with different levels of added noise from 20 dB to 5 dB and observe its effect on the speech recognition accuracy. The experiments are performed with the best setup — the sequence summarizing network.

Table 3: Results of the ASR experiments with different levels of additional noise in the data in WER[%]. The enhancement network is trained on mix of clean, 15 dB and 5 dB data. The ASR is trained on the same set processed by the enhancement. Note that the difference between ∞ dB results and results in Table 2 are caused by the model trained on different (noisy) data.

	enh seqsum		oracle	
test data	**Dev**	**Eval**	**Dev**	**Eval**
∞ dB	21.46	23.44	9.16	8.99
20 dB	18.38	20.55	8.98	7.14
15 dB	19.75	21.22	9.09	7.04
10 dB	22.88	23.63	9.84	7.96
5 dB	28.01	28.79	13.34	10.22

To make the mask estimation network robust to the noise, we trained it on a set composed of utterances with three different levels of added noise of SNRs of 5 dB, 15 dB and ∞ dB (no added noise). The same set of utterances after processing by the trained enhancement was used to train the ASR system. The enhancement and ASR was then used on test data with added noise of ∞ dB, 20 dB, 15 dB, 10 dB and 5 dB.

Results from the experiments can be seen in Table 3. We can see a surprising effect of the 20 dB and 15 dB noise, which actually improves performance compared to the data with no additional noise. This gain is due to the insertion errors discussed in the previous section. When a small noise is also present in the data, the residual signal of the interfering speaker blends with the noise and the ASR system more likely classifies this part as a silence than in the case without any noise. Since in realistic setting, we can expect small amount of noise to occur, the performance drop for the absolutely clean data is not very critical. Increasing the level of the noise degrades the performance as we would expect. However, even when the noise is severe, the speaker extraction still leads to satisfactory results compared to the recognition of unprocessed mixtures. Note that these experiments were performed with a simple ASR system. Using more robust ASR methods such as CNNs, sequence discriminative training, data augmentation etc. could further improve these results.

7. CONCLUSION

In this paper, we explored a method for extracting a speaker from a multichannel mixture of multiple overlapping speakers based on informing a neural network about the target speaker. We extended our previous work by investigating the form of the speaker information presented to the network. We proposed to use a sequence summarizing scheme which enables to learn the extraction of a speaker representation jointly with the network and shown that this approach outperforms extracting the speaker representation separately. The experiments confirmed the efficiency of this method for automatic speech recognition task and studied the effect of additional noise on the performance. In future work, we plan to experiment with optimizing the speaker extraction jointly with the ASR system and investigate this scheme with real recordings such as meetings [32].

8. ACKNOWLEDGMENT

Katerina Zmolikova is Brno Ph.D. Talent Scholarship Holder — Funded by the Brno City Municipality.

9. REFERENCES

[1] Paris Smaragdis, "Convolutive speech bases and their application to supervised speech separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 1, pp. 1–12, 2007.

[2] Aapo Hyvärinen, Juha Karhunen, and Erkki Oja, "Independent component analysis," 2001.

[3] Ozgur Yilmaz and Scott Rickard, "Blind separation of speech mixtures via time-frequency masking," *IEEE Transactions on Signal Processing*, vol. 52, no. 7, pp. 1830–1847, 2004.

[4] Hiroshi Sawada, Shoko Araki, and Shoji Makino, "Underdetermined convolutive blind source separation via frequency bin-wise clustering and permutation alignment," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 3, pp. 516–527, 2011.

[5] Michael I. Mandel, Ron J. Weiss, and Daniel P.W. Ellis, "Model-based expectation-maximization source separation and localization," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 2, pp. 382–394, 2010.

[6] Dang H. Tran Vu and Reinhold Haeb-Umbach, "Blind speech separation employing directional statistics in an expectation maximization framework," in *Acoustics, Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, 2010, pp. 241–244.

[7] Yong Xu, Jun Du, Li-Rong Dai, and Chin-Hui Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 7–19, 2015.

[8] Felix Weninger, John R. Hershey, Jonathan Le Roux, and Bjorn Schuller, "Discriminatively trained recurrent neural networks for single-channel speech separation," in *2014 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2014, pp. 577–581.

[9] Arun Narayanan and DeLiang Wang, "Ideal ratio mask estimation using deep neural networks for robust speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 7092–7096.

[10] Jahn Heymann, Lukas Drude, and Reinhold Haeb-Umbach, "Neural network based spectral mask estimation for acoustic beamforming," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, 2016, pp. 196–200.

[11] Hakan Erdogan, John R. Hershey, Shinji Watanabe, Michael Mandel, and Jonathan Le Roux, "Improved MVDR beamforming using single-channel mask prediction networks," in *Interspeech*, 2016.

[12] John R. Hershey, Zhuo Chen, Jonathan Le Roux, and Shinji Watanabe, "Deep clustering: Discriminative embeddings for segmentation and separation," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, 2016, pp. 31–35.

[13] Yusuf Isik, Jonathan Le Roux, Zhuo Chen, Shinji Watanabe, and John R. Hershey, "Single-channel multi-speaker separation using deep clustering," in *Interspeech*, 2016, pp. 545–549.

[14] Zhuo Chen, Yi Luo, and Nima Mesgarani, "Deep attractor network for single-microphone speaker separation," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, 2017.

[15] Dong Yu, Morten Kolbæk, Zheng-Hua Tan, and Jesper Jensen, "Permutation invariant training of deep models for speaker-independent multi-talker speech separation," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, 2017.

[16] Dong Yu, Xuankai Chang, and Yanmin Qian, "Recognizing multi-talker speech with permutation invariant training," *arXiv preprint arXiv:1704.01985*, 2017.

[17] Katerina Zmolikova, Marc Delcroix, Keisuke Kinoshita, Takuya Higuchi, Atsunori Ogawa, and Tomohiro Nakatani, "Speaker-aware neural network based beamformer for speaker extraction in speech mixtures," in *Interspeech*, 2017.

[18] Marc Delcroix, Keisuke Kinoshita, Takaaki Hori, and Tomohiro Nakatani, "Context adaptive deep neural networks for fast acoustic model adaptation," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, 2015, pp. 4535–4539.

[19] Marc Delcroix, Keisuke Kinoshita, Chengzhu Yu, Atsunori Ogawa, Takuya Yoshioka, and Tomohiro Nakatani, "Context adaptive deep neural networks for fast acoustic model adaptation in noisy conditions," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*. IEEE, 2016, pp. 5270–5274.

[20] Karel Vesely, Shinji Watanabe, Katerina Zmolikova, Martin Karafiat, Lukas Burget, and Jan Honza Cernocky, "Sequence summarizing neural network for speaker adaptation," in *Acoustics, Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, 2016, pp. 5315–5319.

[21] Najim Dehak, Patrick J. Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[22] Ondřej Glembek, Lukáš Burget, Pavel Matějka, Martin Karafiát, and Patrick Kenny, "Simplification and optimization of i-vector extraction," in *Acoustics, Speech and Signal Processing (ICASSP), 2011 IEEE International Conference on*. IEEE, 2011, pp. 4516–4519.

[23] Chao Weng, Dong Yu, Michael L. Seltzer, and Jasha Droppo, "Deep neural networks for single-channel multi-talker speech recognition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 23, no. 10, pp. 1670–1679, 2015.

[24] Emmanuel Vincent, Rémi Gribonval, and Cédric Févotte, "Performance measurement in blind audio source separation," *IEEE transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[25] John Garofolo, "CSR-I (WSJ0) Complete LDC93S6A," `https://catalog.ldc.upenn.edu/ldc93s6a`, 1993.

[26] Jon Barker, Ricard Marxer, Emmanuel Vincent, and Shinji Watanabe, "The third CHiME speech separation and recognition challenge: Dataset, task and baselines," in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding*. IEEE, 2015, pp. 504–511.

[27] Jont B. Allen and David A. Berkley, "Image method for efficiently simulating small-room acoustics," *The Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943–950, 1979.

[28] Emanuel A. P. Habets, "Room impulse response generator," Tech. Rep., Technische Universiteit Eindhoven, 2010.

[29] Keisuke Kinoshita, Marc Delcroix, Takuya Yoshioka, Tomohiro Nakatani, Armin Sehr, Walter Kellermann, and Roland Maas, "The reverb challenge: A common evaluation framework for dereverberation and recognition of reverberant speech," in *Applications of Signal Processing to Audio and Acoustics (WASPAA), 2013 IEEE Workshop on*. IEEE, 2013, pp. 1–4.

[30] Ernst Warsitz and Reinhold 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.

[31] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., "The kaldi speech recognition toolkit," in *2011 IEEE Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, 2011, number EPFL-CONF-192584.

[32] Shoko Araki, Nobutaka Ito, Marc Delcroix, Atsunori Ogawa, Keisuke Kinoshita, Takuya Higuchi, Takuya Yoshioka, Dung Tran, Shigeki Karita, and Tomohiro Nakatani, "Online meeting recognition in noisy environments with time-frequency mask based mvdr beamforming," in *2017 Hands-free Speech Communications and Microphone Arrays (HSCMA)*, 2017, pp. 16–20.