ABSTRACT

The parametric Bayesian Feature Enhancement (BFE) and a data-driven Denoising Autoencoder (DA) both bring performance gains in severe single-channel speech recognition conditions. The first can be adjusted to different conditions by an appropriate parameter setting, while the latter needs to be trained on conditions similar to the ones expected at decoding time, making it vulnerable to a mismatch between training and test conditions. We use a DNN backend and study reverberant ASR under three types of mismatch conditions: different room reverberation times, different speaker to microphone distances and the difference between artificially reverberated data and the recordings in a reverberant environment. We show that for these mismatch conditions BFE can provide the targets for a DA. This unsupervised adaptation provides a performance gain over the direct use of BFE and even enables to compensate for the mismatch of real and simulated reverberated data.

Index Terms— robust speech recognition, deep neuronal networks, feature enhancement, denoising autoencoder

1. INTRODUCTION

Over the last few years, the usage of speech recognition in consumer electronics changed dramatically. Voice controlled personal assistants and systems demand for hands-free usage with larger distances between the speaker and usually a single microphone. These conditions challenge automatic speech recognition (ASR) systems to become more robust against environmental influences like noise and especially against reverberation effects.

The recently held REVERB challenge [1] gave some insights on how ASR systems can become more robust. It revealed that almost all of the best performing systems employ a Deep Neural Network - Hidden Markov Model (DNN-HMM) acoustic model. Additionally, it showed the effectiveness of data-driven feature enhancement methods like Non-negative Matrix Factorization (NMF) [2], a Denoising Autoencoder (DA) [3] or even a Deep Recurrent Neural Network (RNN) [4]. Other systems used parametric feature enhancement methods, e.g. Cross Transform [5] and the Weighted Prediction Error (WPE) algorithm [6]. But the results show a weakness of current DNN respectively data-driven approaches: As the channel mismatch between the training data and the evaluation data becomes larger, the performance drops significantly. A special case is the performance decrease when going from simulated reverberant data to real reverberant data which is visible throughout all approaches.

While a certain performance loss due to a mismatch condition is expected to happen for all systems, the data-driven systems do not yet provide a means to adjust them to new conditions using no or very few data. They have to be trained on conditions similar to the ones at decoding time, while parametric methods can be adjusted to new conditions just by an appropriate parameter setting.

We therefore investigate, if it is possible to use a parametric feature enhancement to adapt a data-driven approach to unseen conditions. For our investigations we focus on single-channel audio and choose the data-driven DA [7] and the parametric Bayesian Feature Enhancement (BFE) [8, 9]. We look at the influence of different mismatch situations on the recognition performance which arise from different room sizes (reverberation time), different distances between the speaker and microphone and differences between simulated reverberant data and actually recorded reverberant data.

In the next section, we describe the models used for feature enhancement. Afterwards, we give an overview over the backend and how it is combined with the two enhancement methods. The dataset is described in Sec. 4. We present the results in Sec. 5 and conclude in Sec. 6. We end by relating this paper to previous work and giving an outlook for further research in Sec. 7.

2. FEATURE ENHANCEMENT

2.1. Bayesian Feature Enhancement

In a reverberant environment the discrete-time microphone signal $y(k)$ results from a convolution of the clean speech signal $x(k)$ with the acoustic impulse response (AIR) $h(k)$ of finite length $L_h$ and additional noise $n(k)$

$$y(k) = \sum_{l=0}^{L_h-1} h(l)x(k-l) + n(k). \quad (1)$$

We then estimate the sequence of clean Log Mel Power Spectral Coefficients (LMPSCs) $x_{1:M}$ from the observed sequence $y_{1:M}$. The estimation is carried out in a Bayesian way [8, 10]. We introduce a state vector

$$z_m := \begin{pmatrix} (x_m)_1^T, \ldots, (x_{m-L_c+1})_1^T, (n_m)_1^T \end{pmatrix} \quad (2)$$

containing the last $L_c$ LMPSCs of the clean speech and the current noise LMPSCs. Its a posteriori probability density function (PDF) $p(z_m|y_m)$ is computed recursively with a prediction step

$$p(z_m|y_{1:m-1}) = \int p(z_m|z_{m-1}, y_{1:m-1})p(z_{m-1}|y_{1:m-1})dz_{m-1}$$
and an update step
\[
p(z_m | y_1:m) \propto p(y_m | z_m, y_1:m-1)p(z_m | y_1:m-1)
\] (3)
The prediction step requires an a priori model \( p(z_m | y_{m-1}, y_1:m-1) \) for the clean speech and noise LMPSCs. We employ a switching linear dynamical model (SLDM) for the speech and assume the noise signal to be a realization of a stationary white Gaussian stochastic process. The update step calls for an observation model \( p(y_m | z_m, y_1:m-1) \) which we choose to be a multivariate Gaussian with time-variant mean vector and covariance matrix.

The observation model relates the LMPSCs of clean speech and noise to the LMPSCs of noisy reverberant speech. As such, it requires a model of the AIR. In [10] we proposed to use the model by Polack [11]. This model assumes the AIR to be a realization of a white Gaussian noise process with an exponentially decaying envelope. Although only a coarse approximation to a real AIR, it has the advantage that it is characterized by a single parameter: The time constant of the exponential decay. This time constant is proportional to the room reverberation time \( T_{60} \), which is the time it takes until the energy in the tail of the AIR decays to \(-60\) dB of the total energy. It depends on the room properties and is independent of the distance between the speaker and the microphone. The parameter can be estimated blindly from reverberant speech with a precision well below \( \pm 100 \) ms (e.g. [12][13]) which is sufficient for our model to deliver good results.

Note that the direct signal and early reflections are not modelled well with this coarse model. BFE has been designed with distant speech in mind and it is to be expected that it is not that effective if the distance between the speaker and the microphone becomes small.

For an in-depth description of this approach we refer to [8, 9].

2.2. Denoising Autoencoder

A (stacked) DA is a network with multiple encoding layers, followed by one affine linear decoding layer when dealing with real-valued data like speech features in this case [7]. The encoding layers have the form
\[
h_i(z_i) = s(W_i z_i + b_i)
\] (4)

\( z_i \) is the input to the \( i \)-th hidden layer, \( W_i \) its weight matrix and \( b_i \) its bias vector. \( s() \) is a non-linearity like a sigmoid, tanh or ReLU.

The goal of the DA is to reconstruct clean (speech) features \( \hat{x} \) from corrupted input features \( x \). Here, the corruption is caused by reverberation and additional background noise. When providing these corrupted features as the input \( z_0 \) to the first layer, we want the output \( \hat{x} \) to be highly similar to the clean features. To learn the required mapping between noisy and clean features, a loss function describing the mean squared error \( ||x - \hat{x}||^2 \) is minimized during training.

Because this loss function is highly non-linear in the weights of the network, stochastic gradient descent is used to find a solution for this optimization problem. One which generalizes well needs a good initialization of the weights \( W_i \). This is achieved with a pre-training. The probably most common approach for this is the one presented by Hinton et al. [14]. In this paper, we use another approach, namely greedy supervised pre-training [15] with the extension of corrupting the input for the trainable layer as described in [16]. The main reason for this choice is that the latter requires no additional generative model and uses the same function as the fine-tuning. Nevertheless, the results are comparable.

We pre-train every layer for 15 epochs with a learning rate of 0.1 while fixing the weights of the already trained layers. During pre-training the clean corpus is used and the input for the trainable layer is corrupted by randomly setting 50% of it to zero. Afterwards we perform fine-tuning with an initial learning rate of 0.1 and a Newbob learning strategy. The input is the multicondition and the output the clean training data. Note that we do not apply additional corruption during the fine-tuning since the multicondition data is already a corrupted version of the training data.

The autoencoder is implemented using Theano [17] and features 3 hidden layers with 2048 sigmoid units. The input spans over 7 consecutive MFCC frames with 13 components. The output also consists of 7 frames during training. For the decode, we average over all seven appearances for every single frame as proposed in [18].

2.3. Adapting the DA with BFE

As mentioned in the introduction and as the results will show, the performance of the DA (and DNN alone) drops significantly when there is a larger mismatch between the training data and the data to be decoded. Therefore we adapt the DA to the new condition: We first enhance the data to be decoded with BFE, resulting in a cleaned-up version of the MFCC features. These are then used as a target to retrain the DA with an initial learning rate of 0.01. After this process, the DA can be used to enhance the data for the new condition.

Note that this approach is not limited to BFE but can be used with any parametric feature enhancement method, making it possible to adapt a DA to new environmental conditions.

3. Backend

For the backend we use the freely available RASR toolkit [19, 20]. We chose a hybrid approach for the recognition, where a DNN calculates the posteriors for each generalized triphone state. The number of decision tree based generalized triphone states is chosen to be 3000 for every setup. The network itself consists of 6 hidden layers with 2048 sigmoid units each. It has a factorized linear bottleneck structure where each weight matrix is factorized into two matrices of dimensions \( 2048 \times 256 \) and \( 256 \times 2048 \). This corresponds to adding an additional linear hidden layer with 256 units between each pair of hidden layers. Such a structure reduces the number of free parameters, accelerates the decoding and training process and was found to be effective against overfitting [21, 22].

To train the network, we first initialize the weights with a greedy layer-wise pre-training. Each layer is pre-trained for 2 epochs with the learning rate set to 0.016. Afterwards, the network is fine-tuned with an initial learning rate of 0.016. The Newbob strategy is used to control the learning rate for the following epochs. The batch size is set to 512. Instead of stochastic gradient descent, we use an optimization algorithm called Mean-Normalized Stochastic Gradient Descent (MN-SGD). This shows faster convergence properties and enables the training of a factorized network without further decomposition methods [22]. Finally, we decode a held-out set with the weights gained from each iteration. The weights with the best WER on the held-out set are then used for the experiments. For decoding, we use the bigram language model supplied with the database (see Sec. 4).

3.1. Combining BFE and the DNN backend

We also experimented with a combination of BFE with a DNN backend, without a DA. The BFE enhanced features are used as the input for the DNN trained on the clean training data. Though we tried different setups (DNN trained on multicondition, DNN retrained on
BFE features), the described method delivered the best results. Since the BFE is a parametric model, we also have to determine its parameters. As mentioned before, the crucial parameter is the reverberation time ($T_{60}$). For the SimDATA set, the parameter is known (see Sec. 4). For the REALDATA set we perform a grid search to find the best parameter. We also confirmed the parameters for the SimDATA using this technique. But again, this parameter could also be determined in an unsupervised fashion from the single-channel speech input. The SLDM necessary for the BFE is trained on the clean utterances of the training set.

3.2. Combining the DA and DNN backend

For the combination of the DA with the DNN backend, we take the DNN trained on the clean data and retrain it with the DA output of its training data and an initial learning rate of 0.01. In contrast to [3], this method delivered the best results for us. But the results without a retraining are only slightly worse. This indicates that the DA outputs features similar to clean ones, but apparently with small differences. The retraining allows the DNN to adapt to these.

For the adapted DAs we use the DNN trained on clean data.

4. DATASET

4.1. Evaluation set

Experiments are carried out on the datasets SimDATA and REALDATA from the Reverb Challenge [1]. The vocabulary size is 5k.

For the SimDATA set utterances by 28 different speakers are taken from the WSJCAM0 corpus [23] and convolved with three different room impulse responses (RIRs). Noise is added at a signal-to-noise ratio of 20 dB. The rooms are named Room1, Room2 and Room3. Additionally, there are two distances between the microphone and the speaker: 50 cm for the near condition and 200 cm for the far condition. For each condition there are 363 utterance and 6k words.

Table 1 shows two important acoustic parameters for the dataset. The first one is the $C_{50}$ parameter (Clarity index). It describes the ratio between the early signal energy (< 50 ms) and the rest and is thus related to the distance between the speaker and the microphone. The second parameter is the reverberation time $T_{60}$ (compare Sec. 2.1).

The REALDATA set consists of 372 utterances with 6.1k words in total. The utterances are from the MC-WSJ-AV corpus [24] and spoken by 10 different speakers. These are a subset of WSJCAM0 utterances rerecorded with real speakers in a noisy and reverberant room. The set is also divided into a far and a near set but the distances are $\sim 100$ cm respectively $\sim 250$ cm this time.

4.2. Training set

The training set of the WSJCAM0 corpus is used. In the case of clean training, the 7861 utterances by 92 speakers are left untouched. For the multicondition training the utterances are convolved with RIRs from up to three different room sizes (small, medium and large) and background noise with a SNR of 20 dB is added. These RIRs are different from the ones used to generate SimDATA, but have comparable room reverberation times. Note that the RIRs change from one utterance to the other in the same order, so the size of the training set remains the same for all experiments and there is an equal number of utterances with different RIRs. The same acoustic parameters as for the evaluation set are shown in Table 1.

<table>
<thead>
<tr>
<th>Room</th>
<th>Distance</th>
<th>$C_{50}$</th>
<th>$T_{60}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Room1</td>
<td>near</td>
<td>29-31 dB</td>
<td>250 ms</td>
</tr>
<tr>
<td></td>
<td>far</td>
<td>21-22 dB</td>
<td></td>
</tr>
<tr>
<td>Room2</td>
<td>near</td>
<td>14-17 dB</td>
<td>500 ms</td>
</tr>
<tr>
<td></td>
<td>far</td>
<td>6-7 dB</td>
<td></td>
</tr>
<tr>
<td>Room3</td>
<td>near</td>
<td>14-16 dB</td>
<td>700 ms</td>
</tr>
<tr>
<td></td>
<td>far</td>
<td>6-7 dB</td>
<td></td>
</tr>
</tbody>
</table>

Table 1. Acoustic parameters for the different conditions.

5. RESULTS

Table 2 summarizes the results obtained with different training conditions on SimDATA and REALDATA. The respective training condition is denoted as a subscript of the model. DA+DNN_{LargeFar} means for example that only the RIR of the large room with a distant speaker is used. If no distance is denoted, both, near and far, have been included. Further, MC (multicondition) includes all conditions and Clean are the clean training utterances.

5.1. Baseline

We start by discussing the baseline results for each model in the first five rows of Table 2. The combination of the DA and the DNN backend outperforms the other setups in every single condition when trained with full multicondition data (DA+DNN_{MC}). It also slightly improves the results when trained on clean data only (DA+DNN_{Clean}) compared to DNN_{Clean}. Additionally, the severe performance degradation caused by a channel mismatch becomes obvious. The word error rate (WER) doubles for more reverberant conditions when the model is trained with clean data only. In this case, BFE significantly improves the performance, achieving results competitive to DNN_{MC} for far conditions and even outperforming it for the REALDATA set. Only for near conditions it breaks down for the reason described in sec. 2.1.

5.2. DA and DNN with channel mismatch

Next, we look at the results for a DNN and a DA+DNN for different channel mismatch conditions. This is the second block of Table 2. If the models have only seen reverberant speech (DA+DNN_{LargeFar}) the WER increases significantly for conditions with less reverberation. If they have only seen slightly reverberated speech (such as DA+DNN_{Small}), the performance decreases for highly reverberated speech. But reverberation is not the only important mismatch. Performance degradation due to a mismatch in the $C_{50}$ parameter is also noticeable. On the other hand, the results show that there is some tolerance for the case where the model saw reverberated speech but with a different reverberation time (DA+DNN_{MediumFar} vs. DA+DNN_{Large}). Additionally the DA is again able to increase the performance in all but one case (DA+DNN_{SmallNear}), mostly by a significant margin. It makes it possible to use only the data from the large room for training and achieve better results than the model

1 Although not listed in the results, we also considered MediumFar, MediumNear, SmallFar and SmallNear, but no new insights can be gained from these setups.
trained with all conditions (DA+DNN\textsubscript{Large} vs. DA+DNN\textsubscript{Small}). Note that there is no mismatch for BFE since it is a parametric model and can be adjusted to new conditions. All results must be compared to the baseline (BFE+DNN\textsubscript{Clean}).

### 5.3. Adaption with BFE

Finally, we show how a parametric method can be used to adapt the DA to unseen conditions, leading to a significant WER reduction. We conduct two different adaptions. One is an adaption to the \textit{SimData} (superscript \textsubscript{Sim}), the other is an adaption to the \textit{RealData} (superscript \textsubscript{Real}). The first one is carried out using all sets of the \textit{SimData}, while the later one uses all sets of the \textit{RealData}.

The last block of Table 2 shows the results for different adapted models. The first three rows are related to \textit{SimData} where one model has only seen slightly reverberated speech, one only highly reverberated far speech and one no reverberated speech at all. Retraining these models brings a significant gain compared to the results with the unadapted models. The biggest improvement can be seen for DA+DNN\textsubscript{Clean} (vs. DA+DNN\textsubscript{Sim Clean}), with a relative WER reduction of nearly 50%. Importantly, all adapted models deliver better results than the BFE baseline. Especially the ability to improve features for the \textit{near} condition remains untouched. This means that the DA does not just learn the same mapping the BFE performs but rather keeps a part of its original mapping while adjusting it to the new condition. Also, even though adapted with \textit{SimData} only, the gain is also visible for the \textit{RealData} set. This indicates that the DA still generalizes instead of only working on the adapted condition.

The last two rows show an adaptation with \textit{RealData}. Again, the performance increases significantly. Even more interesting, the adapted model DA+DNN\textsubscript{Real Clean} outperforms its unadapted equivalent DA+DNN\textsubscript{MC}. This shows, that the proposed adaptation is able to reduce a mismatch between simulated and real reverberant data.

### 6. CONCLUSIONS

The combination of a DA and a DNN delivers good performance when the conditions at decode time have been included in the training data. It can even compensate for a mismatch in reverberation time to a certain extend. But when the mismatch becomes too large, the WER grows quickly. BFE on the other hand can be adjusted to unseen conditions by parameter selection, though the performance is behind the one a DA can achieve in a matched condition. But while it is possible to avoid a mismatch caused by the acoustic properties of the room and the speaker position by just generating enough artificially reverberated training data using different RIRs, there is still a gap between real recordings and simulated data. The results show that BFE as a parametric method can produce cleaned-up feature vectors which serve as targets for the DA for an unsupervised adaptation to the decoding conditions, bridging that gap.

### 7. RELATION TO PRIOR WORK AND OUTLOOK

This work builds on the DA [3, 16, 18, 25] and BFE [8, 9] as feature enhancement techniques. It also uses recent advancements in the acoustic modeling for the backend [19, 22]. We propose to make use of the advantages of both feature enhancement techniques: The possibility to be adjusted to new conditions of a parametric model (BFE) and the modelling power of a neural network (AE) by using the BFE to adapt the DA. In future research, we will try other parametric methods as we suspect that this is not limited to BFE. Further, a combination of different parametric methods for the adaptation might bring an additional performance gain.
8. REFERENCES


