Three ways to adapt a CTS recognizer to unseen reverberated speech in BUT system for the ASpIRE challenge

Martin Karafiát, František Grézl, Lukáš Burget, Igor Szöke and Jan "Honza" Černocký

Brno University of Technology, Speech@FIT and IT4I Center of Excellence, Brno, Czech Republic

cernocky,karafiat,grezl,burget,szoke@fit.vutbr.cz

Abstract

This paper describes several strategies tested in BUT's submission to the IARPA ASpIRE challenge. The ASpIRE task was to develop an automatic speech recognition (ASR) system for wide-band noisy reverberant speech, while only clean CTS (Fisher) data was allowed for ASR training. To solve this task, we have started with augmenting Fisher data with artificially noised and reverberated versions. The most obvious adaptation was (1) to re-train the whole GMM/HMM-based ASR system. Then, two techniques were designed and tested to make the adaptation easier and overcome retraining the whole ASR on large amount of speech: (2) we trained a speech enhancement DNN (also called de-noising auto-encoder), and (3) we adapted the feature extraction based on stacked bottle-neck networks (SBN). While re-training the whole system works the best, only slightly inferior results were obtained with the auto-encoder denoising followed by retraining of the first layers of the SBN hierarchy, letting most of the ASR system trained on clean Fisher unchanged. This shows a promising, efficient and fast way to port ASR systems to new conditions.

Index Terms: speech recognition, reverberation, de-reverberation, neural networks, DNN

1. Introduction

The IARPA Automatic Speech recognition In Reverberant Environments (ASpIRE) challenge[1] is looking for automatic speech recognition (ASR) solutions working on data from reverberant environments and severe mismatch conditions. Unlike other evaluations, where using data from the target or close-to-target domain is authorized, ASpIRE limits the speech training data to Fisher 1 and 2 telephone corpora only. Their artificial modifications are however allowed. The use of other non-speech data (for example real noises, room impulse responses, etc.) is allowed as well. We tackle the single microphone condition of ASpIRE.

The usual ASR system built at BUT is based on a Tandem structure with features generated by a hierarchy of two NNs, called Stacked bottle-neck (SBN) hierarchy (see Fig. 1), and GMM/HMM recognizer. This was also the architecture of choice for the ASpIRE.

The data from an *unseen* reverberated condition was the first challenge. As only Fisher was authorized, we have investigated ways to synthesize (augment) Fisher data in a way that would be close to the target reverberated speech. Using data augmentation has already been successfully used in speech enhancement [1, 2]. It also led to good results in our system for 2014 IARPA Babel evaluation although only artificial noising (no reverberation) was employed [3].

Once we have the augmented data, the issue is how to adapt the recognition system. The most straightforward way is to re-train the whole system (feature extraction and GMM/HMM). However, we were seeking more elegant ways that would allow us to let most of the recognition system intact. Classical techniques such as speech enhancement (including de-convolution that tries to reconstruct clean speech by inverse filtering the reverberant speech) and spectral enhancement were discarded as we wanted to exploit (1) NN-based signal enhancement and (2) the SBN feature extraction which is the very first block of our system. Also, the traditional approaches often cause a degradation if the system is used on clean data.

The enhancement of reverberant speech using NNs was already exploited: for example, in [4], a classical approach of removing the room impulse response is proposed, but the filter is estimated using a NN. NNs have also been used for speech separation [5] instead of popular computational auditory scene analysis (CASA) techniques. NN-based auto-encoder for speech enhancement was proposed in [1] with optimization in [2] and finally, reverberant speech recognition with signal enhancement by a deep auto-encoder was tested in the Chime Challenge and presented in [6].

Working on the feature extraction part of our ASR system was inspired by our previous work on the adaptation of the SBN hierarchy for multi-lingual and semi-supervised training [7]. Here, we have shown that the SBN hierarchy can be considered as a 2-stage system, where the first NN is responsible for low-level feature extraction and the second one "fits" the features to the following discriminatively trained GMM/HMM model. Therefore, we considered interesting to retrain only the first part of the feature extractor, while letting the rest of the ASR system intact.

2. Structure of ASR system

2.1. Initial PLP system

Our speech recognition system is HMM based on cross-word tied-states triphones, it is trained from scratch using standard maximum likelihood training. Final word transcriptions are decoded using 3-gram Language Model (LM) trained only on the transcriptions of training data.

For the initial system, Mel-PLP features are generated in classical way, the resulting number of coefficients is 13. Deltas, double- and triple-deltas are added, so that the feature vector

This work was supported by European Union Horizon 2020 research and innovation programme under grant agreement No. 645323 "BISON", Technology Agency of the Czech Republic project No. TA04011311 "MINT" and by IT4Innovations Centre of Excellence CZ.1.05/1.1.00/02.0070.

[1] https://www.innocentive.com/ar/challenge/9933624

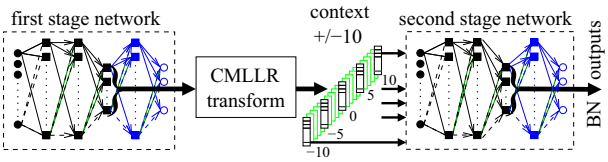

Figure 1: Scheme of Stacked Bottle-Neck Neural Network feature extraction.

has 52 dimensions. Cepstral mean and variance normalization is applied with the means and variances estimated per conversation side. HLDA is estimated with Gaussian components as classes to reduce the dimensionality to 39. The resulting features will be referred to as PLP-HLDA.

2.2. SBN feature extraction

The NN input features are 24 log Mel filter bank outputs concatenated with different fundamental frequency features: "BUT F0" has 2 coefficients (F0 and probability of voicing), "snack F0" is just a single F0 and "Kaldi F0" are 3 coefficients (Normalized F0 across sliding window, probability of voicing and delta). Fundamental frequency variation (FFV) is a 7 dimensional vector. Therefore, the whole feature vector has 37 coefficients. More details on the fundamental frequency features can be found in [8].

Conversation-side based mean subtraction is applied and 11 frames are stacked. Hamming window followed by DCT consisting of 0th to 5th base are applied on the time trajectory of each parameter resulting in 37×6=222 coefficients at the first-stage NN input (see Fig. 1).

The first-stage NN has five hidden layers with 1500 units each except the BN layer. The bottle-neck (BN) layer is the fourth hidden layer and its size is 80 neurons. Its outputs are stacked over 21 frames (+/-10) and down-sampled (every 5th is taken) before entering the second-stage NN. This NN has the same structure and sizes of hidden layers as the first-stage NN except for the BN layer with 30 neurons. The neurons in both BN layers have linear activation functions as they were reported to provide better performance [9]. The NN targets are monophone states obtained by forced alignment of training data with the initial PLP system. The final SBN features are outputs from the second-stage BN layer transformed by Maximum Likelihood Linear Transform (MLLT). HMM states are considered as the classes for the MLLT estimation.

2.3. GMM-HMM SBN System

The final system is based on a feature level fusion of two feature streams: PLP-HLDA (39 dimensions) and SBN features (30 dim.) are concatenated, which results in 69 dimensional feature stream (PLP-SBN). Then, new models are trained by single-pass retraining from PLP basic system. 12 Gaussian components per state were found to be sufficient for PLP-SBN features. The models and features serve as a starting point for training Region Dependent Transforms (RDT) [10]. In RDT framework, an ensemble of linear transformations is trained with the discriminative Minimum Phone Error (MPE) criterion. Each transformation corresponds to one region in feature space partitioned by a GMM. According to our previous experiments [11], GMM with 125 components was chosen. Finally, GMM-HMM system is trained using MPE [12] on top the RDT features.

The resulting system is only the first-pass one, which is used to obtain transcriptions for speaker adaptation. For the speaker adaptation, CMLLR transform is applied to the first-stage BN output as depicted in Fig. 1. Consequently, the second-stage NN is re-trained in Speaker Adaptive Training (SAT) fashion [13].

The adapted SBN features are used to construct new concatenated PLP-SBN features, which are further speaker adapted by another CMLLR transform. On top of the resulting speaker-adapted features, RDT and the GMM-HMM system are also re-trained in SAT fashion. More detailed description of this speaker adaptation strategy can be found in [8].

Table 1: *Data amounts.*

Data-set	No. of conversation sides	Size [h]
Fisher 1+2	23398	1800
Aspire Dev	30	3.4

3. The data and its augmenting

Fisher English database Part 1 and 2 was used for training as no other data was allowed. It contains over 20 thousand of telephone conversational sides. The test data was defined by the ASpIRE challenge. The ASpIRE dev set contains 30 recordings from various rooms and noisy conditions recorded with 16 kHz sampling rate (contrary to Fisher which is 8 kHz telephone data). Therefore, this data was first down-sampled to 8 kHz. For sizes of the data-sets, see Table 1.

3.1. Noising

Our training data was processed by artificially adding the following types of noises:

- real fan stationary noises: 115 samples (4 minutes long) were taken from the Freesound library[2]. These samples belong to categories: fan, AC, hvac, street, ventilation. Their character is stationary (sound of fan or AC).
- real background stationary noises: 170 samples (4 minutes long) from Freesound. These samples belong to categories: city, fan, AC, restaurant, shop, crowd, library, office, workshop. Their character is mainly stationary, with some minor portion of transient noises and babbling.
- real background transient noises: 60 samples (4 minutes long) from Freesound. These samples belong to categories: dishes, motor, workshop, doors, city, keyboard, library, office. The character is mainly transient, with some minor portion of stationary noises.
- babbling noises: 25 samples (4 minutes long), each created by merging speech from 100 random speakers from Fisher database using speech activity detector.
- ASpIRE noises[3]: 140 samples (10-60 second long) of noises extracted from ASpIRE dev data using speech activity detector. This was conforming to the evaluation rules.
- Artificial noises: 7 samples (4 minutes long) of artificial generated noises: various spectral modifications of white noise + 50 and 100 Hz hum.

[2] `http://www.freesound.org`

[3] we thank our colleagues from BBN for sharing them

3.2. Reverberation

We generated artificial room impulse responses (IR) using "Room Impulse Response Generator" tool from E. Habets[4]. The tool can model the size of room (3 dimensions), reflectivity of each wall, type of microphone, position of source and microphone, orientation of microphone toward the audio source, and number of bounces (reflections) of the signal. Each our room model consists of a pair of IR. One is used to reverberate (convolution with IR) the speech signal and the other is used to reverberate the noise signal that are then mixed into a single recording. Just coordinates of audio sources (speech/noise) differ for each of the IRs in such pair. We randomly set all parameters of the room for each room model.

3.3. Composition of the training set

We used `fant` tool [14] to mix reverberated speech and reverberated noise with given SNR. Speech signal was compensated for the delay caused by the reverberation (to match the timing with the original one). The following training datasets with artificially corrupted speech were created:

Large rooms dataset consists of 1800 hours of clean Fisher data augmented with another 3 copies of artificially corrupted Fisher data. IRs were generated for rooms where each dimension was limited to the range of 2–22 meters. Noises were added at SNRs ranging from 0dB to 45dB. The noise types used are: real fan stationary noises, real background stationary noises, babbling noises, and artificial noises.

Small rooms is a dataset similar to *large rooms* with the addition of real background transient noises and ASpIRE noises. After listening to the data from *large rooms* dataset and comparing it with the relatively less reverberant ASpIRE dev data, we also decided to limit the room dimensions to the range of 2–5 meters.

Auto-encoder training dataset is similar to *small rooms*. Two noises were always added into one recording: one random stationary noise and one random transient noise.

Enhanced dataset uses room dimensions 2–5 meters and real fan stationary noises, real background stationary noises, babbling noises, artificial noises added to speech at SNRs ranging from 15–45dB.This data is further enhanced (cleaned) by the auto-encoder described in section 4.1. This training was created in order to learn the artifacts, which can be introduced into the speech signal during the speech enhancement.

4. Strategies to cope with reverberation

4.1. Audio enhancement by DNN auto-encoder

The role of the auto-encoder is to enhance (de-noise and de-reverberate) the speech signal. It is trained on the artificially created parallel clean-noisy Fisher corpora as described in the previous section. The input of the NN is 129 dimensional vectors of log spectra stacked over 31 frames (e.g. 3999 dimensional vector). The desired output is 129 dimensional vectors (again log spectrum) corresponding to the clean version of the central input frame. A standard feed-forward architecture is used: 3999 inputs, 3 hidden layers with 1500 neurons, 129 outputs, tanh nonlinearities in the hidden layers. The NN is initialized in such a way that it (approximately) passes its input to the output and it is trained using conventional stochastic gradient descent to minimize the MSE objective.

We have experimented with different strategies of normalizing NN input and output. To achieve a good performance, utterance level mean and variance normalization is applied to both the NN input and the desired NN output. To synthesize the cleaned-up speech log spectrum, the NN output is denormalized based on the global mean and variance of clean speech. To enhance the ASpIRE data, a simulator of the telephone channel is used before the data enters the NN.

4.2. Re-training the SBN feature extractor

The second tested approach that allows for letting most of the ASR system intact (trained on clean data) is the adaptation of the SBN hierarchy in the feature extractor. Here, we have concentrated only on the parameters of the first-stage NN, to avoid the computationally expensive re-training of CMLLR, the second-stage NN and the GMM/HMM system. To be able to do that, the BN-to-output part of the previously fully trained NN was fixed and the input-to-BN part is adapted (fine-tuned) to the target data.

5. Experiments

We started experimenting with the full system including MPE training. To save the training time for discriminative training, a subset of 800 h of speech was defined for RDT and discriminative GMM acoustic model training. The results are summarized in Table 2. The first column shows on which data the first-stage NN in the feature extractor was trained. "Clean" means that it was kept from the original system trained on clean Fisher. The second column details on which data the GMM/HMM system was trained. Again, "Clean" denotes using parameters obtained on clean Fisher. The third column says whether the test recordings (or more precisely their log-spectra) are enhanced using the auto-encoder before ASR decoding. The fourth column shows the standard word error rate (WER) metric on ASpIRE dev data.

The first two lines present the worst and the best results: a baseline system trained only on clean Fisher without any modification reached WER of 42.3%. The next line is the best obtainable result, where the feature extractor and the whole GMM/HMM system was fully re-trained on full set of augmented data. No signal cleaning is used here. The improvement is almost 16% relative.

The following line shows the effect of signal cleaning using the auto-encoder. With this "auto-encoder only" setup and the original ASR system trained only on clean data, we have reached 37.9%, which is a nice 10% relative improvement over the baseline.

Next, re-training of the first-stage NN on augmented data was tested. We found that in this re-training, it is advantageous to preserve also the clean data in the NN training set. The resulting WER of 38.2% overcomes the baseline but does not reach the auto-encoder performance.

Finally, both signal cleaning and feature extractor re-training were combined (see the last line of Table 2). Log-spectrum processing by the auto-encoder described above, followed by re-training of the first-stage NN, brought us a nice 36.7% WER (13% relative improvement). The performance of the fully retrained system was not reached, but most of the ASR system is unchanged and we could avoid the time-consuming RDT and MPE GMM/HMM re-training.

Last, we have investigated into the effect of MPE training

[4] `http://www.audiolabs-erlangen.de/content/05-fau/professor/00-habets/05-software/01-rir-generator/rir_generator.pdf`

Table 2: *Final RDT MPE CMLLR system results.*

NN	GMM System	Speech enhancement	WER[%]
Clean	Clean	no	42.3
Large rooms	Large rooms	no	**35.6**
Clean	Clean	yes	37.9
Small rooms	Clean	no	38.2
Enhanced	Clean	yes	**36.7**

Table 3: *Effect of MPE.*

NN	GMM System	MPE	Speech enhancement	WER[%]
Large rooms	Large rooms	yes	no	35.6
Large rooms	Large rooms	no	no	36.9
Clean	Clean	yes	yes	37.9
Clean	Clean	no	yes	38.7

of the GMM/HMM system, see Table 3. The MPE lines correspond to Table 2. In the case of full system re-trained (first two lines), we see 3.5% relative improvement, which is the usual gain obtained from MPE with systems based on PLP-SBN features. For the "clean" system with only enhanced audio, MPE system does not bring so much (only 2% relative improvement) as the discriminatively trained system is more sensitive to channel mismatch, despite the auto-encoder's efforts to fix it.

6. Conclusions

We have presented our work towards the ASR of wide-band noisy reverberant speech in ASpIRE challenge. To solve this task, we have started with augmenting Fisher data with artificially noised and reverberated versions. The most obvious, best performing, but also the most costly approach was to re-train the whole GMM/HMM-based ASR system. Then, two techniques were designed and tested to make the adaptation easier and to overcome the retraining of the whole ASR on a large amount of speech: we trained a speech enhancement DNN (also called de-noising auto-encoder), and we adapted the feature extraction based on a stacked bottle-neck hierarchy. With the combination of both approaches, almost as good results as with full retraining were obtained. The nice feature of this approach is that most of the ASR system trained on clean Fisher is unchanged. This shows a promising, efficient and fast way to port ASR systems to new conditions.

In future, we will concentrate on a systematic study of individual contributing factors and application of this approach to different scenarios of the TAČR MINT project targeting meeting speech.

7. References

[1] Y. Xu, J. Du, L.-R. Dai, and C.-H. Lee, "An experimental study on speech enhancement based on deep neural networks," *IEEE Signal processing letters*, vol. 21, no. 1, Jan. 2014.

[2] ——, "Global variance equalization for improving deep neural network based speech enhancement," in *Proc. IEEE China Summit & International Conference on Signal and Information Processing (ChinaSIP)*, 2014, pp. 71 – 75.

[3] M. Karafiat, K. Vesely, I. Szoke, L. Burget, F. Grezl, M. Hannemann, and J. Cernocky, "BUT ASR system for BABEL surprise evaluation 2014," in *Proceedings of 2014 Spoken Language Technology Workshop*, South Lake Tahoe, Nevada, 2014, pp. 501–506.

[4] B. Dufera and T. Shimamura, "Reverberated speech enhancement using neural networks," in *Proc. International Symposium on Intelligent Signal Processing and Communication Systems, ISPACS 2009.*, Jan 2009, pp. 441–444.

[5] T. Yanhui, D. Jun, X. Yong, D. Lirong, and L. Chin-Hui, "Deep neural network based speech separation for robust speech recognition," in *Proceedings of ICSP2014*, 2014, pp. 532–536.

[6] M. Mimura, S. Sakai, and T. Kawahara, "Reverberant speech recognition combining deep neural networks and deep autoencoders," in *Proc. Reverb Challenge Workshop*, Florence, Italy, 2014.

[7] F. Grezl and M. Karafiat, "Combination of multilingual and semi-supervised training for under-resourced languages," in *Proceedings of Interspeech 2014*, Singapore, 2014, pp. 820–824.

[8] M. Karafiát, F. Grézl, M. Hannemann, K. Veselý, I. Szoke, and J. H. Černocký, "BUT 2014 Babel system: Analysis of adaptation in NN based systems," in *Proceedings of Interspeech 2014*. Singapore: IEEE, September 2014.

[9] K. Veselý, M. Karafiát, and F. Grézl, "Convolutive bottleneck network features for LVCSR," in *Proceedings of ASRU 2011*, 2011, pp. 42–47.

[10] B. Zhang, S. Matsoukas, and R. Schwartz, "Recent progress on the discriminative region-dependent transform for speech feature extraction," in *Proc. of Interspeech 2006*, Pittsburgh, PA, USA, Sep 2006, pp. 2977–2980.

[11] M. Karafiát, F. Grézl, M. Hannemann, K. Veselý, and J. H. Černocký, "BUT BABEL System for Spontaneous Cantonese," in *Proceedings of Interspeech 2013*, 2013, pp. 2589–2593. [Online]. Available: http://www.fit.vutbr.cz/research/view_pub.php?id=10423

[12] D. Povey, "Discriminative training for large vocabulary speech recognition," Ph.D. dissertation, University of Cambridge, 2003.

[13] M. Karafiát, F. Grézl, M. Hannemann, and J. H. Černocký, "BUT neural network features for spontaneous Vietnamese in BABEL," in *Proceedings of ICASSP 2014*. Florence, Italy: IEEE, May 2014.

[14] H. Hirsch and H. Finster, "The simulation of realistic acoustic input scenarios for speech recognition systems," in *Proc. Interspeech 2005*, Lisabon, Portugal, 2005.