Chapter 10
Training Data Augmentation and Data Selection

Martin Karafiát, Karel Veselý, Kateřina Žmolíková, Marc Delcroix, Shinji Watanabe, Lukáš Burget, Jan "Honza" Černocký, and Igor Szőke

Abstract Data augmentation is a simple and efficient technique to improve the robustness of a speech recognizer when deployed in mismatched training-test conditions. Our work, conducted during the JSALT 2015 workshop, aimed at the development of: (1) Data augmentation strategies including noising and reverberation. They were tested in combination with two approaches to signal enhancement: a carefully engineered WPE dereverberation and a learned DNN-based denoising autoencoder. (2) Proposing a novel technique for extracting an informative vector from a Sequence Summarizing Neural Network (SSNN). Similarly to i-vector extractor, the SSNN produces a "summary vector", representing an acoustic summary of an utterance. Such vector can be used directly for adaptation, but the main usage matching the aim of this chapter is for selection of augmented training data. All techniques were tested on the AMI training set and CHiME3 test set.

10.1 Introduction

Training (or "source") versus evaluation ("target") data match or mismatch is a well-known problem in statistical machine learning. It was shown [1] that an automatic speech recognizer trained on clean data performs accurately on clean test data but poorly on noisy evaluation data. But it holds also vice versa—a recognizer trained on noisy data performs accurately on noisy but poorly on clean evaluation data.

M. Karafiát (✉) • K. Veselý • K. Žmolíková • L. Burget • J. "Honza" Černocký • I. Szőke
Brno University of Technology, Speech@FIT and IT4I Center of Excellence, Brno, Czech Republic
e-mail: karafiat@fit.vutbr.cz; iveselyk@fit.vutbr.cz; izmolikova@fit.vutbr.cz; burget@fit.vutbr.cz; cernocky@fit.vutbr.cz; szoke@fit.vutbr.cz

M. Delcroix
NTT Corporation, 2-4, Hikaridai, Seika-cho, Kyoto, Japan

S. Watanabe
Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA, USA

S. Watanabe et al. (eds.), *New Era for Robust Speech Recognition*,
DOI 10.1007/978-3-319-64680-0_10

Unfortunately, "clean" and "noisy" are only very broad categories; the source vs. target data mismatch can have many forms and can depend on the speakers, acoustic conditions, and many other factors.

Typical solutions for dealing with a data mismatch include *speech enhancement*, where we modify the (possibly noisy) target data to fit a system trained on clean source data [13, 33], and *model adaptation*, trying to adjust/adapt a model to deal with the mismatch condition [10, 15, 30]. A third technique, investigated in this chapter, is *data augmentation*. Here, we are trying to change the source data in order to obtain data with similar characteristics to the target data. Such generated data (generally a greater amount than the original one, as we are trying to cover a wide variety of target conditions) is called "augmented."

Note that two terms are generally used in the literature: *data augmentation* usually means filling sparse target data by borrowing from rich source data, while *data perturbation* means adding more variation into the target data by using the target data only. In our approach to data augmentation, we usually combine both, i.e., we borrow from source data *and* we modify it to fit the target data characteristics.

10.1.1 Data Augmentation in the Literature

One of the first attempts to augment source data to fit the target data was made by Bellegarda et al. [4]. In experiments conducted on the Wall Street Journal corpus, the goal was to populate the target speaker feature space with transformed data from source speakers. A feature rotation ("metamorphing") algorithm [3] was used: both source and target speaker features are first transformed onto a unit sphere by phoneme-dependent normalization, then a transformation is estimated to map the source features to target ones. For a target speaker, the closest source speakers can be found using a distance metric among phoneme clusters on the sphere. When these source speakers are found, one can remap their features to the target speaker space, thus increasing the data for speaker-dependent model training. The conclusion was that 100 target speaker sentences augmented by 1500 source speakers' sentences led to a speaker-dependent model with the same accuracy as a model trained on 600 target speaker sentences.

More recent approaches can be split into categories depending on the level on which the data augmentation is done:

On the *audio level*, the goal is to perturb the audio to minimize source/target data mismatch. The original voice is not modified in the sense of generating an unseen speaker; the augmentation is done to neglect different acoustic environments by using artificial reverberation, noising, or some other perturbation of the source data. In this scenario, we typically have enough source speakers, but a nonmatching acoustic environment (for example a quiet place versus a crowded public place). The usual procedure includes adding artificial [17] or real [18] noise. Reverberation can also be simulated using real or artificial room impulse responses [18], or both methods can be combined.

The audio itself can also be modified. The approaches covered in [20] include upsampling and downsampling the audio with an appropriate change of reference labels timing, or changing the pitch or tempo of the raw audio using an appropriate audio editor. The results of [20] show that reverberation/noising and resampling/pitch modifications are not completely complementary.

Finally, new speech data can be artificially generated using either statistical or concatenative text-to-speech (TTS) synthesis. This approach may not generate a large variety of new speakers but can still generate unseen sentences to augment training data for less frequent phoneme contexts. The latter technique, where speaker and prosody parameters are generated using hidden Markov models [27], is used more often. For use with automatic speech recognition (ASR) training, one can skip audio generation and use statistical speech synthesis to generate directly perceptual linear prediction (PLP) or mel-frequency cepstral coefficient (MFCC) features [22, 34].

On the *feature level*, the augmentation is done on the level of extracted features: low-level (Mel-banks, PLPs) or high-level (neural network bottle-neck features). A typical example is vocal tract length perturbation (VTLP) [14], modifying the speaker's vocal tract shape by spectral shifts. This is an "inverse" technique to the commonly used vocal tract length normalization (VTLN), where the goal is to normalize different speakers to a generic one. In VTLP, new data is generated by altering the true VTLN factor; for example, the authors of [14] generated five versions of data and tested them on TIMIT phoneme recognition. A nice improvement was observed; moreover, they found a positive effect of perturbing the test data with several warp factors and then averaging the acoustic-model (deep-neural-network, DNN) scores before the decoding.

Stochastic feature mapping (SFM) [5] falls under feature-level data augmentation too. Here, feature transformations are used to create artificial speakers. This approach is partly complementary to VTLP but its main advantage is that features can be easily generated on the fly.

10.1.2 Complementary Approaches

In a broader sense [23], data augmentation can be seen also from a different perspective: We can "fill" the sparsity of the actual training data with either untranscribed data from the target language, synthesized data from the target language, or other language data.

Untranscribed data can be used in the case where we have a few transcribed resources but a vast amount of untranscribed data (for example from Internet sources). In so-called self-training [39] (a variation of unsupervised or semisupervised training), a speech recognition system is first bootstrapped using the available little amount of transcribed data, then used to label the untranscribed data. A confidence measure is used to select the reliably transcribed segments, which are then added to the training set, and the system is retrained. The process can be done iteratively.

Data from *other languages* can be used, which has the great advantage of using real, not artificially prepared, data. However, it is more complicated because of language mismatch, which can be partially overcome by using universal phone sets, phone-to-phone mappings, or hidden-layer unit-to-target mappings in multiple-layer perceptrons [9]. Note, however that multilingual training is still a very active research topic, which is being investigated in several projects, such as the U.S. IARPA-sponsored Babel.[1]

The experimental part of this chapter concentrates only on the signal-processing approaches described in Sect. 10.1.1.

10.2 Data Augmentation in Mismatched Environments

10.2.1 Data Generation

In this section, we describe the noising and reverberation techniques used and the strategies to construct the training dataset.

- *Noise*. Our training data was processed by artificially adding two types of noises *real background noises* were downloaded from various sources, for example Freesound,[2] and "babbling noises" were created by merging speech from random speakers.
- *Reverberation.* We generated artificial room impulse responses (RIRs) using a "room impulse response generator" tool from E. Habets.[3] The tool can model the size of the room (three dimensions), the reflectivity of each wall, the type of microphone, the position of the source and the microphone, the orientation of microphone toward the audio source, and the number of bounces (reflections) of the signal. In each room, we created a pair of RIRs: one was used to reverberate (by convolution with the RIR) the speech signal and the other was used to reverberate the noise. Both signals were then mixed into a single recording. Each pair of RIRs differed only by the coordinates of the audio sources (speech/noise). We randomly set all parameters of the room for each room model.
- *Data augmentation strategies.* For each original clean utterance (independent headset microphones - IHM) from the AMI corpus, a corrupted version of the utterance was created by randomly choosing one of the following four speech corruption methods:

[1] https://www.iarpa.gov/index.php/research-programs/babel.

[2] www.freesound.org.

[3] https://github.com/ehabets/RIR-Generator.

1. One of the *real background noises* (see above) was randomly selected and added to the speech signal at a signal-noise-ratio (SNR) randomly selected from values: of −5, 0, 5, 10, and 15 dB.
2. One of the RIRs was randomly selected and used to reverberate the speech signal. In this case, no noise was added. Note that when adding reverberation, we compensated for the incurred delay to match the timing with the original signal.
3. The third option is a combination of the previous two. A random stationary noise and random reverberation were added. Speech and noise were reverberated by two different RIRs as described above. The two signals were then mixed at a randomly selected SNR level from the same range as before.
4. The same as the previous option, but babbling noise was used instead of stationary.

10.2.2 Speech Enhancement

In addition to the above data augmentation, we investigated two front-end approaches to handle source and target data mismatch caused by noise and reverberation. Here, we compare two main approaches: denoising by a neural-network(NN)-based autoencoder and signal-processing enhancement using the weighted prediction error (WPE).

10.2.2.1 WPE-Based Dereverberation

Reverberation is responsible for nonstationary distortions that are correlated with the speech signal and, consequently, it cannot be suppressed using the conventional noise reduction approaches. Therefore, we used the WPE dereverberation method [35, 36], which was shown to greatly improve ASR in reverberant conditions for several tasks [7, 37]. WPE is discussed in detail in Chap. 2.

WPE is based on long-term linear prediction (LP), but introduces modifications to conventional LP to make it effective for dereverberation. It is well known that multichannel LP can be used for channel equalization [11]; however, using conventional LP for speech signals causes excessive degradation because LP equalizes not only the room impulse responses but also the (useful) speech production process. To address this issue, WPE modifies the conventional LP algorithm in two ways: by modeling speech with a short-term Gaussian distribution with a time-varying variance [21], and by introducing a short time delay in the LP filters that prevents the equalization of the speech production [19].

WPE has a number of characteristics that make it particularly suitable for distant speech recognition: it is based on linear filtering, which ensures a low level of distortion in the processed speech. WPE can be formulated for single-channel or multichannel cases. It has also been shown to be relatively robust to ambient noise.

Note that the WPE algorithm does not require a pretrained model of speech and operates in a per-utterance manner.

More details about this technique can be found in Chap. 2.

10.2.2.2 Denoising Autoencoder

An artificial neural network was also employed as a denoising autoencoder to enhance (denoise and dereverberate) the speech signal. It was trained on the artificially created parallel clean–noisy AMI corpora. The reverberated and noised data described above (option three) was used for this purpose.

The input of the NN consisted of 257-dimensional vectors of log spectra stacked over 31 frames (i.e., a 7967-dimensional vector). The desired output was a 257-dimensional vector (again a log spectrum) corresponding to the clean version of the central input frame. A standard feedforward architecture was used: 7967 inputs, 3 hidden layers each with 1500 neurons, 257 outputs, and tanh nonlinearities in the hidden layers. The NN was initialized in such a way that it (approximately) replicated its input to the output and it was trained using the conventional stochastic gradient descent to minimize the mean squared error (MSE) objective.

We have experimented with different strategies for normalizing the NN input and output. To achieve a good performance, utterance-level mean and variance normalization was applied to both the NN input and the desired NN output. To synthesize the cleaned-up speech log spectrum, the NN output was denormalized based on the global mean and variance of clean speech.

10.2.3 Results with Speech Enhancement on Test Data

In the following experiments, we compare the effectiveness of the two speech enhancement techniques described in the previous sections: the denoising autoencoder and WPE [7] dereverberation. Table 10.1 shows results on the noisy CHiME-3 data obtained with the baseline DNN-based ASR system trained on clean independent-headset-microphone (IHM) AMI data. The table compares results obtained on the original unprocessed noisy test data and those obtained on data enhanced with the two techniques. The results are also compared for systems trained

Table 10.1 Performance of speech enhancement techniques on CHiME-3

Test data enhancement	XE (%WER)	sMBR (%WER)
None	48.86	46.99
WPE	45.36	43.63
Autoencoder	**30.58**	**30.59**

WER word error rate

The bold numbers indicate the best values in the table that help the orientation

Table 10.2 Results on CHiME-3 with different data augmentation variants

DNN training data		Test data	XE	sMBR
Noise type	Reverb	enhancement	(%WER)	(%WER)
None	None	None	48.86	46.99
Babble	Artificial	None	26.41	–
Stationary	Artificial	None	25.8	–
Stationary	None	None	24.26	20.47
Stationary	None	WPE	**22.72**	**19.28**

The bold numbers indicate the best values in the table that help the orientation

using a frame-by-frame cross-entropy (XE) objective and systems further retrained using state-level minimum Bayes risk (sMBR) discriminative training [28].

A relatively small improvement was obtained with WPE. This can be easily explained, as WPE aims to only dereverberate the signals, and relatively low reverberation is present in the CHiME-3 data due to the small distance between the speaker and the microphones. This trend might, however, differ for other test data. On the contrary, the denoising autoencoder provides significant gains, as it was trained to reduce both noise and reverberation.

No improvement was obtained from sMBR training. A possible reason is that new types of errors caused by the presence of noise in the data were not seen during the training, where only the IHM data were used. Note that much larger relative gains from sMBR will be reported in the following sections, where the training data was augmented with artificially corrupted speech data (see, e.g. Table 10.2).

10.2.4 Results with Training Data Augmentation

Table 10.2 presents results obtained when adding different types of noise to the training data (babbling vs. stationary noise). We also tested whether it helps to add reverberation to the training data or if it is sufficient to corrupt the training data only with additive noise. Interestingly, the best results were obtained with no reverberation added to the training data, but with the test data enhanced using the WPE dereverberation technique. WPE enhancement brings almost 1.5% absolute improvement. This indicates that signal-level dereverberation is more effective than training the acoustic model on reverberated speech.

Table 10.2 also shows a nice improvement (over ~3% absolute) from retraining DNNs using sMBR sequence training. Comparing the results in Tables 10.1 and 10.2, we observe that better performance could be achieved by training a recognizer on artificially corrupted data than using a denoising autoencoder trained on the same data to reduce noise.

Table 10.3 shows the results obtained for the different training datasets on the REVERB dev set. As for the CHiME-3 experiment, we observe that adding noise to the training data greatly improved performance. Not surprisingly, in the case of the REVERB data, adding reverberation to the training data also significantly improved

Table 10.3 Results on REVERB dev set with different data augmentation variants (sMBR models)

DNN training data		Test data enhancement	dev	
Noise type	Reverb		Near (%WER)	Far (%WER)
None	None	None	92.48	90.89
Stationary	None	None	52.49	49.72
Stationary	Artificial	None	40.33	37.93
None	None	WPE	56.19	49.28
Stationary	None	WPE	19.75	22.52
Stationary	Artificial	WPE	**19.75**	**20.77**

The bold numbers indicate the best values in the table that help the orientation

performance. However, the improvement brought by the reverberant training data is significantly reduced when using a dereverberation front end.

On the CHiME-3 test, we found no effect from adding reverberation into the training data for acoustic-model training. On the contrary, the WPE dereverberation technique was found effective. We also showed that greater performance improvement could be achieved by retraining the acoustic model on artificially corrupted speech than using a denoising autoencoder trained on the same training data to remove noise. On the REVERB test, adding reverberation was found beneficial, but the gains become smaller when using the WPE dereverberation front end. On the other hand, the impact of adding noisy training data remains. In future, we would like to verify our findings on other databases.

Finally, sMBR was found effective when the system was trained on noised data, in comparison with a system based on an enhancement autoencoder, where no gain from discriminative training was observed.

Note that the effect of data augmentation has also been investigated for the REVERB task in [8], with the significant difference that they used (instead of the AMI corpus) the REVERB training data, which they augmented with noise and various SNRs. In addition, in CHiME-3 and AMI, data augmentation has also been performed by treating each microphone signal recording as an independent training sample. This simple approach was also found to improve performance [26, 38].

10.3 Data Selection

10.3.1 Introduction

Using the methods above, we can generate large amounts of augmented data, from which it is possible to choose the training set. Different approaches to data selection have been explored in the past. Many existing methods are based on inspecting the frequency of speech units to evaluate the benefit of adding an utterance to the training. In [32], the data selection strategy aimed to choose such a subset which has a uniform distribution over phonemes or words. Similarly, the selection method in [31] was guided by the term frequency and inverse document frequency (TF-IDF) of triphones. However, these methods are not very suitable for the case of data created by artificial noising as they do not explore the acoustic diversity.

A different approach was used in [2], where the aim was to select a subset of the data with the most similar acoustic characteristics to the target domain. To achieve this, the vector of the posterior probabilities of components in Gaussian mixture models was used to represent each utterance. The idea of using a fixed-length vector for characterization of utterances was also exploited in [25], where the selection was based on the distributions of i-vectors.

Here we investigate use of a fixed-length "summary vector" representing the acoustic conditions to select training utterances that are the most similar to the test conditions. Unlike the existing data selection approaches, the proposed summary vector extraction exploits a neural network framework. A special NN is used to compensate for the mismatch between clean and noisy conditions. This is realized by appending a compensation network to an NN trained on clean speech. The compensation network is trained to perform utterance-level bias compensation. Consequently, the output of the compensation network summarizes the information about the noise conditions of an entire utterance and can thus be used to select useful training data.

The extracted "summary vector" has the desired property of discriminating specific noise types, which can be proved by visualizing the vectors. Moreover, we have also confirmed experimentally that the proposed "summary vector" can be used to select training data and that it outperforms random training data selection and i-vector-based data selection.

10.3.2 *Sequence-Summarizing Neural Network*

We describe each utterance using a fixed-length vector summarizing the acoustic conditions of the utterance. In other words, a vector is extracted from each utterance that is in some sense similar to i-vectors known from speaker recognition [6]. However, instead of relying on a conventional i-vector extraction, we train a special neural network able to "summarize each utterance."

To extract summary vectors, we train a composite architecture combining two neural networks as sketched in Fig. 10.1 (a similar architecture was previously used for speaker adaptation [29]). It consists of the main NN and the sequence-summarizing NNs (SSNN), both sharing the same input features. To train this scheme for the extraction of summary vectors, we proceed as follows:

1. First, the main network (DNN(·), upper part of Fig. 10.1) is trained on clean data $\mathscr{X}^{\text{clean}}$ as a standard DNN classifier with triphone state targets $\mathscr{Y}$ and a cross-entropy criterion (XE[·]):

$$\hat{\Theta}^{\text{clean}} = \arg\max_{\Theta} \text{XE}\left[\mathscr{Y}, \text{DNN}\left(\mathscr{X}^{\text{clean}}; \Theta\right)\right]. \tag{10.1}$$

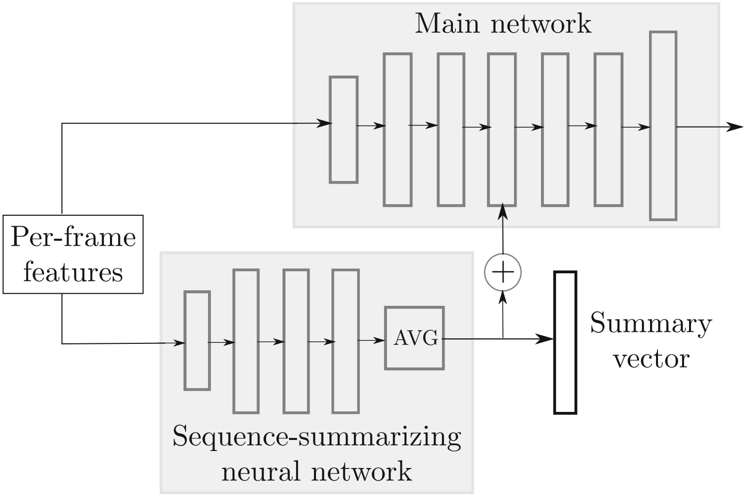

Fig. 10.1 Training of acoustic-condition estimator

The estimated parameters of the main network $\hat{\Theta}^{\mathrm{clean}}$ then stay fixed for the rest of the training.

2. The sequence-summarizing NN is added to the scheme (SSNN(·), lower part of Fig. 10.1), which also receives frame-by-frame speech features (the same as in the main network) as its input. The last layer of the SSNN involves averaging (global pooling over the frames) and produces one fixed-length summary vector for each utterance, which is then added to the hidden-layer activations of the main network. The whole architecture is now trained on noisy data $\mathscr{X}^{\mathrm{noisy}}$ with the same objective as used in the first step:

$$\arg\max_{\Phi} \mathrm{XE}\left[\mathscr{Y}, \mathrm{DNN}\left(\mathscr{X}^{\mathrm{noisy}}, \overline{\mathrm{SSNN}(\mathscr{X}^{\mathrm{noisy}}; \Phi)}; \hat{\Theta}^{\mathrm{clean}}\right)\right]. \tag{10.2}$$

Note that only the parameters of the sequence summarizing neural network Φ are trained at this point.

The idea of the training procedure is that the SSNN should learn to compensate for the mismatch caused by presenting noisy data $\mathscr{X}^{\mathrm{noisy}}$ to the main network, which was previously trained only on clean data $\mathscr{X}^{\mathrm{clean}}$. Thus, the summary vector extracted by the SSNN should contain important information about the acoustic conditions to characterize the noise component of an utterance. In the final application (data selection), we discard the main network and only use the SSNN to extract the summary vectors.

10.3.3 Configuration of the Neural Network

To find the optimal configuration of the summarizing neural network, we performed a set of experiments varying the hidden layer where the summary vector is added (connection layer), the size of this layer (thus the size of the extracted vector), and the amount of data used to train the network. Although the whole composite network as seen in Fig. 10.1 was not intended to be used for decoding in the final application, for these experiments, we used it directly to test the noisy data. This scenario, channel adaptation, allowed us to find the best configuration without the final time-consuming procedure—data selection and system rebuild.

First, the optimal connection layer was evaluated. In these experiments, the sizes of all hidden layers in both DMM and SNN were 2048 and the amount of noised training data was equal to the original clean set. The results on CHiME-3 are shown in Table 10.4. It seems that adding the summary vector to the second hidden layer is the most effective for adapting the clean DNN to the noisy data. Moreover, the results present over 8% absolute WER reduction by adding a summary-vector extractor to the clean DNN. It is a nice improvement, taking into account that the extractor performs just a simple per-utterance bias compensation in the hidden layer of the clean DNN classifier.

The second hidden layer was taken as the connection layer and the optimal size of the summary vector was evaluated. To be able to train summary-vector extractors of different sizes, we had to retrain the original DNN classifier with various sizes of the second hidden layer on the clean data. Table 10.5 shows the WER reduction as a function of dimensionality of the summary vector. It degrades with decreasing dimensionality; therefore we decided to keep its original dimensionality of 2048.

Finally, the effect of adding data for the training of the summarizing network was evaluated. We generated several random selections of noised data and trained

Table 10.4 Optimal connection layer for training of summary-vector extractor (CHiME-3)

Connection layer	XE (%WER)
None	47.72
1	39.89
2	**39.32**
3	40.09
4	41.08
5	40.70

The bold numbers indicate the best values in the table that help the orientation

Table 10.5 Dimensionality of summary-vector extractor (CHiME-3 %WER)

Size of second layer	256	512	1024	2048
Clean DNN	49.03	48.41	47.83	47.72
Joint NN	49.39	42.70	40.15	39.32
Abs. improvement	−0.36	5.71	7.68	**8.40**

The bold numbers indicate the best values in the table that help the orientation

Table 10.6 Data sizes for summary-vector extractor training (CHiME-3 %WER)

Summary-vector dimensionality	Data size		
	1×train	2×train	3×train
1024	40.15	38.99	37.30
2048	39.32	40.24	**37.15**

The bold numbers indicate the best values in the table that help the orientation

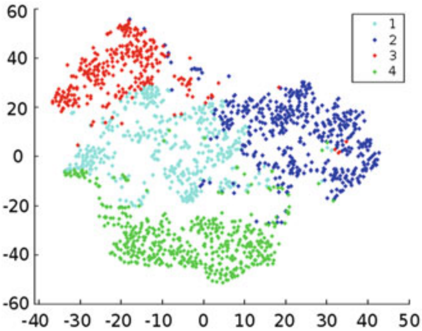

Fig. 10.2 t-SNE plots of summary vectors estimated from CHiME-3 data. The *colors* correspond to the clusters obtained by *k*-means (*left*) and the actual noise conditions (*right*)

the SSNN on them. Table 10.6 shows the positive effect of a sufficient amount of training data (3× original clean set).

10.3.4 Properties of the Extracted Vectors

To see whether the method generates vectors reflecting the noise conditions in the data, we extracted the vectors for CHiME-3 utterances and observed their properties. The CHiME-3 test set contains four different recording environments—bus (BUS), cafe (CAF), street (STR), and pedestrian area (PED). We performed clustering of the extracted vectors into four clusters using *k*-means and compared the obtained clusters to the real environments in the data. Figure 10.2 shows two plots created by t-SNE [12]—the right one shows the four real environments in the data and the left one the clusters created by *k*-means. Although the clusters were created by an unsupervised technique, there are clear similarities with the real ones.

It is also worth comparing the newly proposed summary vector with i-vectors [6] as i-vectors are also known to capture information about the channel. Note that i-vectors were also recently used for adapting DNNs in speech recognition tasks [16, 24]. Figure 10.3 shows i-vectors and summary vectors extracted from CHiME-3 projected onto the first two liner discriminant analysis (LDA) bases. The recording environment labels were used as the classes for LDA. It seems that the environments

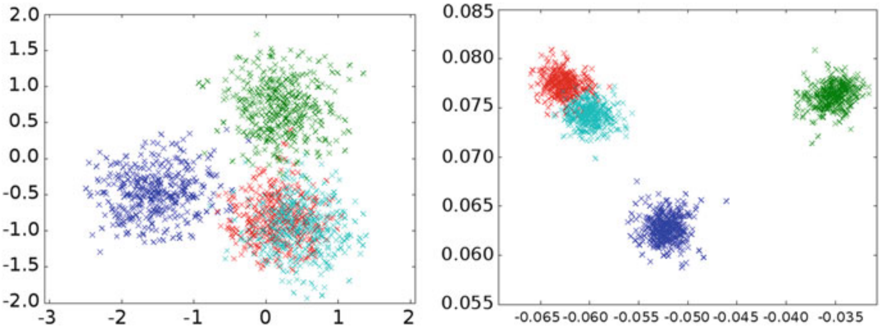

Fig. 10.3 Plot of the first and second LDA bases on CHiME-3 data for i-vectors (*left*) and summary vectors (*right*)

are better separated in the summary-vector space than in the i-vector space.[4] This shows that summary vectors contain information suitable for CHiME-3 recording environment clustering, even though the extractor was trained on different data (a corrupted AMI corpus).

10.3.5 Results with Data Selection

To perform data selection, we extract summary vectors for each generated training utterance and each test utterance. We select a subset of the generated training data by selecting the utterances that are the closest to the test set conditions. For this, we compute the mean over all summary vectors of the test set and measure its distance to the summary vector of each utterance of the training data. Only the closest utterances are kept in the training subset. By this, we aim to select that type of added noise which matches best the noise in the test data. The amount of selected data is equal to the size of the clean training set.

As few different types of noise are present in the test data, computing the mean of summary vectors from the whole test set may not be the best way to represent it. Therefore, further experiments were performed by clustering the summary vectors of the test set and computing the means of these clusters. The training utterances were then selected to have the shortest distance to the summary vector centroid of one of the clusters.

For measuring the distance between vectors, we used the cosine and Euclidean distances. Table 10.7 shows results obtained using these two measures and different numbers of clusters of test data. The results indicate that using the cosine distance is more effective. The best results are obtained using four clusters of test data, which corresponds to the fact that there are four real recording environments in the test set.

[4] Brno University of Technology open i-vector extractor (see http://voicebiometry.org) was used for these experiments.

Table 10.7 Comparison of different selection methods (CHiME-3 %WER)

Distance measure/# clusters	1	4	10
Cosine	25.09	24.72	24.98
Euclidean	26.75	26.55	26.59

Table 10.8 Comparison of random vs. automatic selection results (CHiME-3 %WER)

	Selection		
Dataset	Random	i-Vector	Summary vector
dev	25.8	25.61	**24.72**
eval	45.58	44.02	**43.23**

The bold numbers indicate the best values in the table that help the orientation

Table 10.8 shows the best result obtained with the summary-vector data selection compared to random data selection and selection using i-vectors. About 1% absolute improvement on the dev set and 2% on the eval set was obtained with the proposed data selection method compared to random data selection, showing the effectiveness of the proposed data selection method.

10.4 Conclusions

We have shown that, despite its simplicity, data augmentation is an effective technique to improve the robustness of a speech recognizer when deployed in mismatched training–test conditions. Noising of the data was found to be more effective than NN-based denoising strategies. We have also proposed a new promising approach for selecting data within the augmented set, based on a summarizing neural network that is able to generate one fixed-dimensional vector per utterance. On the CHiME-3 test set, we observed 1% absolute improvement over random data selection and the technique also compared favorably to data selection based on i-vectors.

Acknowledgements Besides the funding for the JSALT 2015 workshop, BUT researchers were supported by the Czech Ministry of Interior project no. VI20152020025, "DRAPAK," and by the Czech Ministry of Education, Youth, and Sports from the National Program of Sustainability (NPU II) project "IT4 Innovations Excellence in Science—LQ1602."

References

1. Ager, M., Cvetkovic, Z., Sollich, P., Bin, Y.: Towards robust phoneme classification: augmentation of PLP models with acoustic waveforms. In: 16th European Signal Processing Conference, 2008, pp. 1–5 (2008)
2. Beaufays, F., Vanhoucke, V., Strope, B.: Unsupervised discovery and training of maximally dissimilar cluster models. In: Proceedings of Interspeech (2010)

3. Bellegarda, J.R., de Souza, P.V., Nadas, A., Nahamoo, D., Picheny, M.A., Bahl, L.R.: The metamorphic algorithm: a speaker mapping approach to data augmentation. IEEE Trans. Speech Audio Process. **2**(3), 413–420 (1994). doi:10.1109/89.294355
4. Bellegarda, J., de Souza, P., Nahamoo, D., Padmanabhan, M., Picheny, M., Bahl, L.: Experiments using data augmentation for speaker adaptation. In: International Conference on Acoustics, Speech, and Signal Processing, 1995, ICASSP-95, vol. 1, pp. 692–695 (1995). doi:10.1109/ICASSP.1995.479788
5. Cui, X., Goel, V., Kingsbury, B.: Data augmentation for deep neural network acoustic modeling. IEEE/ACM Trans. Audio Speech Lang. Process. **23**(9), 1469–1477 (2015). doi:10.1109/TASLP.2015.2438544
6. Dehak, N., Kenny, P., Dehak, R., Dumouchel, P., Ouellet, P.: Front-end factor analysis for speaker verification. IEEE Trans. Audio Speech Lang. Process. **19**(4), 788–798 (2011). doi:10.1109/TASL.2010.2064307. http://dx.doi.org/10.1109/TASL.2010.2064307
7. Delcroix, M., Yoshioka, T., Ogawa, A., Kubo, Y., Fujimoto, M., Ito, N., Kinoshita, K., Espi, M., Hori, T., Nakatani, T., Nakamura, A.: Linear prediction-based dereverberation with advanced speech enhancement and recognition technologies for the REVERB challenge. In: Proceedings of REVERB'14 (2014)
8. Delcroix, M., Yoshioka, T., Ogawa, A., Kubo, Y., Fujimoto, M., Ito, N., Kinoshita, K., Espi, M., Araki, S., Hori, T., Nakatani, T.: Strategies for distant speech recognition in reverberant environments. EURASIP J. Adv. Signal Process. **2015**, Article ID 60, 15 pp. (2015)
9. Egorova, E., Veselý, K., Karafiát, M., Janda, M., Černocký, J.: Manual and semi-automatic approaches to building a multilingual phoneme set. In: Proceedings of ICASSP 2013, pp. 7324–7328. IEEE Signal Processing Society, Piscataway (2013). http://www.fit.vutbr.cz/research/view_pub.php?id=10323
10. Gales, M.J.F., College, C.: Model-Based Techniques for Noise Robust Speech Recognition. University of Cambridge, Cambridge (1995)
11. Haykin, S.: Adaptive Filter Theory, 3rd edn. Prentice-Hall, Upper Saddle River, NJ (1996)
12. Hinton, G., Bengio, Y.: Visualizing data using t-SNE. In: Cost-Sensitive Machine Learning for Information Retrieval 33 (2008)
13. Hu, Y., Loizou, P.C.: Subjective comparison of speech enhancement algorithms. In: Proceedings of IEEE International Conference on Speech and Signal Processing, pp. 153–156 (2006)
14. Jaitly, N., Hinton, G.E.: Vocal tract length perturbation (VTLP) improves speech recognition. In: Proceedings of the 30th International Conference on Machine Learning, Atlanta, GA (2013)
15. Kalinli, O., Seltzer, M.L., Acero, A.: Noise adaptive training using a vector Taylor series approach for noise robust automatic speech recognition. In: Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP'09, pp. 3825–3828. IEEE Computer Society, Washington (2009) doi:10.1109/ICASSP.2009.4960461. http://dx.doi.org/10.1109/ICASSP.2009.4960461
16. Karafiát, M., Burget, L., Matějka, P., Glembek, O., Černocký, J.: iVector-based discriminative adaptation for automatic speech recognition. In: Proceedings of ASRU 2011, pp. 152–157. IEEE Signal Processing Society, Piscataway (2011). http://www.fit.vutbr.cz/research/view_pub.php?id=9762
17. Karafiát, M., Veselý, K., Szőke, I., Burget, L., Grézl, F., Hannemann, M., Černocký, J.: BUT ASR system for BABEL surprise evaluation 2014. In: Proceedings of 2014 Spoken Language Technology Workshop, pp. 501–506. IEEE Signal Processing Society, Piscataway (2014). http://www.fit.vutbr.cz/research/view_pub.php?id=10799
18. Karafiát, M., Grézl, F., Burget, L., Szőke, I., Černocký, J.: Three ways to adapt a CTS recognizer to unseen reverberated speech in BUT system for the ASpIRE challenge. In: Proceedings of Interspeech 2015, pp. 2454–2458. International Speech Communication Association, Grenoble (2015). http://www.fit.vutbr.cz/research/view_pub.php?id=10972
19. Kinoshita, K., Delcroix, M., Nakatani, T., Miyoshi, M.: Suppression of late reverberation effect on speech signal using long-term multiple-step linear prediction. IEEE Trans. Audio Speech Lang. Process. **17**(4), 534–545 (2009)
20. Ko, T., Peddinti, V., Povey, D., Khudanpur, S.: Audio augmentation for speech recognition. In: INTERSPEECH, pp. 3586–3589. ISCA, Grenoble (2015)

21. Nakatani, T., Yoshioka, T., Kinoshita, K., Miyoshi, M., Juang, B.H.: Blind speech dereverberation with multi-channel linear prediction based on short time Fourier transform representation. In: Proceedings of ICASSP'08, pp. 85–88 (2008)
22. Ogata, K., Tachibana, M., Yamagishi, J., Kobayashi, T.: Acoustic model training based on linear transformation and MAP modification for HSMM-based speech synthesis. In: INTERSPEECH, pp. 1328–1331 (2006)
23. Ragni, A., Knill, K.M., Rath, S.P., Gales, M.J.F.: Data augmentation for low resource languages. In: INTERSPEECH 2014, 15th Annual Conference of the International Speech Communication Association, Singapore, September 14–18, 2014, pp. 810–814 (2014)
24. Saon, G., Soltau, H., Nahamoo, D., Picheny, M.: Speaker adaptation of neural network acoustic models using i-vectors. In: 2013 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), pp. 55–59. IEEE, New York (2013)
25. Siohan, O., Bacchiani, M.: iVector-based acoustic data selection. In: Proceedings of INTERSPEECH, pp. 657–661 (2013)
26. Swietojanski, P., Ghoshal, A., Renals, S.: Hybrid acoustic models for distant and multichannel large vocabulary speech recognition. In: 2013 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU). IEEE, New York (2013)
27. Tokuda, K., Zen, H., Black, A.: An HMM-based approach to multilingual speech synthesis. In: Text to Speech Synthesis: New Paradigms and Advances, pp. 135–153. Prentice Hall, Upper Saddle River (2004)
28. Veselý, K., Ghoshal, A., Burget, L., Povey, D.: Sequence-discriminative training of deep neural networks. In: Proceedings of INTERSPEECH 2013, pp. 2345–2349. International Speech Communication Association, Grenoble (2013). http://www.fit.vutbr.cz/research/view_pub.php?id=10422
29. Veselý, K., Watanabe, S., Žmolíková, K., Karafiát, M., Burget, L., Černocký, J.: Sequence summarizing neural network for speaker adaptation. In: Proceedings of ICASSP (2016)
30. Wang, Y., Gales, M.J.F.: Speaker and noise factorization for robust speech recognition. IEEE Trans. Audio Speech Lang. Process. **20**(7), 2149–2158 (2012). http://dblp.uni-trier.de/db/journals/taslp/taslp20.html#WangG12
31. Wei, K., Liu, Y., Kirchhoff, K., Bartels, C., Bilmes, J.: Submodular subset selection for large-scale speech training data. In: Proceedings of ICASSP, pp. 3311–3315 (2014)
32. Wu, Y., Zhang, R., Rudnicky, A.: Data selection for speech recognition. In: Proceedings of ASRU, pp. 562–565 (2007)
33. Xu, Y., Du, J., Dai, L.R., Lee, C.H.: An experimental study on speech enhancement based on deep neural networks. IEEE Signal Process Lett. **21**(1), 65–68 (2014)
34. Yoshimura, T., Masuko, T., Tokuda, K., Kobayashi, T., Kitamura, T.: Speaker interpolation in HMM-based speech synthesis system. In: Eurospeech, pp. 2523–2526 (1997)
35. Yoshioka, T., Nakatani, T.: Generalization of multi-channel linear prediction methods for blind MIMO impulse response shortening. IEEE Trans. Audio Speech Lang. Process. **20**(10), 2707–2720 (2012)
36. Yoshioka, T., Nakatani, T., Miyoshi, M., Okuno, H.G.: Blind separation and dereverberation of speech mixtures by joint optimization. IEEE Trans. Audio Speech Lang. Process. **19**(1), 69–84 (2011)
37. Yoshioka, T., Chen, X., Gales, M.J.F.: Impact of single-microphone dereverberation on DNN-based meeting transcription systems. In: Proceedings of ICASSP'14 (2014)
38. Yoshioka, T., Ito, N., Delcroix, M., Ogawa, A., Kinoshita, K., Fujimoto, M., Yu, C., Fabian, W.J., Espi, M., Higuchi, T., Araki, S., Nakatani, T.: The NTT CHiME-3 system: advances in speech enhancement and recognition for mobile multi-microphone devices. In: Proceedings of ASRU'15, pp. 436–443 (2015)
39. Zavaliagkos, G., Siu, M.-H., Colthurst, T., Billa, J.: Using untranscribed training data to improve performance. In: The 5th International Conference on Spoken Language Processing, Incorporating the 7th Australian International Speech Science and Technology Conference, Sydney, Australia, 30 November–4 December 1998 (1998)