Acoustic Model Adaptation In Reverberant Conditions Using Multi-task Learned Embeddings

Aditya Raikar*, Meet Soni*†, Ashish Panda*, Sunil Kumar Kopparapu*
*TCS Research, Mumbai
†Samsung Research Institute, Bangalore

***Abstract*—Acoustic environment plays a major role in the performance of a large-scale Automatic Speech Recognition (ASR) system. It becomes a lot more challenging when substantial amount of distortions, such as background noise and reverberations are present. Of late, it has been standard to use i-vectors for Acoustic Model (AM) adaptation. Embeddings from Single Task Learned (STL) neural network systems, such as x-vectors and r-vectors, have also been used to a varying degree of success. This paper proposes the use of Multi Task Learned (MTL) embeddings for large vocabulary hybrid acoustic model adaptation in reverberant environments. MTL embeddings are extracted from an affine layer of the deep neural network trained on multiple tasks such as speaker information and room information. Our experiments show that the proposed MTL embeddings outperform i-vectors, x-vectors and r-vectors for AM adaptation in reverberant conditions. Besides, it has been demonstrated that the proposed MTL-embeddings can be fused with i-vectors to provide further improvement. We provide results on artificially reverberated Librispeech data as well as real world reverberated HRRE data. On Librispeech database, the proposed method provides an improvement of 10.9% and 8.7% relative to i-vector in reverberated test-clean and test-other data respectively, while an improvement of 7% is observed relative to i-vector when the proposed system is tested on HRRE dataset.**

***Index Terms*—Acoustic Modeling, Multi-Task Learn ing, ASR, reverberation**

I. INTRODUCTION

Research in Automatic Speech Recognition (ASR) has been constantly evolving and improving ([1] [2]) with the advent of neural network based models thanks to the availability of very large training data and increased computational resources. Specifically, Acoustic model (AM) adaptation, a sub problem of ASR, has received immense interest from researchers in recent past and thus has been well explored. Its effectiveness lies in handling mismatched conditions between the training and test data, which is achieved by adapting on auxiliary information specific to the utterance, such as channel, ambiance, speaker, language and accent, etc. This concept has motivated several approaches, each tackling the train-test mismatch problem differently but with one common goal: a more efficient and robust ASR system. To achieve such goal, AM adaptation using i-vectors [3] has been widely used. The benefit of adaptation through i-vector stems from the fact that i-vectors are able to capture various aspects of "utterance specific" properties such as channel characteristics, noise, reverberation and speaker specific information, which are helpful for adapting acoustic models [4] [5] [6]. Single task learning (STL) based deep speaker embeddings such as x-vectors [7] have been shown to outperform i-vectors in the text-independent speaker verification/recognition task and language-ID task [8]. It was shown in [9] that like i-vectors, x-vectors are also capable of capturing speaker, channel, and phrase related characteristics. Naturally, research on the use of such deep STL embeddings for improving the ASR performance has received a lot of attention recently. The correlation between speaker embedding's performance in speaker recognition task and ASR was studied in [10]. Using i-vectors, x-vectors and deep convolutional neural network (CNN) embeddings [11], it was shown that x-vectors adaptation did not provide any improvement in ASR performance. In [12], x-vector like accent embeddings were used as auxiliary inputs to ASR in order to eliminate the mismatch between native and non-native speech utterances. In [13], a structured overview of various adaptation techniques was presented with a focus on domain, speaker and accent adaptation. Significant improvement in low resource ASR was found in [14], when i-vectors were replaced with x-vectors for adaptation. In [15], AM adaptation by another STL embedding called r-vector has been successfully shown to be effective for reverberant conditions.

Although Multi-task leaning [16] has been around for a long time, MTL embeddings have not been widely studied for AM adaptations. MTL networks have been designed to predict phoneme posterior as well as other tasks such such speaker identity [17], noise [18] and accent [19], however MTL embeddings stacked with feature vectors have not been explored for AM adaptations. The approach where embeddings are stacked with acoustic features provide an advantage where embedding extractor can be trained using a different dataset where meta-data of various tasks (attributes such as speaker label, RIR label, etc.) is available in training data, which may not be present in ASR training data. A multi-task learning neural network can be trained to discriminate between several tasks, such as gender, noise, reverberation and speaker etc. Therefore, it is possible to extract an embedding from it which will encode various parameters related to an acoustic environment. This makes such embeddings eminently suitable for AM adaptations. In this paper, we study AM adaptation using MTL embeddings for large vocabulary ASR systems. Our multi-task learning network is trained for both speaker

*†This work was done while the second author was working with TCS Research.

ISBN: 978-1-6654-6798-8

and room size classification. A 256 dimensional embedding is extracted from this network and is appended with the feature vectors for AM adaptation. The main contributions of this paper are:

- This work explores the less explored area of MTL embeddings for AM adaptations.
- We show that the proposed MTL embeddings are better than both i-vectors and r-vectors for the ASR in reverberant speech.
- We further show that fusion of i-vector and the proposed MTL embeddings provides the best performance gains.

II. Multi Task Learning based Embeddings

The STL based neural embeddings are trained with an objective to discriminate between corresponding single task labels. The r-vector and x-vector fall under such category, where they are trained over room acoustics and speaker information respectively. Such embeddings are usually extracted from the last network layers. The neural embeddings are learned using the TDNN layers [20] [10], followed by pooling layer, that collects the temporal statistics. Finally, the embeddings are then extracted from the subsequent affine layer.

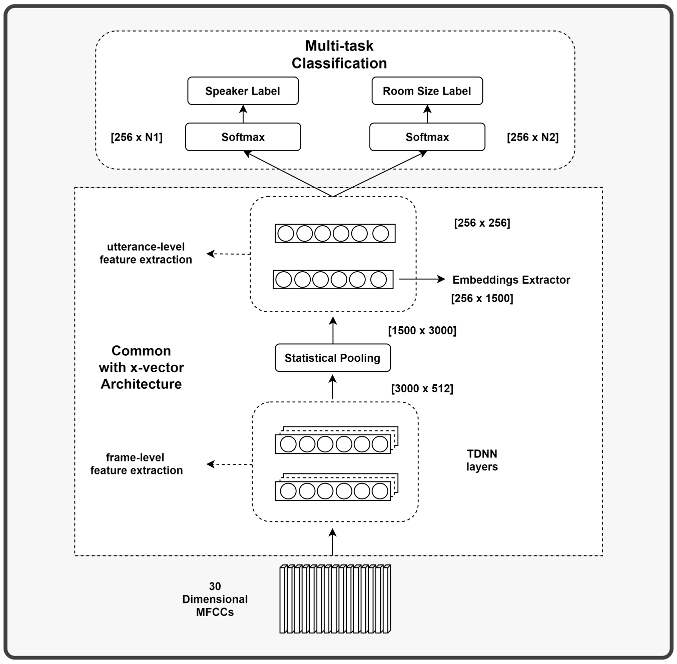

Fig. 1: Proposed extraction of MTL embeddings with two classifiers each for speaker (N1 number of speaker labels) and room size (N2 number of room size labels) respectively.

Unlike the r-vectors and x-vectors, the proposed MTL embeddings are extracted from an architecture that is designed to learn two task classifications: speaker information and room size. Trained this way, the embeddings will encode both speaker and room size information, which are important from ASR perspectives. Unlike original r-vector [15] which takes individual RIR as a classification label, we use room size as our output label. It makes learning task simpler since RIRs of similar room sizes are put in one class. It also enables use of large no. of RIRs without increasing the output size of the network. We compute the loss as follows:

$$\mathcal{L}_{\text{multi-task}} = \lambda_{speaker}\mathcal{L}_{\text{speaker}} + \lambda_{room}\mathcal{L}_{\text{room}} \tag{1}$$

Each task loss $\{ \mathcal{L}_{\text{speaker}}, \mathcal{L}_{\text{room}}\}$ is weighted by loss weighting $\{\lambda_{speaker}, \lambda_{room}\}$ respectively. This formulation provides us with the tool to adjust the level of domination a task will play in AM adaptation. For example, if both losses are given equal weightage, then both speaker and room size have an equal role in AM adaptation. This way we can observe which task is more important from the ASR perspective. We have used the implementation employed in [21], for this work. [1]. The embedding extractor architecture is same as that of the x-vector, up until the embedding layer, which has got 256 hidden units. It is further followed by two hidden layers, with 256 dimension each, which is then projected on to the number of classes. The classification task uses the standard cross entropy loss.

III. Experimental Setup

A. MTL Embedding extractor

In order to train the classifier, a mix of unaugmented (anechoic) and reverberation augmented version of Full Libripseech data was constructed. The data is approximately 960 hours long and there are total 2338 speakers. To construct reverberant classes, we use 80% of the total simulated Room Impulse Responses (RIRs) taken from [22] [23] for reverberant augmentation which have been labeled as small, medium and large, based on the room size. In addition to it, we add anechoic class to the 4 room size labels. A training set is constructed by sampling each utterance to be from an anechoic, small, medium or large room with equal probability. Test set was also prepared, which is a mix of ane- choic and reverberated version of the Librispeech test-clean. We make sure that the simulated RIRs used in the test sets are different from that used in the training sets. All the networks were trained for 100,000 iterations on 350 frames of MFCCs with 30-dimension. The batch size was set as 500 and a small held out set of training utterances is kept for validation. Stochastic gradient descent with learning rate 0.2 was used, along with 0.5 momentum.

B. ASR systems

All experiments are conducted using ASR models trained on Librispeech [24]. Initial experiments use reverberation augmented version of Librispeech 100 hours subset. The best combinations obtained from this 100 hours subset is then tested on the full Librispeech 960 hours dataset with reverberation augmentation. The simulated RIR set used to augment both Librispeech 100 and 960 hours respectively, is same as the test RIR set used to evaluate MTL embedding extractor. For evaluation of trained models, we use test-clean and test-other

[1] https://github.com/cvqluu/MTL-Speaker-Embeddings

subsets of Librispeech. In order to evaluate the effectiveness of room acoustic based AM adaptation, reverberated sets from test-clean and test-other are prepared. For this purpose, we use have used real RIRs: 271 from MIT impulse response survey [25], 188 from voiceHome [26], 2325 from [27] and 327 from [22] [23] Since total number of real RIRs are greater than utterances in test-clean and test-other, we made 3 different rev sets for each respectively. The naming convention followed for such set is $\{$test-c-rev-set-#, test-o-rev-set-#$\}$-$\{\# = 1,2,3\}$ for reverberated Librispeech test-clean and test-other set respectively. Further, we have also shown our results on HRRE database [28]. It consists of 13.4 hours clean speech utterances taken from Aurora-4 which is recorded in 20 different highly reverberant real environments, with different reflection coefficients and speaker-microphone pair distances.

All ASR systems trained using default kaldi recipes [22]. For Librispeech train-clean-100 set, we use kaldi Librispeech recipe [2] and modify it as follows. We train GMM-HMM system till $tri4$ using 100h subset and use it to obtain alignments of training data. Then we use default chain recipe to extract high-resolution MFCC features, train i-vector extractor and i-vector extraction, and perform AM training using Lattice-Free Maximum Mutual Information (LF-MMI) objective. We modify the recipe to use single GPU and train the network for 4 epochs. For Librispeech 960h dataset, we use Kaldi default recipe without cleanup part. Training is done using 2 parellel jobs on two GPUs, while keeping all other parameters same. During decoding, we use standard Language Models (LM) available for each task. For Librispeech evaluation subsets, we use small 3-gram LM and rescore the lattices generated using large 4-gram LM.

IV. Results and Discussion

A. *Experiments on 100 hour subset of Librispeech*

The experimental results reported in Table I provide an insight into the performance of various embeddings in reverberated conditions. We use 100 dimensional i-vectors to set the baseline, extracted in online manner during both training and decoding, which is standard in Kaldi recipes. The x-vectors, r-vectors and the MTL-embeddings are all of 256 dimensions. It can observed that for reverberated speech, x-vector performs better than i-vector and r-vector performs better than both. However, the proposed MTL-embeddings perform the best irrespective of the the weight assigned to speaker information loss and room size loss. Out of the three sets of weights tested by us, assigning weights 0.1 to speaker information loss and 1 to room size loss provided the best performance resulting in 1% absolute performance gain over the r-vectors. The weights assigned in the loss function suggest that room size information is more important for performance of ASR systems in the case of reverberated speech. However, the fact that MTL-embeddings provide similar performance to i-vector and x-vector adaptation in test-clean condition suggests that we are not losing performance because of lesser assigned weight to speaker information in the loss function.

Embedding	Test Data WER (in %)				
	test-clean	test-c-rev-set-1	test-c-rev-set-2	test-c-rev-set-3	Average test-rev
i-vector	5.43	31.12	32.69	32.01	32.94
x-vector	5.46	28.76	30.81	30.08	29.88
r-vector	5.49	28.16	30.09	29.57	29.54
MTL (1,0.1)	5.42	27.87	29.78	29.36	29.00
MTL (1,1)	5.46	27.47	29.63	29.41	28.83
MTL (0.1,1)	**5.42**	**27.26**	**29.32**	**29**	**28.52**

TABLE I: Results (% WER) of online i-vector, x-vector, r-vector and the proposed MTL embedding for various reverberated conditions. MTL(w1,w2) signifies assignment of weight w1 for speaker information loss and w2 for room size loss.

Embedding	Test Data WER (in %)				
	test-clean	test-c-rev-set-1	test-c-rev-set-2	test-c-rev-set-3	Average test-rev
i-vector	5.43	31.12	32.69	32.01	31.94
ivector+xvector	5.23	27.85	29.39	28.95	28.73
ivector+rvector	5.37	28.04	30.02	29.41	29.15
ivector+MTL(1,0.1)	**5.18**	26.75	28.95	**28.12**	27.94
ivector+MTL(1,1)	5.29	**26.62**	**28.51**	28.15	**27.72**
ivector+MTL(0.1,1)	5.3	27.03	29.11	28.83	28.32

TABLE II: Results (% WER) of various fusion techniques. x-vectors, r-vectors and the proposed embeddings were stacked with online i-vectors for AM adaptation. MTL(w1,w2) signifies assignment of weight w1 for speaker information loss and w2 for room size loss.

The next set of experiments explored possibility of fusion with online i-vectors. Table II tabulates the results from experiments where x-vectors, r-vectors and and proposed MTL-embeddings were stacked with i-vectors for AM adaptations. Such fusion techniques invariably improved the performance over online i-vectors. It seems to suggest that i-vectors alone may not be sufficient for AM adaptations in reverberated conditions. While r-vectors provided better performance than x-vectors on their own (Table I), they were worse performers in fusion. Stacking x-vectors with i-vectors resulted in an average WER of 28.73% in reverberated conditions while stacking r-vectors with i-vectors resulted in 29.15%. This shows that x-vectors compliment i-vectors better than r-vectors. The proposed MTL embeddings, however, comfortably outperform both other fusion schemes. Proposed MTL embeddings with equal weights to speaker information and room size loss was the best performer when fused with i-vectors. This fusion resulted in a relative performance gain of 13.27% over i-vector alone and 3.5% relative improvement over i-vector and x-vector fusion. Though the weights assigned to individual task loss is important, it is noteworthy that all three selected weights outperform x-vectors and r-vectors. Another observation is, although the training of MTL embedding extractor and AM adaptation is done using simulated RIRs, still with the help of this information it is able to perform quite good on disjoint real RIR test set, and outperform i-vector by a significant margin.

[2] https://github.com/kaldi-asr/kaldi/blob/master/egs/librispeech/s5/run.sh

B. Multi-task learning vs single-task learning

In the proposed MTL-embedding, the network is trained to learn two tasks, speaker identity and room-size, with certain weights assigned to the loss from each task. If the weight of the loss from room-size is set to zero, then the proposed embedding is akin to x-vector. If the weight from the speaker identity is set to zero, then the proposed embedding is akin to r-vector. Therefore, it would be interesting to see whether multi-task embedding has any advantage over multiple single task embedding stacked together. Table III presents the results from our experiments exploring this. It can be seen that the proposed MTL-embeddings outperform the stacked STL embeddings. It seems to suggest that the MTL-embeddings are better able to encode various aspects of the speech than STL embeddings taken together. Comparing results of stacked single task embeddings in Table III with the performance of MTL embeddings reported in II, it can be observed that the stacked STL embeddings do not outperform the MTL embeddings, irrespective of the weights given to individual task loss, which underlines the advantage of multi-task learning for AM adaptations.

Embedding	**Test Data WER (in %)**				
	test-clean	**test-c-rev-set-1**	**test-c-rev-set-2**	**test-c-rev-set-3**	**Average test-rev**
i-vector	5.43	31.12	32.69	32.01	31.94
Stacked STL	5.36	26.93	29.95	28.68	28.52
ivector+MTL(1,1)	**5.29**	**26.62**	**28.51**	**28.15**	**27.72**

TABLE III: Results (% WER) for experiments comparing stacked x-vector and r-vector and MTL-embeddings. Stacked STL stands for i-vector+x-vector+r-vector and MTL(w1,w2) signifies assignment of weight w1 for speaker information loss and w2 for room size loss

C. Experiments on Librispeech 960h data

1) Results on RIR augmented Librispeech Test Set: In this experiment, we evaluate the performance of MTL embeddings on full Librispeech 960h dataset. The analysis on the smaller Librispeech 100h data showed that fusion of i-vector and other embeddings provided the best improvements. Also, it was shown that MTL(1,1) provided the best performance when used alongside i-vectors for AM adaptations. Therefore, we compare three systems on the larger Librispeech 960h hours data: i-vector+r-vector, i-vector+x-vector and i-vector+MTL(1,1). Experiments were conducted on the reverberated versions of both test-clean and test-other set. Table IV presents the results for reververated test-clean data while Table V presents the same for reverberated test-other data.

It can be observed that the results follow the same trend as in the case of Librispeech 100h training data subset. In both reverberated test-clean and test-other data, the proposed MTL-embeddings provide superior performance reducing the WER by 10.9% and 8.7% relative to i-vector on test-clean and test-other data respectively. Over the stacked i-vectors and r-vectors, the relative reduction in WER are 6.8% and 5.0% on test-clean and test-other data respectively.

Embedding	**Test Data WER (in %)**				
	test-clean	**test-c-rev-set-1**	**test-c-rev-set-2**	**test-c-rev-set-3**	**Average test-rev**
i-vector	4.06	21.11	22.86	22.57	22.18
i-vector+r-vector	3.98	20.08	21.89	21.55	21.17
i-vector+x-vector	**3.87**	19.25	21.16	20.82	20.41
ivector+MTL(1,1)	3.97	**18.70**	**20.31**	**20.21**	**19.74**

TABLE IV: Results (% WER) for experiments on full Librispeech 960h training data and reverberated test-clean set. MTL(w1,w2) signifies assignment of weight w1 for speaker information loss and w2 for room size loss

Embedding	**Test Data WER (in %)**				
	test-other	**test-o-rev-set-1**	**test-o-rev-set-2**	**test-o-rev-set-3**	**Average test-rev**
i-vector	9.68	36.28	35.17	35.93	35.79
i-vector+r-vector	9.54	35.19	33.50	34.55	34.41
i-vector+x-vector	9.44	33.92	32.59	33.53	33.34
ivector+MTL(1,1)	**9.40**	**33.17**	**31.84**	**33.05**	**32.68**

TABLE V: Results (% WER) for experiments on full Librispeech 960h training data and reverberated test-other set. MTL(w1,w2) signifies assignment of weight w1 for speaker information loss and w2 for room size loss

D. Experiments on HRRE test data

Here we present the results on HRRE test set, which contains real reverberant speech data. The dataset comes with 20 real recorded reverberant testsets with different reflection coefccients and speaker-microphine pair. The final results shown for each experiment on HRRE is the average across all the 20 testsets. We analyze the performance of the proposed MTL embeddings with Librispeech 100h and 960h models.

1) Results using Librispeech 100h Model: Table VI and VII present the WERs from the experiments conducted using model trained on Librispeech 100 hours. We first try to observe what combination of weights: w1 and w2 of MTL(w1,w2) embeddings works best on HRRE test set. Here also we observe that out of the three sets of weights w1,w2 = (1,1) provides the best performance, which is in line with the trend observed on artificially reverberated Librispeech test set.

HRRE Test Data				
Embeddings	**i-vector**	**i-vector+MTL(1,0.1)**	**i-vector+MTL(0.1,1)**	**i-vector+MTL(1,1)**
WER	25.16	23.71	23.48	**22.91**

TABLE VI: Results (% WER) for experiments on HRRE dataset tested with Librispeech 100h model. MTL(w1,w2) signifies assignment of weight w1 for speaker information loss and w2 for room size loss

Further we compared the best performing set of weights: w1,w2 = (1,1) and compared with x-vector and r-vector, and here too we see that MTL(1,1) outperform others significantly.

HRRE Test Data				
Embeddings	**i-vector**	**i-vector+x-vector**	**i-vector+r-vector**	**i-vector+MTL(1,1)**
WER	25.16	23.99	24.08	**22.91**

TABLE VII: Results (% WER) for experiments on HRRE dataset tested with Librispeech 100h model. MTL(w1,w2) signifies assignment of weight w1 for speaker information loss and w2 for room size loss

2) Results using Librispeech 960h Model: Table VIII presents the results of HRRE data tested with Librispeech 960h model. Here also we see that the trend seen in artificially reverberated Librispeech data is replicated, with the proposed MTL embeddings outperforming others. It is noteworthy that the r-vector is not performing as well as the x-vector and the proposed MTL. In fact, out of the 20 different sub-sets in HRRE, the proposed method outperforms the r-vector in all the subsets, but 1.

HRRE Test Data				
Embeddings	**i-vector**	**i-vector+x-vector**	**i-vector+r-vector**	**i-vector+MTL(1,1)**
WER	18.81	17.88	18.35	**17.58**

TABLE VIII: Results (% WER) for experiments on HRRE dataset tested with Librispeech 960h model. MTL(w1,w2) signifies assignment of weight w1 for speaker information loss and w2 for room size loss

V. Conclusion and Future work

In this paper, we introduced MTL-embeddings for AM adaptation for improved ASR performance in reverberated conditions. The MTL-embedding extractor was trained to classify two tasks, namely, speaker identity and room size. Training the MTL-embedding extractor followed by AM adaptation using simulated room impulse response, we observe that MTL-embedding used for AM adaptation outperform i-vector on test set built using real RIR, as well as on real reverb environment recorded HRRE dataset. The proposed MTL-embeddings provided superior performance to x-vectors and r-vectors. We also showed, experimentally, that i-vectors and our MTL-embeddings containing complimentary characteristics of the speech signal and work best when they are stacked together for AM adaptations. In future, as an extension, we propose to study richer MTL-embeddings by training the network for more tasks such as noise types, gender, etc. Also, it would be interesting to study, if an optimal set of weights exists, that can be assigned to loss from each tasks in a variety of conditions.

References

[1] George Saon et al., "English conversational telephone speech recognition by humans and machines," *arXiv preprint arXiv:1703.02136*, 2017.

[2] Chung-Cheng Chiu et al., "State-of-the-art speech recognition with sequence-to-sequence models," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 4774–4778.

[3] Najim Dehak et al., "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2010.

[4] George Saon, Hagen Soltau, David Nahamoo, and Michael Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*. IEEE, 2013, pp. 55–59.

[5] Martin Karafiát, Lukáš Burget, Pavel Matějka, Ondřej Glembek, and Jan Černockỳ, "ivector-based discriminative adaptation for automatic speech recognition," in *2011 IEEE Workshop on Automatic Speech Recognition & Understanding*. IEEE, 2011, pp. 152–157.

[6] Karel Veselỳ, Shinji Watanabe, Katerina Žmolíková, Martin Karafiát, Lukáš Burget, and Jan Honza Černockỳ, "Sequence summarizing neural network for speaker adaptation," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5315–5319.

[7] David Snyder, Daniel Garcia-Romero, Daniel Povey, and Sanjeev Khudanpur, "Deep neural network embeddings for text-independent speaker verification.," in *Interspeech*, 2017, pp. 999–1003.

[8] David Snyder, Daniel Garcia-Romero, Alan McCree, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, "Spoken language recognition using x-vectors.," in *Odyssey*, 2018, pp. 105–111.

[9] Desh Raj, David Snyder, Daniel Povey, and Sanjeev Khudanpur, "Probing the information encoded in x-vectors," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 726–733.

[10] Joanna Rownicka, Peter Bell, and Steve Renals, "Embeddings for dnn speaker adaptive training," in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2019, pp. 479–486.

[11] Joanna Rownicka, Peter Bell, and Steve Renals, "Analyzing deep cnn-based utterance embeddings for acoustic model adaptation," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 235–241.

[12] MA Tuğtekin Turan, Emmanuel Vincent, and Denis Jouvet, "Achieving multi-accent asr via unsupervised acoustic model adaptation," in *INTERSPEECH 2020*, 2020.

[13] Peter Bell, Joachim Fainberg, Ondrej Klejch, Jinyu Li, Steve Renals, and Pawel Swietojanski, "Adaptation algorithms for speech recognition: An overview," *arXiv preprint arXiv:2008.06580*, 2020.

[14] Martin Karafiát et al., "Analysis of x-vectors for low-resource speech recognition," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 6998–7002.

[15] Yuri Y Khokhlov et al., "R-vectors: New technique for adaptation to room acoustics," in *INTERSPEECH*, 2019, pp. 1243–1247.

[16] Rich Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[17] Gueorgui Pironkov, Stphane Dupont, and Thierry Dutoit, "Speaker-aware multi-task learning for automatic speech recognition," in *23rd International Conference on Pattern Recognition (ICPR)*, 2016.

[18] Suyoun Kim, Bhiksha Raj, and Ian R. Lane, "Environmental noise embeddings for robust speech recognition," *CoRR*, vol. abs/1601.02553, 2016.

[19] Abhinav Jain, Minali Upreti, and Preethi Jyothi, "Improved accented speech recognition using accent embeddings and multi-task learning," in *Proc. Interspeech 2018*, 2018, pp. 2454–2458.

[20] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.

[21] Chau Luu, Peter Bell, and Steve Renals, "Leveraging speaker attribute information using multi task learning for speaker verification and diarization," *arXiv preprint arXiv:2010.14269*, 2020.

[22] Daniel Povey et al., "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*. IEEE Signal Processing Society, 2011.

[23] Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L Seltzer, and Sanjeev Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5220–5224.

[24] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[25] James Traer and Josh H McDermott, "Statistics of natural reverberation enable perceptual separation of sound and space," *Proceedings of the National Academy of Sciences*, vol. 113, no. 48, pp. E7856–E7865, 2016.

[26] Ewen Camberlein and Romain Lebarbenchon, "voicehome-2 corpus-localization and speech enhancement baseline-code," 2017.

[27] Igor Szöke, Miroslav Skácel, Ladislav Mošner, Jakub Paliesek, and Jan Černockỳ, "Building and evaluation of a real room impulse response dataset," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 4, pp. 863–876, 2019.

[28] Juan Pablo Escudero, Victor Poblete, José Novoa, Jorge Wuth, Josué Fredes, Rodrigo Mahu, Richard Stern, and Néstor Becerra Yoma, "Highly-reverberant real environment database: Hrre," *arXiv preprint arXiv:1801.09651*, 2018.