Investigating the Impact of the Training Data Volume for Robust Speech Recognition using Multi-Task Learning

Gueorgui Pironkov, Stéphane Dupont, Thierry Dutoit
TCTS Lab, University of Mons, Belgium
{gueorgui.pironkov, stephane.dupont, thierry.dutoit}@umons.ac.be

***Abstract*—Dealing with speech corrupted by noise and reverberation is still an issue for automatic speech recognition. To address this, a solution that can be combined with multi-style learning consists of using multi-task learning, where the acoustic model is trained to solve one main task and at least one auxiliary task simultaneously. In noisy and reverberant environment using clean-speech features estimation as auxiliary task has proven its efficiency, while the main task is speech recognition. Still, recognizing speech in these degraded conditions is all the more difficult when the amount of available data during training is limited. Thus, in this paper, we evaluate robust speech recognition based on multi-task learning when the amount of training data is gradually reduced. We show that using multi-task learning improves recognition and more specifically, that its impact is even more significant if the amount of training data is decreasing (up to 12% relative improvement of the word error rate). All training and testing experiments are carried out on parts of the CHiME4 database.**

I. Introduction

The progress brought by Deep Neural Networks (DNN) in recent years has made them a very powerful tool for a wide variety of classification and regression tasks [1]. For Automatic Speech Recognition (ASR) acoustic modeling, DNNs have gradually outperformed the previous state-of-the-art methods based on Gaussian Mixture Models (GMM) [2]. In fact, the improvement brought by DNNs is now reaching levels were studies are starting to argue that near human-level performance can be obtained [3]. Considering ASR as a solved a problem would be an error thought, and this is especially true concerning ASR in noisy and reverberant environment [4]. In order to tackle this problem, different methods have been investigated, including enhancing the input features at the front-end of the ASR system for instance.

Our study though, does not focus on pre-processing features and is instead interested in using Multi-Task Learning (MTL) to improve ASR performance in this corrupted acoustic environment. As opposed to the traditional Single-Task Learning (STL) architecture where a system is trained to solve only one task, MTL consists of training one single system to solve multiple tasks that are related but still different [5]. Using MTL in order to improve speech recognition has already been tested for a variety of situations where ASR is the main task while different auxiliary tasks are added [6], [7]. Despite showing improvement for ASR in clean acoustic conditions, few MTL auxiliary tasks have been found to be helpful when speech is corrupted by noise and reverberation. Generating the clean-speech feature as an auxiliary task can be cited among the most efficient approaches though [8], [9], [10], [11]. Another problem related to ASR is the amount of annotated data, as annotated speech is critical for ASR training and might be an issue for languages with fewer available resources. Thus, in this work, we focus on how MTL can improve ASR compared to STL when the amount of available training data is limited. As MTL could be seen as a regularization method, our hypothesis was that there exists a correlation between the volume of training data and the benefit brought by MTL training. In order to evaluate this correlation we use the simulated speech of the CHiME4 dataset [4]. Indeed, the CHiME4 dataset contains both real and simulated data. However, only simulated data can be used during training here, since we need the clean-speech features to train the MTL auxiliary task.

This paper is organized as follows. First, an overview of the state-of-the-art in MTL for ASR is presented in Section II. The MTL mechanism is then described in depth in Section III. The details concerning the experimental setup used to evaluate the impact of MTL with limited noisy and reverberant speech are presented in Section IV, while the results are presented and discussed in Section V. Finally, the conclusion and ideas for future work are examined in Section VI.

II. Related Work

Multi-task learning has been successfully applied for ASR with a variety of auxiliary tasks [6], as for instance using gender classification [8], [12]. Still, most of these tasks are efficient when speech recognition is done in

clean conditions, that are not deteriorated by noise, nor reverberation. Using MTL for robust ASR is also a field of interest though, and some auxiliary tasks have shown promising results.

For instance, some studies have focused on solving only the problem of reverberation in speech by training on reverberant speech features as input and generating de-reverberated speech features as auxiliary task [13], [14]. Rather than focusing only on reverberation, other studies focus only on noise in their MTL auxiliary task, where the auxiliary task tries to recognize the type of noise present in the corrupted speech [15], [16]. This classification auxiliary task brings a very limited improvement to the main ASR task. However, a far more promising and successfully approach consists of generating the clean-speech features as auxiliary task [8], [9], [10], [11], that is the same clean-speech features to which noise and reverberation is artificially added for training purpose. Implicitly, this means that an access to this clean-speech is required in order to generate the targets of the auxiliary task, making it very difficult to use real data. Instead, such an MTL system is trained using simulated noisy and reverberant data. Finally, it can be noted that an MTL system can be used as a feature extractor as well, where a Bottle-Neck (BN) layer is added to the neural network [17]. In that case, the MTL system performs two distinct tasks: speaker and noise classification, while the BN features extracted though the MTL system are used as input of a traditional STL-ASR system.

Using MTL for ASR may be seen as a regularization method, as for MTL a term is added to the cost function (see Section III) similarly to L1/L2 regularization for instance. Thus, an expected outcome is that the impact of MTL is directly dependent of the training data volume. This remark is confirmed by Caruana results where the main task is mortality risk and the auxiliary tasks are predicting the white blood cell count and the partial pressure of oxygen in the blood [5]. However, these results are not confirmed when MTL is used for ASR. Indeed, in Bell et al. study, an MTL-ASR system is trained on the TED talks database [18] while the auxiliary task focused on predicting monophones [19]. The amount of data is progressively reduced from 100% ($\approx$ 145 hours) to 10%. They show that the improvement brought by MTL compared to STL is not influenced by the amount of data, suggesting that using MTL for ASR does not only bring a regularization effect on the acoustic model training.

Bell et al. work focused on MTL-ASR in clean conditions with large vocabulary continuous speech recognition databases, where the auxiliary task is monophone classification. In this paper we present a similar study but we investigate MTL-ASR in noisy and reverberant conditions with a smaller database ($\approx$ 15 hours for CHiME4), while the auxiliary task is clean-speech generation.

To the best of our knowledge, there have been no attempts to investigate the impact of MTL for ASR when the amount of available annotated training data is limited and corrupted by noise and reverberation.

III. Multi-Task Learning

The core concept of multi-task learning, introduced by Rich Caruana in 1997, consists of training a single system (a neural network here) to solve simultaneously multiple tasks that are different but still related [5]. In the MTL nomenclature, the principal task is referred to as the *main task*, which is the task that would be initially used for an STL training. An example of a MTL model with one main task and N auxiliary tasks is presented in Figure 1. At least one *auxiliary task* is added during training to help improve the convergence of the neural network to the benefit of the main task.

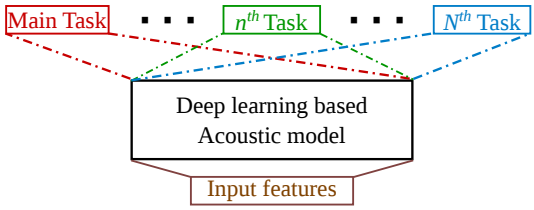

Fig. 1: A Multi-Task Learning network with one main task and N auxiliary tasks.

Two essential characteristics are shared among all MTL systems: a) The same input features are applied for training both the main task and the auxiliary task(s); b) The network parameters (namely weights and biases of neurons), are shared among the main and auxiliary tasks (with the exception of the output layer). These parameters are furthermore updated by backpropagating the sum of the errors associated to each task, with a term:

$$\epsilon_{MTL} = \epsilon_{Main} + \sum_{n=1}^{N} \lambda_n * \epsilon_{Auxiliary_n} \ ,$$

where ϵ_{MTL} corresponds to the sum of all the task errors to be minimized, ϵ_{Main} and $\epsilon_{Auxiliary_n}$ are the errors associated with the *main* and *auxiliary* tasks respectively, whereas λ_n is a nonnegative weight that is associated with each of the *auxiliary* tasks, and finally N is the total number of auxiliary tasks applied during training. The influence of the auxiliary task with respect to the main task is controlled by the value of λ_n. On the one hand, if λ_n is close to 1, then the n^{th} auxiliary task will contribute equally to the error estimation as the main task. On the other hand, for a λ_n closer to 0, the system will behave

as a single-task learning system due to the very small (or nonexistent) influence of the auxiliary task. Only the main task is kept during testing, as the auxiliary task is removed. The selection of a relevant auxiliary task with respect to the main task is the critical point leading to a better convergence of the main task. Sharing the parameters of the system among multiple tasks, instead of computing and training each of the tasks independently, may lead to better results than the independent processing of each task [5].

IV. Experimental Setup

A. Database

The dataset partially used for the robust ASR training and testing is the CHiME4 database [4]. This database, released in 2016, was proposed for a speech recognition and separation challenge in reverberant and noisy environment. This dataset contains 1-channel, 2-channel, and 6-channel microphone array recordings. A total of four different noisy environments (café, street junction, public transport, and pedestrian area) were used to record *real* acoustic mixtures using a tablet device with 6-channel microphones. *Simulated* data is obtained using the WSJ0 database [20]. Noise is added to the WSJ0 clean-speech recordings, the noise being recorded in the four noisy environments described above. As access to the clean-speech is required to extract the targets of the clean-speech extraction auxiliary task, only the simulated dataset is used for training. All three datasets (training, development, and test sets) consist of 16 bit wav files sampled at 16 kHz. The simulated dataset used for training contains speech uttered by 83 speakers for a total of 7138 sentences, which is the equivalent of $\approx$15 hours. This also corresponds to the largest dataset (100%) used for training, that is progressively reduced in order to evaluate the impact of MTL for robust ASR. The development set contains the same volume of simulated and real data, that is a total of 8 different speakers (4 per dataset real/simulated dataset) uttering a total of 3280 utterances ($\approx$5.6 hours). Similarly, the evaluation set consists of a total of 8 speakers uttering 2640 sentences leading to approximately 4.5 hours.

In this paper, clean-speech estimation is considered as auxiliary task, therefore we use only the noise recorded from a single channel (channel n^{o}5) during training. The development and test set noises though are randomly selected among all available channels, making the main ASR and the auxiliary tasks harder but also challenging the generalization ability of the MTL setup.

B. Features

The following traditional ASR pipeline is used to extract the features used as input for training the MTL system, as well as targets for the clean-speech estimation task. Starting from the raw audio wav files, 13-dimensional Mel-Frequency Cepstral Coefficients (MFCC) features are extracted and additionally normalized through Cepstral Mean-Variance Normalization (CMVN). The adjacent $\pm$ 3 frames are spliced for each frame. The obtained 91-dimensional feature vectors are then reduced through a Linear Discriminative Analysis (LDA) transformation to a 40-dimensional feature space. The final step of the pipeline consists of projecting the 40-dimensional features through a feature-space speaker adaptation transformation known as feature-space Maximum Likelihood Linear Regression (fMLLR), with no further dimension reduction at this stage. Finally, additional temporal context is given to the network during training by splicing the surrounding $\pm$ 5 frames concerning the input features fed to the acoustic model (using the 40-dimensional features that are computed through this pipeline). For the targets of the auxiliary task, the same pipeline is used to generate the clean-speech features but there is no $\pm$ 5 splicing.

C. Training the acoustic model

Training and testing the proposed MTL setup for robust ASR was done using the *nnet3* version of the Kaldi toolbox [21].

A traditional feed-forward deep neural network acoustic model is used to evaluate the performance of this auxiliary task while progressively reducing the amount of data. The feed-forward DNN is composed of 4 hidden layers, each of them having 1024 neurons activated through Rectified Linear Units (ReLU). The main ASR task used for STL and MTL as well computes 1972 phone-state posterior probabilities after a softmax output layer. The DNN training is done through 14 epochs with an initial learning rate starting at 0.0015 that is progressively reduced to 0.00015, using the cross-entropy loss function for the main ASR task, and quadratic loss function for the clean-speech estimation auxiliary task (as it is a regression problem). The DNN parameters are updated through stochastic gradient descent (SGD) by backpropagating the error derivatives. The mini-batch size used to process the input features is equal to 512. These hyper-parameters were set through empirical observations.

The experiments presented in this article were also considered using other deep learning algorithms such as Recurrent Neural Networks (RNN) with Long Short-Term Memory (LSTM) cells and Time-Delay Neural Networks (TDNN) as acoustic models instead of a feed-forward DNN. However, the results obtained with the feed-forward DNN were similar or better than with these more complex neural network architectures on the CHiME4 simulated dataset. Also, the feed-forward DNN computational time was much lower for training compared to the RNN-LSTM.

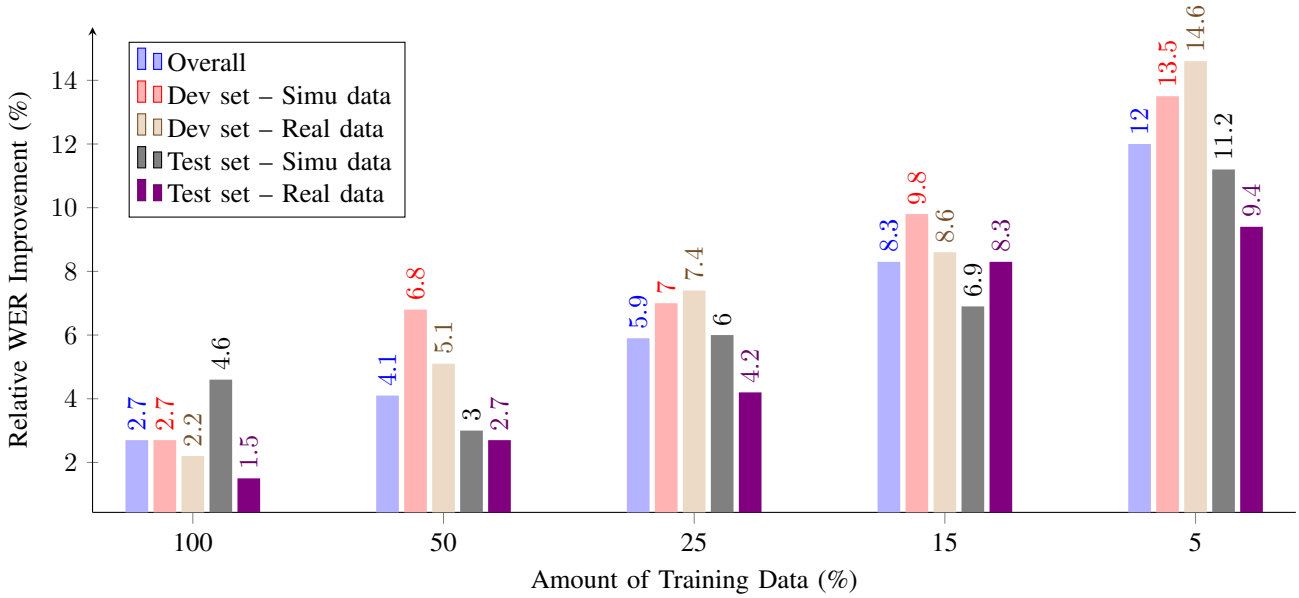

Fig. 2: Evolution of the relative WER improvement brought by Multi-Task Learning compared to single-task learning while the amount of training data is progressively reduced. MTL is done with clean-speech estimation as auxiliary task and with a fixed value of $\lambda = 0.15$. The *Overall* value is computed over all four datasets.

The phone-state probabilities estimated by the feed-forward network are fed to a HMM system and associated with a language model, in order to obtain the most likely transcriptions during decoding, the language model being a 3-gram KN language model trained on the WSJ 5K standard corpus.

In this work, the baseline is obtained by training the setup and its hyper-parameters presented previously in a single-task learning manner. The results are then compared to the MTL results while using the same volume of simulated data for training. The word error rate (WER) is computed for both the development and test sets over all four noisy environments of CHiME4 for real and simulated data.

V. Results

After varying the value of λ, the best WER is reached for $\lambda = 0.15$. In order to evaluate the impact of MTL compared to STL for robust ASR with limited amount of data, we progressively reduce the training set from 100% to 50%, then 25%, 15% and finally only 5%, by randomly removing utterances from the training set. More specifically, we are interested in the relative WER improvement brought by MTL compared to STL depending of the amount of data. The obtained relative WER improvement is represented in Fig. 2, while the detailed WER results for the STL baseline and the MTL setup are shown in Table I.

Before discussing the impact of the amount of training data, we can notice a significant difference between the development and test datasets results (for simulated and real data as well, independently of the amount of data or the STL/MTL training). For instance, when 100% of the data is used for an STL training, the real data of the development set reaches 18.09% WER while for the test set this value goes up to 31.81%. The variability of the recording conditions partially explains this mismatch. The *Lombard effect*[1] might be another explanation. Finally, 83 speakers are used during training, while only 8 are used in the test and development sets. Thus, if the speech of one of the speakers is harder to recognize, its impact will be more degrading on the WER.

Fig. 2 highlights the positive impact of MTL as the training data decreases. The overall relative improvement increases from 2.7% for 100% of the training set up to 12% when only 5% of the training data is used. It should be noted though, that for 5% of kept training data, the WER results significantly drop, going for STL from 34.14% when 15% of the data is kept to 72.86% for 5% of kept data overall. This drop can be explain by the very small amount of utterances used during training in this situation (357 utterances). So for this specific amount of training data, even though MTL improves the overall WER result

[1]Human natural tendency to rise voice when speaking in a noisy and reverberant environment.

Amount of training data	Type of system	Overall	Dev Set			Test Set		
			Mean	Simu	Real	Mean	Simu	Real
100% (7138 ut.)	STL	23.73	18.27	18.45	18.09	29.18	26.55	31.81
	MTL	23.08	17.83	17.96	17.70	28.32	25.32	31.32
50% (3596 ut.)	STL	26.78	21.00	21.16	20.83	32.56	29.55	35.57
	MTL	25.69	19.75	19.72	19.77	31.64	28.67	34.60
25% (1785 ut.)	STL	29.45	23.54	23.56	23.52	35.36	32.07	38.64
	MTL	27.71	21.85	21.90	21.79	33.58	30.16	37.00
15% (1071 ut.)	STL	34.14	27.91	27.56	28.26	40.38	36.61	44.14
	MTL	31.31	25.35	24.87	25.82	37.27	34.07	40.47
5% (357 ut.)	STL	72.86	65.57	62.34	68.80	80.14	74.32	85.96
	MTL	64.14	56.33	53.93	58.73	71.95	66.00	77.90

TABLE I: Word error rate (in %) of Single-Task Learning and Multi-Task Learning when the amount of training data is progressively reduced. MTL is done with clean-speech estimation as auxiliary task and with a fixed value of $\lambda = 0.15$. The *Overall* value is computed over all four datasets, while *ut.* stands for "utterances".

(64.14%), the network is unable to converge and is most likely stuck in a local minimum during the SGD.

Nevertheless, for more than 5% of kept training data, the WER still increases but at a much slower speed, going for an STL system from 23.73% WER for 100% of training data to 34.14% for only 15% of kept data. A significant correlation can be seen between the amount of training data and the benefit of using MTL, as nearly each time that the volume of training data is divided by two, the relative improvement brought by MTL is increased by two percent. Thus, using MTL for robust ASR when the amount of available data is limited might significantly improve recognition. Additionally, this improvement is also noticeable on real and simulated data, while training is done only on simulated data, showing that speech recognition in real-life conditions is benefiting from this auxiliary MTL task trained on simulated data only.

It should also be mentioned that our results could look somehow contradictory with Bell et al. work [19] discussed in Section II. Their results showed that for even 10% of training data the improvement brought by MTL compared to STL was similar to the improvement brought when 100% of their training data was used. But, besides using clean-speech for training as well as another auxiliary task, another significant difference concerns the amount of data. Using 10% of TED talk database implies that around 15 hours of training hours were used, which is the same amount of data corresponding to 100% of the simulated training dataset of CHiME4 that we use in this paper. As Bell et al. work shows that for large amounts of training data, there is still an improvement brought by MTL, suggesting that MTL behaves as more than just a regularization method, we show here that when the amount of training data is low, MTL does behave as a regularization method and even more significantly improves the WER compared to STL.

This remark is confirmed by the evolution of the main ASR task error during training (with 25% of the data) depicted in Fig 3. The error on the training set using STL is lower compared to MTL but the system is over-fitting as the error for STL on the validation set is much higher than for MTL, highlighting the better generalization of MTL to unseen data.

VI. Conclusion

In this paper, multi-task learning acoustic modeling with limited amount of training data for robust speech recognition was evaluated. We used clean-speech generation as auxiliary task, as it has shown its benefits in noisy and reverberant conditions. We show that the relative word error rate improvement brought by MTL compared to STL gradually increases as the amount of available training data gradually decreases. Additionally, training is done using only simulated data in this work (as the clean-speech is required in order to extract the targets of the MTL auxiliary task), but we demonstrate that both real and simulated data benefit from this training approach.

Having confirmed that MTL can improve ASR when the amount of training data is limited, a study we would like to investigate in the future concerns speaker adaptation. As usually for speaker adaptation the amount of data belonging to that speaker is very limited, it will be interesting to use MTL as a robust speaker adaptation tool.

Also, using more than 15 hours of data for training could be evaluated in this noisy and reverberant context as future work, thus showing if the improvement brought by MTL

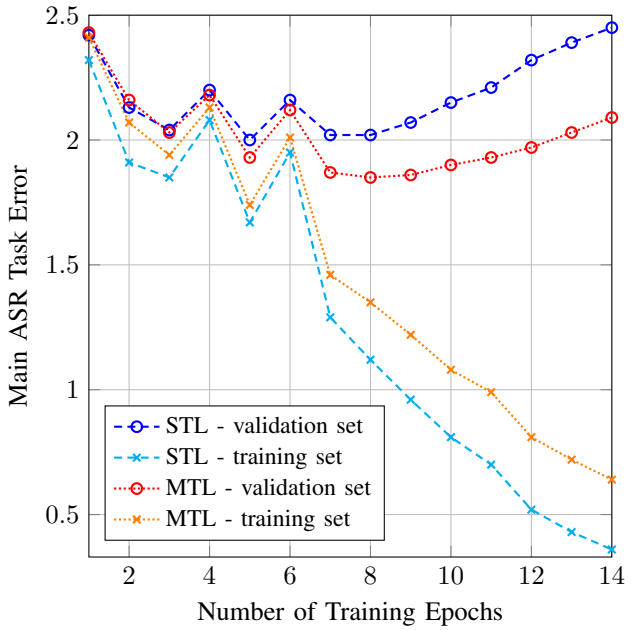

Fig. 3: Main task errors computed per epoch during training, when using 25% of the training set on the training and validations sets.

would reach a limit in comparison to STL for a certain amount of data.

Finally, we also would like to further study robust low-resources ASR. In the low-resources scenario, often additional clean-speech data exists but is not annotated in terms of accurate transcriptions. Using this data to create simulated corrupted speech is easy though. We could then design a system that switches dynamically from MTL to STL depending the annotation level of training samples. If they are annotated the system would behave as the MTL setup we present in the article. If they are not, the main ASR task would be dropped, keeping only the clean-speech estimation task. This setup will allow us to benefit from the non-annotated recordings while training for robust speech recognition.

Acknowledgment

This work has been funded by the Walloon Region of Belgium through the SPW-DGO6 Wallinov Program n°1610152.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *Signal Processing Magazine, IEEE*, vol. 29, no. 6, pp. 82–97, 2012.

[3] W. Xiong, J. Droppo, X. Huang, F. Seide, M. Seltzer, A. Stolcke, D. Yu, and G. Zweig, "Achieving human parity in conversational speech recognition," *arXiv preprint arXiv:1610.05256*, 2016.

[4] E. Vincent, S. Watanabe, A. A. Nugraha, J. Barker, and R. Marxer, "An analysis of environment, microphone and data simulation mismatches in robust speech recognition," *Computer Speech & Language*, 2016.

[5] R. Caruana, "Multitask learning," in *Learning to learn*. Springer, 1997.

[6] G. Pironkov, S. Dupont, and T. Dutoit, "Multi-task learning for speech recognition: an overview," in *Proceedings of the 24th European Symposium on Artificial Neural Networks (ESANN)*, 2016.

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

[8] Y. Lu, F. Lu, S. Sehgal, S. Gupta, J. Du, C. H. Tham, P. Green, and V. Wan, "Multitask learning in connectionist speech recognition," in *Proceedings of the Australian International Conference on Speech Science and Technology*, 2004.

[9] Z. Chen, S. Watanabe, H. Erdogan, and J. R. Hershey, "Speech enhancement and recognition using multi-task learning of long short-term memory recurrent neural networks." in *INTERSPEECH*. ISCA, 2015, pp. 3274–3278.

[10] B. Li, T. N. Sainath, R. J. Weiss, K. W. Wilson, and M. Bacchiani, "Neural network adaptive beamforming for robust multichannel speech recognition," in *Proc. Interspeech*, 2016.

[11] Y. Qian, M. Yin, Y. You, and K. Yu, "Multi-task joint-learning of deep neural networks for robust speech recognition," in *IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU), 2015*. IEEE, 2015, pp. 310–316.

[12] J. Stadermann, W. Koska, and G. Rigoll, "Multi-task learning strategies for a recurrent neural net in a hybrid tied-posteriors acoustic model." in *INTERSPEECH*, 2005, pp. 2993–2996.

[13] R. Giri, M. L. Seltzer, J. Droppo, and D. Yu, "Improving speech recognition in reverberation using a room-aware deep neural network and multi-task learning," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015*. IEEE, 2015, pp. 5014–5018.

[14] Y. Qian, T. Tan, and D. Yu, "An investigation into using parallel data for far-field speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2016*. IEEE, 2016, pp. 5725–5729.

[15] S. Sakti, S. Kawanishi, G. Neubig, K. Yoshino, and S. Nakamura, "Deep bottleneck features and sound-dependent i-vectors for simultaneous recognition of speech and environmental sounds," in *Spoken Language Technology Workshop (SLT), 2016 IEEE*. IEEE, 2016, pp. 35–42.

[16] S. Kim, B. Raj, and I. Lane, "Environmental noise embeddings for robust speech recognition," *arXiv preprint arXiv:1601.02553*, 2016.

[17] S. Kundu, G. Mantena, Y. Qian, T. Tan, M. Delcroix, and K. C. Sim, "Joint acoustic factor learning for robust deep neural network based automatic speech recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2016*. IEEE, 2016, pp. 5025–5029.

[18] A. Rousseau, P. Deléglise, and Y. Esteve, "TED-LIUM: an Automatic Speech Recognition dedicated corpus."

[19] P. Bell, P. Swietojanski, and S. Renals, "Multitask learning of context-dependent targets in deep neural network acoustic models," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 2, pp. 238–247, 2017.

[20] J. Garofolo, D. Graff, D. Paul, and D. Pallett, "CSR-I (WSJ0) Complete LDC93S6A," *Web Download. Philadelphia: Linguistic Data Consortium*, 1993.

[21] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.