Speaker-Aware Multi-Task Learning for Automatic Speech Recognition

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

***Abstract*—Overfitting is a commonly met issue in automatic speech recognition and is especially impacting when the amount of training data is limited. In order to address this problem, this article investigates acoustic modeling through Multi-Task Learning, with two speaker-related auxiliary tasks. Multi-Task Learning is a regularization method which aims at improving the network's generalization ability, by training a unique model to solve several different, but related tasks. In this article, two auxiliary tasks are jointly examined. On the one hand, we consider speaker classification as an auxiliary task by training the acoustic model to recognize the speaker, or find the closest one inside the training set. On the other hand, the acoustic model is also trained to extract i-vectors from the standard acoustic features. I-Vectors are efficiently applied in the speaker identification community in order to characterize a speaker and its acoustic environment. The core idea of using these auxiliary tasks is to give the network an additional inter-speaker awareness, and thus, reduce overfitting. We investigate this Multi-Task Learning setup on the TIMIT database, while the acoustic modeling is performed using a Recurrent Neural Network with Long Short-Term Memory cells.**

I. Introduction

Acoustic models using deep learning algorithms are currently showing state-of-the-art results for Automatic Speech Recognition (ASR) [1]. Deep Neural Networks (DNN), through their many level of non-linear transformations, are able to assimilate concepts of higher abstraction level as the number of hidden layers increases. Recently, more complex architectures than the classic fully-connected feed-forward DNNs take advantage of other hidden layers connections configurations to further improve the recognition accuracy. For example, Convolutional Neural Networks (CNN) apply multiple localized patches that share the same connection weights [2]. Another increasingly effective architecture uses Recurrent Neural Networks (RNN) with Long Short-Term Memory (LSTM) cells [3], adding an extra temporal memory to the network.

Nevertheless, these deep learning algorithms tend to suffer from poor generalization. As the amount of training data is limited, the network learns an accurate representation of the training set only. As a result, the network may not generalize well and lead to lower recognition results when encountering unseen data or real life conditions. This commonly met issue in ASR is also referred to as "overfitting".

In this article, we investigate if a single acoustic model trained to solve multiple related tasks can decrease the overfitting issue met by deep learning algorithms. This approach is known as Multi-Task Learning (MTL) in contrast to the usual Single-Task Learning (STL) training [4]. The core concept is to train a single deep neural network to solve in parallel one main task, plus at least one auxiliary task, using the same input features. More specifically here, we use as main task the classic ASR estimation of phoneme-state posterior probabilities, with two auxiliary tasks simultaneously: 1) speaker classification/recognition 2) i-vector extraction. If the network is able to recognize the speaker and extract the i-vector, while performing its main speech recognition task, the network will then have learned (in its internal representations) valuable information about the inter-speaker variability, their environmental characteristics and the underlying link between speaker and speech. A RNN-LSTM deep learning algorithm is used as acoustic model for our study.

This article is organized as follows. Section 2 presents related work. In Section 3, the MTL mechanism is described. Further details concerning the auxiliary tasks are discussed in Section 4. Section 5 introduces the experimental setup and results are shown in Section 6. Finally, we conclude and present future work ideas in Section 7.

II. Related Work

Regularization methods are often essential for the network's convergence. In addition, they aim at reducing overfitting. The MTL setup proposed in this work focuses also on improving generalization, and thus, can be seen in conjunction to other regularization methods.

Stopping the training prematurely is one option, once the accuracy starts to decrease on a validation set, this method being referred to as "early stopping" [5]. Other regularization methods, such as L1 and L2 regularization, add an extra term to the cost function, thus, easing a sparser hidden architecture [6]. It is also possible to randomly set to zero some units activations, this technique known as "dropout" has led to better generalizing systems. Furthermore, limiting the hidden weights of a DNN in an ordered and bio-inspired manner, leading to a sparse DNN, has shown promising results [7].

The drawback of these regularization methods is that they assume that the network's number of parameters in unnecessarily large, and try to reduce it by suppressing units or connections, thus, not getting advantage of the full network's modeling capacity. Moreover, the generalization capacity of the network is constrained by the recognition task. As a result, there should be a training method with one main task (estimating the phoneme-state posterior probabilities commonly used for ASR), and additionally force the network to solve other useful tasks, therefore taking full advantage of all the network's parameters. This training scheme is know as Multi-Task Learning [4].

Lately, MTL applied to DNN, CNN, RNN or RNN-LSTM acoustic models has shown promising results in several speech and language processing areas: speech synthesis [8], [9], speaker verification [10], multilingual speech recognition [11], [12], [13], spoken language understanding [14], [15], natural language processing [16], etc.

Speech recognition does also profit from MTL, through different auxiliary tasks. Gender classification was first tested as an auxiliary task for ASR, by adding two (male-female) [17] or three (male-female-silence) [18] additional output nodes to a RNN acoustic model. Phoneme classification can be used as an additional auxiliary task of the phoneme-state posterior probabilities, thus, indicating to a DNN which phone-state posteriors may be related [19], [20]. Nevertheless, classifying broader phonetic classes (such as plosive, fricative, nasal, ...) does not seem to be an effective auxiliary task for ASR [18]. Other studies investigate graphemes (symbolic representation of writing rather than speech sound), showing that estimating only the current grapheme as auxiliary task is ineffective [18]. However, adding the left and right grapheme's context improves the main ASR task [21]. Estimating the phoneme left and right context is also a efficient auxiliary task [19].

Adapting the acoustic model to a specific speaker can be improved by MTL too [22]. In this case, a STL DNN is trained in a speaker-independent manner. Then, while the major part of the DNN's parameters are fixed, a small number of the network's parameters are updated using MTL. More specifically, phoneme and senone-cluster estimation are tested as auxiliary tasks for speaker adaptation.

Robustness to noise is a commonly met speech recognition problem that some MTL auxiliary tasks try to address. This could be done by generating enhanced speech as an auxiliary task [17], [23], or more recently by recognizing the noise types as auxiliary task [24].

Finally, speaker-aware ASR models using MTL were proposed lately. The acoustic model is given additional speaker information by training the network to also recognize the speakers [25], or by extracting extra features from a similar setup [26]. In the latter study, a first Bottle-Neck (BN) MTL system using a RNN-LTSM acoustic model classifies the speakers as auxiliary task. Then, the BN layer is concatenated to the standard acoustic features and used as input for a second STL RNN-LSTM system. Extracting i-vectors [27] as an auxiliary task has also shown promising results for a speaker-aware training [28]. I-Vectors' ability to discriminate speakers and their associated environment are powerful tools for speaker verification as well as ASR [29].

Additional information on MTL usage for automatic speech recognition can be found in [30].

In this article, we are also interested in adding speaker-awareness to the training process. But instead of using speaker classification or i-vector extraction as separate auxiliary tasks, both auxiliary tasks are simultaneously applied. Our interest is in forcing the network to learn valuable inter-speaker information, through these two speaker-aware auxiliary tasks, leading to better generalization.

III. Multi-Task Learning

Multi-Task Learning was first investigated in 1997 [4]. As discussed earlier, the core idea for MTL consists of training jointly and in parallel one deep learning model on several tasks that are different, but related. As a rule, the network is trained on one main task, plus at least one auxiliary task. The aim of the auxiliary task is to improve the model's convergence, more specifically to the benefit of the main task. An illustration, where the MTL model has one main task and N auxiliary tasks, is presented in Figure 1. Two fundamental characteristics are shared among all MTL systems. First, all tasks are trained using the same input features. Second, all tasks share the same parameters and internal representations. In this setup, the network's parameters are updated by backpropagating the combination of the respective task errors through the hidden layers of the network, defined as:

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

ϵ_{MTL} being the error combination to be minimized, with ϵ_{Main} and $\epsilon_{Auxiliary_n}$ respectively the main and auxiliary tasks errors, λ_n is a nonnegative weight and N the total number of auxiliary tasks. Varying the λ_n value will modify the auxiliary task(s) influence on the backpropagated error, and thus, on the entire system. If λ_n is closer to 1, then the n^{th} auxiliary task will be as impacting as the main task, whereas for λ_n set to 0, the auxiliary task would not have any influence on training. In most cases, the auxiliary tasks are dropped at test time, keeping only the main task outputs. Selecting relevant auxiliary tasks is crucial, as MTL can improve the model's robustness to unseen data, hence, decrease overfitting impact. On the contrary, if the auxiliary task is not relevant for the main task, the convergence could be worse. Smaller datasets can especially benefit from this method, as generalization is a greater issue with lower resources. Rather than processing each task independently, sharing the network's structure among the different tasks leads to higher performance [4].

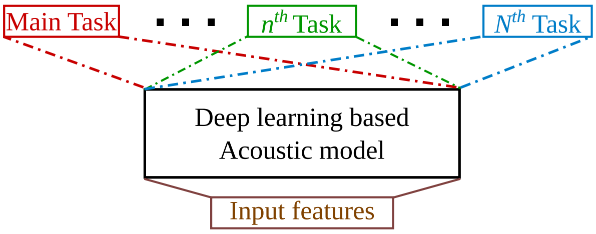

Fig. 1. A Multi-Task Learning network with one main task and N auxiliary tasks.

IV. Auxiliary Tasks

As detailed in Section II, a large and diversified number of auxiliary tasks have been considered for MTL ASR. We propose in this article speaker classification/recognition associated to i-vector extraction as auxiliary tasks.

The primary motivation to use both auxiliary tasks is to draw the networks attention at the correlation between the phone-state posteriors variability and the speakers. Physical (vocal organs, gender, age, ...) as well as non-physical (regional and social affiliation, co-articulation, ...) characteristics lead to inter-speaker variations [31]. Furthermore, if the system is able to differentiate the speaker's characteristics, then, this information can be used for a better interpretation of the distortion brought by one speaker in comparison to another.

A. Speaker Classification

In order to properly apply the MTL setup, we extract from each input example fed the RNN-LSTM the speaker id, and store the information in an auxiliary label vector. The size of this sparse vector is equal to the total number of speakers contained in the training set of the database.

At training time, the network is taught to recognize the speaker, whereas at test time, this speaker may not be present in the training dataset, which is the case in our study. In such case, the network will try to classify the test speakers to the closest existing speakers inside the training set. The more speakers are included in the training dataset, the greater chance there is to find a similar speaker during test time.

Moreover, applying deep learning algorithms for speaker verification has shown encouraging results. Hence, it makes sense to consider this task as an auxiliary task in a MTL setup using a deep learning architecture. For instance, *d-vectors* are extracted by training a STL DNN to recognize speakers with frame level acoustic features [32]. The last layer before the softmax layer is then used for speaker classification by measuring the cosine distance.

B. I-Vectors Extraction

I-Vectors are low-dimensional features able to characterize a speaker and its acoustic environment. They are currently considered as the state-of-the-art in the speaker identification area. I-Vectors propose a smart way to reduce a large-dimensional input to a fixed-size, low-dimensional feature vector, while preserving most of the relevant speaker information. The i-vector extraction method is based on the Joint Factor Analysis framework [33] to define a new low-dimensional space known as the total variability space. A given speech utterance will then be represented in this new space by an i-vector. For a given utterance, the mean super-vector M corresponding to its Gaussian Mixture Model (GMM) can be written as:

$$M = m + Tw \ , \tag{2}$$

where m is the speaker and channel independent super-vector extracted from a Universal Background Model (UBM), T is a low-rank rectangular matrix iteratively estimated over the training corpus known as the total variability matrix, and w is the i-vector. Thanks to this representation, the lower-dimensional vector w can be used as a speaker model, instead of the much larger GMM.

In the MTL setup we are investigating, we use the already estimated i-vectors as targets of this auxiliary task. For this auxiliary task, having different speakers in the training set and test set is not an issue, as the network should be able to extract i-vectors from unseen speakers, which is not the case for the speaker classification auxiliary task, making i-vector extraction a more robust auxiliary task.

V. Experimental Setup

The proposed MTL setup is trained and tested using the free, open-source, speech recognition toolkit Kaldi [34].

A. Database

This MTL approach was investigated on a phone recognition task using the TIMIT Acoustic-Phonetic Continuous Speech Corpus [35].

In order to properly assess this setup, the TIMIT database is divided in three subsets. The standard training set is composed of 462 speakers. A development set of 50 speakers is used to tune the language model weight. Finally, the 24-speaker standard test set is used for evaluation of the model improvement. All speakers are native speakers of American English, from 8 major dialect divisions of the United States, with no clinical speech pathologies. There is no overlapping of the speakers present in one dataset to another, but all 8 dialects can be found in the three datasets. Each of the speakers is reading 10 sentences. Using the the phone label outputs and the supplied phone transcription, we compute and compare the Phone Error Rate (PER) metric.

B. System description

The input acoustic features are 13-dimensional Mel-Frequency Cesptral Coefficients (MFCC) features, which are normalized via Cepstral Mean-Variance Normalization (CMVN). This features are first used before training the ASR system in order to extract 100-dimensional i-vectors, using a 256-component GMM-UBM (through the standard i-vector extraction pipeline of Kaldi). Then, the same MFCC features are processed by a hybrid RNN-LSTM - Hidden Markov Model (HMM) system. The RNN-LSTM generates the phoneme-state posterior probabilities as main task plus the two speaker-aware auxiliary tasks outputs, whereas the HMM deals with the speech's temporal nature.

Random seeds are used for input features shuffling, as well as hidden weights initialization. 40 frames of left context are added to every input. The RNN-LSTM acoustic model is composed of three uni-directional LSTM hidden layers, with 1024 cells per layer and a linear projection of 256 dimensions for each layer [36]. We use sequences of 20 training labels with a delay of 5 labels. The learning-rate decreases from 0.0012 to 0.00012, training is stopped after a maximum of 10 epochs, and 100 feature vectors are processed in parallel in every mini-batch. For the main tasks and the speaker classification auxiliary task, the error is computed using cross entropy. Whereas for the i-vector extraction auxiliary task, we backpropagate the quadratic error as we consider i-vector extraction as a non-linear regression task. Also, a softmax output non-linearity is added for the main task and speaker classification task, but not for the i-vector extraction one. The system is depicted in Figure 2.

During decoding, we use dictionary and language models to establish the most likely transcription. Both auxiliary tasks branches are discarded throughout evaluation, leading to a regular STL system.

We use a RNN-LSTM acoustic model as the auxiliary tasks require access to a wider time window than the phone-state probabilities estimation task. By keeping track of the RNN-LSTM backward connections, we are able to extend the temporal information used for the auxiliary tasks.

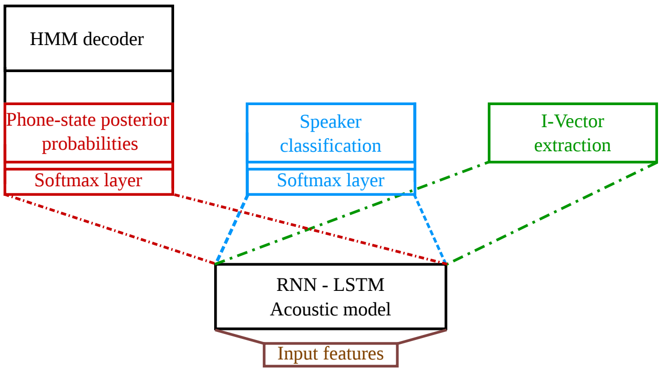

Fig. 2. Illustration of the experimental setup. A RNN-LSTM is trained for three tasks. Phone-state posterior probabilities estimation as main task, plus two auxiliary tasks: speaker classification and i-vector extraction. The estimated posterior probabilities are then fed to a HMM, whereas the auxiliary tasks are discarded during evaluation.

VI. Results

All results presented in this section, were averaged over three runs with random seeds, following Abdel-Hamid et al. work with TIMIT [37].

A. Baseline

A STL RNN-LSTM is first trained to set the baseline. We set the weight coefficients λ to 0 for both auxiliary tasks. This way, the auxiliary tasks do not influence training, and the system is trained in a STL manner, estimating only the phone-state posterior probabilities.

B. Influence of λ coefficients

In order to evaluate the impact of speaker classification associated to i-vector extraction as MTL auxiliary tasks, we variate the two weight coefficients $\lambda_{speaker}$ and $\lambda_{ivector}$, respectively for the speaker classification task and for the i-vector extraction task, with the values presented in Table I.

The evaluated λ coefficients were selected following previous studies applying these auxiliary tasks separately [25], [28]. Using these λ values assured that the three tasks converged and that none of them prevail strongly over the other ones.

TABLE I
Ensemble of the λ values tested, where $\lambda_{speaker}$ refers to the speaker classification auxiliary task and $\lambda_{ivector}$ to the i-vector extraction auxiliary task.

$\lambda_{speaker}$	10^{-3}	10^{-2}	10^{-1}
$\lambda_{ivector}$	10^{-4}	10^{-3}	

C. Results

The obtained results are presented in Table II. Setting $\lambda_{ivector}$ value to 10^{-3} will give worse results than for 10^{-4}, independently of $\lambda_{speaker}$. For this value of $\lambda_{ivector}$, the PER is even higher than with the baseline STL system on the dev set, and slightly better when $\lambda_{speaker}$ is smaller than 10^{-2} on the test set. The results show the importance of a well balanced MTL system between each task.

TABLE II
Impact of combining speaker classification and i-vector extraction as auxiliary tasks for MTL speech recognition.

STL		dev	20.07	
		test	21.70	
	$\lambda_{speaker}$ \ $\lambda_{ivector}$		10^{-4}	10^{-3}
MTL	10^{-3}	dev	**19.27**	20.23
		test	**20.80**	21.57
	10^{-2}	dev	19.70	20.30
		test	20.83	21.60
	10^{-1}	dev	19.97	20.30
		test	21.37	22.07

As Figure 3 outlines it, for a $\lambda_{ivector}$ of 10^{-4}, the PER is significantly reduced in comparison to STL for both the dev set and the test set. The most significant PER decreasing is obtained when $\lambda_{speaker}$ is set to 10^{-3}. Even for a $\lambda_{speaker}$ of 10^{-3} and a $\lambda_{ivector}$ of 10^{-4} both auxiliary tasks were still converging after each iteration. In comparison with STL, the relative improvement on the *dev set* is around 4.0% and 4.2% for the *test set* when $\lambda_{speaker}$ equals 10^{-3} and $\lambda_{ivector}$ equals 10^{-4}, which is as a non-negligible improvement.

D. Individual auxiliary task vs. Combined auxiliary tasks

In previous work, we have investigated both speaker-aware auxiliary tasks for MTL ASR [25], [28], but individually. In Table III we compare the relative improvement[1] brought by speaker classification associated to i-vector extraction as auxiliary tasks in comparison to using only one of these auxiliary tasks.

As discussed in Section IV-A, the speaker classification task could be much more impacted if the speakers present at training time are no longer present at test time. Comparing these auxiliary tasks on a database containing more speakers

[1] The input features used for this comparison are very similar, but not exactly the same. Thus, we compare the relative improvement rather than using directly the associated PERs.

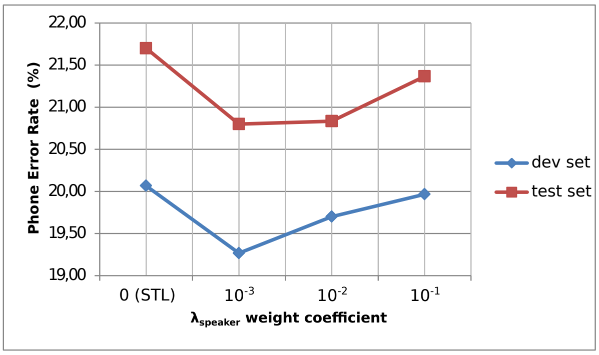

Fig. 3. Phone Error Rate when varying the $\lambda_{speaker}$ weight coefficient. When $\lambda_{speaker}$ is set to 0, $\lambda_{ivector}$ is also set to 0 in order to have a STL training, otherwise $\lambda_{ivector}$ is fixed at 10^{-4}.

may lead to a smaller difference in the relative improvement, as the speaker classification will be more likely to find a closer speaker.

Another explanation could be that, in the speaker verification area, state-of-the-art speaker classification is obtained through i-vector features followed by Probabilistic Linear Discriminant Analysis classification. Thus, asking the network to directly classify the speakers from the the standard acoustic features may be a much more difficult task than using i-vectors as an intermediary.

In this speaker-aware framework, we can see that training simultaneously for both auxiliary tasks is much more helpful for the main task than using individual training the auxiliary tasks.

TABLE III
RELATIVE IMPROVEMENT (%) BROUGHT BY DIFFERENT MTL AUXILIARY TASKS IN COMPARISON TO STL.

MTL auxiliary task	*dev set*	*test set*
Speaker classification alone	0.8	0.3
I-Vectors extraction alone	2.7	3.8
Combining both speaker-aware tasks	**4.0**	**4.2**

VII. CONCLUSION

A novel combination of MTL speaker-aware auxiliary tasks for speech recognition is investigated in this article. A RNN-LSTM acoustic model is trained simultaneously for phone-state posterior probability estimation, speaker classification and i-vector extraction. Generating labels in order to train these tasks is quite easy: there is no further processing required for speaker classification task, whereas the i-vectors used as labels are estimated only one time, just before training. Furthermore, using MTL does not require a significantly important additional amount of computational time as we use the same internal structure for all three tasks. Results show that a non-negligible improvement can be obtained using these auxiliary task jointly.

Future work will focus on investigating other deep learning architectures (CNNs for instance) using this MTL setup. We are also interested in training this setup on databases containing more speakers.

ACKNOWLEDGMENT

This work has been partly funded by the European Regional Development Fund (ERDF) through the DigiSTORM project.

REFERENCES

[1] 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.

[2] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, and G. Penn, "Applying convolutional neural networks concepts to hybrid NN-HMM model for speech recognition," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4277–4280.

[3] O. Vinyals, S. V. Ravuri, and D. Povey, "Revisiting recurrent neural networks for robust ASR," in *Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on*. IEEE, 2012, pp. 4085–4088.

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

[5] L. Prechelt, "Early stopping-but when?" in *Neural Networks: Tricks of the trade*. Springer, 1998, pp. 55–69.

[6] S. J. Nowlan and G. E. Hinton, "Simplifying neural networks by soft weight-sharing," *Neural computation*, vol. 4, no. 4, pp. 473–493, 1992.

[7] G. Pironkov, S. Dupont, and T. Dutoit, "Investigating sparse deep neural networks for speech recognition," in *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*, Dec 2015, pp. 124–129.

[8] Z. Wu, C. Valentini-Botinhao, O. Watts, and S. King, "Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis."

[9] Q. Hu, Z. Wu, K. Richmond, J. Yamagishi, Y. Stylianou, and R. Maia, "Fusion of multiple parameterisations for DNN-based sinusoidal speech synthesis with multi-task learning," in *Proc. Interspeech*, 2015.

[10] N. Chen, Y. Qian, and K. Yu, "Multi-task learning for text-dependent speaker verification," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[11] S. Dupont, C. Ris, O. Deroo, and S. Poitoux, "Feature extraction and acoustic modeling: an approach for improved generalization across languages and accents," in *Automatic Speech Recognition and Understanding, 2005 IEEE Workshop on*. IEEE, 2005, pp. 29–34.

[12] G. Heigold, V. Vanhoucke, A. Senior, P. Nguyen, M. Ranzato, M. Devin, and J. Dean, "Multilingual acoustic models using distributed deep neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 8619–8623.

[13] A. Mohan and R. Rose, "Multi-lingual speech recognition with low-rank multi-task deep neural networks," in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4994–4998.

[14] G. Tur, "Multitask learning for spoken language understanding," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*, vol. 1. IEEE, 2006, pp. I–I.

[15] X. Li, Y.-Y. Wang, and G. Tür, "Multi-task learning for spoken language understanding with shared slots." in *INTERSPEECH*, vol. 20, no. 1, 2011, p. 1.

[16] R. Collobert and J. Weston, "A unified architecture for natural language processing: Deep neural networks with multitask learning," in *Proceedings of the 25th international conference on Machine learning*. ACM, 2008, pp. 160–167.

[17] 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 Tenth Australian International Conference on Speech Science & Technology: 8-10 December 2004; Sydney*, 2004, pp. 312–315.

[18] 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.

[19] M. L. Seltzer and J. Droppo, “Multi-task learning in deep neural networks for improved phoneme recognition,” in *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*. IEEE, 2013, pp. 6965–6969.

[20] P. Bell and S. Renals, “Regularization of context-dependent deep neural networks with context-independent multi-task training,” in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4290–4294.

[21] D. Chen, B. Mak, C.-C. Leung, and S. Sivadas, “Joint acoustic modeling of triphones and trigraphemes by multi-task learning deep neural networks for low-resource speech recognition,” in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 5592–5596.

[22] Z. Huang, J. Li, S. M. Siniscalchi, I.-F. Chen, J. Wu, and C.-H. Lee, “Rapid adaptation for deep neural networks through multi-task learning,” in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[23] 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 *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

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

[25] G. Pironkov, S. Dupont, and T. Dutoit, “Speaker-aware long short-term memory multi-task learning for speech recognition,” in *24th European Signal Processing Conference (EUSIPCO), 2016 (Under Review)*. IEEE, 2016.

[26] T. Tan, Y. Qian, D. Yu, S. Kundu, L. Lu, K. C. SIM, X. Xiao, and Y. Zhang, “Speaker-aware training of LSTM-RNNS for acoustic modelling.”

[27] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, “Front-end factor analysis for speaker verification,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 788–798, 2011.

[28] G. Pironkov, S. Dupont, and T. Dutoit, “I-vector estimation as auxiliary task for multi-task learning based acoustic modeling for automatic speech recognition,” in *INTERSPEECH (Under Review)*, 2016.

[29] A. W. Senior and I. Lopez-Moreno, “Improving dnn speaker independence with i-vector inputs.” in *ICASSP*, 2014, pp. 225–229.

[30] 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.

[31] U. Reubold, J. Harrington, and F. Kleber, “Vocal aging effects on F0 and the first formant: A longitudinal analysis in adult speakers,” *Speech Communication*, vol. 52, no. 7, pp. 638–651, 2010.

[32] E. Variani, X. Lei, E. McDermott, I. Lopez Moreno, and J. Gonzalez-Dominguez, “Deep neural networks for small footprint text-dependent speaker verification,” in *Acoustics, Speech and Signal Processing (ICASSP), 2014 IEEE International Conference on*. IEEE, 2014, pp. 4052–4056.

[33] P. Kenny, G. Boulianne, P. Ouellet, and P. Dumouchel, “Joint factor analysis versus eigenchannels in speaker recognition,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 15, no. 4, pp. 1435–1447, 2007.

[34] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlíček, Y. Qian, P. Schwarz *et al.*, “The kaldi speech recognition toolkit,” 2011.

[35] J. S. Garofolo, L. D. Consortium *et al.*, *TIMIT: acoustic-phonetic continuous speech corpus*. Linguistic Data Consortium, 1993.

[36] H. Sak, A. W. Senior, and F. Beaufays, “Long short-term memory recurrent neural network architectures for large scale acoustic modeling.” in *INTERSPEECH*, 2014, pp. 338–342.

[37] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, “Convolutional neural networks for speech recognition,” *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, no. 10, pp. 1533–1545, 2014.