I-VECTOR ESTIMATION AS AUXILIARY TASK FOR MULTI-TASK LEARNING BASED ACOUSTIC MODELING 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

I-Vectors have been successfully applied in the speaker identification community in order to characterize the speaker and its acoustic environment. Recently, i-vectors have also shown their usefulness in automatic speech recognition, when concatenated to standard acoustic features. Instead of directly feeding the acoustic model with i-vectors, we here investigate a Multi-Task Learning approach, where a neural network is trained to simultaneously recognize the phone-state posterior probabilities and extract i-vectors, using the standard acoustic features. Multi-Task Learning is a regularization method which aims at improving the network's generalization ability, by training a unique network to solve several different, but related tasks. The core idea of using i-vector extraction as an auxiliary task is to give the network an additional inter-speaker awareness, and thus, reduce overfitting. Overfitting is a commonly met issue in speech recognition and is especially impacting when the amount of training data is limited. The proposed setup is trained and tested on the TIMIT database, while the acoustic modeling is performed using a Recurrent Neural Network with Long Short-Term Memory cells.

Index Terms— automatic speech recognition, multi-task learning, LSTM, i-vector, TIMIT

1. INTRODUCTION

Acoustic modeling based on deep learning models is currently showing state-of-the-art results for Automatic Speech Recognition (ASR) [1]. Deep Neural Networks (DNN), through their many levels of non-linearities, 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 configurations of hidden layers connections to further improve the recognition's accuracy. For instance, Convolution Neural Networks (CNN) apply several localized patches that share the same connection weights [2]. Another more and more effective architecture uses Recurrent Neural Networks (RNN) with Long Short-Term Memory (LSTM) cells [3], adding an extra temporal memory to the network.

Quite often though, these deep learning models suffer from poor generalization. As the amount of training data is limited, the network tends to learn an accurate representation of the training set only. As a consequence, when encountering unseen data or real life conditions, the network may not generalize well and lead to lower recognition results. This commonly met issue in ASR, and machine learning more generally, is also referred to as "overfitting".

In this article, we investigate if a single system trained to solve multiple related tasks can decrease the overfitting issue met by deep learning models. 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 learning model 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, whereas the auxiliary task focuses on extracting the associated i-vectors. If the network is able to extract the i-vectors, while performing its main speech recognition task, the network will then learn very valuable information about the inter-speaker variability, their environmental characteristics and the underlying link between speaker and speech. A RNN-LSTM deep learning model 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 task 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.

2. RELATED WORK

Regularization methods are sometimes essential for the network's convergence. Additionally, they aim at reducing overfitting. The MTL method we are investigating in this work focuses also on improving generalization, and thus, can be seen in conjunction to other regularization methods. For instance, it is possible to stop the training earlier, once the ASR accuracy starts to decrease on a validation set [5]. Other regularization methods, such as L1 and L2 regularization, add

a term to the cost function, thus, easing a sparser hidden architecture [6]. It is also possible to randomly set to zero some neuron activations, this technique referred to 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 significant improvement [7].

The drawback of these 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 limited by the recognition task. As a result, there should be a training configuration with one main task (estimating the phoneme-state posterior probabilities commonly used for ASR), and additionally force the network to solve another significant task, therefore taking full advantage of all network's parameters. This training configuration is know as Multi-Task Learning [4].

Lately, MTL applied with 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 kind of auxiliary tasks. Gender classification was primarily 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. Using phoneme classification, as an additional auxiliary task of the phoneme-state posterior probabilities, indicates to a DNN which phone-state posteriors may be related [19, 20]. Nevertheless, using broader phonetic classification (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 unworthy [18]. However, adding the left and right grapheme context improves the main recognition task [21]. Estimating the phoneme context is also a successful auxiliary task [19].

Adapting the acoustic model to a specific speaker can be improved by MTL as well [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 adaptation.

Robustness to noise is a common speech recognition issue 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 type [24].

Finally, speaker-aware ASR models based on MTL were proposed lately. The acoustic model could be given additional speaker information by training the network to recognize the speakers as auxiliary task [25], or by extracting features from a similar setup [26]. In the latter study, a first Bottle-Neck (BN) MTL system using a RNN-LTSM acoustic model applies speaker classification as auxiliary task. Then, the BN layer is concatenated to the standard acoustic features and used as input for a second STL RNN-LSTM.

Additional information on MTL usage for automatic speech recognition can be found in [27].

In this article, we are also interested in adding speaker-awareness to the training process. But instead of using speaker classification, we extract i-vectors [28] as auxiliary task. I-Vectors' ability to discriminate speakers and their associated environment are powerful tools for speaker verification as well as ASR [29]. Our interest is in forcing the network to extract the i-vectors from the standard acoustic features while performing its ASR task, thus learning valuable inter-speaker information, leading to a better generalization.

3. MULTI-TASK LEARNING

Multi-Task Learning emerged 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 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 on the same input features. Second, all tasks share the same parameters and internal representations. The network's parameters are updated by backpropagating the combination of the respective task errors through the hidden layers of the network, defined as:

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

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. 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 near 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. 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].

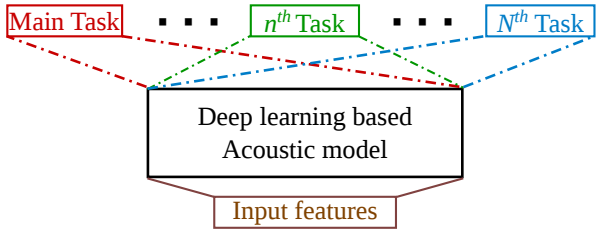

Fig. 1. A Multi-Task Learning network with one main task and N auxiliary tasks.

4. AUXILIARY TASK: I-VECTORS EXTRACTION

I-Vectors are low-dimensional features that are able to characterize a speaker and its acoustic environment. They are know as the state-of-the-art in the speaker identification area. I-Vectors are a smart way to reduce a large-dimensional input to a fixed-size, low-dimensional feature vector, while preserving most of the relevant information. The i-vector extraction method uses the Joint Factor Analysis framework [30] to define a new low-dimensional space referred to as the total variability space. A given speech utterance will then be represented in the 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:

$$M = m + Tw \,, \tag{1}$$

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, rather than the much larger GMM.

In the MTL setup we are investigating we use the already estimated i-vectors that should be concatenated to the the standard acoustic features, as targets of our auxiliary task. The primary motivation here 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 diversity [31, 32]. 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. At training time, the network is taught to extract i-vectors from a limited number of speakers, whereas at test time, this speaker may not be present in the training dataset, which is true in our study. In such case, the network should be able to extract i-vectors from unseen speakers, which is not the case if the auxiliary task was speaker classification, making i-vector extraction a more robust auxiliary task.

5. EXPERIMENTAL SETUP

The proposed MTL setup is trained and tested using the free, open-source, speech recognition toolkit Kaldi [33].

5.1. Database

The MTL approach we propose was investigated on a phone recognition task using the TIMIT Acoustic-Phonetic Continuous Speech Corpus [34].

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.

5.2. System description

The input acoustic features are 13-dimensional Mel-Frequency Cepstral 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 and generates i-vectors as auxiliary task, whereas the HMM deals with the speech's temporal nature.

Random seeds are used for input features shuffling, as well as weight 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 [35]. 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, the error is computed

using cross entropy. Whereas for the auxiliary task, we back-propagate 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 but not for the auxiliary one. The system is depicted in Figure 2.

During decoding, we use dictionary and language models to establish the most likely transcription. The auxiliary task branch is discarded throughout evaluation, leading to a regular STL system.

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

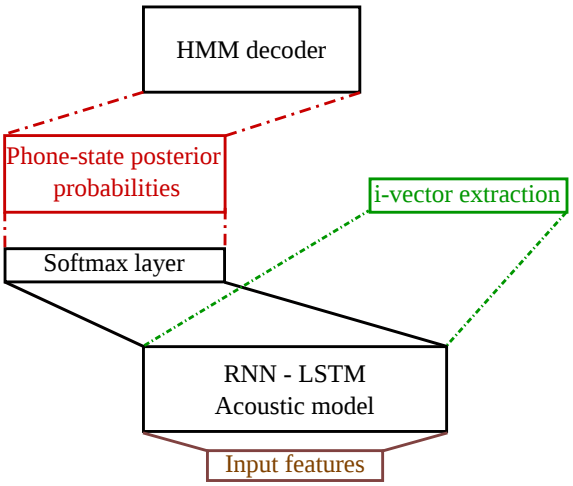

Fig. 2. Illustration of the experimental setup. A RNN-LSTM is trained for two tasks. Phone-state posterior probabilities estimation as main task and i-vector extraction as auxiliary task. The estimated posterior probabilities are then fed to a HMM, whereas the auxiliary task is discarded during evaluation.

6. RESULTS

All results presented in this section, were averaged over three runs with random seeds, following Abdel-Hamid et al. work with TIMIT [36].

6.1. Baseline

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

6.2. Influence of λ coefficients

In order to evaluate the impact of i-vector extraction as a MTL auxiliary task, the weight coefficient λ is set successively to 1.10^{-4}, 5.10^{-4}, 1.10^{-3} and 5.10^{-3}, similarly to Chen et al. study [23]. This may seem very low, but as we are using two different objective functions (cross entropy for the main task and quadratic error for the auxiliary task) the error scale is quite different. Using these λ values assured that both tasks converged and that none of them prevail strongly over the other one.

6.3. Results

The obtained results are presented in Table 1. Using MTL with this auxiliary task improves the PER results in comparison to STL.

As Figure 3 outlines, for a λ of 1.10^{-4}, the PER is significantly reduced in comparison to STL for both the dev set and the test set. However, increasing λ over 5.10^{-3} degrades the results as the main task is no longer benefiting from this auxiliary task. The results are significantly worse with this value of λ showing the importance of a well balanced MTL system between each task. On the other hand, choosing a λ too small would actually lead to neglecting the auxiliary task. In this study, even for a λ of 1.10^{-4} the auxiliary task was still converging after each iteration. The relative improvement on the *dev set* is around 2.7% and 3.8% for *test set* when λ equals 1.10^{-4}, which is as a non-negligible improvement.

Table 1. Impact of i-vector extraction as auxiliary task for MTL speech recognition.

λ coefficient	*dev set* PER (%)	*test set* PER (%)
0 (STL)	20.07	21.70
1.10^{-4}	**19.53**	**20.87**
5.10^{-4}	19.70	20.97
1.10^{-3}	20.10	21.37
5.10^{-3}	27.87	28.80

6.4. I-Vector extraction vs. Speaker classification

In previous work, we investigated another speaker-aware auxiliary task for MTL ASR: speaker classification [25]. In Table 2 we compare the relative improvement[1] brought by i-vector extraction as auxiliary task in comparison to speaker classification. In this speaker-aware frame, we can see that

[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 PER.

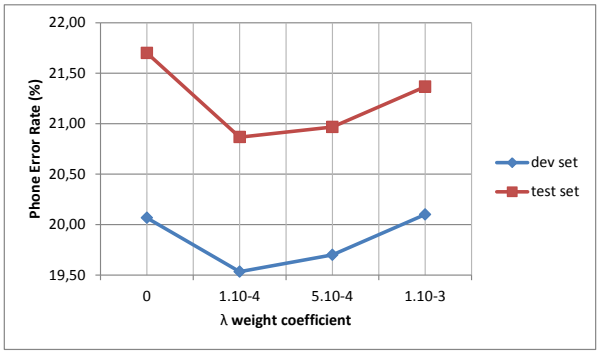

Fig. 3. Phone Error Rate when varying the λ weight coefficient of i-vector extraction as auxiliary task, applied to MTL speech recognition.

the i-vector extraction auxiliary task is much more helpful for the main task than speaker classification. The improvement is even more important on the test set.

As discussed in Section 4, the speaker classification task should 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 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, 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.

Table 2. Relative improvement (%) brought by different MTL auxiliary tasks in comparison to STL.

MTL auxiliary task	*dev set*	*test set*
Speaker classification	0.8	0.3
I-Vectors extraction	**2.7**	**3.8**

7. CONCLUSION

A novel MTL auxiliary task for speech recognition is investigated in this article. A RNN-LSTM acoustic model is trained simultaneously for phone-state posterior probability estimation and i-vector extraction. Using i-vector extraction as an auxiliary task is a quite easy task as it only require to extract one time the i-vectors in order to generate the auxiliary output labels before the training. In comparison to other "speaker-aware" auxiliary tasks, this method is more robust, than speaker classification for instance, as we can apply it to unseen speakers. Furthermore, using MTL does not require a significantly important additional amount of computational time as we use the same internal structure for both tasks. Results show that a small but non-negligible improvement can be obtained using this auxiliary task.

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. Additionally, we will consider combining the i-vector extraction auxiliary task with speaker classification for a MTL system with two speaker-aware auxiliary tasks.

8. ACKNOWLEDGEMENTS

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

9. REFERENCES

[1] Geoffrey Hinton, Li Deng, Dong Yu, George E Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara 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] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, and Gerald 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] Oriol Vinyals, Suman V Ravuri, and Daniel 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] Rich Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

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

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

[7] Gueorgui Pironkov, Stephane Dupont, and Thierry 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] Zhizheng Wu, Cassia Valentini-Botinhao, Oliver Watts, and Simon King, "Deep neural networks employing multi-task learning and stacked bottleneck features for speech synthesis," .

[9] Qiong Hu, Zhizheng Wu, Korin Richmond, Junichi Yamagishi, Yannis Stylianou, and Ranniery Maia, "Fusion of multiple parameterisations for DNN-based sinusoidal speech synthesis with multi-task learning," in *Proc. Interspeech*, 2015.

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

[11] Stéphane Dupont, Christophe Ris, Olivier Deroo, and Sébastien 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] Georg Heigold, Vincent Vanhoucke, Alan Senior, Patrick Nguyen, Marc'Aurelio Ranzato, Matthieu Devin, and Jeffrey 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] Aanchan Mohan and Richard 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] Gokhan Tur, "Multitask learning for spoken language understanding," in *Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings. 2006 IEEE International Conference on*. IEEE, 2006, vol. 1, pp. I–I.

[15] Xiao Li, Ye-Yi Wang, and Gökhan Tür, "Multi-task learning for spoken language understanding with shared slots.," in *INTERSPEECH*, 2011, vol. 20, p. 1.

[16] Ronan Collobert and Jason 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] Youyi Lu, Fei Lu, Siddharth Sehgal, Swati Gupta, Jingsheng Du, Chee Hong Tham, Phil Green, and Vincent 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] Jan Stadermann, Wolfram Koska, and Gerhard Rigoll, "Multi-task learning strategies for a recurrent neural net in a hybrid tied-posteriors acoustic model.," in *INTERSPEECH*, 2005, pp. 2993–2996.

[19] Michael L Seltzer and Jasha 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] Peter Bell and Steve 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] Dongpeng Chen, Brian Mak, Cheung-Chi Leung, and Sunil 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] Zhen Huang, Jinyu Li, Sabato Marco Siniscalchi, I-Fan Chen, Ji Wu, and Chin-Hui Lee, "Rapid adaptation for deep neural networks through multi-task learning," in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[23] Zhuo Chen, Shinji Watanabe, Hakan Erdogan, and John 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] Suyoun Kim, Bhiksha Raj, and Ian Lane, "Environmental noise embeddings for robust speech recognition," *arXiv preprint arXiv:1601.02553*, 2016.

[25] Gueorgui Pironkov, Stephane Dupont, and Thierry Dutoit, "Speaker-aware long short-term memory multi-task learning for speech recognition," in *The 24th European Signal Processing Conference (EUSIPCO)*. IEEE, 2016.

[26] Tian Tan, Yanmin Qian, Dong Yu, Souvik Kundu, Liang Lu, Khe Chai SIM, Xiong Xiao, and Yu Zhang, "Speaker-aware training of lstm-rnns for acoustic modelling," .

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

[28] Najim Dehak, Patrick Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 19, no. 4, pp. 788–798, 2011.

[29] Andrew W Senior and Ignacio Lopez-Moreno, "Improving dnn speaker independence with i-vector inputs.," in *ICASSP*, 2014, pp. 225–229.

[30] Patrick Kenny, Gilles Boulianne, Pierre Ouellet, and Pierre 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.

[31] Ulrich Reubold, Jonathan Harrington, and Felicitas 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] Mohamed Benzeghiba, Renato De Mori, Olivier Deroo, Stephane Dupont, Teodora Erbes, Denis Jouvet, Luciano Fissore, Pietro Laface, Alfred Mertins, Christophe Ris, et al., "Automatic speech recognition and speech variability: A review," *Speech Communication*, vol. 49, no. 10, pp. 763–786, 2007.

[33] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukáš Burget, Ondřej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlíček, Yanmin Qian, Petr Schwarz, et al., "The kaldi speech recognition toolkit," 2011.

[34] John S Garofolo, Linguistic Data Consortium, et al., *TIMIT: acoustic-phonetic continuous speech corpus*, Linguistic Data Consortium, 1993.

[35] Hasim Sak, Andrew W Senior, and Françoise Beaufays, "Long short-term memory recurrent neural network architectures for large scale acoustic modeling.," in *INTERSPEECH*, 2014, pp. 338–342.

[36] Ossama Abdel-Hamid, Abdel-rahman Mohamed, Hui Jiang, Li Deng, Gerald Penn, and Dong Yu, "Convolutional neural networks for speech recognition," *Audio, Speech, and Language Processing, IEEE/ACM Transactions on*, vol. 22, no. 10, pp. 1533–1545, 2014.