Speaker-Aware Long Short-Term Memory Multi-Task Learning for Speech Recognition

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

***Abstract*—In order to address the commonly met issue of overfitting in speech recognition, this article investigates Multi-Task Learning, when the auxiliary task focuses on speaker classification. Overfitting occurs when the amount of training data is limited, leading to an over-sensible acoustic model. Multi-Task Learning is a method, among many other regularization methods, which decreases the overfitting impact by forcing the acoustic model to train jointly for multiple different, but related, tasks. In this paper, we consider speaker classification as an auxiliary task in order to improve the generalization abilities of the acoustic model, by training the model to recognize the speaker, or find the closest one inside the training set. 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

Deep neural networks (DNN), through their ability to assimilate higher levels of abstract concepts using their multiple levels of non-linear nodes, have made deep learning algorithms the best performing modeling techniques for Automatic Speech Recognition (ASR) [1]. Despite the effectiveness of classic fully-connected feed-forward DNNs, more complex methods, profiting from diverse hidden-connections architectures, have led to higher recognition accuracies. Convolutional Neural Networks (CNN) [2] or Recurrent Neural Networks (RNN) using Long Short-Term Memory (LSTM) [3] cells can be mentioned among them. Shared connection weights are applied to different localized patches for CNN, whereas RNN-LSTM contain backward connections, thereby adding a temporal memory.

However, deep learning algorithms sometimes suffer from bad generalization. This issue, commonly known as "overfitting", occurs when the amount of available training data is limited. Thus, the network learns an accurate representation for the training set only. As a result, the learned representation does not necessarily generalize well to unseen datasets and real life conditions.

In this paper, we propose to study whether the overfitting problem can be addressed, by training a single system to solve multiple different tasks. In contrast with the common Single Task Learning (STL) training, this schema is know as Multi-Task Learning (MTL) [4]. The core concept is to train a single deep learning architecture to solve in parallel one main task, plus at least one auxiliary task. In this article, the main task is the usual estimation of phoneme-state posterior probabilities used for ASR, whereas the auxiliary task focuses on recognizing/classifying the speaker. By forcing the network to recognize the speaker, it gains additional contextual information and learns hidden representations characterizing long-term properties of the voice timber. We expect this to improve the system's ability to decode speech. Training is performed using a RNN-LSTM deep learning algorithm.

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.

II. Related Work

In order to improve generalization, several regularization methods can be applied[1]. Early stopping, for instance, involves stopping the training as soon as the recognition accuracy decreases on a validation set [5]. Adding to the cost function a term, that facilitates a sparser internal architecture (a.k.a. L1 and L2 regularization), has led to better generalizing systems [6]. Recently, promising results are obtained by randomly dropping units during training (dropout), leading to a thinned neural network, but only during training [7]. Additionally, some hybrid approaches investigate sparse DNNs by limiting the connections inside the neural network in a bio-inspired, ordered manner [8].

Many of these methods assume that the network is unnecessarily deep and/or wide, and try to reduce the network so each of its units and weights carry a determining information, rather than maximizing the whole network potential provided by its modeling capacity. Moreover, the network's ability to generalize is limited by the recognition task. This bring us to the intuition that if the network is asked to learn some significant information, aside of the phoneme-state posterior probabilities commonly used for ASR, overfitting could be lowered while taking advantage of all of the network's parameters. This is also the main motivation for MTL [4].

Lately, MTL applied to DNN / CNN / RNN or RNN-LSTM acoustic models has shown promising results in several areas of speech and language processing: speech synthesis [9], [10], speaker verification [11], multilingual speech recognition [12],

[1]Beside reducing overfitting, regularization methods are sometimes crucial for the network's convergence. Here, we focus on the generalization contribution.

[13], [14], spoken language understanding [15], [16], natural language processing [17], etc.

Here, our interest focuses on the ASR area, in which several different auxiliary tasks have proven their usefulness. Gender classification was primarily considered as an auxiliary task for ASR, by adding two (male/female) [18] or three (male/female/silence) [19] 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 state posteriors may be related [20], [21]. Nevertheless, using broader phonetic classification (plosive, fricative, nasal, ...) is an ineffective auxiliary task for ASR [19]. Other studies investigate graphemes (symbolic representation of phonemes), showing that estimating only the current grapheme as auxiliary task is unworthy [19]. However, adding the left and right grapheme context improves the main recognition task [22]. Estimating the phoneme context is also a successful auxiliary task [20]. Furthermore, adapting the acoustic model to a specific speaker can be improved by MTL using broader unites as well [23]. 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 [18], [24], or more recently by recognizing the noise type [25].

Additional information on MTL usage for automatic speech recognition can be found in [26].

We propose to use speaker classification as the auxiliary task. This task can be seen as an extension of the gender classification auxiliary task, as we use speaker related information for the MTL. Our interest is in teaching the network that the variations of the phoneme-state acoustic features are due to the numerous speakers (and their very personal characteristics), hence, reducing overfitting.

III. Multi-Task Learning

Studies discussing Multi-Task Learning emerged in 1997 [4]. As stated earlier, the primary 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 or more auxiliary tasks. 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 parameters and internal representations. The network's parameters are updated by backpropagating the combination of the tasks errors through the hidden layers of the network, with a term:

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

ϵ_{MTL} being the error combination to be minimized, with ϵ_{Main} and $\epsilon_{Auxiliary_n}$ respectively the main and auxiliary distinctive 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, decreasing 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 performances [4].

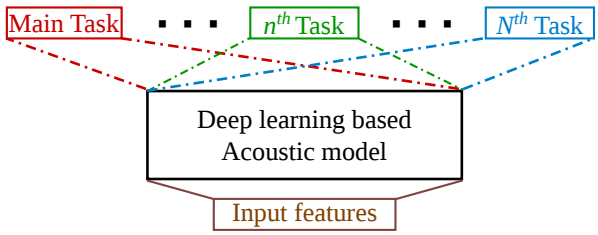

Fig. 1. A Multi-Task Learning network with one main task and N auxiliary tasks.

IV. Auxiliary Task: Speaker Classification

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 as the auxiliary task.

In order to properly apply MTL, we extract from each input example of the RNN-LSTM the speaker id, and store the information in an auxiliary output vector. The size of this sparse vector is equal to the total number of speakers plus one more class. The additional class will be selected if the training input corresponds to a non-speech segment, in other words, if the speaker is silent at that moment. An example with N speakers is depicted in Figure 2.

The primary motivation 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 [27]. Furthermore, if the system is able to recognize the speaker, 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 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 speaker to the closest existing speaker inside the training set. The more speakers are included in the training dataset, the greater the chance there is to find a similar speaker during test time.

Moreover, applying deep learning algorithms for speaker verification has shown encouraging results. For instance, *d-vectors* are extracted by training a STL DNN to recognize speakers with frame level acoustic features [28]. The last layer before the softmax layer is used for speaker classification by measuring the cosine distance.

V. Experimental Setup

The training and testing of the MTL setup are done on the free, open-source, speech recognition toolkit Kaldi [29].

A. Database

The MTL approach we propose was investigated on a phone recognition task using the TIMIT Acoustic-Phonetic Continuous Speech Corpus [30].

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 first, for fine-tunning the hyper-parameters, and second to perform early stopping. 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 phone label outputs and the supplied phone transcription, we compute and compare the Phone Error Rate (PER) metric.

B. Input features

Kaldi's usual feature extraction pipeline is used. First, 13-dimensional Mel-Frequency Cepstral Coefficients (MFCC) features are extracted, and normalized via Cepstral Mean-Variance Normalization (CMVN). The neighboring $\pm$ 3 frames are spliced for each frame. The concatenate features dimensionality is reduced by projecting the features into a 40-dimension feature space using Linear Discriminative Analysis (LDA) transformation. The final features are obtained through feature-space Maximum Likelihood Linear Regression (fMLLR), a feature-space speaker adaptation method.

The fMLLR features are particularly suitable for our auxiliary task, as they normalize inter-speaker variability.

C. System description

The input fMLLR 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 classifies the speakers as secondary task, whereas the HMM deals with the speech's temporal nature. The system is depicted in Figure 2.

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. We use sequences of 20 training labels with a delay of 5 frames. The learning-rate decreases from 0.0012 to 0.00012, training is stopped after a maximum of 15 iterations, and 100 feature vectors are processed in parallel in every mini-batch. For both tasks, the error is computed using cross entropy.

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, speaker recognition, requires a wider time window than the main ASR task. By keeping track of the RNN-LSTM backward connections, we are able to extend the information used for the auxiliary task throughout time.

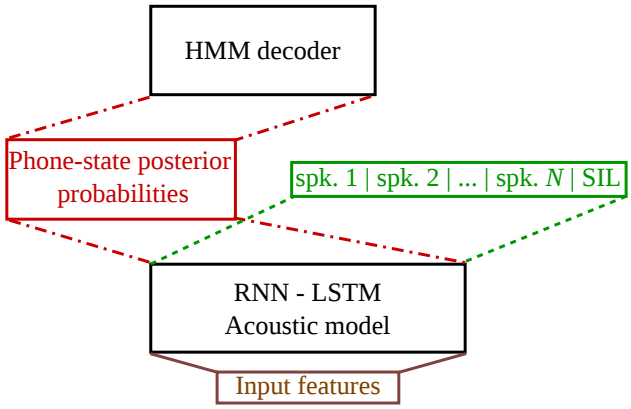

Fig. 2. Illustration of the experimental setup. A RNN-LSTM is trained for two tasks. Phone-state posterior probabilities estimation as main task and speaker recognition as auxiliary task. The estimated posterior probabilities are then fed to a HMM, whereas the auxiliary task is discarded during evaluation. There is on additional "SIL" class used if the speaker is silent at that moment.

VI. Results

All results presented in this section, were averaged over three runs with random seeds, following Abdel-Hamid et al. work on TIMIT [31].

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.

Influence of λ coefficients

In order to evaluate the impact of speaker classification as a MTL auxiliary task, the weight coefficient λ is set successively to 10^{-3}, 10^{-2} and 10^{-1}, as in Chen et al. study [24].

Results

The obtained results are presented in Table I. There is a small but existing improvement brought by MTL. As Figure 3 outlines it, for λ of 10^{-3} the PER is reduced in comparison to STL for both the dev set and the test set. However, increasing λ over 10^{-2} degrades the results as the main task is no longer benefiting from this auxiliary task. The relative improvement for both tasks is around 1.4% when λ equals 10^{-3}, which is as a rather small but non-negligible improvement.

Having only 462 speakers in the training can explain the small improvement brought by this auxiliary task, as it makes it harder for unknown speakers to be classified. Using a database containing more speakers in the training set could improve the PER.

TABLE I
IMPACT OF SPEAKER CLASSIFICATION AS AUXILIARY TASK FOR MTL SPEECH RECOGNITION.

λ coefficient	*dev set* PER (%)	*test set* PER (%)
0 (STL)	18.43	19.93
10^{-3}	**18.17**	**19.67**
10^{-2}	18.60	19.87
10^{-1}	19.00	20.10

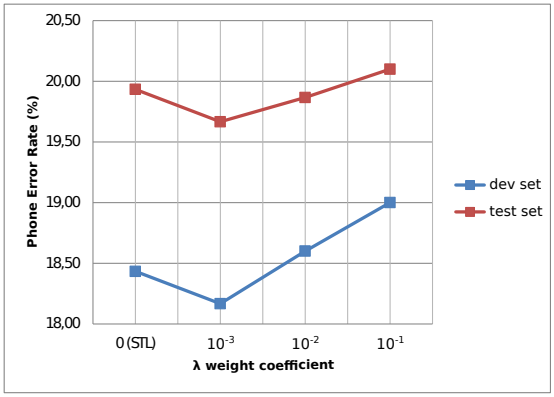

Fig. 3. Phone Error Rate when varying the λ weight coefficient of speaker classification as auxiliary task, applied to MTL speech recognition.

fMLLR features

As explained earlier, our MTL system is trained using fMLLR features, a feature-space speaker adaptation method. Hence, an important question would be to know if this kind of feature preprocessing is decisive for our MTL setup, as we also train the system for speaker awareness through the auxiliary task.

In Figure 4, we compare STL and MTL (λ set at 10^{-3}) with and without fMLLR transformed features. First of all, we can see that the overall PER is better with fMLLR features, with more than 1.5% absolute improvement for both STL and MTL, on both dev et test sets. Second, the improvement brought by MTL, compared to STL, is higher when the fMLLR transform is applied (on average 0.6% relative improvement without fMLLR, versus 1.4% with fMLLR). Thus, there is a compound effect when using both fMLLR transformed features and speaker classification as auxiliary task.

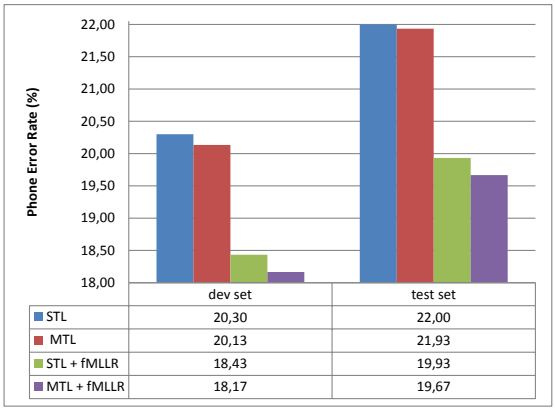

Fig. 4. fMMLR transformation impact on STL and MTL (λ=10^{-3}).

VII. CONCLUSION

In this article we propose a novel MTL auxiliary task for speech recognition. While training a RNN-LSTM for phone-state posterior probability estimation, the network is trained to classify the speakers. Using speaker classification as an auxiliary task is easy as it does not require further processing to generate the auxiliary output labels. It is also simple to reproduce this MTL setup on different databases. 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 generating i-vector as another speaker-aware auxiliary task for ASR.

ACKNOWLEDGMENT

The authors would like to thank Joachim Fainberg and Daniel Povey for the valuable advice on MTL with Kaldi. 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] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, and R. Salakhutdinov, "Dropout: A simple way to prevent neural networks from overfitting," *The Journal of Machine Learning Research*, vol. 15, no. 1, pp. 1929–1958, 2014.

[8] 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.

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

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] 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.

[16] 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.

[17] 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.

[18] 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.

[19] 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.

[20] 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.

[21] 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.

[22] 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.

[23] 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.

[24] 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.

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

[26] 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.

[27] 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.

[28] 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.

[29] 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.

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

[31] 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.