INTERSPEECH 2019
September 15–19, 2019, Graz, Austria

On the Usage of Phonetic Information for Text-independent Speaker Embedding Extraction

Shuai Wang[1,2], *Johan Rohdin*[2], *Lukáš Burget*[2], *Oldřich Plchot*[2], *Yanmin Qian*[1], *Kai Yu*[1], *Jan "Honza" Černocký*[2]

[1]MoE Key Lab of Artificial Intelligence
SpeechLab, Department of Computer Science and Engineering
Shanghai Jiao Tong University, Shanghai, China
[2]Brno University of Technology, Speech@FIT and IT4I Center of Excellence, Brno, Czechia
feixiang121976@sjtu.edu.cn, rohdin@fit.vutbr.cz

Abstract

Embeddings extracted by deep neural networks have become the state-of-the-art utterance representation in speaker recognition systems. It has recently been shown that incorporating frame-level phonetic information in the embedding extractor can improve the speaker recognition performance. On the other hand, in the final embedding, phonetic information is just an additional source of session variability which may be harmful to the text-independent speaker recognition task. This suggests that at the embedding level phonetic information should be suppressed rather than encouraged. To verify this hypothesis, we perform several experiments that encourage or/and suppress phonetic information at various stages in the network. Our experiments confirm that multitask learning is beneficial if it is applied at the frame-level stage of the network, whereas adversarial training is beneficial if it is used at the segment-level stage of the network. Additionally, the combination of these two approaches improves the performance further, resulting in an equal error rate of 3.17% on the VoxCeleb dataset.

Index Terms: phonetic information, text-independent speaker verification, adversarial training

1. Introduction

Speaker recognition aims to verify a subject's identity via his or her speech. Considering the speech content, speaker recognition can be classified into two categories, text-dependent and text-independent. The former demands that in the positive verification trial, the enrollment and test phrases are identical, while the latter doesn't pose such requirement.

In the last two years, the state-of-the-art in the field of the text independent speaker verification has shifted from the generative *i*-vector[1] paradigm that is a subspace factor analysis model operating in the high dimensional space of Gaussian mean super-vectors towards neural networks (NN) that take the frame-level features of an utterance as input and directly produce an utterance level representation — usually referred to as an *embedding* [2, 3, 4, 5, 6, 7, 8]. These embeddings are obtained by the means of *pooling mechanism*, for example taking the mean, over the frame-wise outputs of one or more layers in the NN [2], or by the use of a recurrent NN [3]. An effective approach is to train the NN for classifying a set of training speakers, i.e., using multiclass training [3, 5]. In order to do speaker verification, the embeddings are extracted and used in a standard backend, e.g., probabilistic linear discriminant analysis (PLDA), instead of i-vectors. Such systems have recently been proven superior to i-vectors for both short and long utterance duration in text-independent speaker verification [5, 7].

It's intuitive that phonetic information should be helpful for the text-dependent speaker recognition. It is shown in [9] that multitask learning with phonetic information can benefit the *d*-vector[2] systems that are based on framewise DNN. For the text-independent speaker recognition, many researchers also investigated integrating the phonetic information into the speaker modeling process. Authors in [10, 11] proposed to use posteriors obtained via a phonetically-aware DNN to compute Baum-Welch statistics for *i*-vector extraction, which boosts the performance of *i*-vector systems. More recent work in [12] shows that for the segment-level trained *x*-vector system[7], which is now the state-of-the-art SV framework, it's still quite effective to add phonetic information via frame-level multitask training before the statistics pooling layer.

However, intuitively, for the text-independent speaker recognition tasks, it's doubted that the spoken content should matter for the speaker embeddings, since we don't assume anything about it when processing the enrollment and test speech segments. We hypothesize that, *frame-level phonetic information is helpful for generic feature learning before pooling, while the final speaker embeddings don't need to encode information related to the phonetic content in order to serve as good features for the text-independent SV backend.*

To verify such hypothesis, we analyze several DNN architectures based on *x*-vectors [7] which, apart from segment-level discrimination between speakers, include multitask [12] or adversarial training [13, 14] at the frame or segment level to explicitly encourage or suppress phonetic information. We show that frame-level multitask learning placed before the pooling layer adds fine-grained phonetic information which still benefits the quality of resulting embeddings and SV performance, whereas the segment-level adversarial training performed after pooling suppresses the overall aggregated phonetic information and also improves the system performance. Experiments are carried out on the Voxceleb1 [15] evaluation set, and we achieve an equal error rate (EER) of 3.17% by combining the frame-level multitask and segment-level adversarial strategies.

2. Related work

Phonetic content and speaker identity are the two most important information encoded in the speech signal, speech recognition aims to recognize the phonetic content (more precisely

The work is done during Shuai's internship at BUT speech group.

http://dx.doi.org/10.21437/Interspeech.2019-3036

the words), while speaker recognition aims to recognize the speaker identity. In speech recognition systems, speaker adaptation techniques are usually adopted to utilize speaker-specific information[16, 17]. Similarly, the usage of phonetic information for speaker recognition tasks have also been investigated. By substituting the GMM with a phonetic-aware DNN, the performance of the *i*-vector system is improved significantly in [10, 11]. For the neural network based speaker embedding learning, phonetic information is often considered in a multitask learning framework. The speaker discriminative and phoneme discriminative networks are trained jointly, with some common layers shared[9]. In this paper, we will adopt the TDNN based multitask architecture in [12].

2.1. Speaker embeddings learning with frame-level phonetic information

The architecture in [12] is based on the *x*-vector system[7], the whole system is depicted in Figure 1.

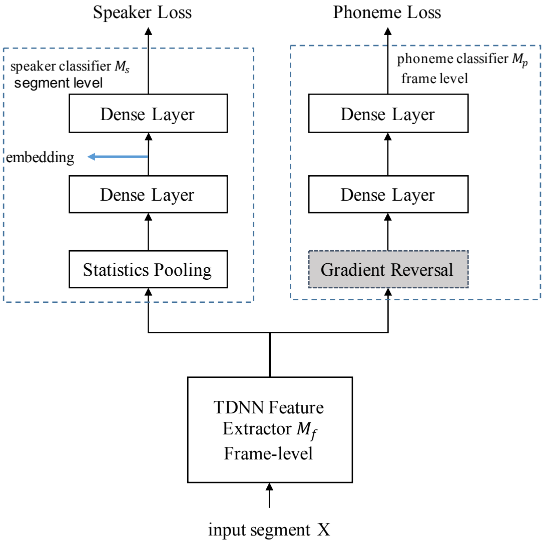

Figure 1: *Structure of the frame-level multitask (without the gradient reversal layer) and adversarial learning for x-vector*

As shown in Figure 1, the neural network consists of three modules, the speaker classifier M_s which is trained at the segment level, the phoneme classifier M_p at the frame level, and the shared time-delay neural network (TDNN) based feature extractor M_f at the frame level. For an input segment of N frames $\mathbf{X} = \mathbf{x}_1, \ldots, \mathbf{x}_N$, the total loss is composed of the speaker loss $\mathcal{L}_s$ and phoneme loss $\mathcal{L}_p$ as $\mathcal{L}_{total} = \mathcal{L}_s + \mathcal{L}_p$, where

$$\mathcal{L}_s = \text{CE}(M_s(M_f(\mathbf{X})), \mathbf{y}^s) \tag{1}$$

$$\mathcal{L}_p = \frac{1}{N}\sum_{i=1}^{N} \text{CE}(M_p(M_f(\mathbf{x}_i)), \mathbf{y}_i^p) \tag{2}$$

where CE(P, Q) denotes the cross entropy loss computed between the two distributions P and Q. $\mathbf{y}^s$ denotes the segment-level speaker labels and $\mathbf{y}^p$ denotes the frame-level phoneme label. To do the opposite of multitask learning, we added a gradient reversal layer (GRL) to the branch of phoneme classifier, aiming to suppress the effect of phonetic information. It's very similar to the speaker invariant training (SIT) proposed in [13], which instead adds the GRL to the speaker branch.

3. Segment-level multitask and adversarial learning with phonetic information

The multitask and adversarial learning framework introduced in Section 2 is quite intuitive. The fine-grained phonetic information of each frame is explicitly encouraged or suppressed. To investigate utilization of phonetic information at the segment level, making the speaker classifier, M_f, and phoneme classifier, M_p, operating at the same granularity, we designed the architecture shown in Figure 2.

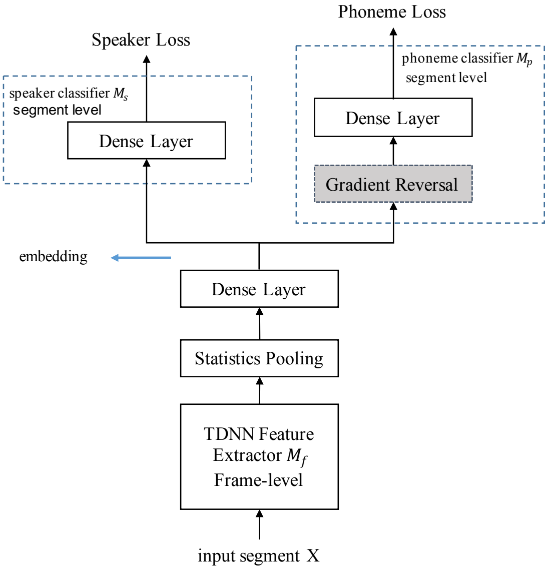

Figure 2: *Structure of the segment-level multitask (without the gradient reversal layer) and adversarial learning for x-vector.*

Here, the *phoneme classifier* should predict the phonetic content in the whole segment. For this, we need to define what the phonetic content of a segment is, or in other words, what are the reference values the speaker classifier should predict. In this work, we simply define the phonetic content of a segment as normalized categorical occurrences of the phonemes. That is, the target values $\mathbf{y}^p = y_1, \ldots, y_C$ for the segment $\mathbf{X}$ with N frames will be computed as $y_c = \frac{N_c}{N}$, where C denotes the total number of phonemes presented in the segment, N_c denotes the frame count of c-th phoneme present in the segment $\mathbf{X}$. Training objective is cross-entropy, as for the frame-wise training.

$$\mathcal{L}_p = \text{CE}(M_p(M_f(\mathbf{x}_i)), \mathbf{y}^p) \tag{3}$$

Note that contrary to the frame-wise training, there is only one instance per segment to evaluate and the reference vector $\mathbf{y}^p$ is not a one-hot vector but may have non-zero values for all phonemes.

4. Experiments

4.1. Experimental set-up

4.1.1. Dataset

Voxceleb1[15] training and Voxceleb2[18] development set are combined to generate the training set for speaker embedding extractor, the data augmentation procedure described in [7] is adopted to increase the amount and diversity of the training data. The final training set contains 2081192 utterances from

7146 speakers, while the standard test set defined in [15] is used for evaluation, which contains 4874 utterances from 40 speakers, the trial list contains 37720 pairs. The training utterances are cut into short segments ranging from 2s - 4s. As [15], we report results in Equal error rate (EER) and minimum detection cost for $\mathrm{Ptar} = 0.01$ ($\mathrm{minDCF}_{0.01}$). The systems evaluated here have, however, much better performance than the ones in [15]. As a result, $\mathrm{minDCF}_{0.01}$ cannot be reliably estimated with the relatively few non-target trials available in the test set (please see Figure 5 and its caption). We therefore report results also on $\mathrm{minDCF}_{0.1}$.

4.1.2. The phoneme recognizer

The frame-level phoneme labels are generated using the official Kaldi [19] Tedlium speech recognition recipe (s5_r3). This recipe uses a TDNN based acoustic model with i-vector adaptation and a RNN based language model. Phoneme posteriors are obtained from the lattices via the forward-backward algorithm and then converted to hard labels. There are 39 phonemes, each coming in four different versions depending on their position in the word, plus a silence (SIL) and noise class (NSN) that has 5 versions each, resulting in 166 *phoneme classes*.

4.1.3. System description

The baseline system is a standard TDNN based *x*-vector, which contains 5 time delay layers and two dense layers. Embeddings are extracted after the first dense layer with a dimension of 512. All the proposed architectures described in Section 2 and 3 are modified from the baseline system. All architectures are implemented in Pytorch[20].

30-dimensional MFCC features are used for the neural network training, an energy-based voice activity detector is used to filter out the silence in the original speech signal.

The extracted embeddings are first processed by linear discriminant analysis (LDA), reducing the dimension from 512 to 128, then a standard PLDA [21] is used to generate the scores.

4.2. Results

4.2.1. Integrating the frame-level phonetic information

As described in Section 2, it's intuitive to jointly train the *x*-vector speaker embedding extractor and a phoneme recognizer by sharing the frame-level layers before the pooling layer. Similar to the findings in [12], integrating frame-level phonetic information could enhance the system's performance, decreasing the EER from 3.73% to 3.38%. However, naively performing the adversarial training at the frame-level with a gradient reversal layer causes significant performance degradation, obtaining an EER of 5.24%.

Table 1: *Systems combining frame-level phonetic information, FRM-MT and FRM-ADV denote two systems described in Section 2, trained using multitask or adversarial objectives, with or without the gradient reversal layer, respectively*

System	EER(%)	$\mathrm{minDCF}_{0.01}$	$\mathrm{minDCF}_{0.1}$
x-vector baseline	3.73	0.389	0.192
FRM-MT	3.38	0.357	0.180
FRM-ADV	5.24	0.502	0.269

To illustrate the effects of the multitask and adversarial learning, the frame-level phoneme accuracy is plotted for both systems in Figure 3. It could be observed that while the phoneme accuracy for the FRM-MT system gradually increases and converges to 48.5%, the corresponding accuracy for the FRM-ADV system quickly converges to around 12% in the first several iterations and barely changes in the following training process. Such phenomenon shows that the adversarial learning forbids the network to become good at discriminating phonemes. However, the gap in accuracy between the FRM-MT FRM-ADV system is not as huge as expected, which might be caused by (1) The data for training the phoneme recognizer and the speaker embedding extractor has mismatch, the generated labels are not always correct. (2) Position-dependent phonemes are used in this work, the label of the same phoneme changes with its position in the word, which makes it even harder to discriminate phonemes perfectly. The results show that in the early stages of the neural network, before the statistics pooling layer, adding fine-grained frame-level phonetic information is beneficial for deep speaker embedding learning.

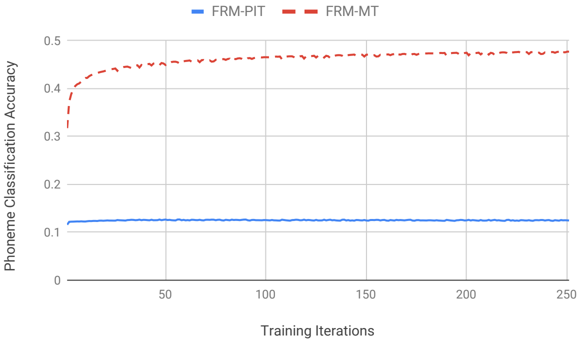

Figure 3: *Prediction accuracy on phonemes using the frame-level multitask or adversarial learning*

4.2.2. Integrating the segment-level phonetic information

As proposed in Section 3, the multitask and adversarial learning could also be performed at the segment level, i.e, after the pooling layer. The results are shown in Table 2. As expected, The SEG-ADV outperforms the baseline in all metrics. More noticeably, The SEG-MT system performs better than the baseline in the Detection costs. This may seem to contradict the hypothesis that phoneme information at the segment level should be harmful to the final embedding. A likely explanation for this is that encouraging phoneme level at the segment-level stage of the network, implicitly encourages phoneme information at the earlier frame-level stages because phoneme information cannot exist at the segment level if it did not exist at the frame level. The advantage of having phoneme information at the frame-level might then be stronger than the disadvantage of having it at the segment level. This needs to be analysed more in future work.

From Figure 4, it could be observed that the phoneme loss for the SEG-MT system gradually decreases, which is not the case for the SEG-ADV system. The segment-level adversarial could remove some global phoneme information as we expected. However, the difference between losses of the SEG-ADV and SEG-MT system is not that large, similar to the possible reasons for the frame-level systems, phoneme labels may not always be correct. Another reason for this phenomenon is the simple normalized categorical counts may not be capable

Table 2: *Systems combining segment-level phonetic information, SEG-MT and SEG-ADV denote two systems described in Section 3, trained using multitask or adversarial objectives, with or without the gradient reversal layer, respectively*

System	EER(%)	minDCF$_{0.01}$	minDCF$_{0.1}$
x-vector baseline	3.73	0.389	0.192
SEG-MT	3.71	0.327	0.175
SEG-ADV	3.35	0.332	0.159

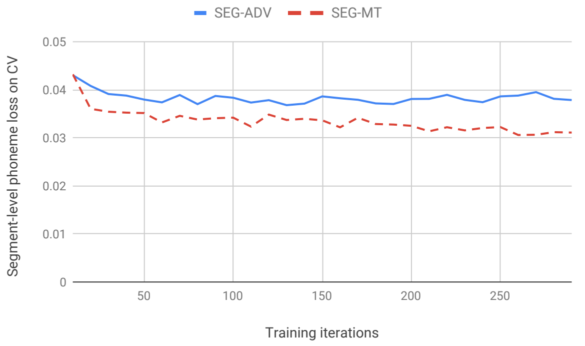

Figure 4: *Phoneme loss using the segment-level multitask or adversarial learning.*

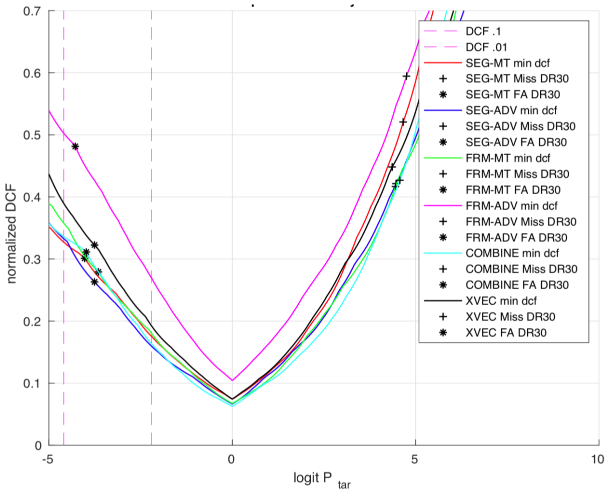

Figure 5: *minDCF as a function of effective prior. FA DR30 refers to the point to the left of which there are fewer than 30 false-alarms. The vertical magenta dashed lines represent the two operating points of minDCF$_{0.01}$ and minDCF$_{0.1}$. Notice that minDCF$_{0.01}$ is on the left side of FA DR30 for all systems which results in this operating point unreliable (for more details see appendix B of [22]).*

enough for capturing the real distribution of phonemes. Further experiments will be left in our future work.

4.2.3. Combining multitask and adversarial learning

Since the previous experimental results show that explicitly encouraging the phonetic information at the frame-level and suppress it at the segment level both improve the performance, it's natural to combine them. The performance comparison is given in Table 3. The results show that combining the frame-level multitask and segment-level adversarial learning further improves the performance.

Table 3: *Systems combining frame-level multitask and segment-level adversarial learning, COMBINE denotes the architecture which performs both strategies*

System	EER(%)	minDCF$_{0.01}$	minDCF$_{0.1}$
x-vector baseline	3.73	0.389	0.192
FRM-MT	3.38	0.357	0.180
SEG-ADV	3.35	0.332	0.159
COMBINE	3.17	0.336	0.163

The minDCF curves comparing all systems mentioned above are shown in Figure 5. Overall, the SEG-ADV is best in the lower values of $\mathrm{P(tar)}$, whereas COMBINE is better for the higher values.

5. Conclusions and future works

In this work we have experimented with multi-task and adversarial training in order to enhance or suppress the phonetic information that can be present in the DNN structure for embedding extraction and analyzed the effects on the task of text-independent speaker verification. We have confirmed that it is helpful for SV to enhance the fine-grained phonetic information in the frame-level part of the DNN via multi-task learning as have been proposed in previous studies. We have then shown that, for most operating points, it is more beneficial to suppress phonetic information at the segment level using adversarial training. Moreover, since these two methods can be applied independently of each other, they can be combined. In our experiments, this either improved or retained the performance for most operating points.

We compare our methods with a baseline *x*-vector system with an EER of 3.73% on the Voxceleb test set, and by applying segment-level adversarial training we improve the performance to 3.35%. Finally we obtain the best result of 3.17% EER by combining segment-level adversarial training and frame-level multi-task training.

In the future work we think it would be interesting to experiment with phonetic units of different granularity like position-independent mono-phones, tri-phones and other senones.

6. Acknowledgement

Shuai, Yanmin and Kai are supported by the Shanghai International Science and Technology Cooperation Fund (No. 16550720300) and the China NSFC project (No. U1736202). Johan, Lukáš, Oldřich and Honza are supported by the European H2020 Marie Sklodowska-Curie cofinanced by the South Moravian Region under grant agreement No. 665860, Google Faculty Research Award program, Czech National Science Foundation (GACR) project ”NEUREM3” No. 19-26934X, Czech Ministry of Interior project No. VI20152020025 ”DRAPAK”, and Czech Ministry of Education, Youth and Sports from the National Programme of Sustainability (NPU II) project ”IT4Innovations excellence in science - LQ1602”.

7. References

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

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

[3] G. Heigold, I. Moreno, S. Bengio, and N. Shazeer, "End-to-end text-dependent speaker verification," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5115–5119.

[4] S.-X. Zhang, Z. Chen, Y. Zhao, J. Li, and Y. Gong, "End-to-end attention based text-dependent speaker verification," in *2016 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2016, pp. 171–178.

[5] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and S. Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification," in *2016 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2016, pp. 165–170.

[6] G. Bhattacharya, J. Alam, and P. Kenny, "Deep Speaker Embeddings for Short-Duration Speaker Verification," in *Interspeech 2017*, 08 2017, pp. 1517–1521.

[7] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2018*. IEEE, 2018.

[8] Z. Huang, S. Wang, and K. Yu, "Angular softmax for short-duration text-independent speaker verification," *Proc. Interspeech, Hyderabad*, 2018.

[9] Y. Liu, Y. Qian, N. Chen, T. Fu, Y. Zhang, and K. Yu, "Deep feature for text-dependent speaker verification," *Speech Communication*, vol. 73, pp. 1–13, 2015.

[10] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, "A novel scheme for speaker recognition using a phonetically-aware deep neural network," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 1695–1699.

[11] P. Kenny, V. Gupta, T. Stafylakis, P. Ouellet, and J. Alam, "Deep neural networks for extracting baum-welch statistics for speaker recognition," in *Proc. Odyssey*, 2014, pp. 293–298.

[12] Y. Liu, L. He, J. Liu, and M. T. Johnson, "Speaker embedding extraction with phonetic information," *arXiv preprint arXiv:1804.04862*, 2018.

[13] Z. Meng, J. Li, Z. Chen, Y. Zhao, V. Mazalov, Y. Gang, and B.-H. Juang, "Speaker-invariant training via adversarial learning," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5969–5973.

[14] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The Journal of Machine Learning Research*, vol. 17, no. 1, pp. 2096–2030, 2016.

[15] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," *arXiv preprint arXiv:1706.08612*, 2017.

[16] G. Saon, H. Soltau, D. Nahamoo, and M. Picheny, "Speaker adaptation of neural network acoustic models using i-vectors," in *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*. IEEE, 2013, pp. 55–59.

[17] M. Karafiát, L. Burget, P. Matějka, O. Glembek, and J. Černockỳ, "ivector-based discriminative adaptation for automatic speech recognition," in *2011 IEEE Workshop on Automatic Speech Recognition & Understanding*. IEEE, 2011, pp. 152–157.

[18] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," in *INTERSPEECH*, 2018.

[19] 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," IEEE Signal Processing Society, Tech. Rep., 2011.

[20] A. Paszke, S. Gross, S. Chintala, G. Chanan, E. Yang, Z. DeVito, Z. Lin, A. Desmaison, L. Antiga, and A. Lerer, "Automatic differentiation in pytorch," in *NIPS-W*, 2017.

[21] S. Ioffe, "Probabilistic linear discriminant analysis," in *European Conference on Computer Vision*. Springer, 2006, pp. 531–542.

[22] N. Brummer, "Measuring, refining and calibrating speaker and language information extracted from speech," Ph.D. dissertation, Stellenbosch: University of Stellenbosch, 2010.