The TalTech Systems for the Short-duration Speaker Verification Challenge 2020

Tanel Alumäe, Jörgen Valk

Tallinn University of Technology, Estonia

tanel.alumae@taltech.ee, jorgen.valk@gmail.com

Abstract

This paper presents the Tallinn University of Technology systems submitted to the Short-duration Speaker Verification Challenge 2020. The challenge consists of two tasks, focusing on text-dependent and text-independent speaker verification with some cross-lingual aspects. We used speaker embedding models that consist of squeeze-and-attention based residual layers, multi-head attention and either cross-entropy-based or additive angular margin based objective function. In order to encourage the model to produce language-independent embeddings, we trained the models in a multi-task manner, using training dataset specific output layers. In the text-dependent task we employed a phrase classifier to accurately reject trials with non-matching phrases. In the text-dependent task we used a language classifier to boost the scores of trials where the language of the test and enrollment utterances does not match. Our final primary metric score was 0.075 in Task 1 and 0.118 in Task 2.

Index Terms: speaker verification, cross-linguality, x-vectors, SdSV Challenge

1. Introduction

In this paper, we describe the Tallinn University of Technology (TalTech) systems developed for the Short-duration Speaker Verification Challenge 2020 [1]. The challenge aims to evaluate new technologies for text-dependent and text-independent speaker verification, focusing on the short duration scenario. Its second special focus is on cross-lingual speaker verification where the phonetic overlap between the enrollment and test utterances has a large variety.

Recently, the field of speaker verification has advanced rapidly, due to the development of neural network-based speaker embeddings called x-vectors [2] and their various improvements. The neural-network based models require large-scale speaker recognition training datasets that have been recently released [3, 4]. Last years have also seen several popular speech verification challenges, such as VOiCES from the Distance Challenge 2019 [5], VoxCeleb Speaker Recognition Challenge [6] and the NIST SRE challenges [7, 8], that have an important role in advancing the field. The SdSV Challenge differs from the other challenges in providing both text-dependent and text-independent speaker verification tasks and in investigating the cross-lingual aspects of speaker verification.

During the challenge, we explored several speaker embedding model architectures, loss functions and data augmentation techniques. This paper gives a detailed description of the models that we used in the final submission. We also describe the models for handling text-dependent verification and language identification. The paper lists the results of the individual and fused models, and also analyzes the contribution of some methods that are are specific to the cross-lingual scenario of the challenge.

2. Datasets

2.1. SdSV Challenge datasets

The SdSV Challenge training and evaluation datasets originate from the DeepMine corpus, collected using crowd-sourcing in Iran [9]. The corpus was recorded in realistic environments. The majority of the utterances are in Farsi (Persian) and a smaller subset in English.

For both Task 1 and 2, the in-domain data provided by the challenge organizers is divided into training, enrollment and test partitions.

The Task 1 training data consists of 101 063 utterances from 963 different speakers. The text of all utterances is drawn from a fixed set of ten Farsi and English phrases. Some speakers in the training data have multiple utterances of all phrases while some have only Farsi phrases. The evaluation enrollment partition contains 12 404 models, all of which contain three utterances of a specific phrase from the same speaker. The phrase identifier of the models in the enrollment data is provided. The evaluation test partition contains 69 542 utterances.

The Task 2 training data consists of 85 764 utterances from 588 unique speakers. All utterances in training set are in Farsi. Unlike Task 1, the transcripts of the utterances are not provided. The evaluation enrollment dataset contains 15 555 models and 110 673 utterances. There is a lot of variety in the enrollment models: the number of utterances per model ranges from 1 to 29 and the total speech duration per model is uniformly distributed between 3 to 120 seconds. The test partition contains 69 350 utterances in both Farsi and English. The duration of the utterances in training, enrollment and test partitions is quite different: while the majority of the utterances in test partition are between two and five seconds in length, the durations of training and enrollment utterances cover a wider range (see Figure 1). We used this observation when designing the backend for this task.

Since there are no official development datasets, we randomly split the official training datasets into in-house training and development sets for both tasks. This was done by taking a random 100 speaker subset of the training set together with the corresponding utterances. The resulting development sets were further split into enrollment and test datasets by drawing a 1000-utterance sample into the test set and using other utterances for generating a custom enrollment dataset that has a similar distribution of utterances per model as evaluation data.

2.2. Other training datasets

The SdSV Challenge uses a fixed training condition with limited training data. In addition to the in-domain training data, only the VoxCeleb1 [3], VoxCeleb2 [4] and the LibriSpeech [10] corpora were allowed to be used for training.

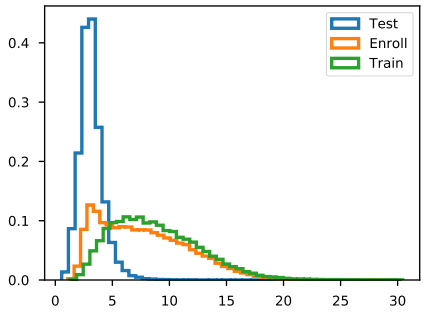

Figure 1: *Histogram of utterance lengths in seconds in the Task 2 data.*

2.3. Non-speech datasets

The SdSV Challenge allows using other non-speech datasets for data augmentation purposes. Our data augmentation strategy uses additive noises and reverberation. For additive noises, we used the music and noise subsets of the MUSAN corpus [11], resulting in over 900 noises and 42 hours of music from various genres. For reverberation, we used simulated small and medium room impulse responses [12] and real room impulse responses from the BUT Speech@FIT Reverb Database [13]. We also used utterances from the allowed speech corpora for generating babble noise.

3. Methods

In this section we provide a description of all the components used in our systems.

3.1. Speaker embeddings

Our backend uses speaker embedding models derived from the x-vector paradigm [2], with several enhancements.

Our models use either 30, 40 or 48 dimensional filterbank features. Utterance level mean normalization is applied. We do not use any speech activity detection. This has two reasons: first, most of the utterances in training and test data are already segmented to contain mostly speech; second, we use attention mechanism in the pooling layer which is known to be relatively insensitive to speech activity detection.

We apply on-the fly data augmentation during training using AugMix [14], a technique recently proposed for image recognition. AugMix has two components: it generates stochastic augmentations of the training data during learning and encourages the predictions produced from the original and augmented training samples to be similar to each other, using Jensen-Shannon divergence (JSD). Although we experimented with both components during the challenge, we ended up using only the stochastic data augmentation component and did not use the JSD-based consistency loss in our final models, since we didn't observe significant benefits of the method. Furthermore, training models using consistency loss makes the training around 2.5 times slower. The stochastic augmentation technique works by generating new augmented copies of training samples on-the-fly. A clean training segment is cloned into several copies. A different randomly drawn series of augmentation transforms, each possibly with random parameters, is applied serially to each of the copies. Then, the augmented copies are

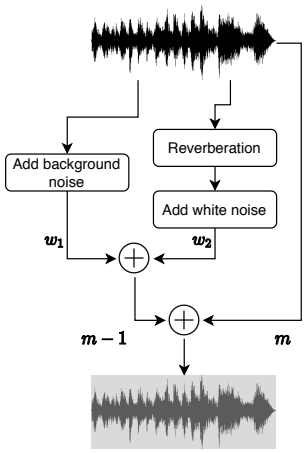

Figure 2: *Sample augmentation pipeline in AugMix (largely based on [14]). The number and type of transforms in each augmentation path is randomly sampled. The augmented signals are mixed using randomly sampled w_1 and w_2. The mixed augmented signal is finally interpolated with the original signal, using a sampled weight m.*

mixed with each other (using randomly sampled weights) and the resulting super-augmentated sample is finally mixed with the original clean sample, using a randomly sampled interpolation coefficient (see Figure 2). The benefit of this method, compared to using pre-generated static augmentations, is that there is a lot of variety in the training data: each training sample is a result of several random transforms, applied in random order and with a random weight.

The architecture of the speaker embedding model used by most of our systems is summarized in Table 1. All speaker embedding models that participated in our submission use the Resnet34 architecture [15, 16] for frame level feature extraction where the basic convolutional blocks with residual connections are replaced with squeeze-and-attention modules [17, 18].

The statistics pooling layer that maps frame-level features to segment level features is replaced in our model with a multi-head attention layer [19] that has been shown to provide superior performance [20, 21, 22, 23]. From among many variants of multi-head attention used in previous studies, we employ the one described in [20]: frame level representations are first mapped to N_{att} outputs ($N_{att} = 128$ in our model), using a 1×1 convolution and a ReLU nonlinearity; from this representation, each attention head (we used $N_{heads} = 5$ heads) computes it's own softmax-based weight distribution over the input utterance; finally, weighted mean and standard deviation are computed over the frame level features for each head, resulting in $N_{heads} \times 512 \times 2$ segment-level representations.

The last two layers of the model are replicated for each training dataset, as depicted in Figure 3. That is, there is a different pre-final dense layer and the final softmax layer for each training corpus, and the number of outputs in the softmax layer is equal to the number of speakers in the particular training dataset. This is different from the usual approach where speakers from all datasets are pooled together during training. This approach is similar to the multilingual bottleneck feature extractors used in speech recognition [24]. This multitask approach is motivated by the fact that the speakers in the three training datasets (VoxCeleb, LibriSpeeech and DeepMine) are expected to sound very different. By separating them into differ-

Table 1: *The neural network architecture used for extracting speaker embeddings. SE/res stands for squeeze-and-attention block and residual connections.*

Layer	*Spatial Size*	*#Channels*	*Kernel*
Input	$F \times T$	1	-
Frame level representations			
Pre-resnet	$F \times T$	64	7×7
Res-block 1	$F/2 \times T/2$	64	$3 \times \begin{bmatrix} 3 \times 3 \\ 3 \times 3 \\ SE/res \end{bmatrix}$
Res-block 2	$F/4 \times T/4$	128	$4 \times \begin{bmatrix} 3 \times 3 \\ 3 \times 3 \\ SE/res \end{bmatrix}$
Res-block 3	$F/8 \times T/8$	256	$6 \times \begin{bmatrix} 3 \times 3 \\ 3 \times 3 \\ SE/res \end{bmatrix}$
Res-block 4	$F/16 \times T/16$	512	$3 \times \begin{bmatrix} 3 \times 3 \\ 3 \times 3 \\ SE/res \end{bmatrix}$
Post-resnet	$1 \times T/16$	512	$F/16 \times 1$
Segment-level representations			
Pooling	1	$5 \times 512 \times 2$	MH-Attention
Embedding	1	512	Dense
Multi-output layers, one for each dataset			
FC	1	512	Dense
Output	1	#Speakers	Softmax

ent softmax branches, we encourage the model to learn dataset-independent features in the shared layers (including the embedding layer) and place the dataset-specific discrimination capability into the final branched layers. We show that this improves the cross-lingual and cross-domain performance of the resulting speaker embeddings.

We used two different objective functions for neural network training: cross-entropy (CE) and additive angular margin (AAM) softmax [25]. The AAM-Softmax parameter s is set to 30 and m to 0.2 radians. Both objective functions were applied in the multi-task fashion, as described before.

We also experimented with phonetic bottleneck features (BNFs). For this, we trained a bottleneck feature extractor on the LibriSpeech data. We used Kaldi [26] to train a DNN acoustic model with factorized time-domain neural network layers. The second-from-last layer in the model is a 40-dimensional linear layer that is used for extracting bottleneck features. We used the model to extract bottleneck features for all the training and evaluation data. The bottleneck features were presented to the speaker embedding model based on the approach proposed in [27]. Bottleneck features are projected to 3-channel spatially contiguous feature maps, having the same dimensionality as input filterbank features. This is done using three dense layers with ReLU nonlinearity. The three resulting channels are then combined with the filterbank features, resulting in a 4-channel input to the convolutional layers.

The models are implemented in PyTorch [28] using a framework developed in our lab. Training segment loading, data augmentation and feature extraction are all performed on the fly, thus reducing both training time and disk space required for training speaker embeddings on large datasets. Feature extraction, and data augmentation together with model training are performed on GPUs. One model can be trained in about 48 hours using three GPUs, without the need to prepare intermediate training data files from raw wave files.

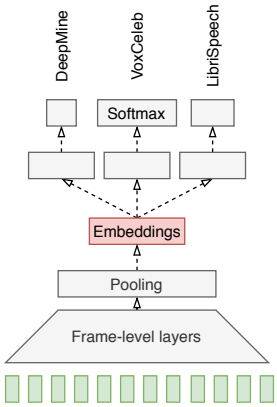

Figure 3: *Speaker embedding extractor with dataset-specific output layers*

3.2. Back-end modeling

For back-end modeling, we use cosine similarity and Probabilistic Linear Discriminant Analysis (PLDA). Cosine similarity is used with embeddings that are trained with AAM-Softmax. The scores of enrollment-test pairs are calculated as the cosine similarity of the two embeddings, after length normalization. PLDA is used with embeddings trained using the CE criterion. The final PLDA model is a linear interpolation of models trained on out-of-domain and in-domain data.

We use the Correlation Alignment (CORAL) technique [29, 30] to transform the embeddings of out-of-domain data to be similar to in-domain embeddings. CORAL does this by aligning the distributions of out-of-domain and in-domain features in an unsupervised way, using the Frobenius norm between the feature covariance matrices.

Before estimating the PLDA model, embeddings are preprocessed using LDA, whitening and length normalization. 200-dimensional LDA and the whitening transform are estimated on in-domain data, while centering is performed on the dataset that is used for training the PLDA. PLDA is estimated on a subset of out-of-domain data (500k utterances from VoxCeleb and 100k utterances from LibriSpeech) and all of in-domain data (except the portion used as in-house development set). Data augmentation is applied to the input audio when computing embeddings for training data. For Task 1, we use subsegmented training data for PLDA training: since the length distributions of test and training data are different, we randomly split training data to segments of 2 to 4 seconds in length. This improved our intermediate results by around 10%.

After scoring, results from all trials are subject to score normalization. We use Adaptive Symmetric Score Normalization [31]. The cohort is selected to be 400 closest utterances from a random 5000 utterance subset of the in-domain training data.

3.3. Phrase classification

Task 1 of the challenge handles text-dependent speaker verification. The task of the system is to verify that both the speaker and the phrase of the test utterance matches with that of the enrollment model. The number of different phrases occurring in evaluation data is limited to the 10 phrases that also occur in training data, i.e., there are no out-of-set test phrases. This allows to detect the Target-Wrong and Impostor-Wrong trials in evaluation trials using a relatively simple phrase classification

Table 2: *Results of the individual systems that participated in the final submission.*

	FBank Dim	*Criterion*	*Fine-tuning*	BNF	DCF_{min}	EER	DCF_{min}	EER
Task 1					*Progress set*		*Evaluation set*	
S	48	CE	-	-	0.090	2.34	0.090	2.37
	40	CE	-	✓	0.123	3.54		
	40	AAM	✓	-	0.094	2.26		
P	Fusion				0.076	2.09	0.076	2.13
Task 2								
S	40	AAM	✓	-	0.139	3.13	0.140	3.11
	30	CE	-	-	0.139	3.17		
P	Fusion				0.117	2.74	0.118	2.73

model. The model that we used is similar to the one used for extracting speaker embeddings, except that 5 TDNN layers are used for frame-level feature extraction and a LSTM layer for pooling. The utterance-level feature is taken from the output of the LSTM corresponding to the last frame of the input utterance. This is followed by a a single dense layer and a final softmax layer. The model was trained on the in-domain training data. On the in-house development set the phrase classification accuracy was 100%. Post-evaluation analysis showed that the equal error rate of the target-wrong trials in the evaluation set is 0.01 which confirms the accuracy of the model.

We used the model to add a bias of -99 to the trial scores where the test utterance did not to match the enrollment phrase.

3.4. Language identification

In Task 2 evaluation data, a subset of the utterances are in English. In order to apply a different scoring strategy to the English test utterances, we built a language identification system for classifying utterances based on the spoken language. For this, we trained a model for extracting language embeddings, using a subset of the of-out-domain English training data and all of in-domain Farsi training data. The architecture of the model is a simplified version of one that we used for extracting speaker embeddings: it has 5 TDNN layers, uses multi-head attention for pooling and cross-entropy for optimization. In addition to the additive noise and reverberation based augmentation, it also uses speed perturbation. The final language identification model uses logistic regression to classify language embeddings, and is trained an a small balanced subset of English and Farsi data.

It was difficult to get a reliable estimate of the language identification performance of the resulting model, since there were no English utterances spoken by Farsi speakers available in the training data. The model classified 100% of the in-house development data correctly but it was unclear how much the model had learned to classify corpus-specific channel effects and how much the actual language characteristics.

We used the language identification model as follows: in the Task 2 systems that used the PLDA backend, we added a small positive bias to the trial scores where the test utterance was classified as being in English.

4. Results

4.1. Main results

Results of the individual systems that participated in our final submission and their fusions for both tasks on the Progress set are listed in Table 2. Results on the Evaluation set were almost identical. The final single systems differed in the dimensionality of the used filterbank features, training criteria of the speaker embedding models, the use of final finetuning of the speaker embeddings model, and in the incorporation of bottleneck features. Fusion was done by simple linear interpolation of the log-likelihood ration scores using uniform weights. We didn't optimize the fusion on the in-house development set since we found it to be too unreliable. The single system submissions are marked with "S" and the primary systems with "P".

It can be seen that models trained using the CE and AAM criteria achieved similar results and their fusion resulted in notable improvement.

We trained models that employ BNFs for both tasks. Although they resulted in improvements on the in-house development data, they gave disappointing results on the Progress set. Therefore we ended up using the model with BNFs only in Task 1, since its inclusion in the final fusion gave small improvements.

Table 3: *Contrastive results on Task 2.*

Multi-output embeddings	Positive bias for English utterances	DCF_{min}	EER
✓	✓	0.140	3.200
-	✓	0.157	3.708
✓	-	0.143	3.254

4.2. Contrastive results

Table 3 lists the performance of some contrastive systems on the Task 2 evaluation set. All results are based on a cross-entropy based frontend and a PLDA-based backend.

All of the speaker embedding models that participated in our submission were trained using multiple dataset-specific output layers (Figure 3). Table 3 shows that replacing the multi-output model with a single output model (i.e., where all speakers are pooled) degrades the primary metric DFC_{min} by 12% and EER by 16% relative.

As described in section 3.4, we used a language identification system in Task 2 to find test utterances that are probably in English, and added a small bias to the corresponding trials, in order to compensate for the language mismatch factor in scoring. Table 3 shows that this method improves Task 2 results, but the gain is small (around 2% relative).

5. Conclusions

This paper described the systems we developed for the SdSV Challenge 2020. For both text-dependent and text-independent tasks we used a fusion of system that mainly differ in the

loss function that was used to train them. All our individual speaker embedding models are based on the Resnet34 architecture with squeeze-and-excitation modules. We also presented post-evaluation analysis of two methods that aim to improve speaker verification performance in the cross-lingual scenario. We showed that when training speaker embedding models on several (possibly out-of-domain) training corpora, using separate output layers for each dataset can result in relatively large performance improvement.

6. References

[1] H. Zeinali, K. A. Lee, J. Alam, and L. Burget, "Short-duration speaker verification (SdSV) challenge 2020: the challenge evaluation plan," arXiv preprint arXiv:1912.06311, Tech. Rep., 2020.

[2] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust DNN embeddings for speaker recognition," in *ICASSP*, 2018, pp. 5329–5333.

[3] A. Nagrani, J. Chung, and A. Zisserman, "VoxCeleb: a large-scale speaker identification dataset," in *Interspeech*, 2017.

[4] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep speaker recognition," in *Interspeech*, 2018.

[5] M. K. Nandwana, J. van Hout, C. Richey, M. McLaren, M. A. Barrios, and A. Lawson, "The VOiCES from a Distance Challenge 2019," in *Interspeech*, 2019.

[6] J. S. Chung, A. Nagrani, E. Coto, W. Xie, M. McLaren, D. A. Reynolds, and A. Zisserman, "VoxSRC 2019: The first VoxCeleb speaker recognition challenge," *ISCA Challenges*, 2019.

[7] S. O. Sadjadi, C. Greenberg, E. Singer, D. Reynolds, L. Mason, and J. Hernandez-Cordero, "The 2018 NIST speaker recognition evaluation," in *Interspeech*, 2019.

[8] S. O. Sadjadi, C. S. Greenberg, E. Singer, D. Reynolds, L. Mason, and J. Hernandez-Cordero, "The 2019 NIST speaker recognition evaluation CTS challenge," *Speaker Odyssey*, 2020.

[9] H. Zeinali, H. Sameti, and T. Stafylakis, "DeepMine speech processing database: Text-dependent and independent speaker verification and speech recognition in Persian and English." in *Proc. Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 386–392.

[10] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "LibriSpeech: an ASR corpus based on public domain audio books," in *ICASSP*, 2015.

[11] D. Snyder, G. Chen, and D. Povey, "MUSAN: A Music, Speech, and Noise Corpus," 2015, arXiv:1510.08484v1.

[12] T. Ko, V. Peddinti, D. Povey, M. L. Seltzer, and S. Khudanpur, "A study on data augmentation of reverberant speech for robust speech recognition," in *ICASSP*, 2017, pp. 5220–5224.

[13] I. Szöke, M. Skácel, L. Mošner, J. Paliesek, and J. Černocký, "Building and evaluation of a real room impulse response dataset," *IEEE Journal of Selected Topics in Signal Processing*, vol. 13, no. 4, pp. 863–876, 2019.

[14] D. Hendrycks, N. Mu, E. D. Cubuk, B. Zoph, J. Gilmer, and B. Lakshminarayanan, "Augmix: A simple data processing method to improve robustness and uncertainty," *arXiv preprint arXiv:1912.02781*, 2019.

[15] W. Cai, J. Chen, and M. Li, "Exploring the encoding layer and loss function in end-to-end speaker and language recognition system," in *Odyssey 2018: The Speaker and Language Recognition Workshop*, 2018.

[16] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *CVPR*, 2016, pp. 770–778.

[17] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *CVPR*, 2018, pp. 7132–7141.

[18] J. Zhou, T. Jiang, Z. Li, L. Li, and Q. Hong, "Deep speaker embedding extraction with channel-wise feature responses and additive supervision softmax loss function," in *Interspeech*, 2019, pp. 2883–2887.

[19] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *arXiv preprint arXiv:1409.0473*, 2014.

[20] Y. Zhu, T. Ko, D. Snyder, B. Mak, and D. Povey, "Self-attentive speaker embeddings for text-independent speaker verification," in *Interspeech*, 2018.

[21] F. A. R. Rahman Chowdhury, Q. Wang, I. L. Moreno, and L. Wan, "Attention-based models for text-dependent speaker verification," in *ICASSP*. IEEE, 2018, pp. 5359–5363.

[22] K. Okabe, T. Koshinaka, and K. Shinoda, "Attentive statistics pooling for deep speaker embedding," in *Interspeech*, 2018.

[23] P. Safari and J. Hernando, "Self multi-head attention for speaker recognition," in *Interspeech*, 2019, pp. 4305–4309.

[24] K. Veselỳ, M. Karafiát, F. Grézl, M. Janda, and E. Egorova, "The language-independent bottleneck features," in *IEEE Spoken Language Technology Workshop (SLT)*, 2012, pp. 336–341.

[25] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4690–4699.

[26] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, "The Kaldi speech recognition toolkit," in *ASRU*, 2011.

[27] T. Zhou, Y. Zhao, J. Li, Y. Gong, and J. Wu, "CNN with phonetic attention for text-independent speaker verification," in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 718–725.

[28] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "PyTorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems 32*, 2019, pp. 8024–8035.

[29] B. Sun, J. Feng, and K. Saenko, "Return of frustratingly easy domain adaptation," in *AAAI*, 2016.

[30] M. J. Alam, G. Bhattacharya, and P. Kenny, "Speaker verification in mismatched conditions with frustratingly easy domain adaptation," in *Odyssey 2018 The Speaker and Language Recognition Workshop*, 2018, pp. 176–180.

[31] P. Matějka, O. Novotnỳ, O. Plchot, L. Burget, M. D. Sánchez, and J. Černockỳ, "Analysis of score normalization in multilingual speaker recognition," *Interspeech*, pp. 1567–1571, 2017.