The IDLab Short-duration Speaker Verification Challenge 2020 System Description

Jenthe Thienpondt, Brecht Desplanques, Kris Demuynck

IDLab, Department of Electronics and Information Systems, imec - Ghent University, Belgium

jenthe.thienpondt@ugent.be, brecht.desplanques@ugent.be

Abstract

In this report, we describe our top-scoring IDLab submission for the text-independent task of the Short-duration Speaker Verification (SdSV) Challenge 2020. The challenge provides a difficult set of speaker verification trials with varying degrees of phonetic overlap. Our submission is based on our recently proposed state-of-the-art ECAPA-TDNN speaker embedding extractor. We further improved performance by using hard prototype mining, a novel approach for computationally efficient hard example mining in conjunction with the AAM softmax loss function. The importance of proper s-normalization imposter cohort selection in this challenge proved crucial, which we further improved by introducing a language-dependent offset. A fusion of five systems with minor topological alterations resulted in a final MinDCF and EER of 0.065 and 1.45% respectively on the SdSVC 2020 evaluation set.

Index Terms: speaker recognition, cross-lingual speaker verification, x-vectors, SdSV Challenge 2020

1. SdSVC IDLab submission

This section is a system description of the IDLab SdSVC final submission. We start with a single system ECAPA-TDNN baseline [1]. The subsequent sections will tackle the problems of domain adaptation and cross-lingual language effects present in the SdSV Challenge data. The final subsection discusses system fusion.

1.1. The ECAPA-TDNN baseline system

All submitted speaker verification systems make use of the ECAPA-TDNN architecture proposed in [1]. This architecture is based on the well-known x-vector topology [2] and introduces several enhancements to extract more robust speaker embeddings. It incorporates Squeeze-Excitation SE blocks [3], multi-scale Res2Net [4] features, multi-layer feature aggregation [5] and channel-dependent attentive statistics poolings [1]. The network topology is shown in Figure 1. The topology of integrated SE-Res2Blocks can be found in Figure 2. Implementation details and performance analysis of this architecture can be found in [1]. We deviate slightly from the original architecture by also incorporating SE-Blocks in the residual connections.

We use all allowed training data, except the VoxCeleb1 test partition and LibriSpeech, for which only the *train-other-500* subset [6] is considered. This amounts to 9077 training speakers. We create 9 additional augmented copies of the training data following the Kaldi recipe [7] in combination with the MUSAN corpus (babble, noise, music) [8] and the RIR[9] dataset (reverb). The remaining augmentations are generated with the open-source SoX (1.25 tempo increase, 0.85 tempo decrease, phaser and flanger) and FFmpeg (alternating opus and aac compression) libraries.

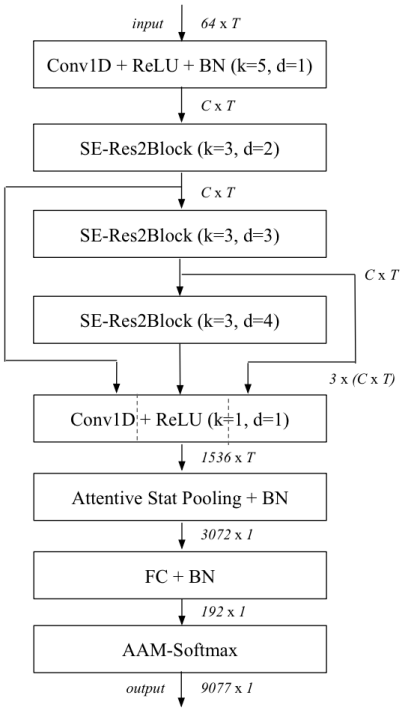

Figure 1: *Network topology of the ECAPA-TDNN. We denote k for kernel size and d for dilation spacing of the Conv1D layers or SE-Res2Blocks. C and T correspond to the channel and temporal dimension of the intermediate feature-maps respectively.*

The input features are 64 dimensional MFCCs from a 25 ms window with a 10 ms frame shift. The MFCCs are normalized through cepstral mean subtraction and no voice activity detection is applied. To avoid overfitting during the ECAPA-TDNN training process, we take a random crop of 2 to 3 seconds of the utterances during each iteration. Similarly, we incorporate SpecAugment [10] as an online augmentation method which randomly masks 0 to 5 time frames and 0 to 8 frequency bands of the training log mel-spectrograms.

We use the Angular Additive Margin (AAM) softmax [11] as training criterion for the model. The system is trained with the Adam optimizer [12] until convergence on a small SdSVC validation subset that contains about 2.5% of the Farsi training utterances. The training protocol uses a cyclical learning rate schedule with the *triangular2* policy [13]. The learning rate is varied between a minimum of 1e-8 and decaying maximum of 1e-3 during cycles of 130k iterations. A weight decay of 2e-5 is applied on all weights of the model except for the AAM softmax layer which uses a weight decay value of 2e-4. We use

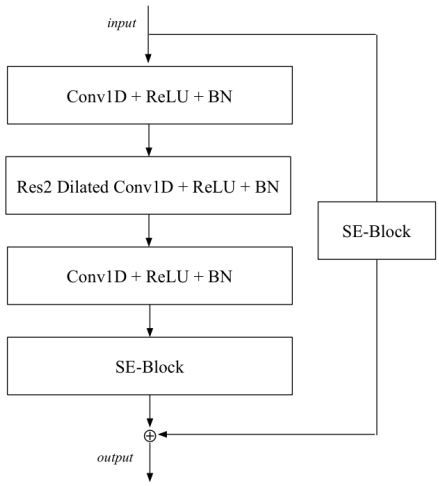

Figure 2: *The SE-Res2Block of the ECAPA-TDNN architecture. The standard Conv1D layers have a kernel size of 1. The central Res2Net [4] Conv1D with scale dimension $s = 8$ expands the temporal context through kernel size k and dilation spacing d.*

a mini-batch size of 128.

The speaker enrollment models are constructed by averaging the corresponding L_2-normalized enrollment embeddings produced by the final fully connected layer of the ECAPA-TDNN. The verification trials are scored by calculating the cosine distance between the enrollment model and the test utterance embedding. Scores are normalized using top-40 adaptive s-normalization [14, 15]. The imposter cohort consists of speakers represented by the average of all their length-normalized training embeddings. The final scores are calibrated with logistic regression [16] on our small SdSVC validation subset.

We consider five implementations with minor topological differences as shown in Table 1. We alternate the embedding size between 192 and 256. The Res2Net multi scale features inside the SE-Res2Blocks are optionally replaced by the standard TDNN 1-dimensional dilated convolutions. *Summed* indicates if the input of each SE-Res(2)Block is the sum of the output of all preceding SE-Res(2)Blocks instead of only considering the output of the preceding block. The number of filters in the convolutional frame layers C is set to 1024, which is reduced to 512 in the bottleneck of the SE-Res(2)Blocks to limit the amount of model parameters. However, system 5 is developed without this constraint and the channel dimension is kept to 2048 for all feature maps in the frame layers.

1.2. Hard prototype mining

To further improve performance on the baseline, we investigate how to exploit the information of the in-domain training data more efficiently. We combine targeting harder samples and putting more importance to target-domain samples with our proposed Hard Prototype Mining (HPM) fine-tuning strategy.

We interpret the weights of the AAM softmax layer as approximations of the class-centers of the training speakers and refer to them as speaker prototypes. As these trainable weights are already a part of the model, there are no additional computations needed. Given batch size n and N training speakers, the AAM softmax loss L with margin m is defined as:

$$L = -\frac{1}{n}\sum_{i=1}^{n} log\frac{e^{s(cos(\theta_{y_i}+m))}}{e^{s(cos(\theta_{y_i}+m))} + \sum_{j=1,j\neq y_i}^{N} e^{s(cos(\theta_j))}} \quad (1)$$

where θ_{y_i} is the angle between the sample embedding $\boldsymbol{x}_i$ with corresponding speaker identity y_i and the speaker prototype $\boldsymbol{W}_{y_i}$. θ_j is the angle with all other L_2-normalized speaker prototypes stored in a trainable matrix $\boldsymbol{W} \in \mathbb{R}^{D\times N}$ with D indicating the embedding size. A speaker similarity matrix $\boldsymbol{S} \in \mathbb{R}^{N\times N}$ can be constructed from $\boldsymbol{W}^T\boldsymbol{W}$, containing the cosine distances between all pairs of speaker prototypes.

A straightforward way of constructing batches would be to only mine samples from the most difficult speaker pairs according to $\boldsymbol{S}$. However, this could lead to oversampling a narrow group of speakers which potentially degrades generalization performance. Consequently, we construct mini-batches by iterating randomly over all N training speakers. Each iteration determines S speakers, irrespective of their similarity, for which U random utterances are sampled from each of their I most similar speakers, including the selected speaker. This implies that $S \times U \times I$ should be equal to the batch size n. When we have iterated over all training speakers, the similarity matrix $\boldsymbol{S}$ is updated and the batch generating process is repeated. Experiments indicate that given a batch size of 128, $S = 16$, $I = 8$ and $U = 1$ result in good performance. We reduce the maximum of the cyclical learning rate to 1e-4 and reduce the cycle length to 60k iterations to fine-tune all models in our submission.

We correct the bias towards the VoxCeleb and LibriSpeech corpus by equalizing the sample probability for each domain. During the construction of the batches, subsequent selections of the S speakers cover a set of all 588 Farsi speakers and 588 random speakers from both the VoxCeleb and LibriSpeech domain. When the set runs empty, the similarity matrix $\boldsymbol{S}$ is updated and 588 new speakers are randomly selected from the out-of-domain data to allow reiteration of the batch generation process. This process assigns more importance towards samples from hard speakers in the target-domain, while still allowing the network to learn from samples of challenging out-of-domain speakers.

1.3. Adaptive s-normalization with language offset

Based on [14], we set the imposter cohort of the adaptive s-normalization to contain in-domain Farsi data only. However, an unknown portion of the test utterances in the SdSVC trials is English. In case of a speaker verification trial with language mismatch, this will result in an overestimated mean imposter score for the Farsi enrollment model, as it will only be compared against Farsi imposters. We introduce a language-dependent offset in the adaptive s-normalization procedure to compensate for this effect.

Given a trial score $s(e,t)$ between the enrollment model e and target utterance t, the language-dependent s-normalized score is defined as:

$$s(e,t)_n = \frac{s(e,t) - \mu(S_t)}{\sigma(S_t)} + \frac{s(e,t) - (\mu(S_e) - \alpha)}{\sigma(S_e)}. \quad (2)$$

with S_i the set of scores of the speaker embedding i against its top-N imposter cohort, with $\mu(S_i)$ the mean of those scores and $\sigma(S_i)$ the standard deviation. α is the language-dependent compensation offset. It is defined as zero if there is no language mismatch detected and in that case regular adaptive s-norm is

Table 1: *EER and MinDCF performance of all individual systems and final fusion on the VoxCeleb1 and SdSVC 2020 test sets. All HPM models use our hard prototype mining technique as explained in Section 1.2. LID denotes usage of our language-dependent s-normalization variant introduced in Section 1.3.*

#	System (# params)	Emb. dim	Res2	Summed	Fine-tune	VoxCeleb1		SdSVC 2020	
						EER(%)	MinDCF	EER(%)	MinDCF
1	ECAPA-TDNN (24M)	192	no	no	baseline	0.94	0.1181	2.38	0.1042
					HPM	0.85	0.0945	1.81	0.0798
					HPM + LID	-	-	1.75	0.0781
2	ECAPA-TDNN (24M)	192	no	yes	baseline	1.03	0.1260	2.34	0.0996
					HPM	0.96	0.1248	1.77	0.0791
					HPM + LID	-	-	1.72	0.0775
3	ECAPA-TDNN (16M)	256	yes	no	baseline	0.86	0.0969	2.32	0.1008
					HPM	0.81	0.1033	1.75	0.0784
					HPM + LID	-	-	1.69	0.0764
4	ECAPA-TDNN (16M)	256	yes	yes	baseline	0.88	0.1101	2.32	0.0994
					HPM	0.88	0.1161	1.69	0.0759
					HPM + LID	-	-	1.63	0.0742
5	ECAPA-TDNN (44M)	256	yes	yes	baseline	0.87	0.0824	2.13	0.0938
					HPM	0.79	0.1010	1.69	0.0759
					HPM + LID	-	-	**1.63**	**0.0739**
	Weighted fusion of 1-5				HPM + LID	-	-	**1.45**	**0.0651**

applied. When during test time the test utterance is detected to be English, we enable the language offset. Given $\mu_{S_{FA}}$ as the expected mean imposter score of Farsi imposters against a Farsi speaker and $\mu_{S_{USA}}$ as the expected mean imposter score of USA-English imposters against a Farsi speaker, we define this compensation offset α as $\mu_{S_{FA}} - \mu_{S_{USA}}$. The mean imposter values can be easily estimated on the speaker prototypes stored in the AAM softmax module by applying s-norm on the relevant prototypes.

To detect the language of the test utterance given its embedding, we train a Language Identification LID module based on a Gaussian Backend (GB) [17] modeled on the L_2-normalized AAM speaker prototypes of the Persian and the USA speakers. However, there will be a mismatch between the English spoken by a native Farsi speaker and a USA citizen. To compensate for this effect we interpolate between the GB mean vector for the USA language class $\boldsymbol{\mu}_{USA}$ and the mean vector corresponding with Farsi $\boldsymbol{\mu}_{FA}$ and set the expected mean embedding of the English model to $0.75\boldsymbol{\mu}_{USA} + 0.25\boldsymbol{\mu}_{FA}$. This adapted language model should be able to robustly detect English spoken by a native Farsi speaker.

1.4. Final submission and results

The IDLab final submission for the SdSVC consists of a fusion of the five proposed ECAPA-TDNN systems fine-tuned with HPM combined with language-dependent s-normalization with the LID labels extracted from System 1. The fusion is realized on the score level by taking a weighted average over the calibrated scores of each individual system. The systems that incorporated Res2 modules were given double the weight in the averaging compared to the other systems.

The final score-based fusion of the single systems fine-tuned with domain-balanced HPM and language-dependent score normalization results in an EER of 1.45% and a MinDCF of 0.0651 as shown in Table 1. Fusion of all systems leads to a relative improvement of 11% and 11.9% for the EER and MinDCF metric respectively on the SdSVC test set over System 5. This shows that minor architectural variations can prove sufficient to learn complementary speaker embeddings. Table 2 gives a comparison of our single systems on the progression and evaluation set of the challenge. There are no significant performance differences between the progression set and the evaluation set.

Table 2: *Performance of the HPM + LID systems from Table 1 on the progression and evaluation set of the SdSV Challenge 2020*

#	SdSV20 Prog.		SdSV20 Eval.	
	EER(%)	MinDCF	EER(%)	MinDCF
1	1.75	0.0754	1.75	0.0781
2	1.72	0.0775	1.72	0.0775
3	1,69	0.0764	1.69	0.0764
4	1.63	0.0745	1.63	0.0742
5	1.63	0,0740	1.63	0.0739
Fusion	1.45	0.06541	1.45	0.0651

2. Conclusions

In this report we provided details about our top-scoring SdSVC 2020 Challenge submission. We proposed HPM as a computationally efficient hard negative mining strategy and language-dependent s-normalization to limit the effects of the imposter score distribution mismatch of cross-lingual trials present in the SdSVC evaluation set. A fusion of five fine-tuned systems based on our ECAPA-TDNN architecture resulted in a final top-scoring submission on the SdSVC evaluation set of 1.45% EER and 0.0651 MinDCF value.

3. References

[1] B. Desplanques, J. Thienpondt, and K. Demuynck, "ECAPA-TDNN: Emphasized channel attention, propagation and aggregation in TDNN based speaker verification," 2020, submitted to Interspeech. [Online]. Available: http://arxiv.org/abs/2005.07143

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

[3] J. Hu, L. Shen, and G. Sun, "Squeeze-and-Excitation networks," in *Proc. IEEE/CVF CVPR*, 2018, pp. 7132–7141.

[4] S. Gao, M.-M. Cheng, K. Zhao, X. Zhang, M.-H. Yang, and P. H. S. Torr, "Res2Net: A new multi-scale backbone architecture," *IEEE TPAMI*, 2019.

[5] Z. Gao, Y. Song, I. McLoughlin, P. Li, Y. Jiang, and L.-R. Dai, "Improving Aggregation and Loss Function for Better Embedding Learning in End-to-End Speaker Verification System," in *Proc. Interspeech*, 2019, pp. 361–365.

[6] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," in *Proc. ICASSP*, 2015, pp. 5206–5210.

[7] D. Snyder, D. Garcia-Romero, G. Sell, A. McCree, D. Povey, and S. Khudanpur, "Speaker recognition for multi-speaker conversations using x-vectors," in *Proc. ICASSP*, 2019, pp. 5796–5800.

[8] D. Snyder, G. Chen, and D. Povey, "MUSAN: A music, speech, and noise corpus," 2015.

[9] 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 *Proc. ICASSP*, 2017, pp. 5220–5224.

[10] D. S. Park, W. Chan, Y. Zhang, C.-C. Chiu, B. Zoph, E. D. Cubuk, and Q. V. Le, "SpecAugment: A simple data augmentation method for automatic speech recognition," in *Proc. Interspeech*, 2019.

[11] J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "ArcFace: Additive angular margin loss for deep face recognition," in *2019 IEEE/CVF CVPR*, 2019, pp. 4685–4694.

[12] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Proc. ICLR*, 2014.

[13] L. N. Smith, "Cyclical learning rates for training neural networks," in *IEEE WACV*, 2017, pp. 464–472.

[14] P. Matejka, O. Novotný, O. Plchot, L. Burget, M. Diez, and J. Černocký, "Analysis of score normalization in multilingual speaker recognition," in *Proc. Interspeech*, 2017, pp. 1567–1571.

[15] S. Cumani, P. Batzu, D. Colibro, C. Vair, P. Laface, and V. Vasilakakis, "Comparison of speaker recognition approaches for real applications." in *Proc. Interspeech*, 2011, pp. 2365–2368.

[16] N. Brümmer and E. de Villiers, "The BOSARIS toolkit: Theory, algorithms and code for surviving the new DCF," 2013.

[17] M. F. BenZeghiba, J. Gauvain, and L. Lamel, "Gaussian backend design for open-set language detection," in *Proc. ICASSP*, 2009, pp. 4349–4352.