ViVoLAB System Description for SdSV Challenge

Victoria Mingote, Antonio Miguel, Alfonso Ortega, Eduardo Lleida

ViVoLab, Aragón Institute for Engineering Research (I3A), University of Zaragoza, Spain
{vmingote,amiguel,ortega,lleida}@unizar.es

Abstract

This paper describes the system description of the system developed by ViVoLAB research group for the Short-duration Speaker Verification (SdSV) Challenge 2020. This challenge is focused on the study of the speaker verification (SV) systems behaviour in a short duration scenario for text-dependent and text-independent tasks. In this description paper, we introduce the different approaches used to create our system taking into account the lexical content since this information is relevant for the text-dependent task. To address this, we have employed some architectures which are focused on maintaining the temporal information of the uttered phrase using deep neural network combined with alignment mechanism or attention mechanisms with phonetic embeddings which helps to condition the attention mask. In this challenge, the use of several databases for training are allowed, but we have developed our architectures using only the training dataset, which corresponds with the evaluation data. Thus, we have implemented a system with the in-domain data. The results obtained on the SdSV evaluation set show that our fused system achieves competitive results using only the in-domain data to train.

Index Terms: speaker recognition, text-dependent, short-duration

1. Introduction

This paper presents the ViVoLab SV systems submitted to the SdSV Challenge in the text-dependent task [1]. During the challenge, we have implemented several systems based on different DNN architectures for the embeddings extraction. To develop these architectures, we have employed different alternatives to keep the order of the phonetic information [2]. With this purpose, we have used architectures based on Convolution Neural Network (CNN) combined with alignment mechanisms or Residual Networks (RN) [3] combined with attention mechanisms [4, 5]. Moreover, we have added more phonetic information to the different architectures with the use of phoneme embeddings extracted from another neural network trained as phoneme classification network [6, 7]. These phoneme embeddings are used as complement to the feature extractor. Additionally, we have explored several combinations between Linear Discriminant Analysis (LDA) and Probabilistic Linear Discriminant Analysis (PLDA) as back-end for the scoring process.

The remainder of this paper is laid out as follows. Section 2 provides a description of the datasets employed for training of the system. In Section 3, we describe the acoustic features used as system input. The different architectures for extracting speaker embeddings and the training strategies are explained in Section 4. Section 5 describes the back-end applied to obtain the evaluation scores. Finally, Section 6 presents the results.

2. Training Datasets

The SdSV Challenge 2020 allows a fixed training set consisting of some specific databases to develop the systems. For text-dependent task, these datasets are VoxCeleb 1[8], VoxCeleb 2 [9], LibriSpeech [10] and DeepMine [11, 12]. We have employed LibriSpeech to train a phoneme classification network which is used to extract phoneme embeddings. For training all the neural network front-end systems, we have used DeepMine train partition. We have not used VoxCeleb 1 and 2 datasets in the training process. The evaluation data corresponds to the DeepMine dataset as well. Thus, we have developed a system only with the in-domain data.

3. Acoustic Features

To develop this work, we have used three different acoustic features for training of the systems.

3.1. Log Mel-Filter Bank

Most of the systems in this work have been developed using as input a feature vector based on mel-scale filter banks. With this feature extractor, we obtain two log filter banks of sizes 24 and 32 which are concatenated with the log energy.

3.2. Mel-Frequency Cepstral Coefficients

As input for the other DNN systems, Mel-Frequency Cepstral Coefficient (MFCC) feature extraction is applied over all audios to convert them to the cepstral features of 20 dimensions, and the first and second-order derivatives are computed over the feature vector. Then, an energy-based voice activity detector is used over them. After frame selection, features are short-time Gaussianized with a 3-second sliding window.

3.3. Correlation

On the other hand, we have carried out some experiments using other less frequent features which are based on the use of the correlation of the input signals. As we will show, these features by themselves are not the best choice to train these systems, but a fusion with them produces remarkable improvements.

4. System Architectures

In this work, four different types of speaker embedding extractors have been developed. For all these embedding extractors, we have used 1 dimension (1D) convolution layers instead of 2 dimension (2D) convolution layers, since this layer allows us to operate in temporal dimension to add context information, and at the same time, the channels are combined at each layer [2]. Each of these architectures is briefly described below.

4.1. RN, Attention and Memory Layers

The first architecture ($archA$) combines RN, Self-Attention Mechanism [4] and Memory layers [5]. Fig.1 depicts this architecture which is composed of two main parts: the backbone and the pooling. The backbone uses two RN blocks with three layers each block and Rectified Linear Units (ReLU) as nonlinearities.

Furthermore, this architecture needs positional information [4] for the self attention layers to provide a good performance. Instead of using temporal positional information as many language modelling applications, we use the output of a phonetic classifier bottleneck [6, 7]. The architecture of the phonetic classifier is an evolution of [6]. In this system we use a modification of efficient net [13] to operate with 1D group convolutions as backbone. Efficient net can produce several temporal scales. We combine them using a modification of [14], where we substitute the linear combinations by the operation concatenation of channels and 1D group convolution. We concatenate this information before each RN block.

Following these RN blocks, the pooling part alternates two MultiHead Attention layers with two Memory layers. In the MultiHead Attention layers, we only employ the encoder part, which can be seen analogously as an alignment method which allows assigning embeddings to several categories. This approach has been found useful for text-dependent tasks. Furthermore, with the integration of the phoneme embeddings in the backbone part, the performance of the attention mechanism improves since the phoneme embeddings help to guide to the attention mask. In addition, the use of memory layers is also proved helpful since these layers are able to store the knowledge obtained for the network during the training process. Using this layer, the input data is compared with all the keys, and the scores obtained are used to select the keys with the highest scores and compute the associated weight vectors. After that, these weights are combined with the memory values of the selected keys, and the output is concatenated with the output of the previous attention mechanism.

Table 1: *Topology for RN, Attention and Memory layers architecture (archA).*

Layer	Layer type	Channels	Output
1	Conv1D	128	128×T
2	ResBlock-ReLU(x3)	160	160×T
3	ResBlock-ReLU(x3)	256	256×T
4	BatchNorm1D	256	256×T
5	DotAtt	256	256×T
6	Memory	256	256×T
7	DotAtt	256	256×T
8	Memory	256	256×T
9	Mean	–	256
10	FC+softmax	–	N
	Cross-Entropy Loss		

4.2. RN, Attention and Memory Layers using Group Convolutions

The second embedding extractor ($archB$) employed is a modification of the previous one. In Table 2, the description of this second network shows that the main difference with the first architecture is the combination of the RN blocks with MaxMinGroup nonlinearity instead of the usual ReLU activations. MaxMinGroup is proposed as a variant of the activation Max-Feature-Map (MFM) [15] which splits the channels into two groups and concatenates the maximum, the minimum and the original values. This activation has the advantage of that the signals are not truncated too soon to zero as in ReLU.

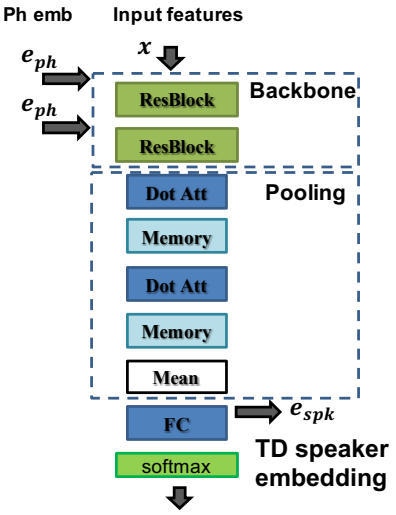

Figure 1: *Architecture for RN, Attention and Memory layers network, composed of a backbone, a pooling and a embedding extraction.*

Furthermore, this second architecture also employs group convolutions instead of the usual convolution. This group convolution layer provides more robustness to deep architectures, and also the computation time and the parameter size are reduced since this layer allows to split the usual convolution in smaller block matrices to operate with them.

Table 2: *Topology for RN, Attention and Memory layers using group convolutions architecture (archB).*

Layer	Layer type	Channels	Output
1	Conv1D	128	128×T
2	ResBlock-MaxMinGroup(x3)	160	160×T
3	ResBlock-MaxMinGroup(x3)	256	256×T
4	BatchNorm1D	256	256×T
5	DotAtt	256	256×T
6	Memory	256	256×T
7	DotAtt	256	256×T
8	Memory	256	256×T
9	Mean	–	256
10	FC+softmax	–	N
	Cross-Entropy Loss		

4.3. RN and LSTM

In order to obtain diversity, the third architecture ($archC$) is based on RN and LSTM. The use of LSTM layers allows us to capture long-term context, which is combined with the short-term context of the convolution layers. To create this architecture, as Table 3 shows, we have used two RN blocks with three layers each block, followed by a Batch Normalization layer and a PReLU. The second part of the configuration includes a bidirectional LSTM and two dense layers.

Table 3: *Topology for RN and LSTM architecture (archC).*

Layer	Layer type	Channels	Output
	Input 57-Melfb	–	57×T
1	Conv1D	256	256×T
2	ResBlock-ReLU(x3)	160	160×T
3	ResBlock-ReLU(x3)	320	320×T
4	BatchNorm1D	–	320×T
5	PReLU1D	–	320×T
6	Bidirectional LSTM	640	640×T
7	Mean	–	1280
8	FC	–	512
9	FC+softmax	–	N
	Cross-Entropy Loss		

4.4. Deep Neural Network with External Alignment Integration

Note that mostly SV systems employ a DNN architecture with an average across time to extract embeddings which dismisses the order of the phonetic information in the utterance. However, as we are working in a text-dependent SV task is relevant to keep the temporal structure of the uttered phrase for training the system correctly, especially when the training data is limited [2, 16]. For this reason, in this architecture ($archD$), we have replaced the global average pooling by a frame-to-state alignment method as a new layer into the DNN architecture. In concrete, a Gaussian Mixture Model (GMM) with a Maximum A Posteriori (MAP) has been employed. This strategy allows us to obtain a supervector as embedding which keeps and encodes the temporal structure of the phrase and the speaker information. To train this architecture, the approximated Detection Cost Function (*aDCF*) [17] is optimized. This function is inspired by DCF [18] and was chosen since is one of the main metrics in the evaluation process for SV tasks.

Table 4: *Topology for Deep Neural Network with External Alignment Integration (archD).*

Layer	Layer type	Channels	Output
	Input 60-MFCC	–	60×T
1	Conv1D-LeakyReLU	64	64×T
2	Conv1D-LeakyReLU	128	128×T
3	Conv1D-LeakyReLU	32	32×T
4	Alignment	T×64	$32 \cdot 64$
5	Cosine	–	N
	aDCF Loss		

4.5. Training Strategies

Initially, we trained the four systems to obtain a model per phrase, so each model was only trained with the data for the correspondence phrase. However, after the first experiments, we decided to train the two bigger architectures presented in 4.1 and 4.2 with all the sentences at the same time to obtain only one model for each architecture and improve the final system performance. Furthermore, we made different combinations of acoustic features to train these two architectures.

5. Back-end

Once the speaker embeddings are extracted from the different architectures, a back-end is applied over each of them. We have employed two different alternatives in function of the training process. When the architectures are trained to produce a model for each phrase, firstly the embeddings are transformed using an LDA into a new feature space of 200 dimensions that maximizes class separability. After this transformation, mean and length normalization are applied. To obtain the final scores, a PLDA binary detector model is trained for each phrase to determine whether pairs of examples are from the same speaker or not. On the other hand, whether the embeddings are extracted from the model trained with all the phrases, the LDA transformation is not applied, and directly we apply the trained PLDA to obtain the verification scores.

6. Results

In this section, we provide some results achieved with the fusion of our different systems, including the best result obtained, and at the same time, we have included the results for each system separately with different configurations. The system fusion adopted to obtain the primary system is a simple linear fusion obtained over the development set, extracted from the DeepMine train partition. This fusion has been made using all the individual systems. Furthermore, the single system corresponds with $archA$ using log Mel FB as features.

Table 5 presents the Equal Error Rate (EER) and NIST 2008 minimum detection cost (DCF08) [19] for the baseline systems [20, 21] and our systems.

Table 5: *Experimental results on DeepMine Database evaluation set, showing EER% and NIST 2008 min cost (DCF08). These results were obtained with train set and varying the training strategies, the acoustic features and the architectures.*

#	Training	Feat.	Arch.	Backend	EER%	DCF08
		Baseline x-vector			9.05	0.529
		Baseline i-vector/HMM			3.49	0.146
1	phrbyphr	Log Mel FB	A	LDA+PLDA	4.58	0.167
2			B		5.73	0.191
3			C		6.42	0.264
4		Corr	A		6.55	0.233
5			B		8.91	0.323
6		MFCC	D		9.41	0.470
7	allphr	Log Mel FB	A	PLDA	3.49	0.129
8			B		3.65	0.128
9		Corr	A		4.72	0.182
10			B		4.71	0.179
		Single system (progress set)			**4.59**	**0.167**
		Single system (evaluation set)			**4.58**	**0.167**
		Primary system (progress set)			**2.56**	**0.088**
		Primary system (evaluation set)			**2.61**	**0.088**

7. Acknowledgements

This work has been supported by the Spanish Ministry of Economy and Competitiveness and the European Social Fund through the project TIN2017-85854-C4-1-R, by the Government of Aragon (Reference Group T36_20R) and co-financed with Feder 2014-2020 “Building Europe from Aragon”, and by Nuance Communications, Inc. The Titan V used for this research was donated by the NVIDIA Corporation.

8. 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] V. Mingote, A. Miguel, A. Ortega, and E. Lleida, “Supervector Extraction for Encoding Speaker and Phrase Information with

Neural Networks for Text-Dependent Speaker Verification," *Applied Sciences*, vol. 9, no. 16, p. 3295, 2019.

[3] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[4] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[5] G. Lample, A. Sablayrolles, M. Ranzato, L. Denoyer, and H. Jégou, "Large memory layers with product keys," in *Advances in Neural Information Processing Systems*, 2019, pp. 8546–8557.

[6] I. Viñals, D. Ribas, V. Mingote, J. Llombart, P. Gimeno, A. Miguel, A. Ortega, and E. Lleida, "Phonetically-Aware Embeddings, Wide Residual Networks with Time-Delay Neural Networks and Self Attention Models for the 2018 NIST Speaker Recognition Evaluation," *Proc. Interspeech 2019*, pp. 4310–4314, 2019.

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

[8] A. Nagrani, J. S. Chung, and A. Zisserman, "VoxCeleb: A Large-Scale Speaker Identification Dataset," in *Proc. Interspeech 2017*, 2017, pp. 2616–2620.

[9] J. S. Chung, A. Nagrani, and A. Zisserman, "VoxCeleb2: Deep Speaker Recognition," in *Proc. Interspeech 2018*, 2018, pp. 1086–1090.

[10] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An ASR corpus based on public domain audio books," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 5206–5210.

[11] 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.

[12] H. Zeinali, L. Burget, and J. Cernocky, "A Multi Purpose and Large Scale Speech Corpus in Persian and English for Speaker and Speech Recognition: the DeepMine Database," in *Proc. ASRU 2019 The 2019 IEEE Automatic Speech Recognition and Understanding Workshop*, 2019.

[13] M. Tan and Q. V. Le, "Efficientnet: Rethinking model scaling for convolutional neural networks," *arXiv preprint arXiv:1905.11946*, 2019.

[14] M. Tan, R. Pang, and Q. V. Le, "Efficientdet: Scalable and efficient object detection," *arXiv preprint arXiv:1911.09070*, 2019.

[15] G. Lavrentyeva, S. Novoselov, E. Malykh, A. Kozlov, O. Kudashev, and V. Shchemelinin, "Audio replay attack detection with deep learning frameworks." in *Interspeech*, 2017, pp. 82–86.

[16] V. Mingote, A. Miguel, A. Ortega, and E. Lleida, "Optimization of the area under the ROC curve using neural network supervectors for text-dependent speaker verification," *Computer Speech & Language*, vol. 63, p. 101078, 2020.

[17] V. Mingote, A. Miguel, D. Ribas, A. Ortega, and E. Lleida, "Optimization of False Acceptance/Rejection Rates and Decision Threshold for End-to-End Text-Dependent Speaker Verification Systems," *Proc. Interspeech 2019*, pp. 2903–2907, 2019.

[18] A. Martin and M. Przybocki, "The NIST 1999 speaker recognition evaluation—An overview," *Digital signal processing*, vol. 10, no. 1-3, pp. 1–18, 2000.

[19] "The NIST Year 2008 Speaker Recognition Evaluation Plan," 2008. [Online]. Available: https://www.nist.gov/sites/default/files/documents/2017/09/26/sre08_evalplan_release4.pdf

[20] H. Zeinali, L. Burget, H. Sameti, O. Glembek, and O. Plchot, "Deep Neural Networks and Hidden Markov Models in i-vector-based Text-Dependent Speaker Verification," in *Odyssey 2016*, 2016, pp. 24–30.

[21] H. Zeinali, H. Sameti, and L. Burget, "HMM-based phrase-independent i-vector extractor for text-dependent speaker verification," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 7, pp. 1421–1435, 2017.