Team 25: Unsupervised Autoregressive Model as a Shared Encoder for Text-Dependent Automatic Speaker Verification

Vijay Ravi, Ruchao Fan, Amber Afshan, Huanhua Lu, Abeer Alwan

University of California Los Angeles, USA

(vijaysumaravi,fanruchao,amberafshan,huanhua,alwan)@ucla.edu

Abstract

In this paper, we describe the system used by Team 25 for task 1 of Short Duration Speaker Verification Challenge (SDSVC, 2020) [1]. A shared-encoder with task-specific decoders is proposed to address text-dependent automatic speaker verification (TD-ASV). An autoregressive predictive coding (APC) encoder is pre-trained in an unsupervised manner using both out-of-domain (LibriSpeech, VoxCeleb) and in-domain (DeepMine) unlabeled datasets to learn generic, high-level feature representation that encapsulates speaker and phonetic content. Two task-specific decoders were trained using labeled datasets to classify speakers (SID) and phrases (PID). Speaker embeddings extracted from the SID decoder were scored using a PLDA. SID and PID systems were fused at the score level. There is a 51.9% relative improvement in minDCF for our system compared to the fully supervised x-vector baseline on the cross-lingual DeepMine dataset. A fusion of the x-vector/PLDA baseline and the SID/PLDA scores prior to PID fusion further improved performance by 15% indicating complementarity of the proposed approach to the x-vector system.

Index Terms: speaker verification, unsupervised-learning, feature-representation, shared-encoder, domain-adaptation.

1. Shared Encoder-Decoder Architecture

1.1. Autoregressive Predicitive Coding (APC) Encoder

Predictive coding has played an important role in speech processing, especially in speech coding using linear prediction coding (LPC) [2]. LPC predicts future audio samples whereas, a recently proposed autoregressive predictive coding [3] predicts the features of a future frame. The idea is to utilize the input sequence itself as labels and predict a frame n steps ahead of the current frame to achieve unsupervised speech representation learning. The model architecture is as shown in Figure 1.

Suppose the input speech sequence is $\mathbf{X} = (x_1, x_2, ..., x_T)$, the time shift of prediction is fixed at n, and the ground truth of the prediction for each frame is $(x_{1+n}, x_{2+n}, ..., x_{T+n})$. In order to prevent the model from learning a trivial solution, we apply a uni-directional neural network structure, as opposed to bi-directional networks, by letting the model be aware of the context only from history. By stacking multiple long short-term memory (LSTM) layers and adding residual connections, we obtain a deep LSTM network. Prior to that, a two-layer feed-forward network is considered as the pre-net network to transform the speech features into a hidden latent space. Together with LSTMs, we denote this combined network as DLSTM. The output of the DLSTM is then fed into a linear layer and transferred to the input space, which means that the dimension will be the same as the input features. Mathematically, the model architecture can be described as follows:

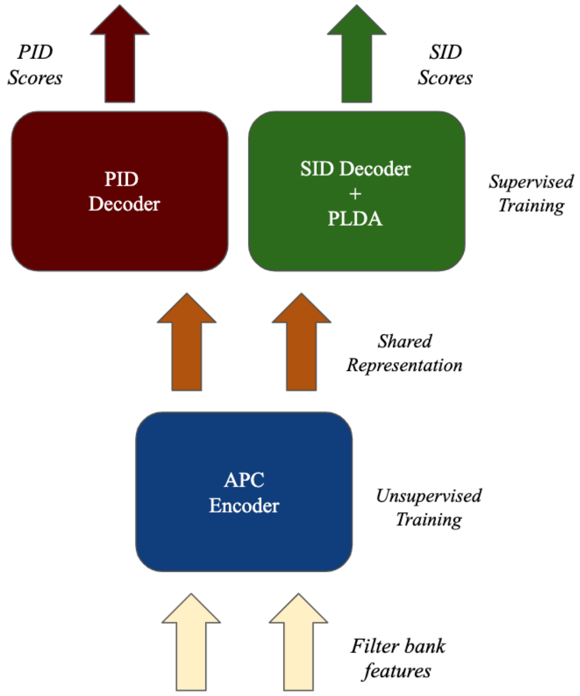

Figure 1: *Encoder-Decoder model architecture proposed in this paper. The encoder is an APC model trained in an unsupervised way to learn a generic, high-level feature representation independent of downstream tasks. The decoders (PID and SID) are trained in a supervised manner.*

$$\mathbf{Y} = W_f DLSTM(\mathbf{X}, W_{lstm}) + b_f \tag{1}$$

where W_{lstm} represents all the parameters in the DLSTM; W_f and b_f denote the weight matrix and bias vector in the last layer, respectively; and $\mathbf{Y} = (y_1, y_2, ..., y_T)$ is the output. Considering the L1 loss as a metric distance for prediction, all the above parameters are obtained by optimizing the following loss function:

$$L_1 = \sum_{t=1}^{T-n} |x_{t+n} - y_t| \tag{2}$$

1.2. Task-specific Decoder

The PID decoder is designed to distinguish between different phrases. In order to obtain a better generalization, we first allow the model to learn a frame-level phonetic representation with a connectionist temporal classification (CTC) [4]. Specifically, the frame-level phonetic representations are calculated by feeding the speech representations in Section 1.1 into a stacked bi-directional LSTM network(BLSTM). Then, the frame-level phonetic representations are transformed into the phoneme space, which consists of 39 phonemes and one blank symbol. With respect to phrase classification, the frame-level phonetic representations are averaged using a statistical pooling layer to form a single feature vector. Two feed-forward layers are used to transcribe the vector to output phrase-ID space, followed by a softmax layer. The overall network is optimized by jointly minimizing the CTC loss and the cross entropy loss:

$$\mathcal{L}_{total} = \mathcal{L}_{CTC} + \lambda \mathcal{L}_{CE}, \tag{3}$$

where, $\mathcal{L}_{CTC}$ is the loss from the prediction of the phonemes and $\mathcal{L}_{CE}$ is the loss arising from the classification of the phrase. λ is a parameter used to control the contribution of the CE loss to the total loss.

The speaker-ID decoder consists of another BLSTM network followed by a statistical pooling layer to extract speaker embeddings. Speech representations obtained from the APC encoder in Section 1.1 are used here as input. The size of the final transformation layer is dependent on the number of speakers in the dataset. The speaker ID decoder is optimized by minimizing the cross entropy loss arising from the classification of speakers.

2. Experimental Details

2.1. Datasets

The specifications of the datasets used in this paper are provided in Table 1. Utterances from LibriSpeech, VoxCeleb1 and VoxCeleb2 [5] and DeepMine Part-1 [6, 7] were used for three different tasks: 1) Unsupervised pre-training of the shared encoder, 2) Phrase ID training, and 3) Speaker ID training. In this section, we provide details of the subsets of data used for each task.

Table 1: *Details of the datasets used.*

Subset	Database	# Utts	# Spks	Duration (in hours)
train-librispeech	Librispeech	140k	5466	478.5
dev-librispeech	Librispeech	2.7k	97	5.3
train-voxceleb	VoxCeleb	1.2M	7350	2637.8
dev-voxceleb	VoxCeleb	73k	7350	151.2
train-deepmine	DeepMine	101k	963	91.5
dev-deepmine	DeepMine	37k	NA	31.6
test-deepmine	DeepMine	69k	NA	61.2

The in-domain training data (*train-deepmine*) contains speech utterances from 963 speakers, some of whom have only Persian phrases. The enrollment (*dev-deepmine*) and test utterances (*test-deepmine*) are drawn from a fixed set of ten phrases consisting of five Persian and five English phrases, respectively. More details of the phrases can be found in [6].

2.1.1. Unsupervised Pre-training of Shared Encoder

The unsupervised pre-training of the shared encoder used the out-of-domain *train-librispeech* subset, 500k utterance from VoxCeleb and the in-domain *train-depmine* subset. Since the APC encoder can be trained with unvoiced frames as well, no speech activity detection (SAD) is applied. A uniform sampling rate of 16 KHz is used across datasets. To prevent overfitting, a combined development set consisting of *dev-librispeech*, *dev-voxceleb* and *dev-deepmine* were used for hyperparameter selection.

2.1.2. Task Specific Decoder Training

For training the phrase ID decoder, 100 hours of LibriSpeech and all utterances of *train-deepmine* were used. *dev-librispeech* and the *dev-deepmine* dataset were used for hyperparameter selection.

The SID decoder was trained using 1.2M utterances (7350 speakers) from the VoxCeleb dataset. Similar to the data processing of the x-vector system in [8], the utterances were cut into 3 second segments and augmented with noise from the MUSAN database [9] resulting in a total of 3.2M utterances ($\sim$ 7k hours).

2.2. Front-End Processing

The Kaldi framework [10] was used for all front-end preprocessing and feature extraction for each of the three tasks. The features are 40 dimensional filterbanks with a frame-length of 25ms and a frame shift of 10ms. Cepstral mean and variance normalization is applied on the features. The energy SAD (from Kaldi), used in the speaker embedding extraction, filters out non-speech frames.

2.3. Model Architecture

2.3.1. APC Encoder

The APC encoder DLSTM is composed of 4 layers of unidirectional LSTMs with each layer consisting of 512 hidden units. The input to the shared-encoder is 40 dimensional filter-bank features. The shared encoder is trained in an auto-regressive manner by minimizing the L1 loss function as described in Section 1.

The pre-net feature embedding network of the encoder DLSTM is made up of 2 fully-connected layers with ReLU activations. The encoder model is initialized using the Xavier uniform initialization and a dropout of 0.1 is applied to the ReLu activation function.

During evaluation, the shared-encoder is used as a feature extractor to extract learned representations for each utterance. These feature representations are the hidden RNN states of the APC model and form a 4-dimensional tensor of the shape (number-layers, batch-size, sequence-length, RNN-hidden-size). In our experiments, 512 dimensional hidden states of all 4 RNN layers of the APC model were used. Features extracted from the APC model are then fed into the task-specific decoder for learning the corresponding speaker and phrase identities.

2.3.2. Task Specific Decoders

Two standalone decoders are trained to classify speech utterances based on speakers and phrase-IDs. Each decoder is trained and evaluated separately.

The phrase ID (PID) decoder is composed of 3 layers of bidirectional LSTMs made up of 512 hidden units. The output of these BLSTM layers is then fed into two different sub-networks to predict phonemes and classify phrases. The phoneme prediction sub-network is a 40 dimensional (39 phonemes + 1 blank space) linear layer. The phrase classification sub-network consists of a pooling layer followed by a fully-connected layer (400 hidden units) and a prediction layer of 11 outputs (10 phrases + 1 no match). Since we utilize out of domain data which do not have phrase-ID labels, we add an extra category for all utterances whose contents do not match the given 10 phrase-IDs of the evaluation data. The value of λ is heuristically set to 0.2.

The speaker ID decoder is made up of 3 layers of bidirectional LSTMs each consisting of 512 hidden units. This is followed by statistical pooling, a fully-connected (dense) layer, and a prediction layer. The dimension of the prediction layer 7350 based on the number of speakers in the training set. During evaluation, the bottleneck features (outputs from the dense layer of the SID decoder) are extracted and used as speaker embeddings. The dimension of the fully-connected dense layer is set at 600 similar to the x-vector system.

2.4. Model Training and Evaluation

The shared encoder was trained for 5 epochs with a learning rate of $2e^{-4}$. The weights and biases of the shared-encoder network were frozen after the training to ensure that the task-specific optimization of the decoders did not modify the shared-encoder the. Both the phrase ID and the speaker ID decoder networks were trained in parallel to minimize their corresponding loss functions. Decoders were trained for 5 epochs with a learning rate of $2e^{-4}$ and the learning rate was annealed by a factor of 0.5 after 3 epochs.

During evaluation,the log likelihood of phrase-ID of test utterance and the corresponding enrollment utterance being the same is computed as the PID score. Speaker embeddings are extracted from the dense layer of the SID decoder. A PLDA classifier is used to compare the extracted speaker embeddings, and predict target/imposter speaker decisions. Speaker embeddings extracted from the speaker ID decoder were centered and projected using LDA. The LDA dimension was tuned on the VoxCeleb training set to 200. After dimensionality reduction, the representations were length-normalized and modeled by the PLDA and the PLDA model was then adapted using the DeepMine training data. The log-likelihood scores of the PLDA model (SID scores) and the PID model were fused to generate the final system prediction.

3. Results and Discussion

Table 2 provides results obtained from the text-dependent speaker verification task of SDSVC. System performance is compared using the normalized minimum detection cost function (minDCF) [11]. We also report the equal error rate (EER).

Two baselines were provided in the challenge evaluation plan for this task: the x-vector system and i-vector/HMM system. The state-of-the-art x-vector method, based on the TDNN architecture of [8], was trained using VoxCeleb1 and VoxCeleb2 databases. Evaluation trials, as per the provided baseline, were scored using the PLDA without any score normalization. The i-vector/HMM method, that also takes into consideration phrase information, was selected as the second baseline. Among the published results, the i-vector/HMM method is the best performing system on DeepMine data.

The proposed system achieves a minDCF of 0.2697 and an EER of 6.28%. This represents a relative improvement of 51.9% in terms of minDCF (0.5611 for the x-vector baseline versus 0.2697 for the proposed method) and 38% in terms of EER (10.13% to 6.28%). In order to have a fair comparison between the x-vector system and the shared-encoder system, we fused the scores of x-vectors and PID. We observed that,

Table 2: *Results for text-dependent task of the SDSV challenge in terms of minDCF and EER.* * *indicates baseline and* $+$ *indicates score-level fusion using linear regression.*

Speaker ID System	**Phrase ID System**	**minDCF**	**EER (%)**
x-vector*	None	0.5611	10.13
i-vector*	HMM	0.1472	3.47
x-vector	PID	0.2170	4.80
SID	PID	0.2697	6.28
SID + x-vector	PID	0.1830	4.18

in this case, the performance of the fused x-vectors was better than the shared encoder system. The minDCF improved relatively by 19.5% (from 0.2697 to 0.2170) and the EER by 23.5% (from 6.28% to 4.8%). Thus, the x-vector system, on its own, is better at capturing speaker discriminatory features, than the SID network of the proposed framework. Nevertheless, on the overall task of TD-ASV, the proposed system performs better than the x-vector baseline. This improvement in performance can be attributed to the unsupervised pre-training of the shared-encoder using unlabeled in-domain data and the use of phonetic information by the proposed system. As a result, our system is better suited for the text-dependent, cross-lingual task of this challenge in comparison to the x-vector baseline.

To further analyze the performance of the proposed system, fusion of the x-vector/PLDA scores and the SID/PLDA scores was performed using linear regression before fusing with PID scores. Equal coefficients of 0.5 were chosen for this linear regression which resulted in a 15% gain in minDCF (0.2170 to 0.1830) and a 12% relative gain in EER (4.8% to 4.18%). These results seem to suggest that the SID system offers complimentary information to the x-vector system. It is possible that the proposed unsupervised method learns useful speaker-discriminative information that was previously discarded when learning representations in a supervised fashion. Combining supervised and unsupervised feature representations can therefore be advantageous in developing robust TD-ASV systems.

The performance of the i-vector/HMM method, on the other hand, exceeded that of the proposed method by 45% (minDCF of 0.1472 vs 0.2697). This system used hidden Markov model (HMM) states to model time sequences and extract i-vectors for each phrase. The i-vector/HMM approach outperforms the proposed method mainly because of its capability to reject target-wrong trials, meaning that if two different phrases were spoken by the same speaker, the HMM Viterbi decoding produced invalid statistics for such trials and consequently they were rejected easily [12]. In contrast, since the PID and the SID systems were fused by a simple score-level fusion, our system may have predicted higher log-likelihoods.

4. Conclusions

In this paper, a novel model architecture comprised of a shared-encoder with task-specific decoders was proposed for TD-ASV. An auto-regressive predictive coding encoder was trained in an unsupervised fashion to learn generic features independent of the downstream task. Task-specific decoders were then optimized for phrase and speaker classification. An improvement of 52% was achieved in terms of minDCF compared to the x-vector baseline.

5. Acknowledgements

This study was supported in part by the NSF.

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. O'Shaughnessy, "Linear predictive coding," *IEEE potentials*, vol. 7, no. 1, pp. 29–32, 1988.

[3] Y.-A. Chung, W.-N. Hsu, H. Tang, and J. Glass, "An unsupervised autoregressive model for speech representation learning," *arXiv preprint arXiv:1904.03240*, 2019.

[4] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd international conference on Machine learning*, 2006, pp. 369–376.

[5] A. Nagrani, J. S. Chung, and A. Zisserman, "Voxceleb: a large-scale speaker identification dataset," in *INTERSPEECH*, 2017.

[6] 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 *Odyssey*, 2018, pp. 386–392.

[7] H. Zeinali, L. Burget, J. Černockỳ *et al.*, "A multi purpose and large scale speech corpus in persian and english for speaker and speech recognition: the deepmine database," *arXiv preprint arXiv:1912.03627*, 2019.

[8] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.

[9] D. Snyder, G. Chen, and D. Povey, "Musan: A music, speech, and noise corpus," *arXiv preprint arXiv:1510.08484*, 2015.

[10] 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," in *IEEE 2011 workshop on automatic speech recognition and understanding*, no. CONF. IEEE Signal Processing Society, 2011.

[11] A. F. Martin and C. S. Greenberg, "The nist 2010 speaker recognition evaluation," in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[12] 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.