End-to-End Language Identification Using a Residual Convolutional Neural Network with Attentive Temporal Pooling

João Monteiro[1,2], Jahangir Alam[1,2], Gautam Bhattacharya[2], Tiago H. Falk[1]
[1]*Institut National de la Recherche Scientifique (INRS-EMT)*
[2]*Centre de Recherche Informatique de Montréal (CRIM)*
Montreal - Canada
joao.monteiro@crim.ca, jahangir.alam@crim.ca, gautam.bhattacharya@crim.ca, falk@emt.inrs.ca

***Abstract*—In this work, we tackle the problem of end-to-end language identification from speech. To this end, we propose the use of a residual convolutional neural network aiming at exploiting the ability of such architectures to take into account large contextual segments of input data. Moreover, in order for variable input lengths to be supported by the proposed setting, a self-attention mechanism is employed on top of the final convolutional layer. This results in a learnable temporal feature pooling scheme that allows for embedding varying duration utterances into a fixed dimension space. Evaluation is performed on data containing ten oriental languages under different test conditions, namely: short-duration recordings, confusing languages trials, as well as a set of trials in which non-target unseen languages are included. End-to-end evaluation of the proposed framework is thus shown to significantly outperform well-known benchmark methods under considered evaluation conditions.**

***Index Terms*—Language identification, Residual convolutional neural networks, Attentive features pooling.**

I. Introduction

Being able to identify spoken languages from speech data is a useful feature in several applications of speech processing. Language recognizers can be used for conditional prediction or hierarchical modelling on downstream tasks in a given pipeline. Speech recognition or speaker verification, for instance, are examples of such cases in which prior information of spoken language will likely boost performance. Commonly, language identification (LID), i.e. identifying the spoken language from a given speech example under the assumption a single language is present, is tackled using similar approaches to speaker verification/recognition [1].

Well known i-vectors [2], for instance, are obtained by first computing a universal background model, which is commonly a Gaussian mixture model, followed by factor analysis on top of statistics of the latents with the aim at obtaining a low dimensional representation that embeds both channel- and speaker-dependent information. Classification is performed with probabilistic linear discriminant analysis (PLDA) [3]. Alternatively, neural networks have been applied in recent years to substitute some components of speaker/language recognition frameworks, such as generating alternative low-dimensional embeddings or performing recognition in an end-to-end fashion, thus eliminating the need of a post-trained binary classifier. Recently proposed x-vectors [4], for example, leverage feed-forward neural networks operating in different time scales to compute low-dimensional embeddings from utterances of varying lengths. Follow-up approaches, in turn, have extended the idea of including context by employing convolutional neural networks across time [5], [6], i.e. performing 2-dimensional convolutions over time-frequency representations of speech, such that full time-dependent information is taken into account for low-dimensional computation, rather than having only short-term time-dependencies modelled through contextual frames, as is the case for x-vectors.

Training in both aforementioned cases has been performed for speaker recognition, i.e. the model is used as a classifier aiming to identify the speaker in a given utterance. The softmax layer outputs parameterize a conditional multinoulli or categorical distribution over speakers and parameters are learned via maximum likelihood estimation through minimization of the cross-entropy loss. At test time, outputs of intermediate layers are used as representations on top of which a binary classifier can be trained for speaker verification for both open- and closed- set testing conditions.

In this contribution, we introduce a convolutional neural network aiming to perform language identification. Language-dependent long-term dependencies are modeled by: (i) convolutions in the time dimension, and (ii) a self-attention layer [7] which weighs last layer time-steps for weighted statistics pooling. Furthermore, aiming to enforce language dependency on models outputs, we add triplet-loss along with the maximum likelihood criterion previously described. The proposed method is evaluated on a dataset composed by recordings from ten oriental languages and shows relevant improvements over strong baselines on several test conditions. Moreover, end-to-end evaluation is also carried out showing that directly

The authors wish to acknowledge funding from the National Research Council of Canada (NRC) through the Canadian Indigenous Languages Technology project under contract 909859, and from the Natural Sciences and Engineering Research Council of Canada (NSERC) through contract/grant RGPIN-2016-4175, and RGPAS-493010-2016. Any opinions, findings, conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect those of the NRC and NSERC.

978-9-0827-9703-9/19/$31.00 ©2019 IEEE

utilizing our model's outputs as scores, i.e. discarding PLDA, matches i-vectors+PLDA's results.

II. Background

A. Residual learning

Residual architectures have been part of several recent relevant results using convolutional neural networks. Firstly introduced in [8], ResNets constitute a set of architectures made up of a series of so-called residual blocks, which determine how a feature transformation should differ from the identity, rather than how it should differ from zero [9]. Residual blocks' transformations present the basic form: $\hat{X} = \mathcal{F}(X) + X$. The residual term comes from the fact that the input is directly used to compute the transformation's output, which in a neural network represents a direct path for gradients to "flow" during loss backpropagation for computation of stochastic gradient descent updates. $\mathcal{F}(X)$ is generally a set of convolutional layers, followed by nonlinear activation functions and normalization layers.

Recent literature has shown that residual blocks contribute in yielding loss landscapes which are easier to train, in the sense that ill-conditioned chaotic landscape regions become less frequent when such architectural feature is employed [10]. Moreover, near identity transformations were studied in depth and guarantees were introduced for the linear and nonlinear $\mathcal{F}(X)$ cases in [11] and [9], respectively.

B. Attention mechanisms

Several attention mechanisms have been introduced recently in architectures aimed at processing temporal data. In general terms, attention blocks learn to conditionally weigh time-steps given inputs representations on some inner layer of a model. Such blocks have been shown to yield high performance across several domains including text [12], [13], speech [5], and image processing [14].

Here, we employ a simple attention scheme usually referred to as self- or intra-attention. Consider $y_{1:T}$ as a set of vectors corresponding to the outputs of a given neural network for some input. A linear transformation W is shared across all time-steps t, and applied to each y_t resulting in a set of scalars $a_{1:T}$, according to:

$$a_t = \tanh\left(W y_t\right). \tag{1}$$

A set of normalized weights summing up to 1 is obtained through the softmax operator:

$$w_t = \frac{e^{a_t}}{\sum_{t=1}^{T} e^{a_t}}, \tag{2}$$

and the attention layer output is finally given by:

$$y = \sum_{t=1}^{T} w_t y_t. \tag{3}$$

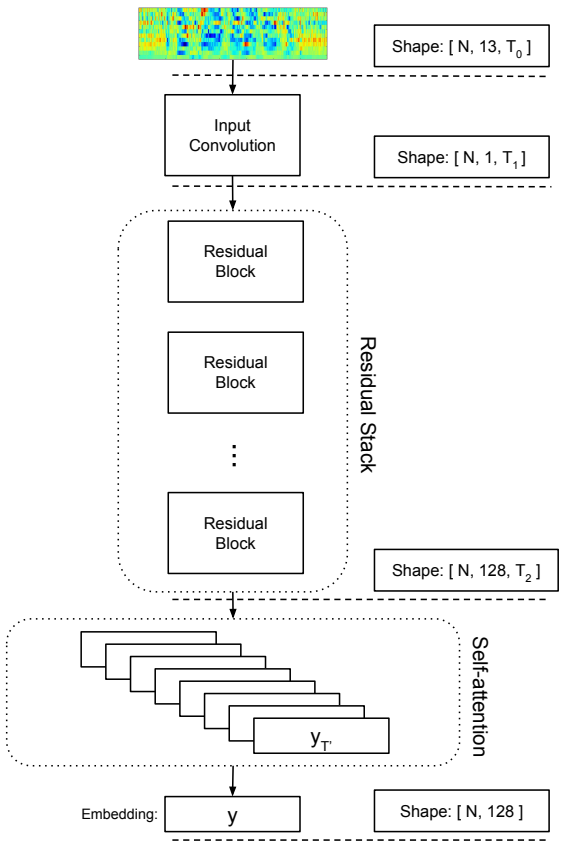

Fig. 1. Model employed for language recognition.

III. Proposed Model

We introduce a convolutional architecture aiming to take long-term contextual information into account, an inherent feature of stacked convolutional layers. We thus propose a residual architecture consisting of a modified ResNet-50 [8] in which 1-dimensional convolutional layers are employed. Inputs correspond to 13 Mel-frequency cepstral coefficients obtained from incoming speech, which are initially treated as single-channel images. An input convolutional layer is responsible for shrinking the cepstral coefficients dimension to 1, and all following convolutional layers operate over the time dimension only. Self-attention is then employed on top of the last convolutional layer, which means the channels dimension corresponding to the last temporal convolution layer gives the final embeddings size, once time-steps are pooled using the self-attention mechanism described in Section II-B. A diagram describing the proposed model is presented in Figure 1.

We further highlight that, differently from other approaches that employ *causal* convolutions for temporal dependency modelling [15], the setting explored herein assumes access to full recordings for computation of each output time-steps. This allows us to compute fixed dimensional language-dependent embeddings relying on full recordings.

During training, two strategies are combined so as to enforce language dependency on embeddings y. We directly

train the model for classification by projecting y into an output layer using a fully connected additional layer, and train the model via maximum likelihood estimation, i.e. with multi-class cross-entropy minimization, as commonly done for speaker recognition [5], [6], [16]. Moreover, aiming to enforce language discriminability, triplet-loss minimization is jointly performed on top of y along with maximum likelihood estimation, employing a distance metric based on the cosine similarity. The most common definition of triplet-loss is given by:

$$T = \max(d_+ - d_- + \alpha, 0), \tag{4}$$

where d_+ and d_- correspond to a distance measure between pairs of embeddings obtained from recordings of the same, and from different languages, respectively. Parameter α is a hyperparameter commonly referred to as margin.

The $\max(x, 0)$, $x \in \mathbb{R}$, operator is used so that triplets, i.e. pairs corresponding to the same and different languages, respectively, that already have low d_+ and high d_- stop influencing training. Here, we follow the approach in [17] and enforce concentration of same language embeddings by using a soft-margin variation of the triplet-loss given by:

$$T = \text{softplus}(d_+ - d_-), \tag{5}$$

where the softplus operator for an argument $x \in \mathbb{R}$ is defined as:

$$\text{softplus}(x) = \log(1 + e^x). \tag{6}$$

Different works have proposed triplet-loss variations using several distances d. Here we employ:

$$d(y_1, y_2) = 1 - \frac{y_1 \cdot y_2}{||y_1||_2||y_2||_2}, \tag{7}$$

where the second term is the cosine of the smallest angle between y_1 and y_2.

Our final training loss is thus defined as the sum of the multi-class cross-entropy with the triplet-loss, and the cross-entropy term will be given by:

$$CE = \log\Big[\sum_{k \in \mathcal{Z}} \exp s_k\Big] - s_z, \tag{8}$$

where $\mathcal{Z}$ is the set of training speakers, and $s_k, k \in \mathcal{Z}$, represents the score corresponding to speaker k, while s_z is the score for the correct speaker in the input recording. Scores are obtained through a fully connected layer using embeddings y as its inputs. Our final training loss will be thus the sum of the two described components: $\mathcal{L} = CE + T$.

IV. Experimental Setup and Results

A. Training

We perform gradient-based minimization of the sum of cross-entropy and soft-margin triplet losses. RMSProp [19] is employed for optimization with its smoothing constant set to 0.99. The global learning rate starts at 0.001 and is halved once the classification error rate, measured on a validation set held out of training, plateaus for 30 epochs. Training is carried out in a single Titan X NVIDIA GPU, with minibatches of size 64. Minibatches are constructed such that two random recordings of each language are sampled sequentially to form same language pairs (positive), and a random recording from a different language is selected to compose the different languages pair (negative). One epoch is considered finished when each language is selected 1000 times to compose positive pairs[1].

TABLE I
Performance comparison of proposed system (last three rows) and benchmarks based on equal error rate (%) and average cost performance (C_{avg}).

	Short-duration		*Full-length*	
	EER (%)	C_{avg}	EER (%)	C_{avg}
i-Vector+LDA [18]	18.04	0.1784	6.12	0.0598
i-Vector+PLDA [18]	17.51	0.1746	5.86	0.0596
TDNN [18]	14.04	**0.1282**	11.31	0.1034
LSTM [18]	15.92	0.1452	12.76	0.1154
LDA+PLDA	14.14	0.1361	3.59	0.0343
LDA+Cosine	15.05	0.1465	4.47	0.0435
End-to-end	**13.26**	0.1291	**2.76**	**0.0257**

B. Dataset

We evaluate our proposed framework using the dataset introduced for the AP18-OLR Challenge [18], which consists of recordings corresponding to speech in 10 different oriental languages. Information about speaker identity, gender, or age were not utilized, nor was phonetic information.

The AP18-OLR database is divided into three subsets, namely train, development and evaluation sets. We further introduce multi-condition training data by augmenting the original train partition with supplementary noisy speech, created by corrupting original examples adding reverberation (reverberation time varies from 0.25s - 0.75s), noise at signal-to-noise ratio (SNR) ranging from 0 to 15 dB, as well as by adding background noise such as music (SNR within 5-15dB), and babble (SNR within 10-20dB). Noise signals were taken from the MUSAN corpus [20] and the room impulse responses (RIRs) used to simulate reverberant effects were taken from [21]. The described data augmentation procedure increases the amount and diversity of train data, which helps in avoiding overfitting of employed models. Moreover, we empirically observed better performance when silence segments were kept in both training and test recordings, thus voice activity detectors are not required. We hypothesize our model is able to learn pause patterns in different languages.

C. Results and discussion

We start the evaluation of our proposed model and training scheme using the development partition of the AP18-OLR Challenge, which corresponds to the evaluation data for the previous year challenge. Trials lists are provided along with

[1]Code is available at: https://github.com/joaomonteirof/e2e_LID

TABLE II
PERFORMANCE COMPARISON OF PROPOSED SYSTEM (LAST THREE ROWS) AND BENCHMARKS BASED ON EQUAL ERROR RATE (%) AND AVERAGE COST PERFORMANCE (C_{avg}). CONFUSING LANGUAGES CORRESPOND TO CANTONESE, KOREAN, AND MANDARIN.

		Short-duration		*Confusing languages*		*Unseen non-target languages*	
		EER (%)	C_{avg}	EER (%)	C_{avg}	EER (%)	C_{avg}
Benchmarks	i-Vector+Cosine	18.02	0.1780	10.71	0.1069	7.77	0.0577
	i-Vector+PLDA	17.50	0.1743	10.66	0.1059	7.51	0.0524
	Tandem+Cosine	15.73	0.1502	13.81	0.1387	8.98	0.0683
	Tandem+PLDA	15.30	0.1461	13.33	0.1324	8.37	0.0596
	LSTM+Cosine	20.10	0.1978	9.11	0.0840	7.78	0.0537
	LSTM+PLDA	19.14	0.1896	8.78	0.0819	7.49	0.0490
	LSTM	24.00	0.2321	7.54	0.0738	7.57	0.0491
Proposed	LDA+Cosine	14.63	0.1432	9.81	0.0967	6.44	0.0463
	LDA+PLDA	13.48	0.1328	8.28	0.0810	5.97	0.0369
	End-to-end	**12.62**	**0.1246**	**6.80**	**0.0669**	**5.65**	**0.0315**

data for both short-duration condition, i.e. recordings with 1 second duration, and full-length test recordings. We thus proceed to evaluation on the test partition which includes three conditions, namely short-duration recordings, trials from known confusing languages only, and a corrupted list of trials in which recordings from languages not represented within train data are included. Three strategies are used for scoring trials for our models: (a) PLDA trained on the embeddings of the full set of training data; (b) The cosine similarity between enrollment language models obtained by averaging embeddings of all training recordings from a given target language, and the embedding of the test recording; (c) End-to-end: The output of the softmax layer corresponding to the claimed language is used as score. Linear discriminant analysis (LDA) was further used to reduce the dimensionality of embeddings from 128 to 64 in the case of PLDA and cosine scoring.

For performance reference on development data (Table I), results, as reported in [18], obtained with i-vectors [2] using both LDA and PLDA backends are used as baselines. Moreover, two neural network based systems, a TDNN [22] and a LSTM [23] with end-to-end scoring, are also used as benchmarks. Results in terms of equal error rate (EER) and average cost performance (C_{avg}) are reported in Table I. More details about both metrics can be found in [18]. Results on evaluation data are presented in Table III. In that case, we provide further results from three benchmark systems, namely: i-vectors, statistics of a GMM-UBM trained on tandem features [24], and a convolutional-recurrent model consisting of 6 convolutional layers followed by a 2-layered bi-directional LSTM, trained with the same setting as the proposed model.

As per results reported in Table I for development data, one can notice that the i-vector+PLDA approach lacks robustness to short-duration recordings, which was observed also in the case of speaker verification [5]. TDNN and LSTM perform better than i-vectors in the short-duration case, which does not hold in the full-length evaluation, indicating such models are not effective on handling longer-term dependencies, due to limited context for the TDNN case, and known training difficulties in the long sequences regime faced by RNNs, including LSTMs. Our proposed approaches significantly outperform all considered baselines in the full-length evaluation, indicating the added context together with the employed attentive pooling effectively improve modelling of long-term dependencies. More importantly, for the end-to-end scoring, i.e. without the use of any extra training step after training the convolutional model, EER in both short-duration and full-length conditions are lower when compared to all evaluated benchmarks in both testing conditions.

Results in Table III for the evaluation data corroborate previous findings in that end-to-end scoring of the proposed method outperforms all compared benchmarks in all evaluation conditions. We specifically point out the fact that a good performance is achieved even when unseen languages are included and end-to-end evaluation is performed, which we attribute to the effect imposed by triplet loss minimization of enforcing both class-separability and concentration of embeddings belonging to the same class. The model trained using the proposed strategy was able to yield improvements in terms of average cost performance of 28.51%, 36.83%, and 39.88% for short-duration, confusing languages, and open-set evaluation conditions, respectively, when compared to an i-vector system with PLDA scoring. We finally highlight that PLDA and cosine similarity backends, which would be valid scoring strategies in an open-set evaluation scenario, i.e. comparing pairs of recordings regardless of the inclusion of corresponding languages within training data, are also able to outperform studied benchmarks.

V. CONCLUSION

We introduced a model along with a training strategy with the goal of performing end-to-end language identification from speech. A modified ResNet-50 is proposed along with a self-attention block, employed for temporal pooling, i.e. weighing representations in different time-steps, which further behaves as a mechanism for the model to learn long-term time dependencies. Evaluation is carried out using a dataset containing recordings corresponding to 10 oriental languages, and varying evaluation conditions. Recognition scores provide empirical evidence that the proposed setting significantly outperforms a set of well known benchmark systems.

REFERENCES

[1] N. Dehak, P. A. Torres-Carrasquillo, D. Reynolds, and R. Dehak, "Language recognition via i-vectors and dimensionality reduction," in *Twelfth annual conference of the international speech communication association*, 2011.

[2] 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.

[3] S. J. Prince and J. H. Elder, "Probabilistic linear discriminant analysis for inferences about identity," in *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, pp. 1–8, IEEE, 2007.

[4] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, "Deep neural network embeddings for text-independent speaker verification," *Proc. Interspeech 2017*, pp. 999–1003, 2017.

[5] G. Bhattacharya, J. Alam, and P. Kenny, "Deep speaker embeddings for short-duration speaker verification," in *Proc. Interspeech 2017*, pp. 1517–1521, 2017.

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

[7] C. Raffel and D. P. Ellis, "Feed-forward networks with attention can solve some long-term memory problems," *arXiv preprint arXiv:1512.08756*, 2015.

[8] 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*, pp. 770–778, 2016.

[9] P. L. Bartlett, D. P. Helmbold, and P. M. Long, "Gradient descent with identity initialization efficiently learns positive definite linear transformations by deep residual networks," *arXiv preprint arXiv:1802.06093*, 2018.

[10] H. Li, Z. Xu, G. Taylor, and T. Goldstein, "Visualizing the loss landscape of neural nets," *arXiv preprint arXiv:1712.09913*, 2017.

[11] M. Hardt and T. Ma, "Identity matters in deep learning," *arXiv preprint arXiv:1611.04231*, 2016.

[12] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, pp. 5998–6008, 2017.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[14] H. Zhang, I. Goodfellow, D. Metaxas, and A. Odena, "Self-attention generative adversarial networks," *arXiv preprint arXiv:1805.08318*, 2018.

[15] S. Bai, J. Z. Kolter, and V. Koltun, "An empirical evaluation of generic convolutional and recurrent networks for sequence modeling," *arXiv preprint arXiv:1803.01271*, 2018.

[16] 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.

[17] A. Hermans, L. Beyer, and B. Leibe, "In defense of the triplet loss for person re-identification," *arXiv preprint arXiv:1703.07737*, 2017.

[18] Z. Tang, D. Wang, and Q. Chen, "Ap18-olr challenge: Three tasks and their baselines," *arXiv preprint arXiv:1806.00616*, 2018.

[19] T. Tieleman and G. Hinton, "Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude," *COURSERA: Neural networks for machine learning*, vol. 4, no. 2, pp. 26–31, 2012.

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

[21] "Open Speech and Language Resources," 2017. http://www.openslr.org/28/.

[22] K. J. Lang, A. H. Waibel, and G. E. Hinton, "A time-delay neural network architecture for isolated word recognition," *Neural networks*, vol. 3, no. 1, pp. 23–43, 1990.

[23] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[24] M. J. Alam, P. Kenny, and V. Gupta, "Tandem features for text-dependent speaker verification on the reddots corpus.," in *INTERSPEECH*, pp. 420–424, 2016.