INTERSPEECH 2020
October 25–29, 2020, Shanghai, China

End-to-end Domain-Adversarial Voice Activity Detection

Marvin Lavechin[1], Marie-Philippe Gill[2], Ruben Bousbib[1], Hervé Bredin[3], and Leibny Paola Garcia-Perera[4]

[1] Cognitive Machine Learning team, École Normale Supérieure/INRIA, PSL, Paris, France
[2] École de Technologie Supérieure, Université du Québec, Montreal, Canada
[3] LIMSI, CNRS, Univ. Paris-Sud, Université Paris-Saclay, Orsay, France
[4] Center for Language and Speech Processing, The Johns Hopkins University, Baltimore, USA

marvinlavechin@gmail.com, bredin@limsi.fr

Abstract

Voice activity detection is the task of detecting speech regions in a given audio stream or recording. First, we design a neural network combining trainable filters and recurrent layers to tackle voice activity detection directly from the waveform. Experiments on the challenging DIHARD dataset show that the proposed end-to-end model reaches state-of-the-art performance and outperforms a variant where trainable filters are replaced by standard cepstral coefficients. Our second contribution aims at making the proposed voice activity detection model robust to domain mismatch. To that end, a domain classification branch is added to the network and trained in an adversarial manner. The same DIHARD dataset, drawn from 11 different domains is used for evaluation under two scenarios. In the *in-domain* scenario where the training and test sets cover the exact same domains, we show that the domain-adversarial approach does not degrade performance of the proposed end-to-end model. In the *out-domain* scenario where the test domain is different from training domains, it brings a relative improvement of more than 10%. Finally, our last contribution is the provision of a fully reproducible open-source pipeline than can be easily adapted to other datasets.

Index Terms: voice activity detection, domain adversarial training, sincnet, long short-term memory

1. Introduction and related work

Voice activity detection is one of the earliest building blocks of every speech processing pipeline, such as speaker recognition and speaker diarization. Learnt embeddings, able to discriminate speech segments from non-speech segments in audio recordings, might be sensitive to domain mismatch and lead to errors and bias in the subsequent processing steps.

A major assumption in the supervised learning scenario is that both the training and test data must be unbiased samples of the same underlying distribution. A mismatch between the training and the test set can lead to a high performance discrepancy. When learning from multiple sources or domains, models can specialize themselves to perform particularly well on some of these domains, and poorly on some others. A viable strategy might consist of muting this domain information to improve robustness of learnt embeddings. Indeed, robustness against different factors such as domain, noise, reverberation or speaker is a key point for practical use and fruitful deployment of most of the automated speech analysis tools.

Approaches aiming at improving invariance and robustness of learnt features have been long studied. In [1], Seltzer and Yu show that noise-aware training improves performance on a speech recognition task. In [2], Ganin and Lempitsky propose a domain adaptation approach to learn discriminative features from a labeled source domain. They ensure that these features are domain shift invariant, so that they can be applied to any unlabeled different (but related) domains. Such as ours, their architecture relies on a first branch responsible for solving the main task, and a secondary branch responsible for classifying the domain. The latter goes through a gradient reversal layer, forcing the network to extract domain-independent features.

In recent years, domain-adversarial approaches to solve such problems have gained interest and have been applied to a wide range of tasks [3, 4, 5, 6, 7, 8]. For instance, in [3], a primary task of senone classification and a secondary task of noise condition are jointly solved on an artificially noise-corrupted version of the Wall Street Journal dataset. A similar approach has been used in [4] where they study this domain-adversarial approach on the speech recognition task in a multilingual setup to extract language invariant representation. In [5], they used the same approach for extracting speaker-invariant representation for the speech emotion recognition task. However, to the best of our knowledge, it has never been shown that domain-adversarial training of DNNs could improve performances on the voice activity detection task.

In parallel, learning acoustic models directly from the raw waveform has been an active area of research [9, 10, 11]. Our approach relies on the SincNet model [11] acting as a feature extractor. We show that such approaches are also sensitive to bias such as the domain, and muting this information can help the model to extract more robust features, and therefore improve performance on new unseen domains.

2. End-to-end voice activity detection

Voice activity detection is the task of detecting speech regions in a given audio stream or recording. It can be addressed as a sequence labeling task where the input is the sequence of handcrafted or learnt feature vectors $\boldsymbol{x} = \{x_1, x_2, \dots, x_T\}$ and the expected output is the corresponding sequence of labels $\boldsymbol{y} = \{y_1, y_2, \dots, y_T\}$ where $y_t = 0$ if there is no speech at time step t and $y_t = 1$ if there is. Because processing long audio files of variable lengths is neither practical nor efficient, we rely on shorter fixed-length sub-sequences for both training and inference.

At training time, fixed-length sub-sequences $\{\boldsymbol{x_i}\}_{i=1}^{N}$ are drawn randomly from the training set to form mini-batches of size N.

As depicted in Figure 1, we propose to work directly from the waveform in an end-to-end manner, and jointly train the fea-

http://dx.doi.org/10.21437/Interspeech.2020-2285

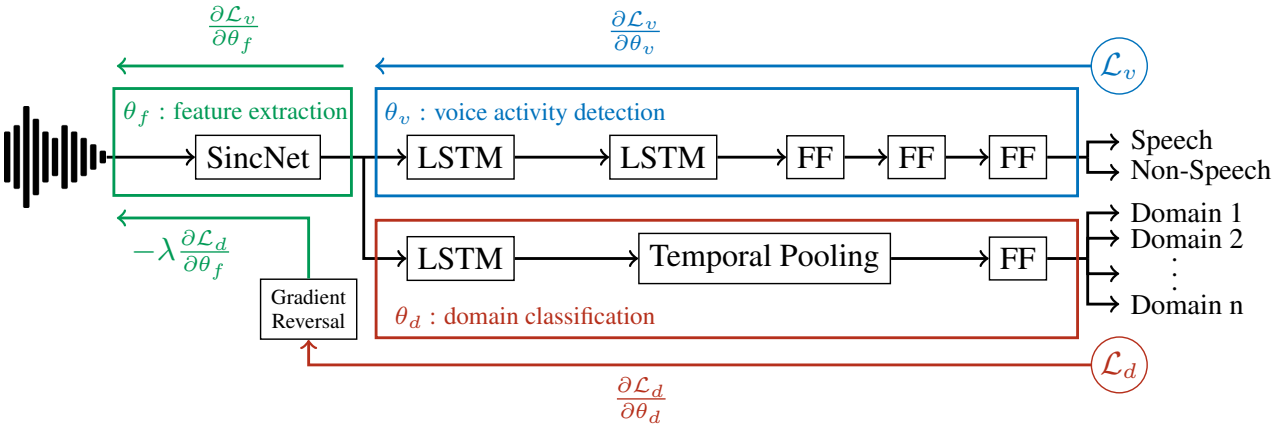

Figure 1: *Proposed architecture. The network takes the raw waveform of a 2s audio chunk as input and passes it to the part responsible for extracting the features, which is based on SincNet convolutional layers [11]. The voice activity detection branch is made of a stack of two bi-directional LSTMs, followed by three feed-forward layers. The domain classification branch is made of one uni-directional LSTM, followed by max-pooling along the time axis and one feed-forward layer outputting the probability distribution over the domains. Depending on the task, one would only use the upper branch (for regular voice activity detection), the lower branch (for domain classification), or combine both with gradient reversal (for domain-adversarial voice activity detection).*

ture extraction and sequence labeling steps. The upper branch is trained to minimize the cross-entropy loss, using standard gradient back-propagation to update the weights of the feature extraction network θ_f and the voice activity detection network θ_v (using abusive notation to highlight function composition):

$$\mathcal{L}_v(\theta_f, \theta_v) = -\frac{1}{N}\sum_{i=1}^{N}\sum_{t=1}^{T} \boldsymbol{y_{it}} \cdot \log \theta_v\left(\theta_f\left(\boldsymbol{x_i}\right)\right)_t \quad (1)$$

At test time, audio files are processed using overlapping sliding sub-sequences of the same length as the one used in training. For each time step t, this results in several overlapping sequences of prediction scores, which are averaged to obtain the final score. Finally, time steps with prediction scores greater than a tunable threshold σ are marked as speech.

3. Domain-adversarial training

In this section, we explore how to build a feature extraction network θ_f less sensitive to domain variability. Our solution consists of adding an additional branch θ_d trained in an adversarial fashion, such as proposed in [12].

The corresponding lower branch in Figure 1 is trained in conjunction with the rest of the network to minimize the mean squared error loss, MSE (using the same abusive notation as above):

$$\mathcal{L}_{\mathrm{d}}(\theta_f, \theta_d) = \frac{1}{N}\sum_{i=1}^{N} \left\| d_i - \theta_d\left(\theta_f\left(\boldsymbol{x_i}\right)\right)\right\|_2^2 \quad (2)$$

where d_i is a one-hot encoding of the domain label of $\boldsymbol{x_i}$. Instead of simply summing the two losses $\mathcal{L}_{\mathrm{v}}$ and $\mathcal{L}_{\mathrm{d}}$ as in a standard multi-task learning scenario, a gradient reversal layer [2] is added in front of the domain classification branch. The back-propagation update rules become:

$$\theta_v \leftarrow \theta_v - \epsilon\frac{\partial \mathcal{L}_v}{\partial \theta_v} \qquad \theta_d \leftarrow \theta_d - \epsilon\frac{\partial \mathcal{L}_d}{\partial \theta_d}$$
$$\theta_f \leftarrow \theta_f - \epsilon\left(\frac{\partial \mathcal{L}_v}{\partial \theta_v} - \boldsymbol{\lambda}\frac{\partial \mathcal{L}_d}{\partial \theta_d}\right)$$

where ϵ is the learning rate and λ is a scaling factor controlling the importance of the domain classification loss $\mathcal{L}_{\mathrm{d}}$ with respect to the main voice activity detection loss $\mathcal{L}_v$.

Contrary to previous studies that assume (unlabeled) target-domain data availability during training on a (labeled) source domain [12], we do not impose this restriction: we do not rely on any labeled or unlabeled target domain data during training. However, we do assume that the number and variability of domains covered by the (labeled) training set is sufficiently large. The intuition behind this statement is that the adversarial branch will reduce the space of possible feature extraction functions θ_f towards a solution that is less sensitive to domain variability; hence, making the whole voice activity detection branch $\theta_v \circ \theta_f$ robust to a new unseen domain (and removing the need to re-train the model for every new target domain).

4. Experiments

4.1. Dataset

We perform all experiments on the single-channel subset of the DIHARD dataset [13]. It contains approximately 47 hours of audio recordings, originally divided into a development set of about 24 hours of audio, and a test set of 23 hours. No training set is provided. For the purpose of this paper, we split the official development set in two parts: two thirds (16 hours, 126 files) serve as training set, the other third (8 hours, 66 files) becomes our development set. The test set (23 hours, 194 files) remains unchanged. For each subset, files are split evenly into 11 different domains covering a wide range of conditions: audio books, broadcast interview, child language, clinical, courtroom, map task, meeting, restaurant, socio-linguistic field recordings, socio-linguistic lab recordings, and web video.

Table 1: *Evaluation of voice activity detection models, in terms of detection error rate (DetER %), false alarm (FA %), and missed detection (Miss %) rates. Results on the development set are reported using small font size. We report two variants: the first one is based on handcrafted features (MFCCs) and the other one is an end-to-end model processing the waveform directly.*

	DetER %		FA %		Miss %	
Baseline [17, 13]	11.2		6.5		4.7	
MFCC [18]	10.5	10.0	6.8	5.4	3.7	4.6
Waveform	**9.9**	9.3	5.7	3.7	4.2	5.6

4.2. Evaluation metric

We use the detection error rate, such as implemented in `pyannote.metrics` [14], to evaluate our systems:

$$\text{detection error rate} = \frac{\text{false alarm} + \text{missed detection}}{\text{total}}$$

where false alarm is the duration of non-speech incorrectly classified as speech, missed detection is the duration of speech incorrectly classified as non-speech, and total is the total duration of speech in the reference.

4.3. Implementation details

Figure 1 depicts the architecture used in all experiments. For SincNet, we use the configuration proposed by the authors of the original paper [11].All long short-term memory (LSTM) and inner feed-forward (FF) layers have a size of 128 and use *tanh* activations. The learning rate is controlled by a cyclical scheduler [15], each cycle lasting for 21 epochs. All models have been trained with a batch of size 64. Data augmentation is applied directly on the waveform using additive noise extracted from the MUSAN database [16] with a random target signal-to-noise ratio ranging from 10 to 20 dB.

4.4. Evaluation protocol

For each experiment, the neural network is trained for up to 300 epochs on the training set. The development set is used to choose the actual epoch and detection threshold σ that minimize the detection error rate. We apply those optimal hyperparameters and report corresponding performance on the test set. Note that, depending on the experiment, the test set domains might have or not been seen during training and development: the latter is marked as *out-domain* in Table 2.

5. Results

5.1. End-to-end voice activity detection

Table 1 summarizes the performance of the proposed end-to-end voice activity detection approach and compares it to a variant where handcrafted features (MFCCs) are used in place of trainable ones, and to the winning submission [17] of the Second DIHARD challenge [13]. The proposed approach outperforms both of them by a significant margin, reaching a detection error rate of 9.9%.

5.2. Domain classification

Before diving into domain-adversarial training, we checked whether the domain classification branch of our model was indeed able to discriminate between domains. Hence, we trained parts of the network that correspond to the domain classification branch, that are parameters θ_f and θ_d such as depicted in Figure 1. These parameters were trained to classify domains of 2 seconds-long sub-sequences.

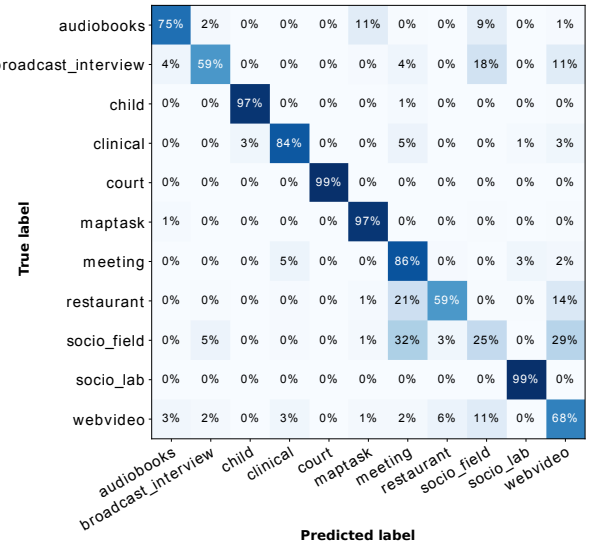

Figure 2: *Confusion matrix of domain predictions (columns) and true domains (rows) on 2 seconds-long sub-sequences of audio. Results are obtained on the test set.*

Figure 2 shows results obtained by this domain classification branch. Overall, the classifier obtained an accuracy of 77% on the test set. The model showed reasonably good performances for most domains. Worst performance was obtained for the *socio field* sub-sequences that were classified as belonging to the *meeting* domain in 32% of the cases, the *webvideo* domain in 29% of the cases, the *broadcast interview* domain in 5% of the cases, and the *socio field* domain in only 25% of the cases.

We tried different positions for the domain classification branch (after SincNet, and after the first or the second LSTM layer) but no differences were observed, neither in terms of classification performance, nor in terms of detection error rate on the speech activity detection task.

5.3. Domain-adversarial voice activity detection

Returning to the speech activity detection task, Table 2 shows performances of our domain-adversarial architecture on target domains that have been seen, or not, during the training phase.

Unsurprisingly, lines **A** and **D** show that the performances are always better when models are tested on target domains that have been seen during training. It is well-known that supervised methods are sensitive to the domain mismatch problem. In our case, this domain mismatch leads to an increase of the detection error rate of 2.1%.

Lines **A** and **B** indicate that adding more annotated data in the training set, even though they come from a different domain, leads to a significant performance boost. This translates into a decrease of 1.4% of the detection error rate.

The models trained adversarially on the domain classification task show a 12% relative improvement compared to the

Table 2: *Comparison of regular and domain-adversarial voice activity detection. Results are calculated on the test set for $\lambda = 1$. Line **A** corresponds to 11 models trained, tuned and tested separately on each of the 11 domains of the dataset. Lines **C** and **B** correspond to models trained, tuned and tested on all domains at once (with or without the adversarial branch, respectively). Line **E** and **D** correspond to models trained and tuned on 10 out of the 11 domains, and tested on the left-out domain (with or without the adversarial branch, respectively). When there is one model per domain (**A**, **D** and **E**), performances of each model for their respective domain have been aggregated across all of the domains.*

	Training & development		Inference		Metric		
	# domains	adversarial	# domains	out-domain	DetER	FA	Miss
A	1		1		11.3	6.8	4.5
B	N		N		9.9	5.7	4.2
C	N	✓	N		10.1	6.1	4.0
D	$N-1$		1	✓	13.4	9.0	4.4
E	$N-1$	✓	1	✓	11.8	7.6	4.2

model trained in the single-task learning setup (lines **D** and **E**).

Moreover, a comparison between lines **A** and **E** shows us that adversarially trained model performances are almost as good as models trained on a single domain. This constitutes a demonstration of the viability of the adversarial strategy for using models on unseen, unlabeled or slightly annotated data.

Finally, lines **B** and **C** indicate that adversarially training on the classification task does not help to improve performances on domains that have been already seen during the training. This result led us to think that the domain can contain useful information to identify speech and non-speech segments, probably for further refining boundaries between speech and noise.

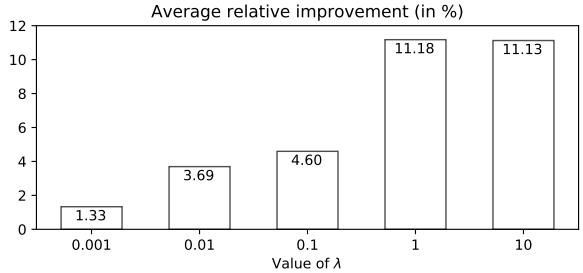

Figure 3: *Relative improvement (in %) in terms of detection error rate for different values of λ compared to the single-task voice activity detection in the leave-one-out setup.*

In Figure 3, we show relative improvement, aggregated across domains, in terms of detection error rate for different values of λ, the weight associated to the domain classification loss. Higher is the value of λ, higher is the average relative improvement, until it gets eventually too high and too much attention is given to the domain classification task. Best results are obtained for $\lambda = 1$ and $\lambda = 10$ that decrease, on average, the detection error rate by 11.15%.

6. Reproducible research

All the code has been implemented (and integrated into) using `pyannote.audio` [19], a python toolkit to build neural networks for the speaker diarization task. A Github page[1] provides instructions to reproduce results, along with additional results such as a per-domain analysis as well as ready-to-use pretrained models.

[1] https://github.com/hbredin/DomainAdversarialVoiceActivityDetection

7. Conclusions

This paper explores the learning of filters using the SincNet model [11], in conjunction with the use of domain-adversarial neural networks [12] for explicitly extracting domain-independent features.

We show that end-to-end voice activity detection leads to a significant improvement compared to models based on hand-crafted features.

Furthermore, when applied on unseen domains, the domain-adversarial multi-task learning strategy greatly improves the performances compared to the standard VAD model which does not use the domain information. Therefore, on the voice activity detection task, it seems that the strategy of muting the domain information appears as viable for extracting robust features that generalize well to new unseen domains. This method potentially reduces the need for labelled data and can improve performances on downstream tasks such as the speaker diarization or the speech recognition task that require robust voice activity detection systems.

Finally, we provide a fully reproducible open-source pipeline that can be easily adapted to other datasets as well as ready-to-use pretrained models.

8. Acknowledgements

The research reported here was conducted at the 2019 Frederick Jelinek Memorial Summer Workshop on Speech and Language Technologies, hosted at L'École de Technologie Supérieure (Montreal, Canada) and sponsored by Johns Hopkins University with unrestricted gifts from Amazon, Facebook, Google, and Microsoft. This work also benefited from the support of the Analyzing Child Language Experiences around the World (ACLEW) collaborative project ANR-17-CE28-0007 LangAge, ANR-16-DATA-0004 ACLEW, ANR-17-EURE-0017, ANR-16-CE92-0025 PLUMCOT of the French National Research Agency (ANR). We would like to thank Neville Ryant for providing the speaker diarization output of the winning submission to DIHARD 2019.

9. References

[1] M. L. Seltzer, D. Yu, and Y. Wang, "An investigation of deep neural networks for noise robust speech recognition," in *ICASSP*, 2013, pp. 7398–7402.

[2] Y. Ganin and V. S. Lempitsky, "Unsupervised domain adaptation by backpropagation," in *ICML*, 2015.

[3] Y. Shinohara, "Adversarial multi-task learning of deep neural networks for robust speech recognition," in *Interspeech*, 2016.

[4] K. Hu, H. Sak, and H. Liao, "Adversarial training for multilingual acoustic modeling," 2019.

[5] M. Tu, Y. Tang, J. Huang, X. He, and B. Zhou, "Towards adversarial learning of speaker-invariant representation for speech emotion recognition," 2019.

[6] S. Sun, C.-F. Yeh, M.-Y. Hwang, M. Ostendorf, and L. Xie, "Domain adversarial training for accented speech recognition," in *ICASSP*, 2018, pp. 4854–4858.

[7] A. Tripathi, A. Mohan, S. Anand, and M. K. Singh, "Adversarial learning of raw speech features for domain invariant speech recognition," 2018, pp. 5959–5963.

[8] A. H. Liu, H. Lee, and L. Lee, "Adversarial training of end-to-end speech recognition using a criticizing language model," in *ICASSP*, 2019, pp. 6176–6180.

[9] T. N. Sainath, R. J. Weiss, A. W. Senior, K. W. Wilson, and O. Vinyals, "Learning the speech front-end with raw waveform cldnns," in *Interspeech*, 2015.

[10] N. Zeghidour, N. Usunier, I. Kokkinos, T. Schatz, G. Synnaeve, and E. Dupoux, "Learning Filterbanks from Raw Speech for Phoneme Recognition," in *ICASSP*, 2018. [Online]. Available: https://hal.archives-ouvertes.fr/hal-01888737

[11] M. Ravanelli and Y. Bengio, "Speaker recognition from raw waveform with sincnet," in *SLT*, 2018, pp. 1021–1028.

[12] Y. Ganin, E. Ustinova, H. Ajakan, P. Germain, H. Larochelle, F. Laviolette, M. Marchand, and V. Lempitsky, "Domain-adversarial training of neural networks," *The Journal of Machine Learning Research*, pp. 2096–2030, 2016.

[13] N. Ryant, K. Church, C. Cieri, A. Cristia, J. Du, S. Ganapathy, and M. Liberman, "The second dihard diarization challenge: Dataset, task, and baselines," in *Interspeech*, 2019, pp. 978–982.

[14] H. Bredin, "pyannote.metrics: a toolkit for reproducible evaluation, diagnostic, and error analysis of speaker diarization systems," in *Interspeech*, 2017. [Online]. Available: http://pyannote.github.io/pyannote-metrics

[15] L. N. Smith, "Cyclical learning rates for training neural networks," in *IEEE Winter Conference on Applications of Computer Vision*, 2017.

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

[17] M. Diez, L. Burget, S. Wang, J. Rohdin, and J. Černocký, "Bayesian HMM Based x-Vector Clustering for Speaker Diarization," in *Interspeech*, 2019.

[18] G. Gelly and J.-L. Gauvain, "Optimization of RNN-Based Speech Activity Detection," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pp. 646–656, 2018.

[19] pyannote.audio contributors, "pyannote.audio: Neural Building Blocks for Speaker Diarization," in *ICASSP*, 2020. [Online]. Available: http://github.com/pyannote/pyannote-audio