DIHARD - CPqD - Hybrid System

Valter A. Miasato Filho[1], *Diego A. Silva*[1], *Luis Gustavo D. Cuozzo*[1]

[1]CPqD, Campinas, Sao Paulo, Brazil

{valterf,diegoa,lcuozzo}@cpqd.com.br

1. Abstract

The CPqD hybrid diarization system was comprised of the LIUM diarization toolkit using data-driven, neural network speech activity detection (SAD). The LIUM diarization system is a four-stage pipeline, in which we add a fifth stage with our SAD mask. The final pipeline is then composed of:

- Speaker change point detection,
- Speaker clustering,
- Viterbi re-segmentation,
- Re-clustering,
- Neural network SAD masking.

The pipeline parameters concerning the LIUM toolkit were meta-optimized via recursive and genetic algorithms, and our SAD model was chosen by evaluating the best accuracy over the validation data between multiple intermediate models, as shown in section 3.

2. Data resources

2.1. DIHARD development dataset

The DIHARD dataset has approximately 19 hours worth of 5-10 minute 16kHz, monaural prompts in 165 FLAC files, comprising a variety of domains shown in Table 1. The collection of all domains were split into training, validation, and test sets with the respective approximate ratios of 50%, 25% and 25%, ensuring that all domains were present under the three partitions.

2.2. Additional development datasets

We used the publicly available AMI [1] and ICSI [2] datasets as additions to the provided development data from the DIHARD challenge. Both datasets are composed of multi-party meeting recordings, from which we used the headset mixes as monaural data. The AMI corpus had poor quality in their original mix due to the noise in some channels being louder than speech in others, so we used the SoX tool [3] to apply dynamic range compression and amplitude normalization in the individual channels before mixing. We split those datasets in training, validation and test sets in roughly estimated proportions of 80%, 10% and 10%, respectively. All three partitions have disjoint sets of speakers.

2.3. DIHARD evaluation data

The evaluation data consists of around 21 hours of data with the same characteristics of the development set, except by the addition of a new domain, consisting of recordings from conversations in restaurants. The same set was used in two different tracks for the challenge: diarization from gold speech segmentation (Track1) and diarization from scratch (Track2).

Table 1: *Development datasets.*

Domain	Duration	Speech%	Ovp%	Spk#
AMI	75:39:25	85.8	16.3	3 to 6
ICSI	71:41:12	85.6	15.2	3 to 10
DIHARD	18:56:50	76.1	6.3	1 to 10
SEEDLINGS	1:50:58	60.1	9.3	2 to 5
SCOTUS	2:04:46	84.0	1.6	5 to 10
DCIEM	2:29:58	68.5	2.0	2
ADOS	2:10:12	61.0	2.3	2 to 3
YP	2:03:25	78.5	1.0	3 to 5
SLX	2:00:26	72.4	5.7	2 to 6
VAST	1:50:20	85.7	11.8	1 to 9
RT04S	2:26:15	93.7	21.7	3 to 10
LIBRIVOX	2:00:30	79.4	0.0	1

3. Algorithm description

3.1. LIUM

Our speaker clustering system was based on `LIUM` [4] diarization system with Gaussian Mixture Models (GMMs) trained for each cluster (speech/non-speech) composed of 32 Gaussians with diagonal covariance. Universal Background Models (UBMs) were adapted for each cluster to obtain models for its speakers with 64 diagonal components each.

A four-step pipeline was used: Bayesian Information Criterion (BIC) speaker change point detection, BIC speaker clustering, Viterbi re-segmentation and Cross-Likelihood Ratio (CLR) re-clustering. Each step had its own set of parameters to extract features, described in [5].

Meta-optimizing was used for the large quantity of hyper parameters present in the LIUM diarization system subsystem. Recursive and genetic algorithms were used to search for near-optimal solutions [6, 7], evaluated against the validation partition of DIHARD dataset for DER minimization. This optimizing was run in three times and each taking an average of 24 hours to complete.

3.2. Neural network topology

Our network is comprised of seven time-convolving layers. Each layer is described in Figure 1, with w standing for the width of the convolution and d for the dilation. All layers have $D = 512$ filters for convolution, and have *ReLUs* as nonlinearities. Batch normalization [8] is applied in between layers for more stable training and faster convergence. To avoid fine-tuning gradient descent parameters, the Adam optimizer [9] was employed and gradients were clipped for having the maximum norm of 1.

The outputs of the network are all connected to the last layer with time-distributed weights. The embedding output is a fully

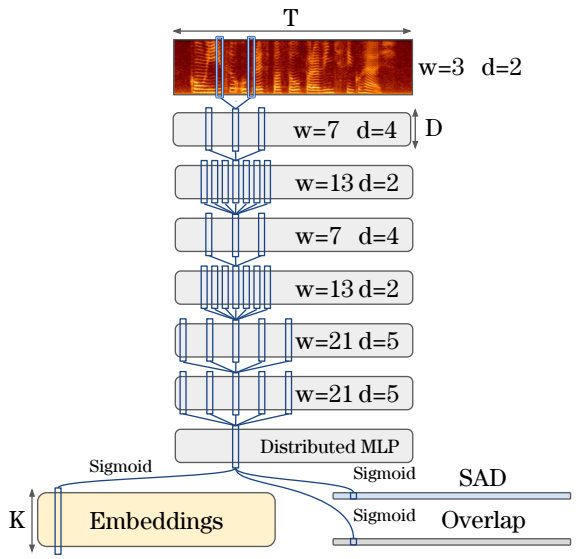

Figure 1: *Neural network topology.*

connected layer with $K = 100$ activations constrained by the sigmoid non-linearity, and its cost function is derived from [10]. The final vectors are then divided by their norm. The SAD and overlap outputs are both single sigmoids for binary classification.

Only the SAD output was relevant for this system.

3.3. Neural network training

The input of our network is the log spectrum of the audio prompts in which speaker diarization is to be performed. We chose a window of $25ms$ with a shift of $30ms$ to perform the short-time Fourier transform. This configuration was inspired by [11] and was used for faster learning and inference. The block size in number of timesteps was $T = 1024$, which accounts for roughly $30s$ of context.

For balancing speech activity and overlap data, we apply sample weights based on a running ratio of the amount of positive/negative examples. The margin value for the affinity matrix loss was set to $m = 0.2$.

We sample 512 batches of 64 examples from different files through 200 iterations, each taking an average of $3880s$ to complete. The intermediate model with the best SAD accuracy was used for composing the hybrid system.

3.4. Data augmentation

We applied two data augmentation techniques for our datasets: noise addition in the AMI and ICSI datasets and noise suppression in the samples provided for the DIHARD challenge. The noise addition was performed with the FaNT tool [12], using external noise samples. We applied CHIME3 [13] samples over AMI, and QUT-NOISE-TIMIT [14] samples over ICSI, both with random signal-to-noise ratios between 5dB and 15dB. The noise supression was used with the corresponding module from the WebRTC project [15] in the DIHARD development set.

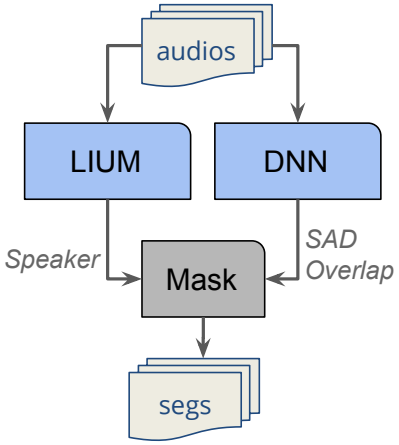

Figure 2: *System diagram.*

4. Diarization system

The diagram of the hybrid system is described in Figure 2, which is composed by the LIUM pipeline and speech activity segments from the trained neural network. Segments were generated with the SAD output, and no smoothing was applied.

5. Hardware Description and Timing

The models were trained on a 32 Intel(R) Xeon(R) CPU E5-2686 @ 2.30GHz machine with Ubuntu OS 16.04 equipped with eight instances of the NVIDIA Tesla K80 Graphics Processing Units over Amazon (AWS) p2.8xlarge instance. The training and development process was based on the Keras framework with Tensorflow backend and NVIDIA®CUDA 9.0 version. The floating point precision for running the experiments was the default 32-bit precision from the toolkits.

In training time, the neural network was run on a single GPU, with the feature extraction and batch generation steps processed on shared CPUs. The GPU time was observed as the bottleneck of the process. The total training time for a single model was roughly 9 days, with the possibility of training a total of 8 systems at the same time.

The benchmarked inference time over the full evaluation set was computed over GPU processing. The usage of GPU in this case was arguably suboptimal. To leverage its computing power in our pipeline, we chose to generate features for the full duration of a single file prior to forwarding it through the network. The feed forward step took 12 minutes to complete over the evaluation set in this scheme.

LIUM inference process was run in CPU and the processing time in DIHARD evaluation dataset:

- All stages: 11min

For benchmarking a single file, we chose a regular desktop machine with a dual-core Intel® Core™ i3-6100 @3.7 GHz with 8GB of RAM. In this scheme, we generated features and feed-forwarded them through the network to avoid excessive RAM consumption. Our test file had 44 minutes of duration. The intermediate feature files held 32MiB worth of disk storage. The full pipeline (LIUM + DNN) took 2min 22s with a peak RAM consumption of 700MiB. The mask process took approximately 5 seconds.

6. References

[1] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal *et al.*, "The ami meeting corpus: A pre-announcement," in *International Workshop on Machine Learning for Multimodal Interaction*. Springer, 2005, pp. 28–39.

[2] A. Janin, D. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke *et al.*, "The icsi meeting corpus," in *Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03). 2003 IEEE International Conference on*, vol. 1. IEEE, 2003, pp. I–I.

[3] L. Norskog and C. Bagwell, "Sox-sound exchange," http://sox.sourceforge.net/, 2013, version 14.

[4] S. Meignier and T. Merlin, "LIUM spkdiarization: an open source toolkit for diarization," in *CMU SPUD Workshop*, vol. 2010, 2010.

[5] M. Rouvier, G. Dupuy, P. Gay, E. Khoury, T. Merlin, and S. Meignier, "An open-source state-of-the-art toolbox for broadcast news diarization," Idiap, Tech. Rep., 2013.

[6] J. H. Holland, *Adaptation in natural and artificial systems: an introductory analysis with applications to biology, control, and artificial intelligence*. MIT press, 1992.

[7] S. Sivanandam and S. Deepa, *Introduction to genetic algorithms*. Springer Science & Business Media, 2007.

[8] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*, 2015, pp. 448–456.

[9] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[10] V. A. Miasato Filho, D. A. Silva, and L. G. D. Cuozzo, "Multiobjective long-short term memory neural networks for speaker diarization in telephone interactions," in *2017 Brazilian Conference on Intelligent Systems (BRACIS)*. IEEE, 2017, pp. 181–185.

[11] D. Povey, V. Peddinti, D. Galvez, P. Ghahrmani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for ASR based on lattice-free MMI," *Submitted to Interspeech*, 2016.

[12] H. G. Hirsch, "Fant: filtering and noise adding tool," *Niederrhein University of Applied Sciences, http://dnt.-kr. hsnr. de/download. html*, 2005.

[13] J. Barker, R. Marxer, E. Vincent, and S. Watanabe, "The third 'chime'speech separation and recognition challenge: Dataset, task and baselines," in *Automatic Speech Recognition and Understanding (ASRU), 2015 IEEE Workshop on*. IEEE, 2015, pp. 504–511.

[14] D. B. Dean, S. Sridharan, R. J. Vogt, and M. W. Mason, "The qut-noise-timit corpus for the evaluation of voice activity detection algorithms," *Proceedings of Interspeech 2010*, 2010.

[15] A. B. Johnston and D. C. Burnett, *WebRTC: APIs and RTCWEB protocols of the HTML5 real-time web*. Digital Codex LLC, 2012.