USC-SAIL System for DIHARD III: Domain Adaptive Diarization System

Tae Jin Park
Signal Analysis and Interpretation Laboratory (SAIL)
University of Southern California
CA, USA
taejinpa@usc.edu

Raghuveer Peri
Signal Analysis and Interpretation Laboratory (SAIL)
University of Southern California
CA, USA
rperi@usc.edu

Arindam Jati
Signal Analysis and Interpretation Laboratory (SAIL)
University of Southern California
CA, USA
jati@usc.edu

Shrikanth Narayanan
Signal Analysis and Interpretation Laboratory (SAIL)
University of Southern California
CA, USA
shri@sipi.usc.edu

***Abstract*—DIHARD challenge focuses on the hard diarization problem and the DIHARD dataset includes a number of challenging domains that are hard to obtain low diarization error rates. We propose a novel approach to deal with domain mismatch problems by estimating the domain of the given input session. We take advantage of three different embedding extractors trained on different datasets. Based on these multiple embedding extractors, our domain adaptive speaker diarization system employs two different approaches: Hard decision and soft decision. In the hard decision method, we estimate the given session into one of the three categories and select an embedding extractor suited to that category. On the other hand, in the soft decision method, we train our proposed neural affinity score fusion network that estimates the desirable weights for the affinity scores we obtain from the three embedding extractors. We show the performance gain from each method and how our domain estimator models are trained to obtain such improvement. In addition, we introduce the auto-tuning spectral clustering method to develop a parameter-free diarization system.**

Index Terms: Speaker Diarization, Domain Adaptation, DIHARD3

I. Notable Highlights

Speaker diarization often suffers from sparse training dataset since training dataset for speaker diarization is not as abundant as training dataset for other applications such as automatic speech recognition (ASR). To tackle this issue, we propose a domain adaptive speaker diarization system that estimates the most suitable speaker embedding extractor or estimates the weights between the speaker embedding extractors that are trained on different dataset. We show the performance gains based on the methods we propose for Track 1.

II. Data Resources

In USC-SAIL DIHARD3 speaker diarization system, we employ three different x-vector models [1] trained on different datasets with a few modifications in the x-vector model architecture. To be more precise, x-vector CH is trained on SRE challenge dataset, x-vector VOX is trained on Voxceleb dataset and x-vector CHIME is trained on Voxceleb dataset and then adapted on CHIME-6 dataset [1]. We use window length of 1.5 s, hop-length of 0.25 s and minimum window length of 0.5 s. We employ cosine similarity to measure the similarity between two speaker embedding vectors. For the hard decision method, we only use affinity matrix that is obtained from a single x-vector model while we use weighted sum of three affinity matrices in the soft decision method. Based on the affinity matrix we get from domain adaptation process, we employ the same clustering method for all the experiments and sessions in DIHARD 3 challenge.

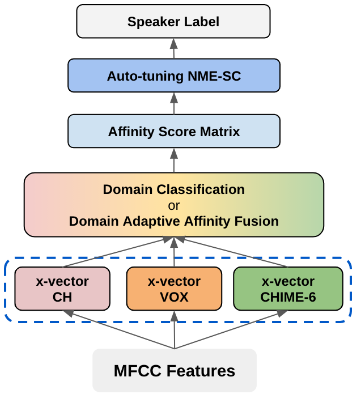

Fig. 1. Overall structure of USC-SAIL diarization system for DIHARD3

III. Detailed description of algorithm

A. Domain Adaptive Processing

1) Hard Decision Method: As previously mentioned, we employ three different speaker embeddings, as we found different embeddings to be optimal for different domains. Motivated by the findings from Table II, we developed a

[1] https://kaldi-asr.org/models.html

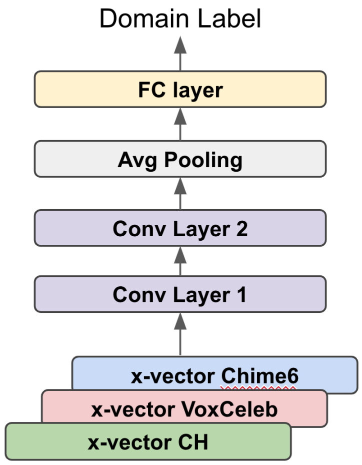

Fig. 2. Diagram of the proposed domain classifier.

TABLE I
CONFUSION MATRIX FOR DOMAIN CLASSIFICATION ON HELD-OUT TEST SESSIONS

		Predicted		
		CH	VOX	CHIME-6
True	CH	17	0	0
	VOX	3	7	0
	CHIME-6	4	0	15

classifier that can predict the speaker embedding to use for each session. We classify each session into one of the three categories, and employ the corresponding speaker embedding to perform speaker diarization. Our approach can be thought of as a crude domain classifier, where our goal is not to accurately predict the domain, but to broadly assign each session to its optimal speaker embedding. This kind of domain grouping has been previously explored in [2], but in this work we grouped them based on the optimal speaker embedding to use. The actual grouping of domains is mentioned in the section IV-A. Hereafter we will refer to the classifier we developed as domain classifier.

We employ a deep neural network as our domain classifier. It takes the concatenation of the three speaker embeddings as input. It consists of two 1-D convolutional layers followed by an average pool layer resulting in fixed dimensional embeddings. The embeddings are then passed through a fully-connected layer with linear activation which assigns probabilities to each class. We use batch normalization followed by ReLU activation and a dropout layer with probability 0.4 after both the convolutional layers. The network was trained on the Development Set using cross-entropy loss. We hold out roughly 20% of sessions from each domain to test our models, and split segments from the remaining sessions into train and validation splits. Since the speaker embeddings are extracted at segment level, our model assigns probabilities for each segment belonging to the 3 classes. During inference, the probabilities of the segments in the session are averaged to obtain the mean probability of the session belonging to each of

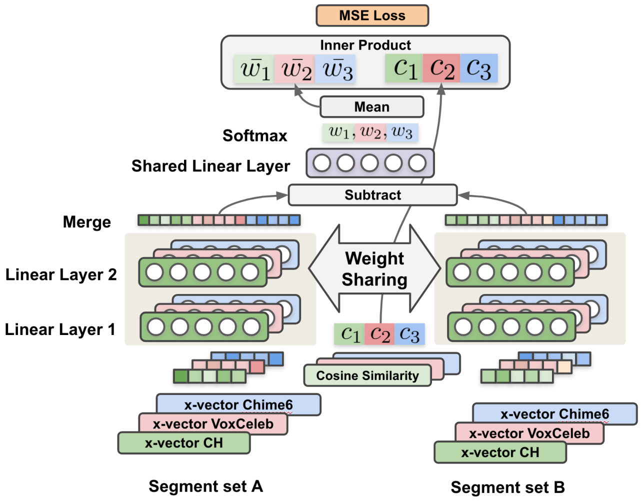

Fig. 3. Neural affinity score fusion model for domain adaptive processing

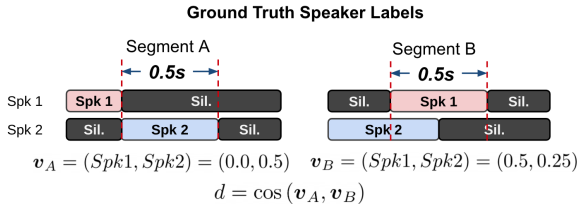
Ground Truth Speaker Labels
Segment A
0.5s
Segment B
0.5s
Spk 1
Spk 2
Spk 1
Sil.
Sil.
Spk 2
Sil.
Sil.
Spk 1
Sil.
Spk 2
Sil.
$\boldsymbol{v}_A = (Spk1, Spk2) = (0.0, 0.5)$
$\boldsymbol{v}_B = (Spk1, Spk2) = (0.5, 0.25)$
$d = \cos(\boldsymbol{v}_A, \boldsymbol{v}_B)$

Fig. 4. Example of training data label generation.

the 3 classes. In addition, instead of assigning class labels that have maximum probability value, we first scale each probability value, with the scaling factor computed to maximize the F1 score of predictions, before choosing the maximum value. The intuition behind the scaling of probabilities is to assign higher weights to some classes for optimal classification performance.

Table I shows the confusion matrix of session-level predictions on the held-out test sessions. This corresponds to an unweighted average F1 score of 0.85. We find that CH sessions were all correctly classified, while there are instances where classes belonging to the other sessions were also classified as CH. We conjecture that this is due to the class imbalance in the development set of the data, with more sessions are from the CTS domain compared to other domains.

2) Soft Decision Method: For the soft decision method, we employ a neural network architecture that is similar to Siamese-network [3] and we refer to this neural network model as neural affinity score fusion (NASF) modol. As described in 3, NASF model accepts three different x-vector embeddings that are extracted from the same audio segment. The embeddings are forward propagated through three separate fully-connected layers then merged to get a concatenated embedding. For the fully-connected layers, 128 units are used for each layer with ReLU activation. The difference between two concatenated embeddings from two different feed-forward networks is fed to the last shared linear layer. The whole model is trained by calculating mean square error (MSE) loss with a ground truth speaker label for each segment. The ground truth speaker labels are created using the approach described in Fig. 4. We employ the concept of speaker vector that is created by calculating the portion of each speaker for the given segment window. Since the ground truth speaker label vectors are bound to be positive, cosine similarity d between two speaker vectors ranges from 0 to 1. Therefore, cosine similarity values from the speaker embeddings in the NASF model are min-max normalized to (0, 1) scale. For each session, 40,000 pairs of audio segments are randomly extracted for both train and inference.

We use weight sharing network and feed three different set of x-vector embeddings (CH, VOX and CHIME-6) from two different segments. Thus, there are six different embedding vectors (two sets of three) that are fed to NASF module. The NASF module outputs weight between these three x-vector inputs and we use this weight to calculate the weighted sum of three affinity matrices. The final weighted affinity matrix is then fed to clustering module to obtain speaker labels.

B. Clustering

We use the auto-tuning spectral clustering method appeared in [4] which does not require development set for tuning the clustering algorithm. The auto-tuning spectral clustering method employs nomarlized maximum eigengap (NME) to find the best binarization parameter p for spectral clustering process. Thus, the auto-turning spectral clustering approach determines p for each session and applies different p values over the different sessions. In addition to parameter p, the number of speaker is also determined by finding the maximum gap of the obtained eigenvalues. The auto-tuning spectral clustering approach not only shows the better performance over traditional probabilistic linear discriminant analysis (PLDA) coupled with agglomerative hierarchical clustering (AHC) [5] method but also shows better performance over spectral clustering method based on manual tuning of p on a development set. In CHIME-6 challenge [2], the auto-tuning spectral clustering method was employed by the challenge winner, STC [6] team, showing a superior clustering performance over AHC method. Especially for DIHARD challenge, since we do not have enough development dataset for parameter tuning, auto-tuning spectral clustering approach showed benefit over clustering methods that require parameter tuning. We employ sparse-search where we only allow maximum of 20 threshold values to be searched. The detailed algorithmic description can be found in [4] and the source code can be downloaded from a git repository[3].

IV. Results on the development set

A. Development Set

TABLE II
Development Set DER for each domain and x-vector type.

Track 1	x-vector Type		
Domain	**CH**	**VOX**	**CHIME-6**
Audiobooks	0.36	0.44	**0**
Broadcast Interview	4.19	6.89	**3.09**
Clinical	23.66	22.05	**16.63**
Court	**4.63**	9.29	5.09
CTS	**15.05**	19.57	19.13
Maptask	6.6	**5.94**	6.64
Meeting	31.44	31.79	**28.3**
Restaurant	57.71	56.04	**52.59**
Socio Field	16.21	14.24	**13.78**
Socio Lab	**8.45**	10.36	9.61
Webvideo	41.32	**39.24**	40.66
Total	19.89	20.52	**19.44**

TABLE III
Development set results for Track 1.

	CORE		FULL	
Type	DER	JER	DER	JER
x-vector CHIME-6	22.14	48.85	19.44	42.76
Oracle Hard Decision	21.25	46.56	18.65	41.18
Dev-set Soft Decision	19.68	43.36	17.39	38.29

TABLE IV
Evaluation set results for Track 1.

	CORE		FULL	
Type	DER	JER	DER	JER
x-vector CHIME-6	22.140	48.850	20.310	43.700
Hard Decision	22.250	48.370	19.660	42.520
Soft Decision	**19.760**	**43.030**	**18.190**	**38.330**

Table II shows the DER achieved by each x-vector type for each domain in DIHARD 3 development set. Note that result in Table II is obtained from the FULL set, Track 1. In Table II, DER for each doamin is shown for each x-vector embedding type. For hard-decision classifier, we group the domains to the best performing x-vector type based on the development set result. Thus, the domains are grouped as follows:

- CH domains: Court, CTS, Socio Lab
- VOX domains: Maptask, Webvideo
- CHIME-6 domains: Audiobooks, Broadcast Interview, Clinical, Meeting, Restaurant, Socio Field

In table II, the last row shows the total DER value from each x-vector embedding extractor. Without selecting an embedding extractor for each session, x-vector CHIME-6 shows the best performance among three x-vector embedding extractors. In Table III, the result with the ground truth domain labels (oracle hard decision) and the result with dev-set optimized soft decision method are shown. Note that since we use DIHARD3 dev-set to train our proposed domain adaptive system, the results in Table III can be an outcome of overfit models.

B. Evaluation Set

The Table IV shows the final evaluation results we obtain from x-vector CHIME-6 model, hard decision model and soft decision model. For FULL set, hard decision model shows a slightly improved performance over the system based on x-vector CHIME-6. Soft decision method shows improved results for both CORE and FULL sets showing DER of 19.76% and 18.19%, respectively.

V. Hardware requirements

- Total number of CPU cores used: 6 cores and 12 threads

[2] https://chimechallenge.github.io/CHIME-6

[3] https://github.com/tango4j/Auto-Tuning-Spectral-Clustering

- Description of CPUs used: Intel i7-6850K 3.60 GHz
- Total number of GPUs used: 1 GPU
- Description of GPUs used: Nvidia GTX 1080 Ti, 11.3 TFLOPs, 11GB Memory
- Total available RAM: 128 GB
- Used disk storage: 3 GB
- Machine learning frameworks used: Pytorch 1.3.1

REFERENCES

[1] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, “X-vectors: Robust DNN embeddings for speaker recognition,” in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Process.*, Apr. 2018, pp. 5329–5333.

[2] M. Sahidullah, J. Patino, S. Cornell, R. Yin, S. Sivasankaran, H. Bredin, P. Korshunov, A. Brutti, R. Serizel, E. Vincent *et al.*, “The speed submission to dihard ii: Contributions & lessons learned,” *arXiv preprint arXiv:1911.02388*, 2019.

[3] G. Koch, R. Zemel, and R. Salakhutdinov, “Siamese neural networks for one-shot image recognition,” in *Proceedings of the Workshop on Deep Learning in International Conference on Machine Learning, ICML*, 2015.

[4] T. J. Park, K. J. Han, M. Kumar, and S. Narayanan, “Auto-tuning spectral clustering for speaker diarization using normalized maximum eigengap,” *IEEE Signal Processing Letters*, vol. 27, pp. 381–385, 2019.

[5] G. Sell, D. Snyder, A. McCree, D. Garcia-Romero, J. Villalba, M. Maciejewski, V. Manohar, N. Dehak, D. Povey, S. Watanabe *et al.*, “Diarization is hard: Some experiences and lessons learned for the jhu team in the inaugural dihard challenge.” in *Interspeech*, 2018, pp. 2808–2812.

[6] I. Medennikov, M. Korenevsky, T. Prisyach, Y. Khokhlov, M. Korenevskaya, I. Sorokin, T. Timofeeva, A. Mitrofanov, A. Andrusenko, I. Podluzhny *et al.*, “The stc system for the chime-6 challenge,” in *CHiME 2020 Workshop on Speech Processing in Everyday Environments*, 2020.