Deep Embedding Clustering for Speaker Diarization

TensorSlow Team
Aditya Singh (170056) and Shashi Kant Gupta (16807645)

Abstract

Speaker diarization has received significant interest within the speech community due to its promise to improve automatic speech transcription considerably. Commonly used approach to this problem include using embedding vectors such as d-vectors, i-vectors, or x-vectors with Spectral Clustering. We propose using Unsupervised Deep Embedding Clustering to cluster data in a more semantically meaningful latent representation with pre-trained Auto Encoders [1] for improved imbalanced data separation. Stacked layers of Auto Encoders have been trained in a residual fashion in place of De-noising Auto Encoders for enhanced learning. We use VoxConverse and AMI Corpus split datasets to test our model. Our model shows considerable improvement over the Spectral Clustering approach. The project can be accessed at https://github.com/shashikg/speaker_diarization_ee698.

1 Introduction and Problem Statement

With the increase in digital media data volumes, speaker diarization finds significant use in cases like multimedia information retrieval, meeting transcription analysis, film and movie transcription, speaker turn analysis, and adaptive speaker processing. Most of the speaker diarization systems include two crucial modules. The first module performs pre-processing on the raw audio signal and extracts high dimensional features known as embedding vector. These embeddings are extracted for multiple sliding window segments of the sampled input audio file. The second module uses these embedding vectors to group them into different clusters and assign unique speaker labels to each of them [2–6] (See **Fig. 1**). Advanced diarization systems also use additional modules to handle the cases for speaker overlap and produce refined clusters [7].

Many of the state-of-the-art speaker diarization systems use recent deep learning frameworks to extract the feature embeddings called as d-vectors[2, 3] and x-vectors[4–6], and cluster them using conventional clustering methods like K-means, spectral [8], AHC, and affinity propagation clustering algorithm [9]. Recent developments in robust deep neural networks based clustering method [1] has motivated its application in speaker diarization for improved label assignment. Initializing of the DEC architecture with a pre-trained auto encoder produces semantically meaningful projection of raw data, making it easily separable even for real world datasets which are typically imbalanced. Section 2 describes the baseline to our model performance. Section 3 explains the architecture of the proposed DEC model. It is followed by section 4 defining the datasets and evaluation criteria. At the end, section 5 discusses over the results.

2 Baseline

For baseline, we use the algorithm described in [2] to annotate the audio files. We extract x-vector embeddings instead of the originally proposed d-vectors embeddings owning to superior performance of x-vector in speaker verification tasks [10].

Prior to x-vector extraction, we detect speech segments from raw audio signal using Silero Voice Activity Detection module [11]. These speech segments are then sliced into overlapping windows as depicted in **Fig 1**. SpeechBrain ECAPA-TDNN (Time Delay Neural Network) implementation [12]

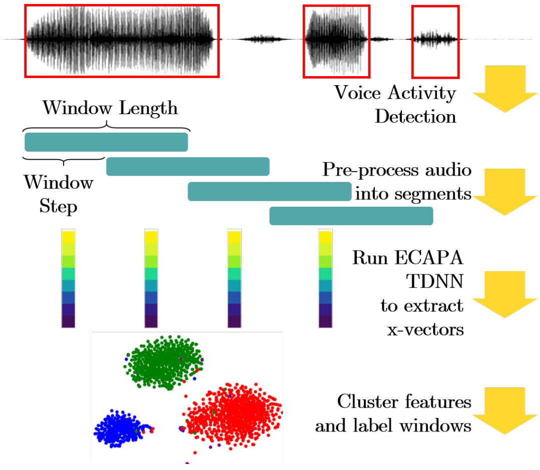

Voice Activity Detection
Window Length
Window Step
Pre-process audio into segments
Run ECAPA TDNN to extract x-vectors
Cluster features and label windows

Figure 1: Steps involved in ECAPA x-vector based diarization system

Table 1: Diarization performance of our proposed and spectral clustering on different dataset

VoxConverse Dataset			
Method	DER	FA	Miss
Spectral	17.76	2.36	2.03
Ours	12.99	2.36	2.03
Spectral (Oracle VAD)	17.98	1.07	2.55
Ours (Oracle VAD)	**11.70**	**1.07**	**2.55**

AMI Dataset			
Method	DER	FA	Miss
Spectral	21.99	9.77	5.47
Ours	23.93	9.77	5.47
Spectral (Oracle VAD)	14.96	2.64	5.83
Ours (Oracle VAD)	**13.14**	**2.64**	**5.83**

is next used for embedding features extraction on these windows. It takes time invariant sampled audio window, extracts 80 dimensional Mel Frequency Cepstral Coefficents (MFCC) features, and passes it through pretrained ECAPA-TDNN network to get 192 dimensional x-vector.

The feature stack for all the audio windows are then clustered using two unsupervised approaches namely spectral clustering on embedding extracted using windows of ground truth speech segments (Spectral (Oracle VAD) in **Table** 1), and spectral clustering on embeddings extracted from speech segments determined by Silero VAD (Spectral in **Table** 1). **Table** S1 in the Supplementary Section contains comparison over different techniques for selecting the baseline.

3 Proposed Model

For our proposed model, we deploy Deep Embedding Clustering [1] in place of Spectral Clustering. This was motivated by the fact that speech signals with multiple speakers are generally imbalanced with each speaker speaking for different duration of time. Directly used, a randomly initialized deep network of the DEC algorithm would perform poorly because of the algorithm's sensitivity to centroid initialization [13]. As a solution to this issue, the encoder part of a Deep Auto Encoder trained on similar data is used to initialize weights for the DEC structure. The bottleneck layer returns features that are semantically meaningful and easier to separate during unsupervised clustering [1].

We trained a Deep Auto Encoder on the extracted ECAPA x-vectors from the training split of the VoxConverse dataset. It was performed using unsupervised reconstruction loss in a residual fashion [14], in which we forced each of the decoder layers to reconstruct the corresponding encoder layers. The hidden dimensions of the Deep Auto Encoder was fixed to 500–500–2000–30, where 30 is the bottleneck latent space dimension (for details refer to the **Supplementary Section 1**). Post training, we extracted the encoder module from the Deep Auto Encoder to be used for DEC initialization.

While testing on an audio file, the model first extracts ECAPA x-vectors for a window length of 1500 ms with 750 ms window step. It uses these x-vectors to fine-tune the pre-trained Deep Auto Encoder to get a better reconstruction on that specific audio file. After fine-tuning, the ECAPA x-vectors are passed through the encoder module to extract the latent space feature vectors. The optimal number of clusters (N) are then determined using these latent space feature vectors. The model then initialize N cluster centroid points and then trains the fine tuned DEC encoder weights, and the cluster centroids via back-propagation on a clustering loss in which the model computes an auxiliary distribution and then uses it to minimise the KL divergence with the target distribution [1]. The auxiliary distribution i.e. the soft assignment of cluster labels are estimated using the Student's t-distribution to predict the similarity between the centroid μ_j and latent features z_i (**Eq. 1**, note that we have taken the α parameter of Student's t-distribution to be 1). Since the method is fully unsupervised, the target

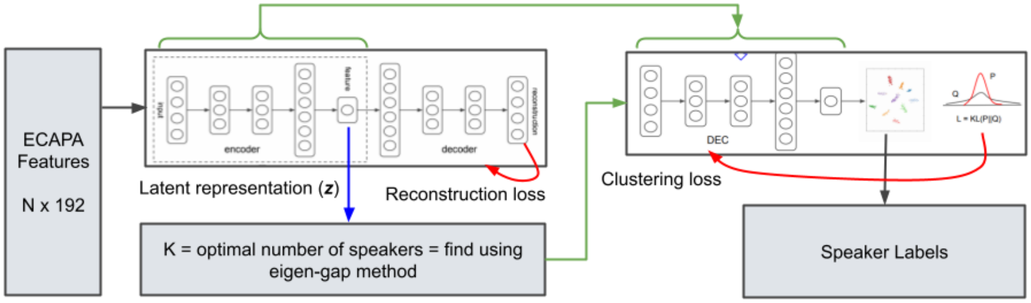


Figure 2: Our proposed clustering method flowchart

distribution p_{ij} were estimated using the predicted soft-assignments (**Eq.** 2 as per the suggestion in DEC paper [1]). For each test file, the DEC encoder network is again re-initialized to the pre-trained Deep Auto Encoder's encoder weights. Originally, the DEC paper suggested initializing the centroid point using K-Means clusters. Instead, we used spectral clustering for centroid initialization. **Fig.** 2 shows the overall flowchart of the method.

$$q_{ij} = \frac{(1 + ||z_i - \mu_j||^2)}{\sum_k (1 + ||z_i - \mu_k||^2)} \tag{1}$$

$$p_{ij} = \frac{(q_{ij}^2 / f_j)}{\sum_k (q_{ij}^2 / f_k)}, \quad \text{here} \quad f_j = \sum_i q_{ij} \tag{2}$$

4 Experiment

4.1 Dataset

We use the VoxConverse [15] and AMI Corpus test split [16] speaker diarization dataset for training and evaluation. VoxConverse dataset consists of 216 multi-speaker audio files with over 20 hours of speech covering more than 8200 speakers. The VoxConverse data is split into two sets of size 166 (train) and 50 (test). The VoxConverse train split is used for training the Deep Auto Encoder. AMI Corpus test split contains 16 files with a total length of over 13 hours. Along with the VoxConverse test split, it is to test the performance of our model against the baseline.

4.2 Evaluation Criteria

We calculate the Diarization Error rate (DER) on the test dataset to evaluate our approach against Spectral Clustering method. The audio is sampled at 16kHz and a batch size of 512 windows is used while extracting ECAPA x-vectors. The overlapped region is ignored while calculating the DER since our model is incapable of returning multiple labels for a single window. We use a collar of 250 ms, which is a standard evaluation criteria used in most diarization work [17]. For DEC evaluation, we train the DEC encoder for 100 iteration to first fine-tune the weights on the specific audio data via the auto encoder's reconstruction loss, and further train it for 50 iterations to minimize the KL Divergence between target and auxiliary distribution using Adam Optimizer with a learning rate of 1e-4. These hyper parameters used for test evaluation are chosen based on the performance on the 'train' split.

5 Results and Discussions

Table 1 lists the diarization error rate (DER) for spectral clustering and our proposed method.The results include both with and without oracle voice-audio-detection (VAD) performance. With oracle VAD, our method performed better than spectral clustering for both of the dataset. But when using VAD based on the Silero model, there is a slightly poor performance on AMI Corpus against the

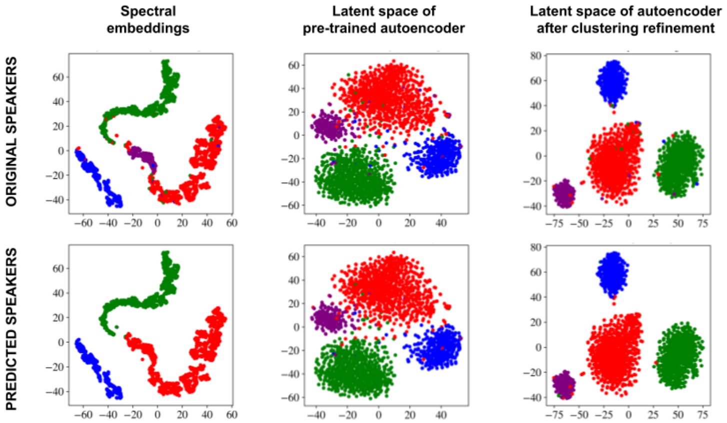


Figure 3: Comparison of t-SNE clusters on spectral embeddings and deep embeddings learned by our proposed method. This example demonstrates superior performance of proposed model on speaker diarization data with imbalance. Here each of the speaker speaks for different duration of time. Spectral clustering happens to ignore the smaller purple cluster but DEC accurately detects it.

spectral clustering method. Comparing the results on AMI dataset for oracle VAD vs predicted VAD, we observe the effect of imperfect voice activity detection module on the model performance.

An explanation to poorer performance of DEC due to the VAD can be linked to its superior cluster formation ability. For audio with routine pauses by the speaker, the silence gets separated into a different cluster, and the DEC outputs different number of speakers than are actually present. Spectral Clustering's relaxed cluster formation usually assigns the silent pauses as speech to the cluster of the respective speaker taking the pauses. Hence cluster assignment takes a toll for DEC in such cases. However, better performance of DEC over Spectral clustering for Oracle VAD indicates higher relevance of our proposed method with improvement in voice activity detection modules.

We also looked at the number of files for which the proposed method shows better performance than the spectral clustering approach. **Table 2** lists the result for the same.

Dataset	VAD Used	Number of files where		Total files
		DEC performs better	Both perform similar	
VoxConverse	Oracle VAD	35	7	50
	Silero VAD	32	6	50
AMI Corpus	Oracle VAD	13	1	16
	Silero VAD	8	1	16

Table 2: Performance of DEC and Spectral based on number of audio files with better DER

To visualize the model performance against Spectral Clustering, we plotted the t-SNE projection of the clusters formed by the spectral embeddings versus the deep embeddings (refer to **Fig 3**). The shown example reflects performance on imbalanced diarization speech segments with all the speakers having different number of audio windows. The spectral clustering merged the smaller purple cluster with the red cluster (both indicating different speakers), while our proposed method segregated it into a well separated cluster. The second column of the figure shows the cluster formed by the initial latent space vectors extracted using the pre-trained encoder while the third column shows the latent space vectors extracted after fine-tuning the encoder module according to the given test audio file. Clearly, the deep embedded clustering method outperforms spectral clustering with improved concentrated clusters for each speaker.

6 Acknowledgement

We thank Dr. Vipul Arora for providing us the opportunity and guidance to work on this project as part of our advanced machine learning training in the course EE698R.

References

[1] Junyuan Xie, Ross Girshick, and Ali Farhadi. Unsupervised deep embedding for clustering analysis, 2016. arXiv: 1511.06335.

[2] Q. Wang, C. Downey, L. Wan, P. A. Mansfield, and I. L. Moreno. Speaker diarization with lstm. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5239–5243, 2018. doi: 10.1109/ICASSP.2018.8462628.

[3] Li Wan, Quan Wang, Alan Papir, and Ignacio Lopez Moreno. Generalized end-to-end loss for speaker verification, 2020. arXiv: 1710.10467.

[4] D. Garcia-Romero, D. Snyder, G. Sell, D. Povey, and A. McCree. Speaker diarization using deep neural network embeddings. In *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4930–4934, 2017. doi: 10.1109/ICASSP.2017.7953094.

[5] 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)*, pages 5329–5333, 2018. doi: 10.1109/ICASSP.2018.8461375.

[6] Joon Son Chung, Jaesung Huh, Arsha Nagrani, Triantafyllos Afouras, and Andrew Zisserman. Spot the conversation: Speaker diarisation in the wild. *Interspeech 2020*, Oct 2020. doi: 10.21437/interspeech.2020-2337.

[7] Yusuke Fujita, Naoyuki Kanda, Shota Horiguchi, Kenji Nagamatsu, and Shinji Watanabe. End-to-end neural speaker diarization with permutation-free objectives, 2019. arXiv: 1909.05952.

[8] Ulrike von Luxburg. A tutorial on spectral clustering, 2007. arXiv: 0771.0189.

[9] Brendan J. Frey and Delbert Dueck. Clustering by passing messages between data points. *Science*, 315(5814):972–976, 2007. ISSN 0036-8075. doi: 10.1126/science.1136800.

[10] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. Ecapa-tdnn: Emphasized channel attention, propagation and aggregation in tdnn based speaker verification. *Interspeech 2020*, Oct 2020. doi: 10.21437/interspeech.2020-2650.

[11] Silero Team. Silero vad: pre-trained enterprise-grade voice activity detector (vad), number detector and language classifier. https://github.com/snakers4/silero-vad, 2021.

[12] Mirco Ravanelli, Titouan Parcollet, Aku Rouhe, Peter Plantinga, Elena Rastorgueva, Loren Lugosch, Nauman Dawalatabad, Chou Ju-Chieh, Abdel Heba, Francois Grondin, William Aris, Chien-Feng Liao, Samuele Cornell, Sung-Lin Yeh, Hwidong Na, Yan Gao, Szu-Wei Fu, Cem Subakan, Renato De Mori, and Yoshua Bengio. Speechbrain. https://github.com/speechbrain/speechbrain, 2021.

[13] Yaling Tao, Kentaro Takagi, and Kouta Nakata. Rdec: Integrating regularization into deep embedded clustering for imbalanced datasets. In Jun Zhu and Ichiro Takeuchi, editors, *Proceedings of The 10th Asian Conference on Machine Learning*, volume 95 of *Proceedings of Machine Learning Research*, pages 49–64. PMLR, 14–16 Nov 2018.

[14] Lian-Feng Dong, Yuan-Zhu Gan, Xiao-Liao Mao, Yu-Bin Yang, and Chunhua Shen. Learning deep representations using convolutional auto-encoders with symmetric skip connections. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3006–3010, 2018. doi: 10.1109/ICASSP.2018.8462085.

[15] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. Voxceleb: A large-scale speaker identification dataset. *Interspeech 2017*, Aug 2017. doi: 10.21437/interspeech.2017-950.

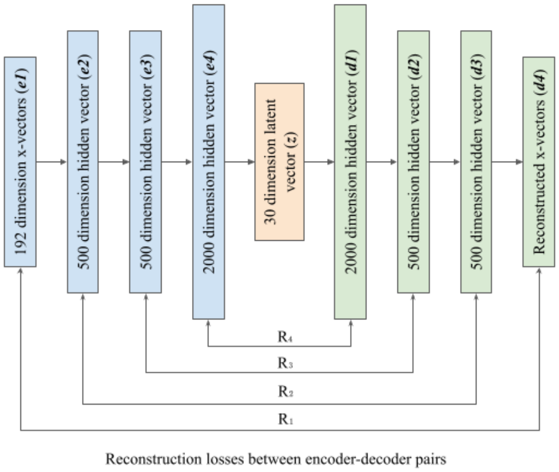
[16] Jean Carletta. Unleashing the killer corpus: experiences in creating the multi-everything ami meeting corpus. *Language Resources and Evaluation*, 41(2):181–190, 2007. ISSN 1574-020X. doi: 10.1007/s10579-007-9040-x.

[17] Tae Jin Park, Naoyuki Kanda, Dimitrios Dimitriadis, Kyu J. Han, Shinji Watanabe, and Shrikanth Narayanan. A review of speaker diarization: Recent advances with deep learning, 2021.

Supplementary

1 Training details of AutoEncoder

The auto encoder was trained on the ‘train’ split of VoxConverse dataset. For this we used ECAPA features calculated for 1500 ms length audio taken at the intervals of 250 ms. We reduced the step length (250 ms instead of 750 ms) to get more number of data points to train the auto encoder. The reconstruction loss was calculated between each of the encoder-decoder pair of the layer. This is represented in the **Fig. S1**.



192 dimension x-vectors (*e1*)
500 dimension hidden vector (*e2*)
500 dimension hidden vector (*e3*)
2000 dimension hidden vector (*e4*)
30 dimension latent vector (*z*)
2000 dimension hidden vector (*d1*)
500 dimension hidden vector (*d2*)
500 dimension hidden vector (*d3*)
Reconstructed x-vectors (*d4*)
R₄
R₃
R₂
R₁
Reconstruction losses between encoder-decoder pairs

Figure S1: Diagram showing the architecture of the auto encoder and several reconstruction losses on the encoder-decoder pair of the layers.

The final loss was taken as the weighted sum between all the reconstruction losses giving some partial weights to each of the reconstruction loss (**Eq. 3**). This weighting pattern ensures that we give higher penalty to the reconstruction loss at the higher depth.

$$Loss = \sum_{k=1}^{k=4} (5-k) \times R_k \tag{3}$$

We trained the auto encoder on a batch size of 256 using Adam optimiser with initial learning rate = 0.001. After training for 60 epochs the learning rate was reduced to 0.0001 and the model was trained for another 60 epochs. The model was trained on a single NVIDIA Tesla P100 GPU provided Kaggle data science platform. Total time taken for training was around 1 hour.

2 Performance of other clustering methods

Dataset	Method	DER	False alarm	Miss	Confusion
VoxConverse	Oracle Kmeans	23.99	3.87	6.37	13.74
	Oracle Spectral Clustering	18.83	3.87	6.37	8.58
	Eigen-gap Spectral Clustering	24.44	3.87	6.37	14.19
AMI Corpus	Oracle Kmeans	19.59	9.77	5.47	4.35
	Oracle Spectral Clustering	17.65	9.77	5.47	2.41
	Eigen-gap Spectral Clustering	21.99	9.77	5.47	6.75

Table S1: Diarization performance of various other clustering methods on VoxConverse and AMI test data. Eigen-gap Spectral Clustering selected as the baseline. Oracle K means and Oracle Spectral Clustering use the number of speakers information from the ground truth rttm files. Baseline uses eigen gap maximization to determine number of clusters.

3 Example of annotated speaker labels for an audio signals

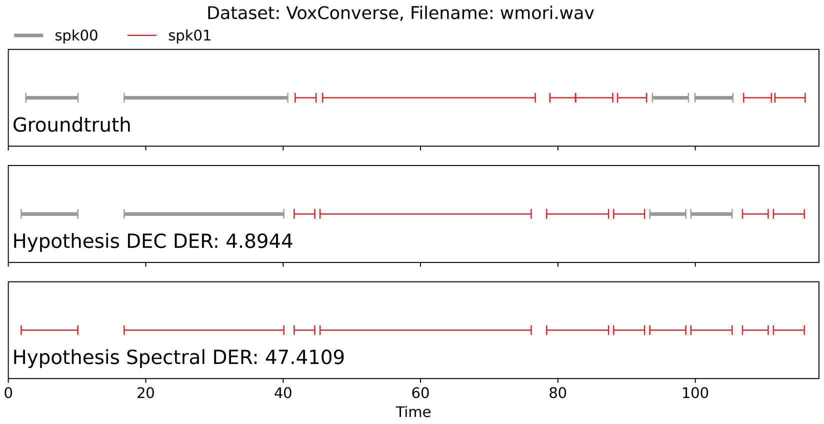


Figure S2: An example showing ground truth vs hypothesis time series for an audio file. Spectral clustering fails to detect the number of speakers correctly, while DEC correctly classifies the audio due to better clusters formation compared to Spectral Clustering.