Towards Unsupervised Speaker Diarization System for Multilingual Telephone Calls Using Pre-trained Whisper Model and Mixture of Sparse Autoencoders

Phat Lam[1], Lam Pham[2], Truong Nguyen[1], Dat Ngo[3], Thinh Pham[4], Tin Nguyen[1], Loi Khanh Nguyen[1], and Alexander Schindler[2]

[1] Ho Chi Minh University of Technology, Vietnam
[2] Austrian Institute of Technology, Austria
[3] University of Essex, United Kingdom
[4] Ho Chi Minh City University of Science, Vietnam

Abstract. Existing speaker diarization systems typically rely on large amounts of manually annotated data, which is labor-intensive and difficult to obtain, especially in real-world scenarios. Additionally, language-specific constraints in these systems significantly hinder their effectiveness and scalability in multilingual settings. In this paper, we propose a cluster-based speaker diarization system designed for multilingual telephone call applications. Our proposed system supports multiple languages and eliminates the need for large-scale annotated data during training by utilizing the multilingual Whisper model to extract speaker embeddings. Additionally, we introduce a network architecture called Mixture of Sparse Autoencoders (Mix-SAE) for unsupervised speaker clustering. Experimental results on the evaluation dataset derived from two-speaker subsets of benchmark CALLHOME and CALLFRIEND telephonic speech corpora demonstrate the superior performance of the proposed Mix-SAE network to other autoencoder-based clustering methods. The overall performance of our proposed system also highlights the promising potential for developing unsupervised, multilingual speaker diarization systems within the context of limited annotated data. It also indicates the system's capability for integration into multi-task speech analysis applications based on general-purpose models such as those that combine speech-to-text, language detection, and speaker diarization.

Keywords: Unsupervised speaker diarization· Whisper · Mixture of sparse autoencoders · Deep clustering · Telephone call.

1 Introduction

Sound-based applications have drawn significant attention from the research community and have become an integral part in the forefront of driving innovation. These applications involve advanced audio processing techniques to analyze and interpret various types of sound data (e.g. acoustic scenes [25],[24]),

sound events [18], machinery sound [19], human speech [14]), enabling the core functionality in many intelligence systems. In human speech analysis, speaker diarization plays a crucial role by identifying and segmenting audio streams based on speaker identity, making it essential for various applications such as communication (e.g. customer support calls), security (e.g. voice tracking), healthcare (e.g. patient monitoring), smart home (e.g. personal assistants), etc. Typically, a cluster-based speaker diarization system consists of five modules. The traditional approach to such a system is illustrated at the top of Fig. 1. The preprocessing module first converts raw audio into a suitable format, followed by the voice activity detection (VAD) module extracting speech segments. These segments are then divided into fixed-length speaker segments. The speaker embedding extractor converts these segments into vectors representing speaker characteristics, and a clustering algorithm assigns speaker labels. Among these modules, speaker embedding and clustering are crucial components to enhance the performance of a cluster-based speaker diarization system [22].

Regarding the speaker embedding extractor, numerous approaches have been proposed for speaker embedding extraction, including metric-based models (GLR [8], BIC [32], etc), probabilistic models (GMM-UBM [28], i-vectors [6], etc), and neural network-based models (d-vectors [33], x-vectors [30], etc.). All these methods require a substantial amount of annotated data, especially for neural network-based approaches, to optimize speaker feature extractors. However, training these extractors on one type of dataset could reduce the model's ability to generalize to diverse or unseen data, particularly from different domains. In addition, datasets for speaker diarization mainly support one single language, due to the labor-intensive and time-consuming nature of collecting data and insufficient availability of data from diverse languages, limiting the effectiveness of speaker diarization systems in multilingual speech analysis applications.

Concerning the clustering module, common methods such as Agglomerative Hierarchical Clustering (AHC) [9], k-Means [35], Mean-shift [31] have been proposed. However, these methods operate directly on the input vector space and rely heavily on distance-based metrics, without leveraging representation learning techniques to uncover deeper patterns. While some deep learning-based frameworks, such as DNN [12], GAN [21], and Autoencoder [10], incorporate representation learning for speaker embeddings, they often require pre-extracted embeddings (e.g. x-vectors) that fit on certain datasets and are primarily evaluated on single-language datasets, typically English.

To address existing limitations, we aim to develop an unsupervised speaker diarization system that does not rely on large training datasets and supports multiple languages. For speaker embedding extraction, we use the multilingual Whisper model. This model is trained on diverse audio data for relevant tasks such as speech recognition, language identification, and translation. However, its applicability in speaker diarization task remains unexplored. Thus, leveraging Whisper's scalability and robustness, we explore its potential to produce high-quality speaker embeddings for diarization, assuming that as a general-purpose model, Whisper can learn representations that incorporate various aspects of

Fig. 1. *The high-level architectures of (A) Traditional cluster-based speaker diarization system and (B) Our proposed unsupervised speaker diarization system*

large training data (e.g., phonetic content, acoustic features) that may be useful for diarization task, despite being primarily designed for speech recognition and speech-to-text transcription [30]. For speaker clustering, we propose an unsupervised deep clustering network called Mixture of Sparse Autoencoders (Mix-SAE) to cluster the extracted embeddings. Overall, our key contributions can be summarized as follows:

– We explored the Whisper model's capability in the diarization task by using it as an alternative to conventional speaker embedding extractors, eliminating the need for annotated training data in developing diarization systems.
– Inspired by the work in [5], we proposed the Mix-SAE network for speaker clustering, which enhances both speaker representation learning and clustering by using a mixture of sparse autoencoders with pseudo-label supervision.
– Through extensive experiments, we demonstrated that speaker diarization can be effectively integrated into Whisper-based systems, enabling comprehensive and multilingual speech analysis applications that combine speech-to-text, language identification, and speaker diarization. An example of a Whisper-based speech analysis application can be found at [1].

The remainder of this paper is organized as follows: The overall proposed speaker diarization system is described in Section 2. Next, Section 3 comprehensively describes our proposed deep clustering framework (Mix-SAE). Experimental settings and results are discussed in Section 4. The conclusion is represented in Section 5.

2   The Overall Proposed System

Our proposed system pipeline is comprehensively described at the bottom of Fig. 1. Generally, the system comprises three main blocks: Front-end prepro-

---

[1] https://huggingface.co/spaces/AT-VN-Research-Group/SpeakerDiarization

Table 1. The Pre-trained Whisper Models

Version	Parameters	Embedding Dimension
Tiny	39M	384
Base	74M	512
Small	244M	768
Medium	769M	1024
Large	1550M	1280

cessing, Speaker embedding extraction and Unsupervised clustering. The next subsections represent each block of the overall pipeline in detail.

2.1   Front-end preprocessing

Firstly, the input audio is divided into fixed-length segments of W seconds and re-sampled to 16 kHz using Librosa toolbox [11]. To match the Whisper encoder's input requirements, zero-padding is applied to the segments. Next, a voice activity detection (VAD) [26] is performed using an energy-based threshold to extract speech segments, which are then converted into spectrograms via Short-time Fourier Transform (STFT) with the setting of 400 filters, 10-ms window size, and a 160-sample hop size, respectively. These spectrograms are used as inputs to the Whisper encoder for speaker embedding extraction.

2.2   Speaker embedding extraction using Whisper model

In our work, we explore using the Whisper model as an alternative to conventional speaker embedding extractors, leveraging its scalability and diverse training data. We aim to utilize Whisper's robustness and generalization to capture various speaker characteristics across languages and domains. This approach allows us to obtain speaker embeddings directly from Whisper, bypassing the need for specific training datasets. For each speech segment, we generate the speaker embedding by feeding its spectrogram into the Whisper model. The final one-dimensional speaker embedding is derived by averaging the 2D tensor output from the last residual attention block of the Whisper encoder along the second axis, with its dimension varying by Whisper model versions, as shown in Table 1.

2.3   Unsupervised Clustering

Given the speaker embeddings extracted from the Whisper model, the unsupervised clustering block groups together speech segments that are likely to be from the same speaker. In this work, we propose a new unsupervised deep clustering method called Mixture of Sparse Autoencoders (Mix-SAE). The proposed network uses a Mixture of Experts (MoE) architecture applied to Sparse Autoencoders (SAE), as detailed in Section 3. After clustering, we assign speakers to each segment and generate the diarization prediction by organizing the segments according to these assignments.

Fig. 2. *The Sparse Autoencoder Architecture (SAE)*

3   Mixture of Sparse Autoencoder Deep Clustering Network (Mix-SAE)

Our proposed Mix-SAE architecture, shown in Fig. 3, consists of two main parts: A set of k-sparse autoencoders, each representing a speaker cluster; and a gating projection that interprets the outputs produced by each autoencoder and assigns the input to a specific sparse autoencoder via its trainable weights.

3.1   Individual Sparse Autoencoder (SAE)

Consider one sparse autoencoder $\mathcal{A}$, represented at Fig. 2. The sparse autoencoder $\mathcal{A}$ has $2L + 1$ layers, including one encoder ($\mathcal{E}$) with L layers, one decoder ($\mathcal{D}$) with L layer and one latent layer. We denote $a_j^{(l)}$ as the activation of hidden unit j at the l-th hidden layer, $z_j^{(i)}$ is the input of i-th sample that leads to hidden unit j. Inspired from [17], we obtain the average activation of hidden unit j at l-th layer over one batch of N samples, which is written as:

$$\hat{\rho}_j^{(l)} = \frac{1}{N} \sum_{i=1}^{N} \left[g\left(a_j^{(l)}(z_j^{(i)}) \right) \right] \tag{1}$$

where the mapping $g(.)$ uses the sigmoid function, which aims to scale the activation parameter to $[0;1]$ and avoid too large value of $\hat{\rho}_j^{(l)}$. The sparsity constraint ensures the average activation $\hat{\rho}_j^{(l)}$ is close to the sparsity parameter ρ, which is quite small. This helps the model learn meaningful features while avoiding copying or memorizing the input by enforcing a limited number of activation neurons in the hidden layer. To achieve the approximation $\hat{\rho}_j \approx \rho$, we leverage Kullback–Leibler divergence penalty term [17]. The KL penalty term applied for the l-th hidden layer that has $n^{(l)}$ hidden units can be written as:

$$\mathcal{L}_{\text{pen}}^{(l)} = \sum_{j=1}^{n^{(l)}} \text{KL}(\rho || \hat{\rho}_j^{(l)}) = \sum_{j=1}^{n^{(l)}} \rho \log \frac{\rho}{\hat{\rho}_j^{(l)}} + (1 - \rho) \log \frac{1 - \rho}{1 - \hat{\rho}_j^{(l)}} \tag{2}$$

Then, the penalty term is calculated for all hidden layers of the autoencoder $\mathcal{A}$ (except the latent layer) by taking the sum of KL terms as:

$$\mathcal{L}_{\text{pen}} = \sum_{l=1}^{2L} \sum_{j=1}^{n^{(l)}} \rho \log \frac{\rho}{\hat{\rho}_j^{(l)}} + (1 - \rho) \log \frac{1 - \rho}{1 - \hat{\rho}_j^{(l)}} \tag{3}$$

We also apply MSE loss for the pair of input data $\boldsymbol{x}$ and reconstruction data $\overline{\boldsymbol{x}}$ for one batch of N samples as:

$$\mathcal{L}_{\text{MSE}} = \frac{1}{2N} \sum_{i=1}^{N} ||\boldsymbol{x}_i - \mathcal{D}(\mathcal{E}(\boldsymbol{x}_i))||_2^2 \tag{4}$$

Given the KL penalty and MSE losses, we define the final objective function for the optimization of one individual sparse autoencoder $\mathcal{A}$:

$$\mathcal{L}_{\text{SAE}} = \mathcal{L}_{\text{MSE}} + \beta \mathcal{L}_{\text{pen}} \tag{5}$$

where β is the parameter to control the effect of sparsity constraint on the objective function.

3.2   k-Sparse Autoencoders

Given the problem of clustering a set of M points $\{\boldsymbol{x}^{(i)}\}_{i=1}^{M} \in \mathbb{R}^m$ into K clusters, the classical k-Means algorithm uses a centroid to represents each cluster in the embedding space, the centroids are mostly calculated by taking the average of all points belonging to that cluster. Inspired by [5] and [20], we use k-autoencoders to represent k clusters, with each autoencoder's latent space acting as a cluster centroid. In this paper, we use sparse autoencoders instead of standard ones, resulting in k-sparse autoencoders as shown in Fig.3. This approach allows data points in the same cluster have their own autoencoder, making feature learning more efficient compared to using a single autoencoder for all data [5]. In our deep clustering network, all k-sparse autoencoders share the same settings and loss function $\mathcal{L}_{\text{SAE}}$ from Equation 5.

3.3   Gating Projection

The role of the Gating Projection ($\mathcal{G}$) is to assign weights $\hat{\boldsymbol{p}} = [\hat{p}_1, \hat{p}_2, ..., \hat{p}_k]$ to the outputs of k-sparse autoencoders based on the input data. Given the weights of $\hat{\boldsymbol{p}} = [\hat{p}_1, \hat{p}_2, ..., \hat{p}_k]$, the Gating Projection is also utilized to assign labels for clusters during the inference phase. In this work, the Gating Projection leverages an MLP architecture with a single linear layer, followed by Leaky ReLU activation and the final softmax layer. Given the input data $\boldsymbol{x}$, the Gating Projection ($\mathcal{G}$) produces weights $\hat{\boldsymbol{p}} = [\hat{p}_1, \hat{p}_2, ..., \hat{p}_k]$ as:

$$\hat{\boldsymbol{p}} = [\hat{p}_1, \hat{p}_2, ..., \hat{p}_k] = Softmax(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}) \in \mathbb{R}^k \tag{6}$$

where $\boldsymbol{W} \in \mathbb{R}^{k \times m}$, $\boldsymbol{b} \in \mathbb{R}^k$ are the trainable weights and bias of the linear layer in the gating projection.

Fig. 3. *The overall architecture of Mix-SAE clustering network*

Fig. 4. *The Pre-training step of Mix-SAE clustering network*

3.4 Training strategy

The training strategy for our proposed Mix-SAE clustering network includes two steps: Pre-training and Main-training.

In the Pre-training step as shown in Fig. 4, we first train a single main sparse autoencoder $\mathcal{A}_{\text{pre}}$ as shown in the upper part of Fig. 4, for the entire dataset using the loss function described at equation 5. After training the main sparse autoencoder $\mathcal{A}_{\text{pre}}$, one off-the-shelf cluster algorithm such as AHC or k-Means, is utilized to obtain initial pseudo-labels $\boldsymbol{P}^{[0]}$ from the learned latent representation of the sparse autoencoder $\mathcal{A}_{\text{pre}}$. Next, we initialize the parameters of k-sparse autoencoders by sequentially training the j-th sparse autoencoder $\mathcal{A}_j$ with the subset of points such that $\boldsymbol{P}^{[0]}[c = j]$, as shown in the lower part of Fig. 4, where c denotes the cluster index, $j = 1, 2, ..., k$. Notably, the training process of k-sparse autoencoders also use Equation 5 as the loss function.

The next Main-training step is described in Fig. 3. This step involves the joint optimization of the k-sparse autoencoders with initialized parameters ob-

tained from the Pre-training step, and the predicted probabilities from the gating projection. Given k-sparse autoencoders $\{\mathcal{A}_1(\theta_1), \mathcal{A}_2(\theta_2), ..., \mathcal{A}_k(\theta_k)\}$, where θ_j is the parameters of encoder $(\mathcal{E}_j)$ and decoder $(\mathcal{D}_j)$ of sparse autoencoder $\mathcal{A}_j$, $j = 1, 2, ..., k$, and the parameters $(\boldsymbol{W}, \boldsymbol{B})$ of the gating projection $\mathcal{G}$, the main objective function of the proposed Mix-SAE network for one batch of N samples $[\boldsymbol{x}^{(1)}, \boldsymbol{x}^{(2)}, ..., \boldsymbol{x}^{(N)}]$ is defined as:

$$\mathcal{L}_{\text{main}}(\theta_1, \theta_2, ..., \theta_k, \boldsymbol{W}, \boldsymbol{B}) = \mathcal{L}_{\text{rec}} + \alpha\mathcal{L}_{\text{ent}} \tag{7}$$

where α is the parameter to constrain the effect of both terms on the main objective function.

The term $\mathcal{L}_{\text{rec}}$ is the weighted sum of reconstruction error over k-sparse autoencoders. This term ensures that the sparse autoencoders could have information on inter-cluster reconstruction error to further strengthen feature learning within their own clusters. We define this term as:

$$\mathcal{L}_{\text{rec}} = -\frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{k}\hat{p}_j^{(i)}\exp\left[-\frac{1}{2}\left(\boldsymbol{x}^{(i)} - \mathcal{D}_j(\mathcal{E}_j(\boldsymbol{x}^{(i)}))\right)^2\right]$$
$$\text{s.t. } \sum_{j=1}^{k}\hat{p}_j^{(i)} = 1, \quad \forall i = 1, 2, \dots, N. \tag{8}$$

where $\mathcal{D}_j(\mathcal{E}_j(\boldsymbol{x}^{(i)}))$ is the output of the j-th sparse autoencoder given the input sample $\boldsymbol{x}^{(i)}$; the probability $\hat{p}_j^{(i)}$, which is computed from $(\boldsymbol{W}, \boldsymbol{B})$ in Equation 6, is the weight from the gating projection assigned to the j-th reconstruction loss.

The term $\mathcal{L}_{\text{ent}}$ is referred to as the pseudo-label guided supervision loss. We denote the pseudo-labels for one batch of N samples at epoch t as: $\mathbf{P}^{[t]} = [\boldsymbol{p}_1^{[t]}, \boldsymbol{p}_2^{[t]}, ..., \boldsymbol{p}_N^{[t]}]$, where $\boldsymbol{p}_i^{[t]} \in \mathbb{R}^k$. The supervision loss is defined as the Cross-Entropy loss between the pseudo-labels $\boldsymbol{p}^{[t_{\text{u}}]}$ previously updated at epoch t_{u} and the prediction of the gating projection $\hat{\boldsymbol{p}}^{[t_{\text{u}}]}$ at the current epoch t:

$$\mathcal{L}_{\text{ent}} = -\frac{1}{N}\sum_{i=1}^{N}\boldsymbol{p}_i^{[t_{\text{u}}]}\log\hat{\boldsymbol{p}}_i^{[t]} \tag{9}$$

The entropy loss $\mathcal{L}_{\text{ent}}$ uses pseudo-labels to provide additional learning signals, simulating a semi-supervised setting. This guides the model towards correct clustering and enhances feature learning [13]. Notably, pseudo-labels are periodically updated after τ epochs during optimization by the predictions of the gating projection $\mathcal{G}$ at the current epoch t using Equation 6. This process aims to reinforce reliable pseudo-labels while correcting noisy ones over time.

After the Main-training step, final cluster label can be inferred via the gating projection $\mathcal{G}$. Given each data sample $\boldsymbol{x}$, the probability vector $\hat{\boldsymbol{p}}$ is calculated using equation 6. Then, the cluster label is determined as:

$$\hat{c} = \operatorname*{argmax} \hat{\boldsymbol{p}} = \operatorname*{argmax}_{j=1,2,..k} \hat{p}(c = j|\boldsymbol{x}) \tag{10}$$

Overall, steps in the training strategy of our proposed Mix-SAE clustering network can be summarized in Table 2.

Table 2. Mix-SAE Deep Clustering Network

Algorithm 1: Mix-SAE mini-batch training strategy

Input: One batch of N points $\boldsymbol{X} = \{\boldsymbol{x}^{(i)}\}_{i=1}^{N} \in \mathbb{R}^m$.

Output: One of k cluster labels for N input points.

Components:

- A set of k-autoencoders: $\{\mathcal{A}_1, \mathcal{A}_2, ..., \mathcal{A}_k\}$

$$\boldsymbol{x} \to \bar{\boldsymbol{x}}_j = \mathcal{D}_j(\mathcal{E}_j(\boldsymbol{x})), \quad j = 1, 2, ..., k.$$

- The Gating Projection $\mathcal{G}$ that produces pseudo-labels and assigns input to suitable autoencoders.

$$\boldsymbol{p} = Softmax(\boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}) \in \mathbb{R}^c.$$

● **Pre-training:**

- Train a single autoencoder $\mathcal{A}_{\text{pre}}$ for the entire dataset with the objective function (5).

- Use one off-the-shelf cluster algorithm to initialize pseudo-labels $\boldsymbol{P}^{[0]}$ for the entire dataset.

for $j = 1$ to k **do**:

   Train j-th sparse autoencoder with data points $\boldsymbol{P}^{[0]}[c = j]$.

end for

● **Main-training:**

for $t = 1$ to T **do**:

   Train the set of k sparse auencoders and the gating projection $\mathcal{G}$ jointly using the main objective function (7).

   if $t \bmod \tau = 0$ **then**:

      Update new pseudo-labels $\boldsymbol{P}^{[t_{\text{u}}]}$ for the batch $\boldsymbol{X}$:

$$t_{\text{u}} \leftarrow t$$

$$\boldsymbol{P}^{[t_{\text{u}}]} = \underset{\text{axis} = 1}{\operatorname{argmax}} [Softmax(\boldsymbol{W}\boldsymbol{X} + \mathbf{B})]$$

Get final cluster result: Get the final cluster result for the batch $\boldsymbol{X}$ via the gating projection $\mathcal{G}$:

$$\hat{\boldsymbol{P}} = \underset{\text{axis} = 1}{\operatorname{argmax}} [Softmax(\boldsymbol{W}\boldsymbol{X} + \boldsymbol{B})]$$

---

4 Experimental Settings And Results

4.1 Evaluation Datasets

To evaluate the performance and generalization of our proposed system to diverse data sources, we gather data from two benchmark corpora CALL-HOME [2],[4],[1] and CALLFRIEND [16], [15]. Each corpus includes various language subsets like English, German, French, Spanish, and Japanese, with multiple telephone conversations from different sources. For evaluation, we use two-speaker subsets of the above benchmark corpora (the most common case in telephone call applications), to form a combined dataset called SD-EVAL. The SD-EVAL dataset comprises 127 recordings totaling around 6.35 hours and is divided into four language-specific subsets: English (EN), Spanish (SPA), German (GER), and French (FR). Each subset has 25 to 35 recordings, each lasting 2 to 5 minutes.

4.2 Evaluation Metrics

We evaluated the proposed sysetm with diarization error rate (DER).

4.3 Experimental settings

The proposed method was implemented with deep learning framework Py-Torch [23]. The network architecture consists of autoencoders with hidden layers

Table 3. Diarization Error Rate (DER) (%) of different systems on SD-EVAL dataset (Whisper version: Tiny, no tolerance)

Methods	W = 0.2 s				W = 0.4 s				W = 0.6 s				W = 0.8 s				W = 1.0 s			
	EN	FR	GER	SPA	EN	FR	GER	SPA	EN	FR	GER	SPA	EN	FR	GER	SPA	EN	FR	GER	SPA
k-Means	44.77	51.42	49.11	48.25	43.75	51.92	43.84	47.08	38.72	46.88	40.97	42.77	40.23	46.61	44.11	44.38	42.06	47.72	46.13	44.66
AHC	38.42	46.72	41.41	42.93	47.64	52.81	46.33	50.69	40.50	48.69	42.90	43.15	38.55	45.91	43.02	43.44	42.91	47.81	47.63	44.80
SpectralNet [29]	36.18	44.62	40.02	46.03	40.44	51.63	41.22	47.52	37.06	44.68	41.29	42.69	36.11	44.67	44.16	46.42	41.88	46.08	44.31	47.23
DCN [34]	32.15	**35.77**	36.51	36.98	37.42	38.92	42.17	43.01	32.08	37.57	38.84	40.77	33.02	43.72	44.23	40.55	40.17	45.96	40.21	38.51
DAMIC [5]	27.78	36.22	36.93	35.21	27.97	35.96	36.14	35.11	28.11	36.67	34.66	33.31	27.22	**36.91**	34.78	34.22	26.95	**36.91**	36.11	34.65
k-DAE [20]	29.12	37.91	41.23	37.00	30.53	39.81	37.10	37.29	32.72	38.84	34.96	35.23	33.33	38.55	34.24	35.51	30.36	37.32	36.22	35.02
Mix-SAE-V1	32.18	38.61	36.07	36.78	29.02	**35.92**	36.51	35.04	27.28	37.01	34.98	34.03	27.90	37.51	34.42	33.83	28.00	37.88	36.18	34.29
Mix-SAE-V2	28.72	43.22	40.66	36.32	29.62	40.07	36.71	35.72	27.81	36.83	34.90	33.54	27.98	39.68	34.62	33.21	27.93	38.05	36.73	**33.82**
Mix-SAE	**26.51**	36.12	**35.00**	**34.91**	26.88	37.30	**35.64**	**34.33**	27.08	36.70	**34.55**	**32.82**	**27.24**	38.39	**34.17**	**32.03**	26.85	37.57	**35.33**	**33.82**

[256, 128, 64, 32] for the encoder and mirrored for the decoder, using Leaky ReLU activation and Batch Normalization followed each hidden layer. The latent vector size is also k (equal to the number of speakers), with mini-batch size $N = 16$. We use k-Means^{++} [3] to initialize pseudo-labels in the Pre-training step.

Regarding hyperparameters, we set sparsity parameter $\rho = 0.2$, sparsity constraint $\beta = 0.01$, pseudo-label supervision $\alpha = 1$. The training process uses learning rate 0.001 and weight decay 5×10^{-4}. The Pre-training step involves 50 epochs for the main autoencoder $\mathcal{A}_{\text{pre}}$ and 20 epochs for each of k-sparse autoencoders. The Main training step runs for 29 epochs and updates pseudo-labels after 10 epochs.

4.4   Results and Discussion

Speaker clustering methods: We evaluate several speaker clustering methods using embeddings from the tiny Whisper model, including k-Means, Agglomerative Hierarchical Clustering (AHC), SpectralNet, autoencoder-based methods such as DCN, DAMIC, k-DAE, and our proposed Mix-SAE. Experiments were conducted with segment sizes (W) ranging from 0.2s to 1.0s. As shown in Table 3, Mix-SAE consistently outperforms other methods, achieving the best performance in English with a DER of 26.51%. This can be attributed to high-quality embeddings from Whisper's extensive English training data. While other methods, especially autoencoder-based ones like DCN and k-DAE, show variability with segment size, Mix-SAE remains stable across different W values, demonstrating its efficiency in capturing speaker features from variable-length segments (e.g. the proposed system achieves DER scores of 26.51%, 26.88%, 27.08%, 27.24%, 26.85% on English and 35.00%, 35.64%, 35.55%, 34.17%, 34.55% on German with $W = 0.2, 0.4, 0.6, 0.8, 1.0$, respectively). For an ablation study, we establish two other systems: Mix-SAE-V1 (Mix-SAE without sparsity loss in equation 5), Mix-SAE-V2 (Mix-SAE w/o pseudo-label loss in equation 7). Results in Table 3 demonstrate the role of both sparsity loss and pseudo-label loss in improving the overall performance. For instance, an improvement of 5.67%

Fig. 5. *Evaluation: (a) DER scores using speaker embeddings from different Whisper versions; (b) Compare DER score versus complexity across deep clustering methods*

and 2.21% is obtained in the case of English with $W = 0.2$s when Mix-SAE is compared to Mix-SAE-V1 and Mix-SAE-V2, respectively.

The quality of speaker embeddings: We assessed the impact of speaker embeddings on diarization performance, as shown in Fig. 5a, using different versions of the Whisper model (Tiny, Base, Small, Medium, Large) with W set to 0.2s in English. Larger Whisper models provided superior embeddings, leading to better performance, with the best DER score of 17.75% (0.25s tolerance). This highlights the potential of using general-purpose like Whisper for multilingual and unsupervised speaker diarization systems as well as integrating speaker diarization as a component in Whisper-based speech analysis applications.

The model complexity: Fig. 5b shows the trade-off between model complexity and diarization performance (DER) across deep clustering methods. Our Mix-SAE achieves 26.51% DER with 334k parameters, striking a good balance between accuracy and efficiency. Additionally, when combined with Whisper Tiny (39M), the system is promising for integration into edge devices for sound applications [7], [27].

Visualization and the effect of Pre-training step: We visualized 2-speaker embeddings after the Pre-training step in our Mix-SAE by applying t-SNE. As Fig. 6 shows, the sparse autoencoders effectively learn underlying patterns from extracted speaker embeddings and map them into latent space where the embeddings of two speakers were relatively well-separated. These clustering results serve as pseudo-labels for optimizing the deep clustering network at the next Main-training step.

5   Conclusion

This paper has presented an unsupervised speaker diarization system for multilingual telephone call applications. In this proposed system, the traditional

(a) English ($W = 0.2s$)   (b) French ($W = 0.2s$)

(c) German ($W = 0.2s$)   (d) Spanish ($W = 0.2s$)

Fig. 6. *t-SNE visualization of speaker embeddings after the pre-training step (Whisper version: Tiny)*

feature extractor was replaced with the Whisper encoder, benefiting from its robustness and generalization on diverse data. Additionally, the Mix-SAE network architecture was also proposed for speaker clustering. Experimental results demonstrate that our Mix-SAE network outperforms other compared clustering methods. The overall performance of our system highlights the effectiveness of our approach when exploring Whisper embedding for the diarization task to develop unsupervised speaker diarization system in the contexts of limited annotated training data. Furthermore, the results also enhances the system's ability to integrate into Whisper-based multi-task speech analysis application. Overall, this work indicates a promising direction toward developing generalized speaker diarization systems based on general-purpose models in future work.

Acknowledgment

References

1. Alexandra, C., Graff, D., Zipperlen, G.: Cabank spanish callhome corpus (1996). https://doi.org/10.21415/T51K54
2. Alexandra, C., Graff, D., Zipperlen, G.: Cabank english callhome corpus (1997). https://doi.org/10.21415/T5KP54
3. Arthur, D., Vassilvitskii, S.: k-means++: the advantages of careful seeding. In: Proceedings of the Eighteenth Annual ACM-SIAM Symposium on Discrete Algorithms. p. 1027–1035. SODA '07, Society for Industrial and Applied Mathematics, USA (2007)
4. Canavan, A., Graff, D., Zipperlen, G.: Cabank german callhome corpus (1997). https://doi.org/10.21415/T56P4B
5. Chazan, S.E., Gannot, S., Goldberger, J.: Deep clustering based on a mixture of autoencoders. In: 29th International Workshop on Machine Learning for Signal Processing (MLSP). pp. 1–6 (2019). https://doi.org/10.1109/MLSP.2019.8918720
6. Dehak, N., et al.: Front-end factor analysis for speaker verification. IEEE Transactions on Audio, Speech, and Language Processing 19(4), 788–798 (2011). https://doi.org/10.1109/TASL.2010.2064307
7. Froiz-Míguez, I., et al.: Design, implementation, and practical evaluation of a voice recognition based iot home automation system for low-resource languages and resource-constrained edge iot devices: A system for galician and mobile opportunistic scenarios. IEEE Access 11, 63623–63649 (2023)
8. Gangadharaiah, R., et al.: A novel method for two-speaker segmentation. In: Proc. INTERSPEECH (2004), https://api.semanticscholar.org/CorpusID:12436529
9. Han, K.J., Narayanan, S.S.: A robust stopping criterion for agglomerative hierarchical clustering in a speaker diarization system. In: Proc. INTERSPEECH (2007), https://api.semanticscholar.org/CorpusID:17876640
10. Li, Y., Wang, W., Liu, M., Jiang, Z., He, Q.: Speaker clustering by co-optimizing deep representation learning and cluster estimation. IEEE Transactions on Multimedia 23, 3377–3387 (2020)
11. McFee, B., Raffel, C., Liang, D., Ellis, D.P., McVicar, M., Battenberg, E., Nieto, O.: librosa: Audio and music signal analysis in python. In: SciPy. pp. 18–24 (2015)
12. Milner, R., Hain, T.: Dnn-based speaker clustering for speaker diarisation. In: Proc. INTERSPEECH (2016), https://api.semanticscholar.org/CorpusID:26152646
13. Min, Z., Ge, Q., Tai, C.: Why the pseudo label based semi-supervised learning algorithm is effective? (2023)
14. Mofrad, M.H., et al.: Speech recognition and voice separation for the internet of things. In: Proceedings of the 8th International Conference on the Internet of Things. pp. 1–8 (2018)
15. Mondada, L., Granadillo, T.: Cabank spanish callfriend corpus. https://doi.org/10.21415/T5ZC76
16. Mondada, L., et al.: Cabank french callfriend corpus. https://doi.org/10.21415/T5T59N
17. Ng, A., et al.: Sparse autoencoder. CS294A Lecture notes 72(2011), 1–19 (2011)
18. Nguyen, T.N.T., et al.: A general network architecture for sound event localization and detection using transfer learning and recurrent neural network. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 935–939 (2021). https://doi.org/10.1109/ICASSP39728.2021.9414602

19. Nguyen, T., Pham, L., Lam, P., Ngo, D., Tang, H., Schindler, A.: The impact of frequency bands on acoustic anomaly detection of machines using deep learning based model. arXiv preprint arXiv:2403.00379 (2024)
20. Opochinsky, Y., et al.: K-autoencoders deep clustering. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 4037–4041 (2020). `https://doi.org/10.1109/ICASSP40776.2020.9053109`
21. Pal, M., et al.: Speaker diarization using latent space clustering in generative adversarial network. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 6504–6508 (2020)
22. Park, T.J., Kanda, N., Dimitriadis, D., Han, K.J., Watanabe, S., Narayanan, S.: A review of speaker diarization: Recent advances with deep learning. Computer Speech & Language **72**, 101317 (2022)
23. Paszke, A., et al.: Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems 32, pp. 8024–8035. Curran Associates, Inc. (2019), `http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf`
24. Pham, L., Ngo, D., Salovic, D., Jalali, A., Schindler, A., Nguyen, P.X., Tran, K., Vu, H.C.: Lightweight deep neural networks for acoustic scene classification and an effective visualization for presenting sound scene contexts. Applied Acoustics **211**, 109489 (2023)
25. Pham, L., Nguyen, T., Lam, P., Ngo, D., Jalali, A., Schindler, A.: Light-weight deep learning models for acoustic scene classification using teacher-student scheme and multiple spectrograms. In: 4th International Symposium on the Internet of Sounds. pp. 1–8 (2023). `https://doi.org/10.1109/IEEECONF59510.2023.10335258`
26. Quatra, M.L., et al.: Vad - simple voice activity detection in python ([Online] Available: https://githubcom/MorenoLaQuatra/vad)
27. Ramírez, A., Foster, M.E.: A whisper ros wrapper to enable automatic speech recognition in embedded systems (2023)
28. Reynolds, D.A., Quatieri, T.F., Dunn, R.B.: Speaker verification using adapted gaussian mixture models. Digital Signal Processing **10**(1), 19–41 (2000). `https://doi.org/https://doi.org/10.1006/dspr.1999.0361`
29. Shaham, U., Stanton, K., Li, H., Nadler, B., Basri, R., Kluger, Y.: Spectralnet: Spectral clustering using deep neural networks (2018)
30. Snyder, D., et al.: X-vectors: Robust dnn embeddings for speaker recognition. In: IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). pp. 5329–5333 (2018). `https://doi.org/10.1109/ICASSP.2018.8461375`
31. Stafylakis, T., Katsouros, V., Carayannis, G.: Speaker Clustering via the mean shift algorithm. In: Proceedings of the Speaker and Language Recognition Workshop (Speaker Odyssey). pp. 186 – 193. ISCA, Brno, Czech Republic (2010)
32. Tritschler, A., Gopinath, R.A.: Improved speaker segmentation and segments clustering using the bayesian information criterion. In: EUROSPEECH (1999), `https://api.semanticscholar.org/CorpusID:15220583`
33. Wan, L., Wang, Q., Papir, A., Moreno, I.L.: Generalized end-to-end loss for speaker verification (2020)
34. Yang, B., Fu, X., Sidiropoulos, N.D., Hong, M.: Towards k-means-friendly spaces: Simultaneous deep learning and clustering (2017)
35. Zhang, A., Wang, Q., Zhu, Z., Paisley, J., Wang, C.: Fully supervised speaker diarization (2019)