Semi-Supervised Online Speaker Diarization using Vector Quantization with Alternative Codebooks

Mahmoud El-Hindi, Michael Muma, Abdelhak M. Zoubir
[1]*Technische Universität Darmstadt, Germany*
Email: {melhindi, muma, zoubir}@spg.tu-darmstadt.de

***Abstract*—Speaker diarization systems process audio files by labelling speech segments according to speakers' identities. Many speaker diarization systems work offline and are not suited for online applications. We present a semi-supervised, online, low-complexity system. While, in general, speaker diarization operates in an unsupervised manner, the presented system relies on the enrollment of the participating speakers in the conversation. The diarization system has two main novel aspects. The first one is a proposed online learning strategy that evaluates processed segments according to their usefulness for learning a speaker, i.e. update a speaker model with it. The segment is evaluated using two metrics to determine whether to use the segment to update the system. The second novel aspect is a proposed vector quantization approach that models the score not only depending on the target speaker codebook but also takes an alternative codebook into account. We also present an approach to compute the alternative codebook. Simulation results show that the proposed system outperforms a comparable system without the proposed online learning strategy and shows benefits, especially for short training lengths.**

***Index Terms*—speaker diarization, vector quantization, online learning**

I. Introduction

Speaker diarization systems aim to determine from an audio recording the instances at which a speaker or multiple speakers were active. The application of speaker diarization includes, e.g., recorded conversations in the context of broadcast news, phone conversations, or business meetings. Traditionally, speaker diarization systems are built as offline systems which have access to the complete audio file. This enables the diarization system to segment the entire audio file and assign labels to speakers. From the audio file, speaker representations are extracted, such as binary key vector [1], i-vector [2], d-vector [3] or x-vector [4]. Recently, research has explored the usage of end-to-end neural diarization, where a single neural network processes the audio file directly and assigns the speaker labels [5],[6]. However, in some applications, the system needs to generate the speaker labels in real-time or with a small amount of delay (i.e., online). This requirement makes an offline system and its corresponding techniques unusable. For example, offline speaker diarization systems can process multiple times over the same data and iteratively split and merge over all the segments since all segments are known in advance [7]. In contrast, online speaker diarization systems have to make all decisions using previous and current input data. Traditionally, this is addressed by the usage of generic models as in [8],[9]. Possible generic models are the universal background model (UBM), Gender Background Models (GBM) or a set of sample speaker models (SSM) [9]. However, these approaches assume that no data or speaker models of the speakers are available a priori. Therefore, unsupervised approaches try to assign the audio segments to speakers without having seen any enrollment data that contains label information [10], [11]. In this paper, we focus on semi-supervised approaches which assume that a small amount of enrollment data for each speaker is available at the beginning of a conversation. This assumption is reasonable, e.g. in the context of business meetings, as participants introduce themselves at the beginning of the meeting. While the enrollment data is labeled, the subsequent segments need to be processed in an unsupervised manner. During processing, the semi-supervised system is not only required to assign a label to a given speech segment, but it also needs to evaluate whether to use the assigned speech segment for further training of the speaker model. As we will discuss in the next section, simply using all assigned speech segments for further learning is problematic, because learning from falsely labeled segments might even degrade the speaker models. This observation motivated our novel approach that is introduced in Section III. An evaluation is given in Section IV, while conclusions are drawn in Section V.

II. Semi-Supervised Online Speaker Diarization

In the sequel, we give a brief overview of semi-supervised approaches and discuss why using all segments for online learning is problematic.

A. General Overview

In general, semi-supervised online learning speaker diarization systems perform two tasks. The first is to label each segment of the audio file with the ID of the corresponding speaker. As it is assumed that all participants enroll themselves at the beginning of the conversation, the segment labeling boils down to a *closed set speaker recognition* task [10].

While the first task can be addressed by supervised learning approaches, the second one is unsupervised, since semi-supervised approaches perform online learning during the processed conversation without any reference label information. As the conversation goes on, the system needs to update and enhance the enrolled speaker models, since – as shown later – the short enrollment data of only a few seconds is not sufficient to yield accurate models. In the context of this

paper, it is also assumed that no further information other than from a single microphone can be used. Hence, no direction of arrival information or visual information, e.g., from a camera is available. Recently the authors in, [10] introduced a speaker diariaztion system that uses all incoming audio segments to update their model. In the next section, we shed light on situations where such an approach is not favorable.

B. Always Learning Approach

The authors in [10] present an online speaker diarization system for meetings that utilizes incremental maximum a-posteriori (MAP) adaptation. During the processing, non-speech segments are removed while the remaining speech segments are divided into sub-segments of a fixed duration T_s. The diarization task is then applied to all sub-segments. For each sub-segment, one of the M speakers is assigned according to

$$\hat{s}_j = \arg\max_{l \in 1,..,M} \sum_{k=1}^{K} \mathcal{L}(\mathbf{o}_k \mid s_l) \tag{1}$$

where s_l donates the model of speaker l, $\mathbf{o}_k$ is the k-th speech feature vector in the sub-segment with K speech feature vectors and $\mathcal{L}(\mathbf{o}_k|s_l)$ is the log-likelihood of the feature vector $\mathbf{o}_k$ given the speaker model s_l, with $l \in 1, ..., M$. The speaker j with the highest log-likelihood is then selected as the corresponding speaker for the segment. Subsequently, the speaker model s_j is updated using the segment and assigned label.

In [10], the authors present two approaches to train the speaker model. One is a sequential MAP adaptation and the other one is an incremental MAP adaptation. In both cases, the learning strategy remains always to use every labeled segment to update the speaker model. This strategy, however, has the risk of degrading model performance since an error in the labeling will result in training the speaker model with data from a wrong speaker. This problem is not as severe for a well-performing model, but as we will show in our simulations, it has a significant impact on a model with poor initial performance. Ironically, the online training of a speaker diarization system is particularly of interest when the initial performance is not as good as desired. To address this challenge, we propose a novel semi-supervised online speaker diarization system that ignores certain audio segments to avoid performance degradation.

III. Selective Semi-supervised Online Learning

Figure 1 provides an overview of our proposed system. Similar to [10], the system consists of two main steps, which are first to label the segment with a speaker ID, and second, to learn the speaker models. The first step is conducted similarly as in [10]. However, the second step differs in our proposed system. In [10], the authors proposed an incremental MAP adaptation to update the speaker model for semi-supervised online speaker diarization. The underlying learning strategy of [10] is to use each classified segment to update the corresponding speaker model. In what follows, this will be referred to as the *always learning strategy*. In contrast, we propose a different learning strategy, which evaluates each segment before it decides whether to use the segment for learning or not. The proposed system has two main novel components. One is the overall learning strategy and the other one is a novel vector quantization (VQ) modulation that we use to evaluate segments. To process a segment, the proposed system takes as input the currently examined segment of the conversation. Each segment is processed using a voice activity detection (VAD) algorithm. The next step is to compute the LLR score of each speaker and the VQ score of each speaker. The used VQ score is computed using a novel method. This novel method is described further in Section III. Based on the LLR score, the diarization task, which is to select the recognized speaker, is carried out. This is done according to Eq. (1), which selects the speaker with the highest LLR. Parallel to this, also the learning decision is evaluated using the LLR and VQ scores. Depending on the result of the learning decision, the speaker model of the corresponding speaker is updated.

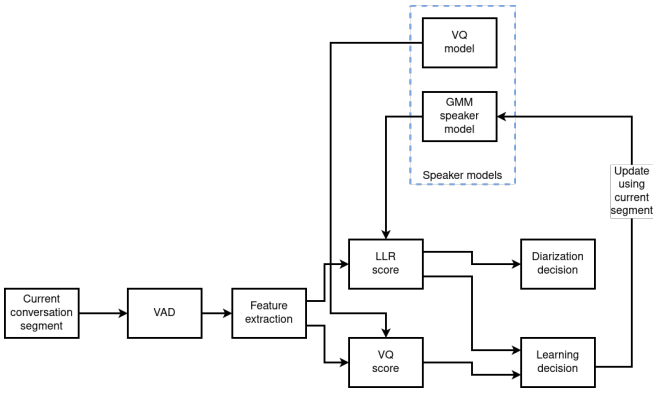


Fig. 1: System overview of the proposed system for the unsupervised online learning for semi-supervised online speaker diariazation

A. Selective Learning Strategy

In classical VQ, the Euclidean distance between a codebook and feature vectors is computed. A smaller score, in general, indicates a match between the codebook and hence the speaker and the speech feature vectors under test [12]. The range of the classical VQ score is always greater or equal to zero, and higher values indicate a mismatch between the underlying codebook and the examined feature vectors. There are several approaches to generate the codebook of the target speaker, for example, the usage of the K-means [13] or the Linde-Buzo-Gray (LBG) clustering [14] algorithms. In our novel approach, the proposed variant of the VQ is computed by additionally using an alternative codebook. In our modified VQ approach, the score is modeled as the difference between the Euclidean distance of the alternative codebook and the Euclidean distance of the target codebook. With this modeling, the VQ score can take positive as well as negative values. A high positive value indicates that the feature vectors belong to the target speaker, while a negative value indicates that the feature vectors do not belong to the target speaker or at least that they are not very characteristic for this speaker.

B. Alternative Codebook Construction

In this subsection, the derivation and the construction of the alternative codebook are described. To generate the alternative codebook inspired by [15], we design a novel procedure that is based on the idea of minimizing a cost function. Optimally, the objective would be similar to:

$$\min_{\text{CVQ}^{\text{alt}}} \sum_{n=0}^{N-1} \left(\text{CVQ}^{\text{alt}}(\mathbf{o}_n) - \text{CVQ}^{\text{Target}}(\mathbf{o}_n)\right)^2,$$

where $\text{CVQ}^{\text{Target}}(\mathbf{o}_n)$ is the vector quantization of the target speaker codebook of the feature vector $\mathbf{o}_n$ and $\text{CVQ}^{\text{alt}}(\mathbf{o}_n)$ is the vector quantization of the alternative speaker codebook of the feature vector $\mathbf{o}_n$. The target speaker codebook is assumed as given and the goal (i.e. optimization target) is to find an optimal alternative speaker representation. When computing the vector quantization noise from the codebook, the entry that exhibits the smallest distance is selected. The other codebook entries are not considered. For solving such an optimization problem which requires selecting a codebook entry with the smallest Euclidean distance, there currently exists no efficient method. To avoid the occurrence of the minimum, we, therefore, reformulate the problem. First, we eliminate the selection of the entry with the minimum Euclidean distance by replacing the vector quantization of a codebook with

$$\sum_{m=1}^{M} e^{-\alpha(\mathbf{o}_n-\mathbf{c}_m)^\top(\mathbf{o}_n-\mathbf{c}_m)}.$$

where $\mathbf{c}_m$ represents the codebook entry m of the alternative speaker and α is a tuning factor. The idea behind this formulation is to reduce the influence of the entries that do not match the feature vector. Ideally, only the entry which matches the feature vector will influence the sum. Using this expression, we model a similar value as the vector quantization of the alternative speaker codebook but have a mathematical description that we can still optimize. The vector quantization of the target speaker codebook is modeled with $\tilde{a}_n = e^{-\alpha a_n}$, where $a_n = \text{CVQ}^{\text{Target}}(\mathbf{o}_n)$ is the target speaker codebook vector quantization value of the vector $\mathbf{o}_n$. As the target codebook and the feature vectors $\mathbf{o}_n$ are given, $\tilde{a}_n$ is fixed.
Using this, we can formulate the optimization task as follows:

$$\min_{\mathbf{c}_m} \sum_{n=0}^{N-1} \left(\tilde{a}_n - \sum_{m=1}^{M} e^{-\alpha(\mathbf{o}_n-\mathbf{c}_m)^\top(\mathbf{o}_n-\mathbf{c}_m)}\right)^2.$$

The advantage of this formulation is that we can compute the derivative of the expression and optimize the variable $\mathbf{c}_m$ to obtain the alternative speaker codebook. The next step is to take the derivative with respect to $\mathbf{c}_m$, which results in

$$\begin{aligned}
&\frac{\partial}{\partial \mathbf{c}_m} \sum_{n=0}^{N-1} \left(\tilde{a}_n - \sum_{m=1}^{M} e^{-\alpha(\mathbf{o}_n-\mathbf{c}_m)^\top(\mathbf{o}_n-\mathbf{c}_m)}\right)^2 \\
&= \sum_{n=0}^{N-1} 2\left(\tilde{a}_n - \sum_{m=1}^{M} e^{-\alpha(\mathbf{o}_n-\mathbf{c}_m)^\top(\mathbf{o}_n-\mathbf{c}_m)}\right) \\
&\quad \left(e^{-\alpha(\mathbf{o}_n-\mathbf{c}_m)^\top(\mathbf{o}_n-\mathbf{c}_m)}\left(2\alpha(\mathbf{o}_n-\mathbf{c}_m)\right)\right).
\end{aligned}$$

The left bracket in the sum is a scalar and can be interpreted as a weight that describes how well we quantize the vector $\mathbf{o}_n$ with respect to the target codebook. If the target codebook has a low quantization noise for the feature vector, but the alternative codebook has a high quantization noise this feature vector will have a higher weight. If the target codebook has a high quantization noise value and also the alternative codebook, this feature vector will have a lower weight. Hence, this scalar can be interpreted as a weight for the feature vector. The weight is low if the alternative codebook has a similar quantization noise value as the target speaker codebook. On the other hand, the weight is high if the two codebooks have a high quantization noise value difference, resulting in an emphasis on adapting towards this feature vector. The right bracket of the sum is a vector and can be interpreted as the direction in which we should adapt. Overall, we have obtained a formulation that can be used as an update coefficient for the codebook entry.
In [15], the authors compute the codebook in an iterative fashion, which updates each codebook entry using a gradient. We also use this approach to compute the codebook. Each codebook entry is updated with:

$$\mathbf{c}_m(t+1) = \mathbf{c}_m(t) + \eta \mathbf{q}_m(t)$$

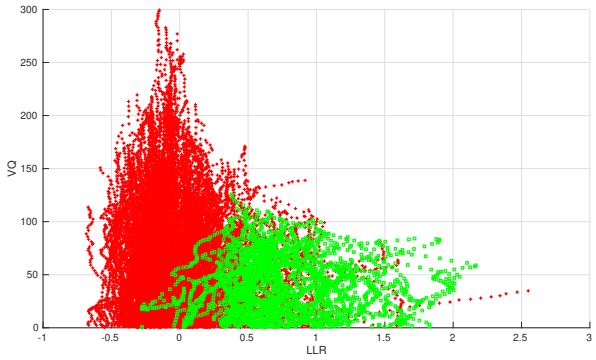
where η is the step-width, t the iteration index and $\mathbf{q}_m(t)$ is

$$\begin{aligned}
\mathbf{q}_m(t) = \quad & \textstyle\sum_{n=0}^{N-1} 2\left(\tilde{a}_n - \sum_{m=1}^{M} e^{-\alpha(\mathbf{o}_n-\mathbf{c}_m(t))^\top(\mathbf{o}_n-\mathbf{c}_m(t))}\right) \\
& \left(e^{-\alpha(\mathbf{o}_n-\mathbf{c}_m(t))^\top(\mathbf{o}_n-\mathbf{c}_m(t))}\left(2\alpha(\mathbf{o}_n-\mathbf{c}_m(t))\right)\right)
\end{aligned}$$

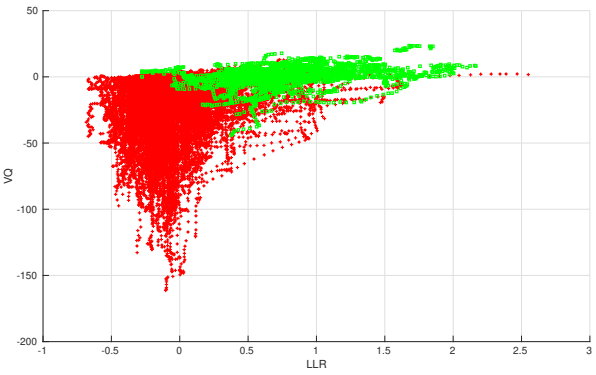
C. Example

In this section, we will show a simple example to illustrate the advantage of the introduced novel VQ. For this, we processed a conversation file from the LibreCSS dataset [16]. For one speaker, we computed the LLR scores of the speech in the conversation and the VQ score for the classical VQ approach and the proposed modification. In Figure 2a, we show the result of the LLR scores and the classical VQ scores. The green marker corresponds to the target speaker and the red marker corresponds to the other participants in the conversation. The scores of LLR and the proposed modified VQ are provided in Figure 2b. Again, the green marker corresponds to the target speaker and the red marker corresponds to the other participants in the conversation. We see that in both figures a higher LLR score corresponds to the target speaker, but some data points belonging to the other participants have a high LLR. If we now try to take the classical VQ into consideration, we see in Figure 2a that a low VQ score occurs for a lot of data points of the other participants. Hence, taking the classical VQ score is not beneficial to detect the target speaker with a low error rate. In contrast, we see in Figure 2b that a high VQ score occurs only for a few data points of the other participants and is beneficial to detect the target speaker with a low error rate. To quantify the overlap between the other participants (red) and the target speaker (green), we can compute the Bhattacharyya coefficient [17] on a 25×25

grid. For Figure 2a we get a value of 0.4358 and for Figure 2b a value of 0.3630, which shows that the novel VQ approaches has smaller overlap. Hence, our proposed approach can be applied to identify more distinguishing segments of the target speaker more reliably.



(a) Example classical VQ and LLR score



(b) Example proposed modified VQ and LLR score

Fig. 2: Example plot of classical VQ and LLR scores (a) and the proposed modified VQ and LLR scores (b). The red marker represents the other speaker in the conversation and the green marker represents the target speaker

IV. Simulations

This section describes the used simulation setup and presents the obtained simulation results for the semi-supervised online speaker diarization. The evaluation aims to show the effect of the different learning strategies and the influence of the amount of available enrollment data.

A. Setup

The speaker diarization is performed using a conventional GMM model as a universal background model system (UBM). To build the UBM, the voxceleb dataset [18] was used. The UBM is trained using the Expectation-Maximization (EM) algorithm with 64 Gaussian components. All the speaker models are derived from the UBM by using the MAP adaptation with the relevance factor set to the value 16, which is a common choice in the literature [19]. From the speech signals, we always extract a feature vector composed of 12 Mel-frequency cepstral coefficients (MFCC) and 12 delta Mel-frequency cepstral coefficients (delta MFCC). Thereby we use a window length of 32ms and a frameshift of 15ms for the feature vector extraction. The speaker diarization task experiments are performed on the LibriSpeechCCS dataset [16]. This dataset is derived from the LibriSpeech dataset [20] by playing utterances via loudspeakers to simulate conversation and capturing the audio signal using a far-field microphone. The conversation consists of eight different speakers. In this setup, it is possible to accurately document when which speaker spoke and control the amount of overlap in the conversation. As this work is an initial discussion of online learning, we only selected the conversation with zero speaker overlap. In [16], there are two subsets with zero overlap, one generated with short-utterance silence which we will refer to as the 0S subset. The other subset is generated with longer inter-utterance silence and we will refer to it as the 0L subset. To initialize each speaker, we extract the amount of training data from the conversation file and cut the selected segment out of the conversation. This segment is randomly drawn from the conversation for each speaker. Further, the experiments are repeated ten times, each time drawing the segments randomly. While this is not optimal, it enables us to decrease the influence of the initialization segment on the performance and more objectively examine the effect of the learning strategy.

B. Results

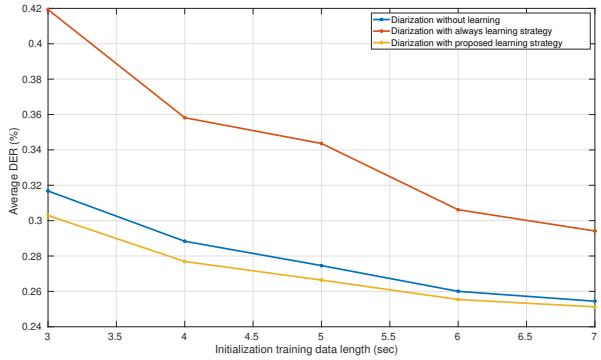
The results of the sub dataset 0L are shown in Figure 3a. Here, the y-axis represents the average diarization error rate (DER), while the x-axis represents the initialization training data length. The average DER is computed by averaging over the DER value of each simulation run and file in the data set. The DER is computed with:

$$\mathbf{DER} = (\mathrm{FA} + \mathrm{Miss} + \mathrm{SpkConf})/(\mathrm{File\ length}),$$

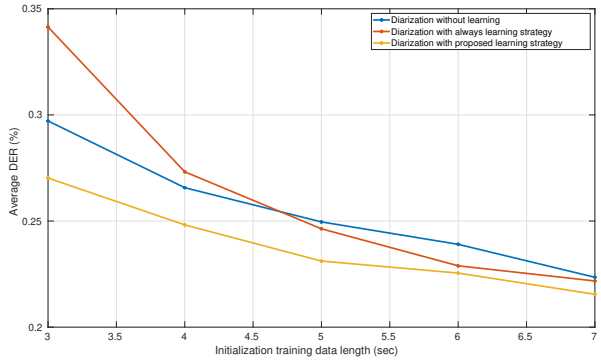
where FA is the false alarm, Miss is the missed detection of speech, and SpkConf is the wrong assignment of the speaker label. In general, we see that, as expected, with increasing length, the performance improves for all three strategies. The always learning strategy, for all the data lengths, exhibits the worst performance and is always outperformed by the diarization without any online learning. This shows that the approach of learning each classified segment is not advantageous in this setting, but rather degrades the performance. The proposed learning strategy outperforms the other two learning strategies for all the data lengths. In particular, for three seconds of initialization training data length, our approach shows a performance improvement of more than 10% compared to the always learning strategy.

The results of the sub-dataset 0S are shown in Figure 3b. Similarly to the sub-dataset 0L, the performance improves for all three strategies with increasing length. And, generally speaking, compared to the 0S sub-dataset, the overall performance improves. Further, we observe that with an increasing data length the always online learning strategy is able to outperform the speaker diarization without any learning. The proposed online learning strategy outperforms the other two

approaches and, again, we observe that the performance improvement obtained using the proposed approach is especially high for short training lengths. Overall the simulation results show the advantage of the proposed learning strategy.



(a) Simulation results for the 0L sub-dataset



(b) Simulation results for the 0S sub-dataset

Fig. 3: Simulation results for sub-dataset 0S (a) and 0L (b) for the three different learning strategies. The x-axis represents the initialization training data length in seconds, while the y-axis represents the average detection error rate

V. Conclusion & Summary

We proposed an unsupervised online learning system for semi-supervised online speaker diarization. The presented approach is characterized by its low complexity and, thus would enable a low latency. The proposed unsupervised learning strategy can improve the speaker model despite initial performance shortcomings. We showed that a learning strategy that uses each classified segment for learning can deteriorate the performance. This deterioration occurs especially when the initial performance is weak. In contrast, the proposed algorithm can outperform the always learning strategy on average and is robust against performance deterioration. The presented work serves to show a new concept of online learning for speaker diarization systems. The goal of online learning is to further improve the performance of the diarization system and reduce the amount of necessary enrollment data. Further, we presented a novel modification of the VQ which allows a different interpretation of VQ scores. The novel VQ models the score as the difference between target and alternative model results. To model the alternative model in VQ, we proposed a new approach that computes an alternative codebook.

Acknowledgment

The authors wish to thank Prof. H. Puder and Dr. M. Lugger for their input.

References

[1] H. Delgado, X. Anguera, C. Fredouille, and J. Serrano, "Improved binary key speaker diarization system," in *Pro 23rd Eur. Signal Process. Conf. (EUSIPCO)*. IEEE, 2015, pp. 2087–2091.

[2] A. W. Zewoudie, J. Luque, and J. Hernando, "The use of long-term features for gmm-and i-vector-based speaker diarization systems," *EURASIP Journal on Audio, Speech, and Music Processing*, vol. 2018, no. 1, pp. 1–11, 2018.

[3] Q. Wang, C. Downey, L. Wan, P. A. Mansfield, and I. L. Moreno, "Speaker diarization with lstm," in *2018 IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*. IEEE, 2018, pp. 5239–5243.

[4] D. Snyder, D. Garcia-Romero, G. Sell, A. McCree, D. Povey, and S. Khudanpur, "Speaker recognition for multi-speaker conversations using x-vectors," in *ICASSP 2019-2019 IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*. IEEE, 2019, pp. 5796–5800.

[5] Y. Fujita, N. Kanda, S. Horiguchi, Y. Xue, K. Nagamatsu, and S. Watanabe, "End-to-end neural speaker diarization with self-attention," in *2019 IEEE Autom. Speech Recognit. Underst. Workshop (ASRU)*. IEEE, 2019, pp. 296–303.

[6] S. Horiguchi, P. Garcia, Y. Fujita, S. Watanabe, and K. Nagamatsu, "End-to-end speaker diarization as post-processing," in *ICASSP 2021-2021 IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*. IEEE, 2021, pp. 7188–7192.

[7] M. H. Moattar and M. M. Homayounpour, "A review on speaker diarization systems and approaches," *Speech Communication*, vol. 54, no. 10, pp. 1065–1103, 2012.

[8] S. Kwon and S. Narayanan, "A study of generic models for unsupervised on-line speaker indexing," in *2003 IEEE Autom. Speech Recognit. Underst. Workshop (ASRU)*. IEEE, 2003, pp. 423–428.

[9] K. Markov and S. Nakamura, "Never-ending learning system for on-line speaker diarization," in *2007 IEEE Autom. Speech Recognit. Underst. Workshop (ASRU)*. IEEE, 2007, pp. 699–704.

[10] G. Soldi, M. Todisco, H. Delgado, C. Beaugeant, and N. W. Evans, "Semi-supervised on-line speaker diarization for meeting data with incremental maximum a-posteriori adaptation." in *Odyssey*, 2016, pp. 377–384.

[11] G. Soldi, C. Beaugeant, and N. Evans, "Adaptive and online speaker diarization for meeting data," in *Proc. 23rd Eur. Signal Process. Conf. (EUSIPCO)*. IEEE, 2015, pp. 2112–2116.

[12] F. K. Soong, A. E. Rosenberg, B.-H. Juang, and L. R. Rabiner, "Report: A vector quantization approach to speaker recognition," *AT T Technical Journal*, vol. 66, no. 2, pp. 14–26, 1987.

[13] E. W. Forgy, "Cluster analysis of multivariate data: efficiency versus interpretability of classifications," *biometrics*, vol. 21, pp. 768–769, 1965.

[14] Y. Linde, A. Buzo, and R. Gray, "An algorithm for vector quantizer design," *IEEE Trans Commun*, vol. 28, no. 1, pp. 84–95, 1980.

[15] T. Lehn-Schiøler, A. Hegde, D. Erdogmus, and J. C. Principe, "Vector quantization using information theoretic concepts," *Natural Computing*, vol. 4, no. 1, pp. 39–51, 2005.

[16] Z. Chen, T. Yoshioka, L. Lu, T. Zhou, Z. Meng, Y. Luo, J. Wu, X. Xiao, and J. Li, "Continuous speech separation: Dataset and analysis," in *ICASSP 2020-2020 IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*. IEEE, 2020, pp. 7284–7288.

[17] A. Bhattacharyya, "On a measure of divergence between two statistical populations defined by their probability distributions," *Bull. Calcutta Math. Soc.*, vol. 35, pp. 99–109, 1943.

[18] J. S. Chung, A. Nagrani, and A. Zisserman, "Voxceleb2: Deep speaker recognition," *arXiv preprint arXiv:1806.05622*, 2018.

[19] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, "Speaker verification using adapted gaussian mixture models," *Digital signal processing*, vol. 10, no. 1-3, pp. 19–41, 2000.

[20] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*. IEEE, 2015, pp. 5206–5210.