NOVEL ARCHITECTURES FOR UNSUPERVISED INFORMATION BOTTLENECK BASED SPEAKER DIARIZATION OF MEETINGS

Nauman Dawalatabad   Srikanth Madikeri

C Chandra Sekhar   Hema A Murthy

Novel Architectures for Unsupervised Information Bottleneck based Speaker Diarization of Meetings

Nauman Dawalatabad, *Student Member, IEEE,* Srikanth Madikeri, *Member, IEEE,*
C. Chandra Sekhar, *Member, IEEE,* and Hema A. Murthy, *Senior Member, IEEE,*

Abstract—**Speaker diarization is an important problem that is topical, and is especially useful as a preprocessor for conversational speech related applications. The objective of this paper is two-fold: (i) segment initialization by uniformly distributing speaker information across the initial segments, and (ii) incorporating speaker discriminative features within the unsupervised diarization framework. In the first part of the work, a varying length segment initialization technique for Information Bottleneck (IB) based speaker diarization system using phoneme rate as the side information is proposed. This initialization distributes speaker information uniformly across the segments and provides a better starting point for IB based clustering. In the second part of the work, we present a Two-Pass Information Bottleneck (TPIB) based speaker diarization system that incorporates speaker discriminative features during the process of diarization. The TPIB based speaker diarization system has shown improvement over the baseline IB based system. During the first pass of the TPIB system, a coarse segmentation is performed using IB based clustering. The alignments obtained are used to generate speaker discriminative features using a shallow feed-forward neural network and linear discriminant analysis. The discriminative features obtained are used in the second pass to obtain the final speaker boundaries. In the final part of the paper, variable segment initialization is combined with the TPIB framework. This leverages the advantages of better segment initialization and speaker discriminative features that results in an additional improvement in performance. An evaluation on standard meeting datasets shows that a significant absolute improvement of 3.9% and 4.7% is obtained on the NIST and AMI datasets, respectively.**

Index Terms—**Speaker diarization, information bottleneck, phoneme rate, varying length segment, speaker discriminative features, two-pass system.**

I. INTRODUCTION

GIVEN an audio signal, *speaker diarization* involves answering the question of *"Who spoke When?"* [1]. A speaker diarization system annotates audio with relative speaker labels. The task involves estimating the number of speakers and assigning speech segments to different speakers. Speaker diarization has been used in various domains, such as telephone conversations, broadcast news, and meetings [1]. Diarization of conversational audio meetings is considered to be a challenging task owing to the spontaneity in the conversation. Diarization systems are often used as frontends in applications that include automatic speech recognition, spoken keyword spotting, and speaker recognition [2].

N. Dawalatabad, C. C. Sekhar and H. A. Murthy are with the Indian Institute of Technology Madras, India. (e-mail: {nauman, chandra, hema}@cse.iitm.ac.in).
S. Madikeri is with the Idiap Research Institute, Martigny, Switzerland. (e-mail: srikanth.madikeri@idiap.ch).

Information Bottleneck (IB) based approach to speaker diarization has shown competitive performance for meeting recordings [3], [4], [5]. Owing to its non-parametric nature, IB based diarization has a very low Real Time Factor (RTF) value [3], [6]. RTF is the time taken by a system to process 1 second of speech data. Since diarization systems are mostly used in the pre-processing stage of many conversational speech applications, it is desirable to have diarization systems with low RTF value.

In general, an approach to unsupervised diarization of a conversational speech includes speech segment initialization followed by the bottom-up agglomerative clustering of the segments [1]. The major challenges in building an unsupervised speaker diarization system lie in the initialization of segments for clustering, to obtain speaker discriminative features, deciding on the number of speakers, and detection of overlapped speaker segments [1], [7]. A good segment initialization and speaker discriminative features help to improve the performance of the diarization system. In this paper, we address these two challenges: (i) better segment initialization for IB based system, and (ii) use of conversation-specific speaker discriminative features in the diarization process.

In the conventional IB based system, the initial speaker segments are obtained by dividing a conversation into short segments of equal duration. Each segment is modeled as a Gaussian and then clustered using IB criterion. It is essential to start with a good segmentation to obtain a good clustering solution. In the first part of the paper, we propose a varying length segment initialization technique for the IB based system. Initial segmentation ensures that the phoneme rate is fixed in each segment rather than the duration. The phoneme rate is used as the side information in segment initialization such that the number of phonemes in each segment is approximately constant. This ensures that the boundaries of the segments are at the phone boundaries. The proposed IB based system using the varying length segment initialization is referred to as VarIB system.

Speaker diarization is a clustering task that involves grouping the speaker segments into different clusters. The features used for the representation of speakers must be discriminative. In the second part of this work, we focus on obtaining speaker discriminative features to improve the output of the IB clustering. In [8], we proposed the Two-Pass Information Bottleneck (TPIB) based speaker diarization using Multi-layer Feed-forward Neural Network (MFNN). The TPIB system refines the speaker boundaries by performing the IB clustering process twice. The speaker discriminative information from

the output of the IB system is used to train MFNN based speaker classifier and obtain speaker discriminative features. These features are then used again in the IB system to refine the speaker boundaries.

By design, the RTF of the TPIB system is at least twice the IB system. Training MFNN is inherently an iterative process. To reduce the RTF of the TPIB system, one has to play with the hyperparameters of MFNN (e.g., the number of hidden layers, hidden nodes, etc.). In this work, we propose a computationally simpler alternative to use Linear Discriminant Analysis (LDA) [9]. LDA model can be trained significantly faster than MFNN, thereby reducing the overall runtime of the TPIB system. We also propose a new cluster stopping criterion for LDA based systems. From here onwards, we refer to TPIB systems with MFNN and with LDA as TPIB-NN and TPIB-LDA, respectively. Finally, we propose a TPIB system with the varying length segment initialization using both MFNN and LDA. This system is referred to as the VarTPIB system.

The aim of this work is to propose different architectures for information bottleneck based speaker diarization. The following are the contributions to the paper.

1) *Segment initialization*: A varying length segment initialization (VarIB) technique using phoneme rate for information bottleneck based speaker diarization is proposed.

2) *Speaker discrimination*: The TPIB-NN system was proposed in [8]. TPIB-LDA system is proposed in this paper as an alternative to TPIB-NN to reduce the RTF. A novel stopping criterion for the TPIB-LDA system is also proposed. We also conduct an experimental study on the combination of latent features obtained using LDA and MFNN.

3) *Joint segment initialization and speaker discrimination*: We propose VarTPIB approach, a combination of VarIB and TPIB approaches. It leverages both the benefits, i.e., good segment initialization and better speaker discrimination.

The first two approaches focus on two different ways to improve the speaker diarization system, whereas the third approach jointly leverages the advantages of the first two proposed approaches.

The rest of the paper is organized as follows. Section II presents a review of related approaches to diarization. Section III describes the baseline IB based speaker diarization system. Varying length segment initialization is proposed in Section IV. Section V describes the TPIB and VarTPIB based systems. Different variants of the TPIB and VarTPIB systems are described in this section. Results of experimental studies and analyses of results are presented in Section VI. Finally, Section VII concludes the paper.

II. Approaches to Speaker Diarization

In this section, we discuss different approaches to speaker diarization reported in the literature related to the proposed approaches in this paper.

The performance of speaker diarization systems is dependent on the initialization of the segments to be clustered. In [10], the focus is to have good initial cluster models, leading to a significant reduction in the diarization error rate with minimal human supervision. The approaches in [11], [12] use

lexical information along with acoustic cues to improve the speaker diarization performance. It is shown that the lexical information helps to add speaker discriminative information. This approach employs a sequence-to-sequence neural network trained on both lexical and acoustic features. Recently, we proposed a varying length segment initialization technique using the stroke onset rate information to diarize percussion instruments in Carnatic music concerts [13]. It is shown that segment length initialization based on stroke rate improves the performance of the percussion instrument diarization system.

The accuracy of a speaker diarization system is also dependent on the feature representations of speakers. A set of features that better discriminates one speaker from another speaker is preferred for diarization. Short term spectral features such as Mel Frequency Cepstral Coefficients (MFCC) are widely used for the task of speaker diarization [1]. The Mel Filterbank Slope (MFS) features and Linear Filterbank Slope (LFS) features have been shown to be better at speaker discrimination in comparison to MFCC features, owing to the emphasis of higher order formants [14]. Group delay based features have also shown to be better at speaker discrimination over traditional MFCC [15]. The i-vector based approaches have been used to build better speaker models [16], [17]. Linear Discriminant Analysis (LDA) has been used for speaker diarization of telephone data in [18]. Iterative probabilistic LDA adaptation for speaker diarization has been used for French TV recordings in [19].

Recently, data-derived features extracted using Deep Neural Networks (DNNs) are used. In [20], an Artificial Neural Network (ANN) is trained to determine whether a pair of utterances are from the same speaker or not. As the ANN is trained to be speaker discriminative, the features extracted from the ANN can be used to add discriminative feature information to the process of diarization. The approach in [21] used the triplet ranking method for diarization and linking. In [22], a DNN is used to get speaker embeddings. This approach is similar to [20], but the Long Short Term Memory (LSTM) network trained with a triplet loss is used to get the embeddings. In [23], a DNN is trained on large amounts of data to get speaker embeddings. In [24], an LSTM model is used to obtain speaker embeddings and perform spectral clustering on the same. In [25], a bidirectional LSTM is used to measure the similarity between speech segments. The approach in [26] uses SincNet [27] in a Vanilla Transfer Learning (VTL) setup to obtain speaker discriminative embeddings. End-to-end diarization techniques [28], [29] have also been proposed where a single neural network directly outputs the diarized speaker boundaries. DNN based embeddings such as x-vectors [30], d-vectors [31], and c-vectors [32] are speaker discriminative. In [33], a fully-supervised speaker diarization system using Unbounded-Interleaved State Recurrent Neural Network (UIS-RNN) is used in the clustering stage. This technique assumes that high quality time-stamped speaker labels are available for training the UIS-RNN. However, in all cases, the DNNs need to be trained on huge amounts of labeled data.

An incremental transfer learning based approach was proposed recently by us, which uses the "Remember-Learn-
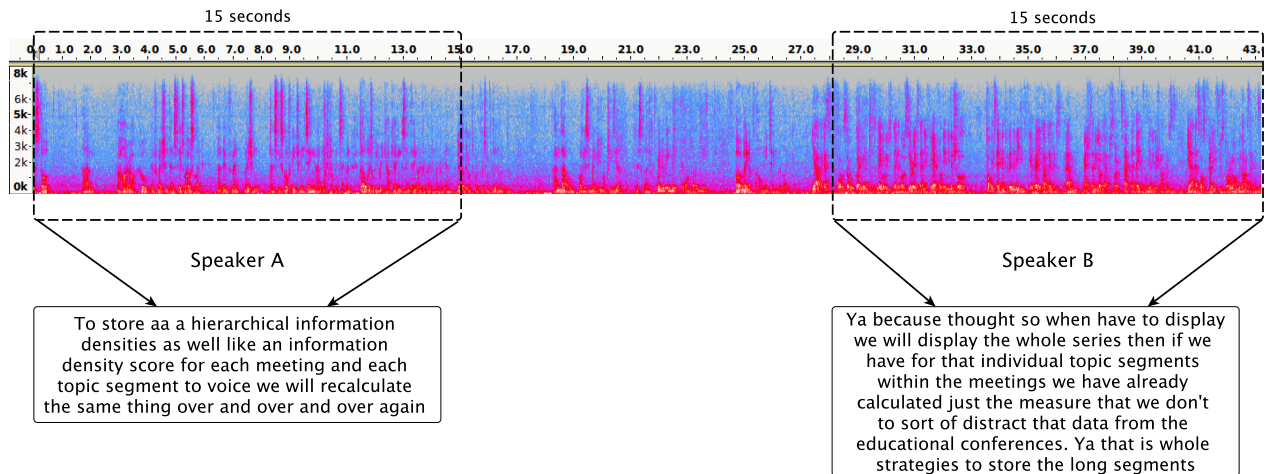
Fig. 1: Illustration of different speaking rates for different speakers. In a window of 15 seconds, speaker A spoke 35 words whereas speaker B spoke around 56 words.

Transfer" approach to continuously learn speaker discriminative information [34]. This reduces the RTF of TPIB (with MFNN) system [8] by 33% (relative) but with minor degradation in the speaker error rate. It uses transfer learning [35], [36] to retain the previously learned parameters, and adapts the parameters for the current conversation. In [37], attempts have been made to use *"Speaker2Vec"* embeddings, where the system is trained on unlabeled data. They have also suggested Neural Predictive Coding (NPC) [38] where a Convolutional Neural Network (CNN) is trained in an unsupervised manner on separate data. Despite several advancements in approaches to diarization, most of the recent approaches are based on deep embedding based techniques where embeddings are obtained in a supervised fashion, which requires a huge amount of speaker labeled data (thousands of hours [39]) for training.

III. INFORMATION BOTTLENECK BASED APPROACH

In this section, we briefly describe the agglomerative Information Bottleneck (aIB) based approach to speaker diarization [3].

The agglomerative information bottleneck based approach for speaker diarization is based on the information bottleneck principle [40]. An audio conversation is divided into fixed duration segments. Let $\mathbf{X}$ be a random variable representing a set of segments in the conversation. Let $\mathbf{Y}$ be a random variable denoting a set of relevance variables that specify relevant information about the speaker in each segment. Let $\mathbf{C}$ be a clustering solution to $\mathbf{X}$. Each segment is assumed to contain only one speaker and is modeled by a Gaussian distribution. In the Gaussian Mixture Model (GMM) based modeling, the speaker information is represented by the components of the GMM. Thus, the relevance variable $\mathbf{Y}$ represents the components of the GMM trained on the speech segments of an audio conversation. Bottom-up clustering is performed in the posterior space of the Gaussian components. The IB based approach clusters a set of segments $\mathbf{X}$ into a set of clusters $\mathbf{C}$

such that it captures as much relevant information as possible about $\mathbf{Y}$. Thus, the objective function $\mathcal{F}$ is given by

$$\mathcal{F} = \mathrm{I}(\mathbf{Y}, \mathbf{C}) - \frac{1}{\beta}\mathrm{I}(\mathbf{C}, \mathbf{X}) \qquad (1)$$

where $\mathrm{I}(.)$ denotes the mutual information and β is a Lagrange multiplier. The mutual information preserved $\mathrm{I}(\mathbf{Y}, \mathbf{C})$ is maximized while the mutual information $\mathrm{I}(\mathbf{C}, \mathbf{X})$ is minimized.

The Normalized Mutual Information (NMI) given by $\frac{\mathrm{I}(\mathbf{Y}, \mathbf{C})}{\mathrm{I}(\mathbf{X}, \mathbf{Y})}$ is used as the stopping criterion [3]. Once clustering terminates, the output is resegmented using Kullback-Leibler Hidden Markov Model (KL-HMM) [4]. The IB based diarization is used as the baseline benchmark for comparison of the work reported in this paper.

IV. VARYING LENGTH SEGMENT INITIALIZATION FOR IB (VARIB) BASED DIARIZATION

The first step in IB based diarization [3] is the segmentation of an audio conversation into short segments of equal size. Posteriors estimated from these segments are then clustered using IB based approach, as explained in Section III. The segment length does not change throughout the clustering process. Thus, the segments must be appropriately initialized in the IB based system. Uniform segment initialization is not always an optimal strategy. For instance, it can be seen from Fig. 1 that in 15 seconds speaker A spoke around 35 words, whereas speaker B spoke around 56 words in the same amount of time. The speaking rate can vary significantly across different speakers. The fixed length segment initialization may lead to a non-uniform (unequal) number of phonemes across segments. This may lead to the non-uniform presence of speaker information across initial segments. As a consequence, the parameters of the Gaussian model and the estimates of posteriors for clusters with fewer phonemes can be poor.

Motivated from [10], [11], and [13], we propose a varying length segment initialization technique to address this problem. [10] focuses on good initialization of clusters. In [11], it is shown that the lexical information adds speaker discriminative
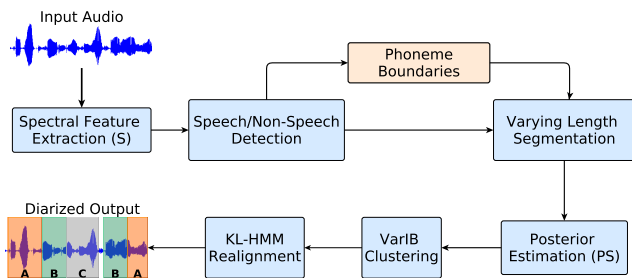
Fig. 2: Block diagram of VarIB based speaker diarization system.

Algorithm 1: Varying Length Segment Initialization

Input:

Phoneme boundaries, minLen, maxLen, and phnRate

Output:

X_{var}: *List of varying length segments*

Main Loop:

$ptr = 0$
while $ptr < len(audio)$ **do**
    $start = ptr$
    $end = ptr + minLen$
    if $end \geq len(audio)$ **then**
        $\{start, len(audio)\} \Rightarrow X_{var}$
        $break$ // *for the final segment*
    end
    $list_A = getPhnList(start, end)$
    if $len(list_A) \geq phnRate$ **then**
        $\{start, end\} \Rightarrow X_{var}$
        $ptr = end$
    else
        $list_B = getPhnList(end, start + maxLen)$
        if $(len(list_A) + len(list_B)) \leq phnRate$ **then**
            $end = list_B(len(list_B))$
            $\{start, end\} \Rightarrow X_{var}$
            $ptr = end$
        else
            $\Delta phnRate = phnRate - len(list_A)$
            $end = list_B(\Delta phnRate)$
            $\{start, end\} \Rightarrow X_{var}$
            $ptr = end$
        end
    end
end

---

information to the diarization process. In [41], group-delay processing is used to obtain the onset boundaries of strokes in a percussive instrumental music recording. The information about the instrument is concentrated in the strokes. In [13], the stroke rate is used as side information to initialize the segments for diarization such that the number of strokes (information about the instrument) across segments is approximately the same.

The speaker's information is present in the phonemes. Phoneme rate is used to perform varying length segment initialization such that the number of phonemes is approximately the same across segments. Phoneme boundaries are obtained using *PhnRec* phoneme recognizer [42]. A range of durations in the interval [*minLen*, *maxLen*] is permitted. The chosen duration approximately corresponds to a fixed number of phonemes, as defined by *phnRate*, a hyperparameter. Here, *minLen* and *maxLen* are the minimum and maximum lengths of segments allowed, respectively. The objective function $\mathcal{F}_{var}$ is given by

$$\mathcal{F}_{var} = I(\mathbf{Y}, \mathbf{C}) - \frac{1}{\beta} I(\mathbf{C}, \mathbf{X_{var}}) \qquad (2)$$

where $\mathbf{X_{var}}$ represents the varying length segments. The complete block diagram of the proposed VarIB system is shown in Fig. 2.

The length of each segment is bounded between *minLen* and *maxLen* such that it has the *phnRate* number of phonemes. The number of varying length segments is dependent on these three hyperparameters. These hyperparameters ensure that the number of segments is neither too small nor too large. Since the speaker's information is present in the set of phonemes he/she speaks, distributing an equal number of phonemes across the segments may lead to the same amount of information in each segment in a recording. Since one Gaussian is used to model a segment, the proposed initialization approach ensures that a similar amount of information is used in modeling a Gaussian component from every segment. Hence, the Gaussian models and the posteriors obtained using these models are expected to be better than those in the fixed length segment initialization in the IB based system.

The algorithm for varying length segment initialization is given in Algorithm 1. Here, *ptr* is a pointer that points to the current time instant in the audio. The function $getPhnList(t_1, t_2)$ returns the start and end times (boundaries) of phonemes between time instants t_1 and t_2. Given phoneme boundaries, *minLen*, *maxLen*, and *phnRate*; the algorithm returns the boundaries of varying length segments. It may be noted that only phoneme boundaries are used.

### A. Posterior Estimation using Varying Length Segments

A Gaussian component is modeled for each of the varying length segments. The components collectively form a GMM. The frame-level posteriors are obtained as given below.

$$P(y_i|f_k) = \frac{a_i \mathcal{N}(f_k, \mu_i, \Sigma_i)}{\sum_{j=1}^{N} a_j \mathcal{N}(f_k, \mu_j, \Sigma_j)} \qquad (3)$$

where f_k denotes a feature vector of a frame. Here, μ_i, Σ_i, and a_i are the mean vector, covariance matrix, and the mixture coefficient for the i-th component in the GMM, respectively. a_i is given by,
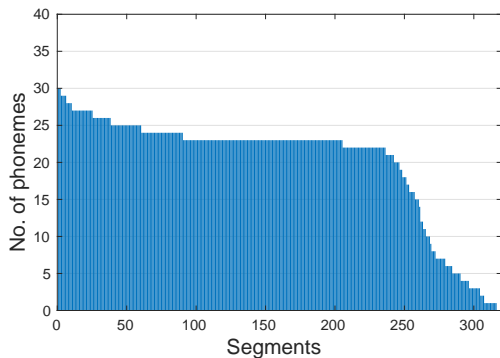
$$a_i = \frac{T_i}{\sum_{j=1}^{N} T_j} \qquad (4)$$

where T_i is the duration of the i-th segment in seconds, and N is the total number of varying length segments to be clustered. Segment level posteriors are obtained by averaging the frame-level posteriors for the frames in the segment. The segment
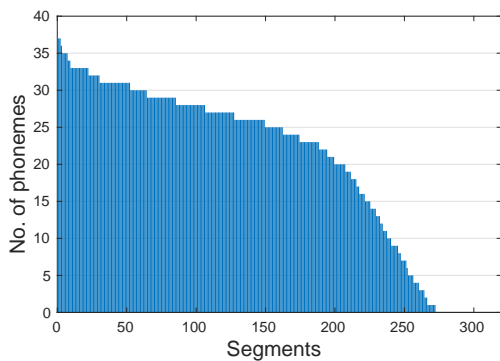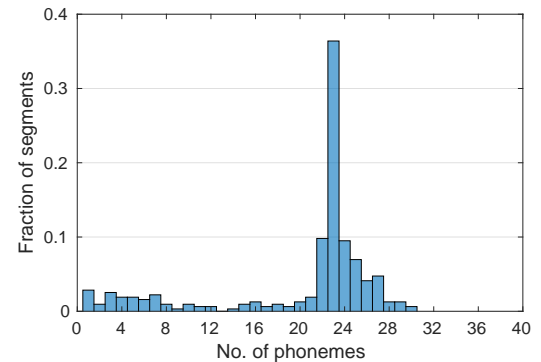
(a) No. of phonemes in fixed length segments
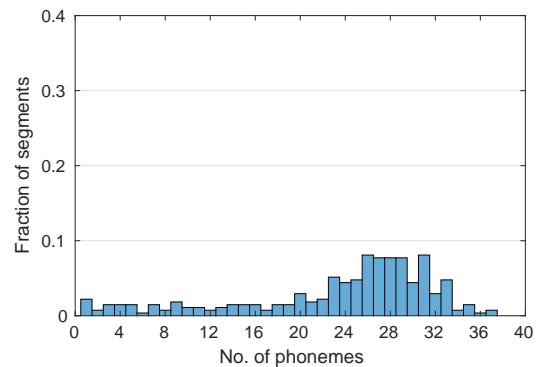
(b) Distribution for fixed length segments in IB

(c) No. of phonemes in varying length segments

(d) Distribution for varying length segments in VarIB

Fig. 3: Distribution of information across segments for fixed length segment initialization and varying length segment initialization.

level posteriors are then used in IB based clustering, followed by one step of KL-HMM based realignment to adjust the speaker boundaries.

B. Information Distribution Across Segments formed using Fixed and Varying Duration Segment Initialization

Figures 3(a) and 3(c) show the number of phonemes in different segments for fixed length (2.5 sec) segment initialization and for varying length segment initialization ($phnRate$=23) on a sample meeting recording taken from the development dataset, respectively. Segments are sorted in descending order of the number of phonemes in a segment. The difference in the number of segments for the two cases is due to the respective segment initialization method. It can be seen that the number of phonemes is distributed unevenly in the fixed duration segment initialization. In contrast, in the varying length segment initialization, most of the segments have approximately the same number of phonemes.

Figures 3(b) and 3(d) correspond to the distributions of segments based on number of phonemes for Figures 3(a) and 3(c), respectively. Long tails on the left and right correspond to low speaking rates and high speaking rates, respectively. It can be seen that the distribution in Fig. 3(d) is peaky (less variance) compared to the distribution in Fig. 3(b). The distribution in Fig. 3(d) peaks at 23 number of phonemes, indicating that most segments comprise of 23 (here, $phnRate$) phonemes. The region to the immediate left of the peak represents segments

from slow ($< phnRate$) speaking parts, whereas the region right to the peak shows short segments in fast ($> phnRate$) speaking regions. The segments in the faster speaking regions are set to $minLen$ duration, whereas other segments are of lengths between $minLen$ and $maxLen$.

It should be noted that both the distributions include short speech segments of duration less than $minLen$ in the proposed system and segments less than the fixed duration in the baseline IB system. These segments are the residual segments that result when big chunks of audio are divided into short uniform or varying length segments, as can be seen from Figures 3(b) and 3(d), respectively. The distribution of residual segments is observed in the extreme left tail of the distributions. However, the number of such short residual segments is small and is less than (or equal to) the number of speech regions in a conversation audio.

V. TWO-PASS IB (TPIB) AND VARYING LENGTH TWO-PASS IB (VARTPIB) BASED FRAMEWORK

In this section, we propose TPIB and VarTPIB approaches to include speaker discriminative features within the IB clustering framework. This section first describes the general Two-Pass IB (TPIB) and Varying Length Two-Pass IB (VarTPIB) based frameworks for speaker diarization, as shown in Fig. 4. We then describe different discriminative models (MFNN and LDA) used in this framework.
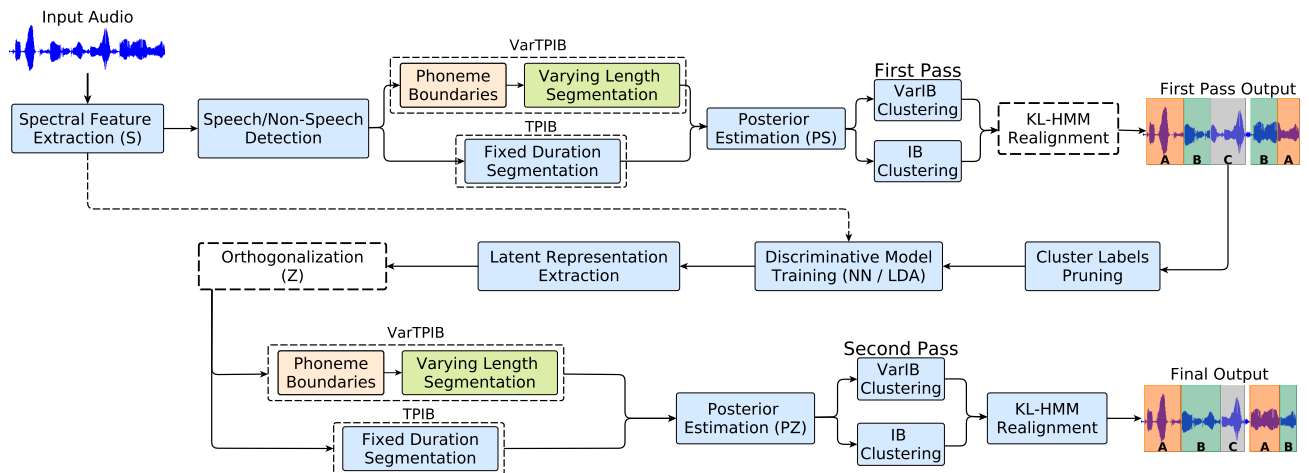
Fig. 4: Two-Pass IB (TPIB) and Varying Length Two-Pass IB (VarTPIB) based frameworks for speaker diarization. The KL-HMM based realignment after the first pass and the orthogonalization are performed only when a neural network model is used as the speaker discriminative model. The VarTPIB system uses phoneme rate as side information for segment initialization, while the TPIB framework uses a fixed duration segment initialization.

The TPIB-LDA approach presented in this section is an extension to our previous work [8] where the TPIB-NN (as discussed later in this section) system was first introduced. The VarTPIB approaches proposed in this section is the combination of VarIB and TPIB approaches. In general, the TPIB and VarTPIB framework consists of the following stages.

- *Spectral Feature Extraction:* This stage takes a conversational audio as input and extracts acoustic features. Speech/non-speech segments can be identified using a Speech Activity Detection (SAD) [43].
- *Segment Initialization:* Input audio is divided into short segments of uniform duration in the TPIB framework. The segment length is usually fixed to 2.5 seconds [3]. VarTPIB approach initializes segments based on phoneme rate information as given in Algorithm 1.
- *Posterior Estimation:* Each of the N segments in the TPIB approach is initially modeled by a single Gaussian. These N Gaussians are used to build a GMM. The posterior for each frame of speech f_k for the i-th GMM component is computed using Equation (3) with mixture coefficients set to the value of $1/N$. The posteriors are then averaged over the frames in the segment to obtain segment level posteriors. Finally, each segment is represented by an N-dimensional posterior vector (P_s). The frame level and the segment level posterior estimation for VarTPIB is the same as explained in Section IV-A.
- *First Pass:* The segment level posteriors obtained are then clustered in a bottom-up fashion using the IB (or VarIB) approach as described in Section III (or IV). The clustering is stopped when NMI reaches a threshold value. Once clustering is completed, the speaker boundaries are refined using the KL-HMM [4] based approach.
- *Discriminative Model Training:* This stage takes the spectral features and the cluster labels for a feature vector from the first pass as input. Assuming that the number of speakers obtained during the first pass is correct, a classifier is trained

to discriminate between the speakers obtained in the first pass. Clusters smaller than a threshold duration (< 3 sec) obtained in the first pass are pruned out and not considered in training the classifier. The discriminative model is chosen such that the time taken to train the model is small. Different discriminative models used in this work are discussed in Section V-A.
- *Latent Representation Extraction:* The discriminative model trained in the previous step is used to project the spectral features onto a latent space where the features are discriminative. The projection onto the latent space varies based on the discriminative model used.
- *Orthogonalization:* The discriminative features are then subjected to Principal Component Analysis (PCA). At the end of this process, the feature directions are orthogonal, leading to a diagonal covariance matrix, a requirement for IB based clustering[1]. The features obtained after orthogonalization are referred to as the Latent Features (LF). The latent features form a stream like the spectral feature stream.
- *Posterior Estimation for Latent Feature Stream:* In the TPIB framework, the LF stream for conversation audio is first uniformly segmented. The posteriors are then obtained as given in Equation (3), where a_i is set to $1/N$ for TPIB system. The frame-level posteriors are then used to obtain segment level posteriors (P_z). Segment level posteriors (P_z) for VarTPIB framework is calculated from the LF stream as mentioned in Section IV-A.
- *Second Pass:* The IB (or VarIB) based clustering is performed on P_z in the second pass. Finally, one step of KL-HMM [4] based realignment is performed to refine the speaker boundaries that forms the final output of the TPIB/VarTPIB framework.

---

[1]PCA is used only for orthogonalization, and not for dimension reduction.

A. Speaker Discriminative Models in the TPIB and VarTPIB Frameworks

In this section, we describe two methods used in the *Discriminative Model Training* stage of TPIB/VarTPIB framework.

1) Mutli-Layer Feed-Forward Neural Network (MFNN): After obtaining the speaker boundaries from the first pass, an MFNN is trained to classify the speakers in a conversation. The inputs to the MFNN are the spectral features of the frames. The labels obtained from the first pass are used as the desired output. To keep the RTF of the entire system low, a shallow MFNN with two hidden layers is used. Sigmoidal activation functions are used in the hidden nodes in the first hidden layer and linear nodes are used in the second hidden layer. The MFNN is first trained using gradient descent to classify speakers with cross-entropy loss function. Once the MFNN is trained, the spectral features of a frame of speech are given as input to the MFNN to obtain corresponding latent representation from the second hidden layer. The latent features obtained after orthogonalization are denoted as LF_{NN}. The LF_{NN} obtained from the trained MFNN are used as a feature stream during the second pass of IB clustering. The remaining steps are identical to those presented in Section V. The two pass IB/VarIB system with discriminative features obtained using an MFNN is referred to as TPIB-NN/VarTPIB-NN. Although care has been taken in TPIB-NN/VarTPIB-NN to keep the number of model parameters small, the run-time of TPIB-NN/VarTPIB-NN system is comparatively high because of the back-propagation algorithm used to train the MFNN. However, the RTF is still well below 1.0 (as mentioned later in Table V).

2) Linear Discriminant Analysis (LDA): In this section, we propose to reduce the RTF of TPIB-NN and VarTPIB-NN systems by using a non-iterative discriminative method. The LDA can be performed in significantly less amount of time than the MFNN based method. The TPIB/VarTPIB system using LDA as speaker discriminator is denoted as TPIB-LDA/VarTPIB-LDA. The spectral feature extraction and the posterior estimation stages are identical to those discussed in Section V. The following stages are different from that of TPIB-NN/VarTPIB-NN system (i.e., Section V-A1).

In TPIB-LDA/VarTPIB-LDA, the IB clustering process in the first pass is stopped when the total number of clusters reaches a fixed number. This is because the upper limit on the reduced dimension in the LDA is one less than the number of classes (speaker clusters). When the number of speakers is small, the LDA based dimension reduction can lead to a significant loss of information. In the proposed system, LDA is used to obtain discriminative latent features and not for dimension reduction. The number of clusters is chosen empirically based on the performance on the development data. This is not required in case of TPIB-NN/VarTPIB-NN system as the number of nodes in the second hidden layer can be kept as high as the input spectral dimension itself. Details of the stopping criterion for LDA based systems is discussed in Section V-B. The realignment can result in a significant reduction in the number of speaker clusters as
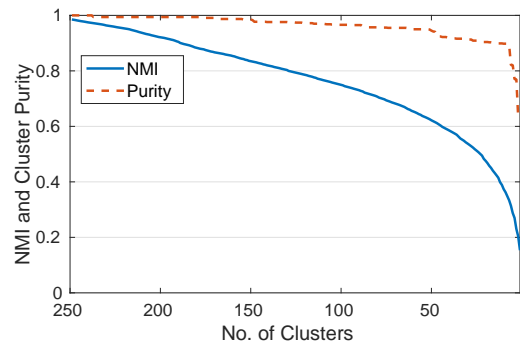

Fig. 5: NMI and cluster purity at different iterations (number of clusters) in IB based bottom-up clustering for a recording.

smaller clusters may get assigned to a bigger cluster. Thus, unlike TPIB-NN/VarTPIB-NN system, KL-HMM realignment is not performed after the first pass of IB clustering in TPIB-LDA/VarTPIB-LDA. This is indicated by a dotted block in Fig. 4.

The cluster labels from the first pass along with the spectral feature stream are used, and LDA is applied. We then extract the latent representation by projecting spectral features onto the LDA space. The obtained latent representation is referred to as LF_{LDA}. The input spectral features are already independent, and the LDA performs a linear transformation which retains the independence among the features even in the latent space. Hence, latent representations are not subjected to orthogonalization. The second pass of IB based clustering is performed on LF_{LDA} with NMI as the stopping criterion.

B. Stopping Criterion for TPIB-LDA and VarTPIB-LDA Systems

A proper stopping criterion in diarization is crucial as it affects the final diarization error rate (DER) [7]. Stopping the clustering process at the number of clusters less than the actual number of speakers is termed as over-clustering, and the opposite is referred to as under-clustering [1]. Under-clustering is preferred to over-clustering, as over-clustering can lead to merging of many speakers into a single cluster [7], [44]. Fig. 5 shows the NMI and cluster purity[2] at different iterations in IB based bottom-up clustering. It can be seen that as the number of clusters decreases, the NMI and cluster purity keeps on decreasing. The clusters become impure (multiple speakers are in one cluster) as the bottom-up clustering proceeds. This leads to an increase in the DER.

The number of speakers in a recording can be as small as two. Unlike MFNN, there is less flexibility in LDA with respect to the choice of the number of dimensions, as the number of dimensions is decided by the number of classes. The LDA reduces the dimension maximum to the *number of classes - 1*. For example, the LDA could reduce the MFCC feature dimension from 19 to 1 when there are only two speakers in a conversation. This dimension reduction is highly lossy. Hence, to avoid the loss of speaker information, we

---

[2]Cluster purity is calculated using open source tool *pyannote metric* [45].
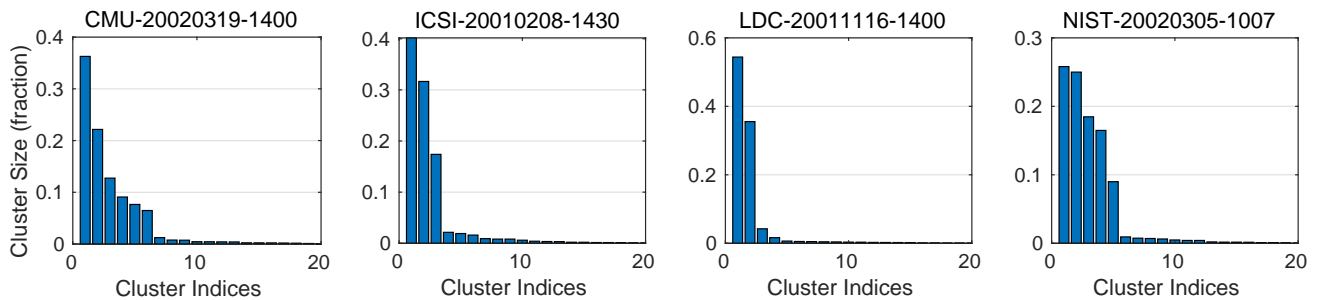
Fig. 6: Size of each cluster given as a fraction of the length of recording for 4 different recordings in development dataset for TPIB-LDA system. The clusters are sorted in the descending order of their sizes.

propose to stop the first pass of the TPIB-LDA and VarTPIB-LDA systems at a larger number of clusters (here, 20). This under-clustering in the first pass of IB clustering eliminates the problem of loss of speaker information.

Since LDA performs linear discrimination among the speakers in the output of the first pass, it can separate similar speakers when the first pass is under-clustered. Fig. 6 shows the sizes of different speaker clusters sorted in descending order for 4 recordings taken from the development set. The plots show the cluster sizes when the number of clusters is 20. It can be seen that all the recordings have two or three dominating speakers. The top four or five clusters contribute to more than 95% of the speaker segments while the rest of the clusters have negligible size. The small size clusters have small prior and hence do not contribute much when LDA is performed.

C. Conversation-Specific Speaker Discriminative Features

Most of the approaches to find speaker discriminative features for speaker diarization task directly use techniques used in the speaker recognition domain. The amount of training data required to define a global speaker discriminative space robustly can be quite large. In the task of speaker recognition, it is necessary to find a global set of speaker discriminating features for a large number of speakers. Finding a universal set of speaker discriminative features is a difficult problem. Further, the notion of the universal number of speakers is not well defined which makes the task more complicated. On the other hand, in a speaker diarization task, the system is only required to discriminate between speakers in the given conversation recording. The number of speakers in a conversational recording is very small (typically of the order of 2-10). Therefore, only the relative discriminating characteristics need to be found. Hence, finding speaker discriminative features for speaker diarization task is relatively simple.

In the TPIB/VarTPIB architectures, the discriminator model is trained to discriminate the speakers in the current conversation recording. Hence, the latent features obtained from the model are expected to be speaker discriminative specific to the current recording. It is important to note that the proposed architectures do not use any external data (labeled or unlabeled) to train the models to obtain speaker discriminative features. Since discriminative models are trained on a recording that is to be diarized, the TPIB/VarTPIB systems do not

TABLE I: AMI meeting datasets.

AMI-1	ES2008c, ES2013a, ES2013c, ES2014d, ES2015a, IS1001c, IS1007a, IS1008c, IS1008d, IS1009c
AMI-2	ES2010b, ES2013b, ES2014c, ES2015b, ES2015c, IS1004b, IS1006c, IS1007c, IS1008a, IS1009d

suffer from an unseen class (speakers) problem. Conversation-specific speaker discriminative features are obtained during the diarization process itself.

VI. EXPERIMENTAL STUDIES

In this section, we demonstrate the effectiveness of the proposed architectures on standard meeting conversation datasets.

A. Datasets, Features and Evaluation Measure

Datasets: Experiments are performed on the following standard NIST-RT datasets from Linguistic Data Consortium [46]: RT-04Dev, RT-04Eval, and RT-05Eval. The subsets of Augmented Multi-Party Interaction (AMI) corpus [47] are also used in our experiments. All datasets contain recordings of conversational audio meetings. The list of meeting IDs from AMI datasets recorded at Idiap (IS) and Edinburgh (ES) is given in Table I. The datasets are also used in [8] and [34] for evaluation. The number of speakers in the NIST dataset varies from 3 to 10. There are four speakers in each of the AMI recordings. RT-04Dev is used as the development dataset to tune hyperparameters of all models, and the remaining datasets are used for testing. Multiple distant microphone (MDM) data is used for each meeting after beamforming using *BeamformIt* [48].

Features: Since the focus of this work is on the different architectures and not on the feature side, we demonstrate the effectiveness of the proposed approaches on most widely used standard Mel Frequency Cepstral Coefficients (MFCC) features. MFCC of 19 dimensions extracted from 26 filterbanks[3] are used as spectral features [3] [4] [14]. These MFCCs are obtained with 25ms window length and 10ms shift.

---

[3]Overall SERs across all datasets on baseline IB system is better than that reported in [8] as we use 26 filterbanks in this paper as opposed to 40 filterbanks in [8].
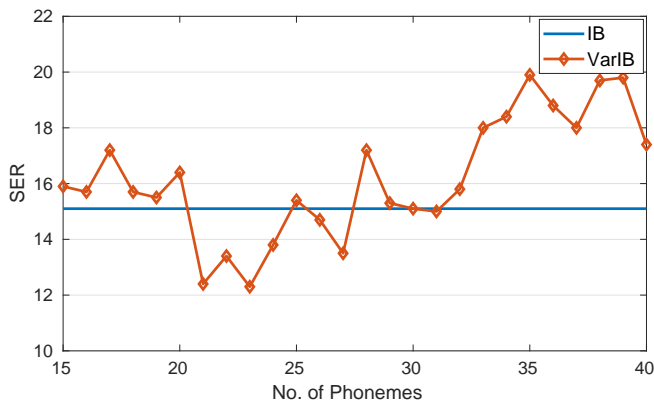
Fig. 7: SERs on development data as a function of window length in terms of number of phonemes (i.e. *phnRate*) in VarIB system. A reference line (in blue) denotes SER of IB based diarization system.

Evaluation Measure: The Diarization Error Rate (DER) is the sum of Missed Speech (MS), False Alarm Speech (FS), and Speaker Error Rate (SER). MS and FS depend on errors in speech activity detection, whereas SER arises due to speaker mismatch. As the primary focus of this paper is on improving the clustering module, the speech/non-speech hypotheses are obtained from the ground truth. Hence, similar to [8], [14], we use SER as the evaluation metric. We used the standard diarization evaluation tool from NIST [49] to calculate SER with a forgiveness collar of 25ms. Similar to [3], [8], [14], we also included the overlapped speech in evaluation in all our experiments.

B. Experimental Setup

We use the open source IB toolkit[4] [50] in all our experiments. The values for NMI and β for all the systems for both the passes are set to 0.4 and 10, respectively.

VarIB system: We use *PhnRec*[5] to detect the phoneme boundaries. Note that *PhnRec* is used only to detect boundaries as the actual identities of the phonemes are not needed. Fig. 7 shows SERs on development dataset for different number of phonemes (*phnRate*) in VarIB system. The blue line is a reference line denoting SER on the development dataset (15.1%) for the baseline IB based diarization system. The *phnRate* is varied between 15-40 phonemes, which corresponds to 4-12 syllables approximately. A reduction in SER is observed between 21-24 phonemes. The lowest SER is observed at 23 phonemes. The values of *minLen* and *maxLen* are set to 2 seconds and 5 seconds, respectively. These hyperparameters are kept the same across VarIB, VarTPIB-NN, and VarTPIB-LDA systems.

TPIB-NN and VarTPIB-NN systems: A shallow two-layer MFNN is used for both TPIB-NN and VarTPIB-NN systems. Based on the best SER obtained on the development dataset, 34 hidden nodes were kept in the first layer while the second

---

[4]Code is available at: https://github.com/idiap/IBDiarization
[5]Code is available at: https://speech.fit.vutbr.cz/software/phoneme-recognizer-based-long-temporal-context
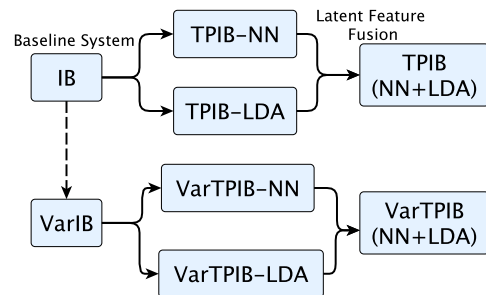

Fig. 8: Flow diagram for experiment protocol. IB system is used as the baseline. TPIB/VarTPIB systems have NN and LDA variants. Fusion of latent features from NN and LDA is also studied.

layer was set to 19 hidden nodes. The *tanh* activation function is used in the first hidden layer, whereas the second layer is kept linear. We use the PyTorch[6] [51] implementation of stochastic gradient descent (SGD) with cross-entropy loss to train the MFNN models. All the weights of MFNN are initialized using the initialization technique described in [52].

TPIB-LDA and VarTPIB-LDA systems: Only the first pass of TPIB-LDA and VarTPIB-LDA systems use a fixed number of clusters as a stopping criterion. For both these systems, the first pass IB clustering is stopped when the number of clusters reaches 20. The second pass uses NMI as the stopping criterion.

Real Time Factor: All RTFs are calculated on 2.6 GHz CPU with two threads without using GPUs. The reported RTFs are calculated by averaging the RTFs across ten independent runs.

Experiment protocol: The complete flow of experiment design is shown in Fig. 8. The IB system is used as a baseline. The performance of TPIB-NN and TPIB-LDA systems are compared with IB system individually. We have also performed an experiment with the fusion of latent features from NN and LDA models. This denoted as TPIB (NN+LDA) in Fig. 8. The performance of the VarIB system is compared with the IB system. Similar to the IB setup, experiments are performed on VarTPIB-NN, VarTPIB-LDA, and VarTPIB (NN+LDA) systems.

All the results reported in this paper are based on the best performing hyperparameters tuned on the development dataset.

C. Results, Analysis, and Discussion

1) Speaker Error Rate:
The results of all the experiments in terms of SER are given in Table II. The performance in terms of SER for all the proposed systems is compared with the baseline IB system. All the systems use MFCC as an input in the first pass. Speaker discriminative models and Latent Features (LF) used before the second pass are also mentioned in the table.

It can be seen in Table II that the VarIB system performs better than the fixed length initialization in the baseline IB

---

[6]https://pytorch.org/

TABLE II: Speaker Error Rates (SERs) for different systems. The best case performance for each dataset is given in bold font. The SER of all the proposed architectures is compared with the baseline IB system.

| System	Segment Initialization	Discriminative Model(s)	Features	Dev	Test Set			
				RT-04Dev	RT-04Eval	RT-05Eval	AMI-1	AMI-2
IB	Fixed	-	MFCC	15.1	13.5	16.4	17.9	23.5
Proposed Systems								
VarIB	Varying	-	MFCC	12.3	12	15.3	17.8	22.6
TPIB	Fixed	MFNN	LF_{NN}	14.2	12.6	14.2	16.1	23.6
		LDA	LF_{LDA}	14.7	11.6	13.2	15.7	24.5
		MFNN+LDA	$LF_{NN} + LF_{LDA}$ (0.2,0.8)	13.1	12.6	12.6	15.4	21.9
		MFNN+LDA	$LF_{NN} + LF_{LDA}$ (Avg.)	14.2	12.4	14.5	16.3	22.2
VarTPIB	Varying	MFNN	LF_{NN}	12	**9.9**	14.2	17.5	**20.9**
		LDA	LF_{LDA}	13.8	12.8	**12.5**	14.8	21.3
		MFNN+LDA	$LF_{NN} + LF_{LDA}$ (0.6,0.4)	**11.6**	11.7	15.1	**13.2**	21.1
		MFNN+LDA	$LF_{NN} + LF_{LDA}$ (Avg.)	12.6	11.9	13.9	15.1	21.1
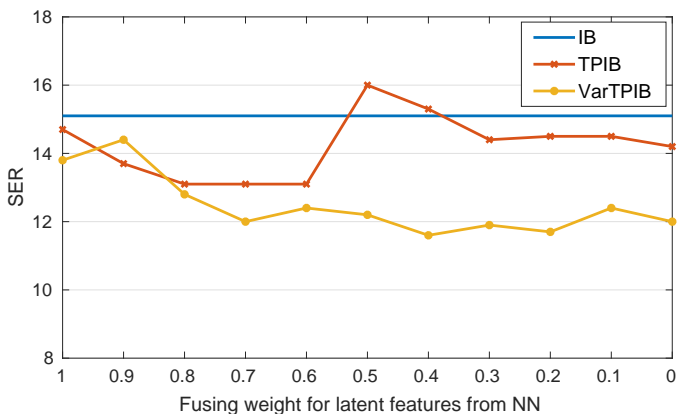
Fig. 9: SER for different feature fusing weights for LF_{NN} and LF_{LDA}. X-axis denotes weight for LF_{NN} (i.e. w_N) and the remaining $w_L = 1 - w_N$ is assigned to LF_{LDA}. The SERs are shown on the development dataset for different systems. The blue color line denotes SER of the IB based system.

system for all datasets. The VarIB system shows a 2.8% absolute improvement on the development set. Best-case absolute improvement of 1.5% and 1.1% is observed on RT-04Eval and RT-05Eval datasets, respectively. Consistent improvements are observed across all datasets. This suggests that equal distribution of speaker information in terms of phonemes across different segments leads to a better clustering solution. The other set of systems, including TPIB-NN and TPIB-LDA, consists of those that use the fixed length segment initialization. Absolute improvements of 2.2% and 3.2% are observed on RT-05Eval dataset for TPIB-NN and TPIB-LDA systems, respectively. It can be seen that both systems outperform the baseline IB system on most datasets.

We also conduct an experiment where both MFNN and LDA are used after the first pass as speaker discriminative models. Both the models were trained independently using the first pass output labels and the spectral features. Since performing LDA is fast, the overall RTF of the TPIB-NN system does not change significantly. The posteriors obtained from MFNN and LDA using LF_{NN} and LF_{LDA} were fused before the second pass as follows [5],

$$P_t(\mathbf{y}) = P(\mathbf{y}|\mathbf{F_t}).w_N + P(\mathbf{y}|\mathbf{L_t}).w_L \qquad (5)$$

where $\mathbf{F_t}$ and $\mathbf{L_t}$ are feature vectors at time $\mathbf{t}$ from LF_{NN} and LF_{LDA}, respectively. $\mathbf{y}$ is a vector representation for y_i ($i = 1 \ldots N$) and y_i refers to the i-th Gaussian component of the respective feature streams. Here, w_N and w_L are the weights assigned to LF_{NN} and LF_{LDA}, respectively, such that $w_N + w_L = 1$. The final posterior representation $P_t(\mathbf{y})$ is then used for the second pass IB clustering. The combination of features shows improvement over individual latent feature streams on most datasets. It shows absolute improvements of 3.8% and 2.5% compared to the baseline IB system on RT-05Eval and AMI-1 datasets, respectively. This indicates the presence of complementary information in the latent features derived from different discriminative models.

In order to test the robustness of the proposed system against different feature weight combinations, we also report the average SER across all the feature fusing weights. It can be seen that on average (under different feature weight combinations), the proposed systems perform significantly better than the baseline IB system on all datasets. The best-case absolute improvement observed is 1.9% on RT-05Eval dataset. Notice that the SERs are averaged over different fusing weights and hence the reported improvement is significant.

The final set of proposed systems are VarTPIB-NN and VarTPIB-LDA. The VarTPIB-NN showed an improvement of 3.6% and 2.6% on RT-04Eval and AMI-2 datasets, respectively. The VarTPIB-LDA showed an improvement of 3.9% on RT-05Eval dataset. The combination of latent features from MFNN and LDA results in a significant improvement of 4.7% on AMI-1 dataset. The latent feature combination in the case of VarTPIB systems did not always show an improvement over both the individual systems. However, the observed SERs are always better than the baseline IB, TPIB, and VarIB systems on most datasets. Similar to the systems using fixed length segment initialization (TPIB), the averaged SER over all the fusing weights for the VarTPIB system is better than the baseline IB on all datasets. It shows a significant absolute improvement of 2.5% and 2.8% over the baseline IB system on RT-05Eval and AMI-1 datasets, respectively. As shown in Table II, the SERs of VarTPIB systems are better than that

TABLE III: Comparison of proposed system with the state-of-the-art AHC on x-vectors with VB realignment. SERs on best performing proposed system, i.e., VarTPIB (0.6, 0.4) is mentioned. Lowest SERs achieved by the proposed approaches is also mentioned in the last column.

Dataset	Baseline IB system	State-of-the-art x-vector system	VarTPIB (0.6, 0.4)	Best SER on proposed systems
RT-04Dev (Dev. set)	15.1	10.4	11.6	11.6
Test Set				
RT-04Eval	13.5	10.9	11.7	9.9
RT-05Eval	16.4	10.4	15.1	12.5
AMI-1	17.9	9.7	13.2	13.2
AMI-2	23.5	10.5	21.1	20.9

of VarIB and TPIB systems, showing clearly that the varying length segment initialization in tandem with a two-pass setup improves the performance significantly.

Fig. 9 shows SERs for different values of w_N and w_L weights for fixed length (TPIB's LF_{NN} and LF_{LDA}) and varying length (VarTPIB's LF_{NN} and LF_{LDA}) systems on development dataset. It can be seen that both the systems perform better than the baseline for most of the values of fusing weights. This confirms the robustness of all the proposed systems to feature fusing weight combinations. It can be seen that the VarTPIB system performs consistently best among all the three systems.

Comparison of proposed systems with the state-of-the-art x-vector based system:

In this section, we compare the performance of the proposed systems with the state-of-the-art Agglomerative Hierarchical Clustering (AHC) [53] on x-vector (deep speaker embedding) system [30]. This system clusters the x-vector embeddings in an agglomerative bottom-up approach. Note that the x-vector model is trained on huge amounts of labeled data in a supervised manner, whereas the proposed systems (and the baseline IB system) do not use any speaker labeled data.

We follow the open-source Kaldi recipe for building the x-vector system [30]. It uses around 5k hours of data (including clean speech and augmentation data) for training the time delay neural network (TDNN) [54] in a supervised fashion. The hyperparameters for x-vector extraction, i.e., the window length and the period, are kept at 3.5 seconds and 1.75 seconds, respectively. Probabilistic Linear Discriminant Analysis (PLDA) [55] [56] models are whitened using corresponding NIST and AMI datasets. The PLDA score threshold is also calibrated on the development dataset. The speaker boundaries obtained after AHC on x-vector is subjected to Variational Bayes (VB) realignment [57]. Similar to [30], the parameters for VB are learned using the VoxCeleb dataset. All the hyper-parameters are tuned to give the best results on the development dataset.

As expected, the x-vectors based diarization system performs significantly better than the unsupervised baseline IB system on all datasets. It can be seen from the Table III that the best system, i.e., the VarTPIB system with fusing weights of (0.6, 0,4) show comparable performance to the x-vector system
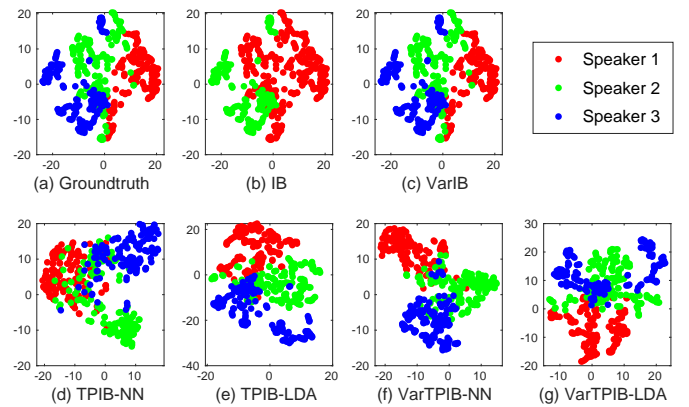

Fig. 10: t-SNE plots for segments in a recording from development set. Different colors represent different speakers.

on some datasets (RT-04Dev and RT04Eval). The lowest SER performance (last column) of the proposed systems shows how close the proposed systems can reach near to the x-vector system's performance. It can be seen that the best SERs are comparable to that of the state-of-the-art x-vector system on most datasets (except for AMI-2 dataset).

Detailed analysis on the performance related to the IB approach is given in [3]. The GMM in an IB approach is estimated using data from the same meeting. The variations in the background noise in a recording and the overlapping speech affect the IB clustering process [3]. Since the proposed systems are based on the IB approach, they show similar behavior. The AMI-2 dataset contains noisy recordings that show high SER on the IB approach. For example, IS1006c and ES2013b meetings show a high SER of 40% and 38% for the IB system, respectively. This is mainly attributed to the lack of robustness to the noisy conditions for IB approach. On the other hand, the TDNN model in the x-vector system is trained on 1000s of hours of speaker labeled data that is augmented with noise datasets (MUSAN and RIRs)[7] hence it is robust to different types of noise. Note that the proposed systems do not use any speaker labeled data to obtain speaker embeddings, unlike the x-vector based system.

2) On Speaker Discrimination:
To observe the speaker separation and the correctness in the speaker's relative labels, we plot the t-Distributed Stochastic Neighbor Embedding (t-SNE) projections [58] of 3 speaker segments along with their speaker identities (relative) taken from all systems. For the t-SNE study, the actual speaker identities corresponding to the ground-truth speaker labels were obtained using the standard NIST evaluation tool. This was cross-verified by listening to the segments for the correctness. Fig. 10 shows the t-SNE projections on the 2d plane for short speaker segments in different feature spaces for all systems. Fig. 10(a) shows the ground truth in MFCC space where three different speakers are present. Fig. 10(b) also shows the MFCC space, but IB system has detected only two speakers. The VarIB correctly identifies three speakers

[7]https://openslr.org

TABLE IV: Module-wise RTFs (x10) for different systems on development dataset.

System	Phoneme Boundaries	Posterior Computation 1	IB Clustering 1	KL-HMM Realignment 1	Discriminative Model Training	Posterior Computation 2	IB Clustering 2	KL-HMM Realignment 2
IB	-	0.19	0.46	0.09	-	-	-	-
VarIB	0.2	0.17	0.36	0.09	-	-	-	-
TPIB-NN	-	0.17	0.50	0.09	0.97	0.16	0.46	0.09
TPIB-LDA	-	0.18	0.48	**0**	**0.01**	0.18	0.50	0.07

TABLE V: Overall RTF (x10) for different systems on development dataset.

System	RTF (x10)
IB	0.74
x-vector	2.13
VarIB	0.82
TPIB-NN	2.44
TPIB-LDA	1.42
VarTPIB-NN	2.58
VarTPIB-LDA	1.61

in the MFCC space itself, as it can be seen from Fig. 10(c). The t-SNE plot for TPIB-NN system (Fig. 10(d)) shows that the three speakers are correctly detected, but the speaker spaces do overlap. The VarTPIB-NN system shows a better separation than TPIB-NN as seen in Fig. 10(f). Similar trends were also observed for LF_{LDA} space in the case of TPIB-LDA (Fig. 10(e)) and VarTPIB-LDA (Fig. 10(g)). Overall, the proposed discriminative frameworks emphasize on the differences between speakers in a given conversation.

3) Real Time Factor:
The module-wise split in RTFs scaled by a factor of 10 (for readability) for IB, VarIB, TPIB-NN and TPIB-LDA systems are shown in Table IV. The VarTPIB-NN and VarTPIB-LDA systems show similar trends in module-wise RTFs as TPIB-NN and TPIB-LDA, respectively. Hence, we do not report the module-wise split in RTF for VarTPIB-NN and VarTPIB-LDA systems. It can be seen that the significant part of the RTF, i.e., 0.97 in TPIB-NN system comes from MFNN training, which is reduced significantly to 0.01 for TPIB-LDA system. Early stopping of the first pass and no KL-HMM realignment after the first pass also contributes to the reduction in runtime for TPIB-LDA.

The overall RTFs scaled by a factor of 10 for different systems are given in Table V. The RTF of x-vector based system during test time is 2.13, which includes the runtime of x-vector extraction from the trained TDNN model, PLDA score estimation, performing AHC on x-vectors, and VB based refinement. The IB and VarIB systems are the fastest among all the systems with a similar RTF of 0.74 and 0.82, respectively. The VarIB system improves the SER without significantly degrading the RTF of IB system. The RTF of TPIB-NN and VarTPIB-NN systems are in a similar range of 2.4 - 2.6. Note that the given RTFs are scaled by a factor of 10. The RTF of TPIB-LDA and VarTPIB-LDA systems are also in a similar range of 1.4 - 1.6. The TPIB-LDA and VarTPIB-

LDA systems show 41.8% and 37.6% relative improvements in RTF compared to TPIB-NN and VarTPIB-NN systems, respectively.

Overall, the proposed systems have RTFs well below 1 (or 10 in scaled RTF value) and are only 2-3 times slower than IB based system with significant improvements in SER.

VII. SUMMARY AND CONCLUSION

We propose different architectures for IB based diarization system focusing on two critical challenges in diarization; (i) Segment initialization, and (ii) Speaker discriminative representation. In the first part, we propose a varying length segment initialization technique for the IB system using phoneme rate information. Here, the segments are initialized such that different segments have a similar number of phonemes. We show that a proper initialization of segments results in a better SER with negligible degradation in runtime. In the second part of the work, we show how the TPIB-NN and TPIB-LDA systems help to incorporate the speaker discriminative features during the diarization process itself. This in turn improves the diarization performance significantly. Moreover, the runtime of TPIB-NN is further improved by the TPIB-LDA system. In the third part of the work, we combine the VarIB and TPIB systems to propose VarTPIB systems that leverage the advantages of both, i.e., better segment initialization and speaker discriminative features. The VarTPIB systems show significant improvement in the performance compared to the baseline IB system, and the proposed VarIB and TPIB systems. In conclusion, we show that a good segment initialization along with the speaker discriminative features results in a good clustering solution.

As the focus of this work is towards different frameworks, we used standard MFCC in all our experiments. However, other spectral features like MFS and LFS [8] [14], and group delay based features [15] can also be used in the proposed architectures. The proposed systems are generic enough to be used along with deep embeddings instead of standard spectral features. Hence, in future we would also like to study different data-derived features such as i-vectors [59], d-vectors [31], x-vectors [39], and c-vectors [32] in the proposed architectures. We also plan to extend the TPIB and the VarTPIB systems to perform multiple passes of IB and VarIB based clustering to further refine the speaker boundaries.

REFERENCES

[1] X. Anguera Miro, S. Bozonnet, N. Evans, C. Fredouille, G. Friedland, and O. Vinyals, "Speaker Diarization: A Review of Recent Research," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 2, pp. 356–370, 2012.

[2] "IBM Watson: Who's speaking? : Speaker Diarization with Watson Speech-to-Text API," https://www.ibm.com/blogs/bluemix/2017/05/whos-speaking-speaker-diarization-watson-speech-text-api/.

[3] D. Vijayasenan, F. Valente, and H. Bourlard, "An Information Theoretic Approach to Speaker Diarization of Meeting Data," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 7, pp. 1382–1393, 2009.

[4] ——, "KL Realignment for Speaker Diarization with Multiple Feature Streams," in *Proceedings of INTERSPEECH*, 2009, pp. 1059–1062.

[5] ——, "An Information Theoretic Combination of MFCC and TDOA Features for Speaker Diarization," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 2, pp. 431–438, 2011.

[6] S. Madikeri, D. Imseng, and H. Bourlard, "Improving real time factor of information bottleneck-based speaker diarization system," *Idiap*, 2015.

[7] M. Sinclair and S. King, "Where are the Challenges in Speaker Diarization?" in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 7741–7745.

[8] N. Dawalatabad, S. Madikeri, C. C. Sekhar, and H. A. Murthy, "Two-Pass IB Based Speaker Diarization System Using Meeting-Specific ANN Based Features," in *Proceedings of INTERSPEECH*, 2016, pp. 2199–2203.

[9] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.

[10] C. Yu and J. H. L. Hansen, "Active Learning based Constrained Clustering for Speaker Diarization," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 11, pp. 2188–2198, Nov 2017.

[11] T. J. Park and P. Georgiou, "Multimodal Speaker Segmentation and Diarization Using Lexical and Acoustic Cues via Sequence to Sequence Neural Networks," in *Proceedings of INTERSPEECH*, 2018, pp. 1373–1377.

[12] T. J. Park, K. J. Han, J. Huang, X. He, B. Zhou, P. Georgiou, and S. Narayanan, "Speaker Diarization with Lexical Information," in *Proceedings of INTERSPEECH*, 2019, pp. 391–395.

[13] N. Dawalatabad, J. Kuriakose, C. C. Sekhar, and H. A. Murthy, "Information Bottleneck based Percussion Instrument Diarization System for Taniavartanam Segments of Carnatic Music Concerts," in *Proceedings of INTERSPEECH*, 2018, pp. 1215–1219.

[14] S. Madikeri and H. Bourlard, "Filterbank Slope based Features for Speaker Diarization," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 111–115.

[15] M. Yadav, A. K. Sao, A. D. Dileep, and P. Rajan, "Group Delay Functions for Speaker Diarization," in *Twenty Second National Conference on Communication (NCC), IEEE*, 2016, pp. 1–5.

[16] S. Shum, N. Dehak, E. Chuangsuwanich, D. Reynolds, and J. Glass, "Exploiting Intra-Conversation Variability for Speaker Diarization," in *Proceedings of INTERSPEECH*, 2011, pp. 945–948.

[17] S. Madikeri, P. Motlicek, and H. Bourlard, "Combining SGMM Speaker Vectors and KL-HMM approach for Speaker Diarization," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015, pp. 4834–4838.

[18] I. Lapidot and J. F. Bonastre, "Integration of LDA into a Telephone Conversation Speaker Diarization System," in *IEEE 27th Convention of Electrical Electronics Engineers in Israel (IEEEI)*, 2012, pp. 1–4.

[19] G. Le Lan, D. Charlet, A. Larcher, and S. Meignier, "Iterative PLDA Adaptation for Speaker Diarization," in *Proceedings of INTERSPEECH*, 2016, pp. 2175–2179.

[20] S. H. Yella, A. Stolcke, and M. Slaney, "Artificial Neural Network Features for Speaker Diarization," in *IEEE Spoken Language Technology Workshop (SLT)*, 2014, pp. 402–406.

[21] G. Le Lan, D. Charlet, A. Larcher, and S. Meignier, "A Triplet Ranking-based Neural Network for Speaker Diarization and Linking," in *Proceedings of INTERSPEECH*, 2017, pp. 3572–3576.

[22] H. Bredin, "TristouNet: Triplet Loss for Speaker Turn Embedding," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 5430–5434.

[23] D. Garcia-Romero, D. Snyder, G. Sell, D. Povey, and A. McCree, "Speaker Diarization using Deep Neural Network Embeddings," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 4930–4934.

[24] Q. Wang, C. Downey, L. Wan, P. A. Mansfield, and I. L. Moreno, "Speaker Diarization with LSTM," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5239–5243.

[25] Q. Lin, R. Yin, M. Li, H. Bredin, and C. Barras, "LSTM Based Similarity Measurement with Spectral Clustering for Speaker Diarization," in *Proceedings of INTERSPEECH*, 2019, pp. 366–370.

[26] H. Dubey, A. Sangwan, and J. H. L. Hansen, "Transfer Learning Using Raw Waveform Sincnet for Robust Speaker Diarization," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6296–6300.

[27] M. Ravanelli and Y. Bengio, "Speaker Recognition from Raw Waveform with SincNet," in *IEEE Spoken Language Technology Workshop (SLT)*, 2018, pp. 1021–1028.

[28] Y. Fujita, N. Kanda, S. Horiguchi, K. Nagamatsu, and S. Watanabe, "End-to-End Neural Speaker Diarization with Permutation-Free Objectives," in *Proceedings of INTERSPEECH*, 2019, pp. 4300–4304.

[29] Y. Fujita, N. Kanda, S. Horiguchi, Y. Xue, K. Nagamatsu, and S. Watanabe, "End-to-end Neural Speaker Diarization with Self-attention," *ArXiv e-prints*, p. arXiv:1909.06247, Sep. 2019.

[30] G. Sell, D. Snyder, A. McCree, D. Garcia-Romero, J. Villalba, M. Maciejewski, V. Manohar, N. Dehak, D. Povey, S. Watanabe, and S. Khudanpur, "Diarization is Hard: Some Experiences and Lessons Learned for the JHU Team in the Inaugural DIHARD Challenge," in *Proceedings of INTERSPEECH*, 2018, pp. 2808–2812.

[31] L. Wan, Q. Wang, A. Papir, and I. Lopez-Moreno, "Generalized End-to-End Loss for Speaker Verification," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 4879–4883.

[32] G. Sun, C. Zhang, and P. C. Woodland, "Speaker Diarisation using 2D Self-attentive Combination of Embeddings," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 5801–5805.

[33] A. Zhang, Q. Wang, Z. Zhu, J. Paisley, and C. Wang, "Fully Supervised Speaker Diarization," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6301–6305.

[34] N. Dawalatabad, S. Madikeri, C. C. Sekhar, and H. A. Murthy, "Incremental Transfer Learning in Two-pass Information Bottleneck Based Speaker Diarization System for Meetings," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2019, pp. 6291–6295.

[35] E. S. Olivas, J. D. M. Guerrero, M. M. Sober, J. R. M. Benedito, and A. J. S. Lopez, *Handbook Of Research On Machine Learning Applications and Trends: Algorithms, Methods and Techniques - 2 Volumes*. Hershey, PA: Information Science Reference - Imprint of: IGI Publishing, 2009.

[36] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. The MIT Press, 2016.

[37] A. Jati and P. Georgiou, "Speaker2Vec: Unsupervised Learning and Adaptation of a Speaker Manifold Using Deep Neural Networks with an Evaluation on Speaker Segmentation," in *Proceedings of INTERSPEECH*, 2017, pp. 3567–3571.

[38] A. Jati and P. Georgiou, "Neural Predictive Coding using Convolutional Neural Networks towards Unsupervised Learning of Speaker Characteristics," *ArXiv e-prints*, Feb. 2018.

[39] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-Vectors: Robust DNN Embeddings for Speaker Recognition," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5329–5333.

[40] N. Tishby, F. C. Pereira, and W. Bialek, "The Information Bottleneck Method," *in NEC Research Institute TR*, 1998.

[41] P. M. Kumar, J. Sebastian, and H. A. Murthy, "Musical Onset Detection on Carnatic Percussion Instruments," in *Twenty First National Conference on Communications (NCC)*. IEEE, 2015, pp. 1–6.

[42] "BUT Phoneme Recognizer," https://speech.fit.vutbr.cz/software/phoneme-recognizer-based-long-temporal-context.

[43] L. Rabiner and R. Schafer, *Digital Processing of Speech Signals*, ser. Prentice-Hall signal processing series. Prentice-Hall, 1978.

[44] N. Evans, S. Bozonnet, D. Wang, C. Fredouille, and R. Troncy, "A Comparative Study of Bottom-Up and Top-Down Approaches to Speaker Diarization," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 2, pp. 382–392, Feb 2012.

[45] H. Bredin, "pyannote.metrics: A Toolkit for Reproducible Evaluation, Diagnostic, and Error Analysis of Speaker Diarization Systems," in *Proceedings of INTERSPEECH*, 2017.

[46] "Linguistic Data Consortium," https://catalog.ldc.upenn.edu/.

[47] J. Carletta, S. Ashby, S. Bourban, M. Flynn, M. Guillemot, T. Hain, J. Kadlec, V. Karaiskos, W. Kraaij, M. Kronenthal, G. Lathoud, M. Lincoln, A. Lisowska, I. McCowan, W. Post, D. Reidsma, and P. Wellner, "The AMI Meeting Corpus: A Pre-announcement," in *Proceedings of the Second International Conference on Machine Learning for Multimodal Interaction*, ser. MLMI'05, 2006, pp. 28–39.

[48] X. Anguera, C. Wooters, and J. Hernando, "Acoustic Beamforming for Speaker Diarization of Meetings," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 2011–2022, 2007.

[49] "The NIST Rich Transcription," https://www.nist.gov/itl.

[50] D. Vijayasenan and F. Valente, "DiarTk: An Open Source Toolkit for Research in Multistream Speaker Diarization and its Application to Meetings Recordings," in *Proceedings of INTERSPEECH*, 2012, pp. 2170–2173.

[51] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga, A. Desmaison, A. Kopf, E. Yang, Z. DeVito, M. Raison, A. Tejani, S. Chilamkurthy, B. Steiner, L. Fang, J. Bai, and S. Chintala, "Pytorch: An imperative style, high-performance deep learning library," in *Advances in Neural Information Processing Systems*, 2019, pp. 8024–8035.

[52] X. Glorot and Y. Bengio, "Understanding the Difficulty of Training Deep Feedforward Neural Networks," in *Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics*, 2010, pp. 249–256.

[53] L. Rokach and O. Maimon, *Clustering Methods*. Springer US, 2005, pp. 321–352.

[54] M. Sugiyama, H. Sawai, and A. H. Waibel, "Review of TDNN (time delay neural network) Architectures for Speech Recognition," *IEEE International Sympoisum on Circuits and Systems*, pp. 582–585, 1991.

[55] S. J. D. Prince and J. H. Elder, "Probabilistic Linear Discriminant Analysis for Inferences About Identity," in *IEEE 11th International Conference on Computer Vision*, 2007, pp. 1–8.

[56] G. Sell and D. Garcia-Romero, "Speaker diarization with PLDA i-vector scoring and unsupervised calibration," in *IEEE Spoken Language Technology Workshop (SLT)*, 2014, pp. 413–417.

[57] R. Zheng, C. Zhang, S. Zhang, and B. Xu, "Variational bayes based i-vector for speaker diarization of telephone conversations," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 91–95.

[58] L. van der Maaten and G. Hinton, "Visualizing Data using t-SNE ," *Journal of Machine Learning Research*, vol. 9, pp. 2579–2605, 2008.

[59] S. Madikeri, I. Himawan, P. Motlicek, and M. Ferras, "Integrating Online I-vector extractor with Information Bottleneck based Speaker Diarization System," in *Proceedings of INTERSPEECH*, 2015, pp. 3105–3109.