Bayesian HMM clustering of x-vector sequences (VBx) in speaker diarization: Theory, implementation and analysis on standard tasks☆

Federico Landini [a,*,1], Ján Profant [b,1], Mireia Diez [a], Lukáš Burget [a]

[a] *Brno University of Technology, Faculty of Information Technology, Speech@FIT, Czechia*
[b] *Phonexia, Czechia*

ARTICLE INFO

ABSTRACT

The recently proposed VBx diarization method uses a Bayesian hidden Markov model to find speaker clusters in a sequence of x-vectors. In this work we perform an extensive comparison of performance of the VBx diarization with other approaches in the literature and we show that VBx achieves superior performance on three of the most popular datasets for evaluating diarization: CALLHOME, AMI and DIHARD II datasets. Further, we present for the first time the derivation and update formulae for the VBx model, focusing on the efficiency and simplicity of this model as compared to the previous and more complex BHMM model working on frame-by-frame standard Cepstral features. Together with this publication, we release the recipe for training the x-vector extractors used in our experiments on both wide and narrowband data, and the VBx recipes that attain state-of-the-art performance on all three datasets. Besides, we point out the lack of a standardized evaluation protocol for AMI dataset and we propose a new protocol for both Beamformed and Mix-Headset audios based on the official AMI partitions and transcriptions.

1. Introduction

In recent years, speaker diarization works are proliferating. This is due to two main factors: first, new datasets and challenges are providing new benchmarks that bring the interest of the community and foster healthy competition. The most relevant examples are the DIHARD series addressing diarization in a wide variety of challenging domains (Ryant et al., 2018, 2019), CHIME6 (Watanabe et al., 2020) providing a very demanding multi-channel benchmark for diarization, or the recent VoxConverse (Chung et al., 2020) exploring diarization on several kinds of YouTube videos. Second, the success of the new end-to-end paradigm for speaker recognition is starting to be adopted for diarization tasks. Unlike standard diarization approaches which normally deal with diarization tasks using oracle voice activity detection (VAD), end-to-end diarization systems deal also with the VAD task. The end-to-end approaches have the advantage of being able to cope with overlapped speech (Fujita et al., 2020; Huang et al., 2020; Medennikov et al., 2020). Even if these methods are still limited, e.g. they are restricted to scenarios with a fixed number of speakers, and mainly tested on artificially created short recordings (Fujita et al., 2020; Kinoshita et al., 2020), and mostly do not achieve state-of-the-art results (Horiguchi et al., 2020), this research line is very promising and indeed prolific.

However, due to the difficulties that end-to-end approaches still have to overcome for diarization tasks, in recent speaker diarization evaluations the best performing systems were based on a more conventional approach, the clustering of x-vectors (Sell et al., 2018; Landini et al., 2020; Xiao et al., 2020).

In this paper, we show how our current VBx diarization approach, which clusters x-vectors using a Bayesian hidden Markov model (BHMM) (Diez et al., 2020), combined with a ResNet101 x-vector extractor (He et al., 2016) achieves superior results on CALLHOME (Martin and Przybocki, 2001), AMI (Carletta et al., 2006) and DIHARD II (Ryant et al., 2019) datasets.

Besides establishing new baselines for these representative datasets, we perform a thorough analysis comparing our results to the best numbers found in the literature. In the case of the AMI dataset, this proved to be a challenging task. Most works published on the AMI corpus choose their own partition, references and audio setup for evaluation, making the comparison between works very complicated. We identified works that were presenting superior performance and reproducible setups and we evaluated our system in their respective setups. In this paper, we further provide our own evaluation protocol, which comprises lists for train/dev/eval partition, references and audios. Our setup is based on the official partition of the AMI corpus. We believe that this setup could serve as a new standard facilitating a fair comparison of diarization systems on the AMI corpus.

The VBx diarization approach has been presented before (Diez et al., 2019), but the paper did not provide any derivation of update formulae, as it was introduced merely as a special case and simplification of its big-brother BHMM with eigen-voice priors (Diez et al., 2020). The more complex BHMM model from Diez et al. (2020) operates directly on frame-by-frame standard Cepstral features. It is based on a Bayesian HMM where states correspond to speakers and transitions correspond to speaker turns. To robustly model speaker distributions it uses an i-vector like model (Dehak et al., 2011; Kenny, 2008): the distribution of each speaker is represented by a Gaussian mixture model (GMM). The GMM is constrained to live in a low-dimensional eigen-voice subspace and each speaker can be therefore robustly represented by a fixed-length i-vector like latent variable.

The VBx approach used in this paper is based on a similar BHMM model and a similar idea for modeling speaker distributions. However, it is used for directly clustering x-vectors, which uses a much simpler probabilistic linear discriminant analysis (PLDA) based model for modeling speaker distributions. The model is essentially the same as the one in Diez et al. (2020) but using only a single Gaussian component to model speaker distributions. In fact, when VBx was introduced in Diez et al. (2019), it was suggested that the same model and the same inference from the BHMM model working on a frame-basis (Diez et al., 2020) could be reused just by replacing the GMMs that modeled the speaker-specific distributions by single Gaussians. However, naively re-implementing the algorithm used in Diez et al. (2020) is not effective as the design changes made to obtain the VBx model lead to significant simplification of the inference formulae and derivations. Therefore, in this paper, we present the same derivations and update formulae as in Diez et al. (2020), but now for the simpler VBx. This derivation should be much easier to follow for readers interested in this specific model. It also allows us to elaborate on how to make this simplified model much more computationally efficient.

It is important to realize that even if these two diarization models have very similar mathematical foundations, the complete diarization system for each of the models is of very different nature: In the case of VBx, the most expensive part of the acoustic modeling is carried out by the external x-vector extractor. In contrast, in the original BHMM (Diez et al., 2020) the Cepstral frame-by-frame features were modeled by a GMM, which was the integral part of the model. Therefore, both systems require very different training strategies (data selection, augmentations, preprocessing, etc.), which does not allow an easy and fair comparison of both methods. The comparison of both models was already done in the context of DIHARD II in our previous work (Diez et al., 2019) where both systems were tuned for optimal performance in the challenge, and where VBx was shown to provide superior performance. Thus, the point of this paper is not to compare the VBx with the frame-by-frame BHMM but to compare the VBx with the current state-of-the-art.

All our code is made publicly available: the recipe for training the x-vector extractor (same architecture for both 8 kHz and 16 kHz), the trained extractors and the pipeline for applying BHMM diarization using agglomerative hierarchical clustering (AHC) as initialization https://github.com/BUTSpeechFIT/VBx.

2. VBx diarization model

This section introduces the VBx model which is used in all the experiments in this paper. The derivation of the inference formulae is also provided. While this derivation is essentially the same as the one that can be found in Diez et al. (2020), it only addresses the simple model used in this paper. In fact, the following text is the same as Section 2 from Diez et al. (2020), rewritten and simplified to address only the model considered in this paper. We intentionally reused the text, structure and symbols from the mentioned paper in order to make the treatment of the model here as self-contained as possible and to facilitate the comparison of the simplified model with the more complex full model proposed in Diez et al. (2020).

The main differences between the model presented here and the one described in Diez et al. (2020) are:

- In Diez et al. (2020), the observations are frame-by-frame Mel frequency cepstral coefficients (MFCC) while here the observations are x-vectors. As a result, VBx operates at a different granularity, and some parameters have different relevance when compared to the model in Diez et al. (2020):
    - The former downsampling factor is not used in this model
    - The fetch factors (introduced in Section 2.5) gain relevance
- In Diez et al. (2020), each speaker (HMM state) is modeled using a GMM with parameters constrained by eigen-voice priors (like in i-vector or JFA models) while here each speaker is modeled with a single Gaussian with parameters constrained by a PLDA model. This results in:

– A simplified form for the speaker-specific distributions (Section 2.3)
– Simpler updates of the approximate posterior speaker distributions in the diarization inference (Section 2.5.1)
– Simpler form for the total forward probability in the HMM model (Eq. (23))

As an extra reference for this section, consider the technical report (Diez and Burget, 2021). This report is intended to be used as complementary material, as it gives the full derivations of the update formulae from the diarization inference, but contains also a summary of different variables that can help readers follow the notation.

2.1. Model overview

As described in the previous section, we expect a sequence of x-vectors extracted from consecutive short segments of speech as input to our diarization method, which aims to cluster these x-vectors according to their speaker identity.

Our diarization model assumes that the input sequence of x-vectors is generated by an HMM with speaker-specific state distributions. To facilitate the discrimination between speakers, the speaker- (or HMM state-) specific distributions are derived from a PLDA (Kenny, 2010) model pre-trained on a large number of speaker-labeled x-vectors. More details on how the speaker-specific distributions are derived from the PLDA are given in Section 2.3. For now, it is sufficient to note that the speaker distributions will be represented only by a latent vector $\mathbf{y}_s$ of the same dimensionality as the x-vectors.

We use an ergodic HMM with one-to-one correspondence between the HMM states and the speakers, where transitions from any state to any state are possible. Note that our model does not consider any overlapped speech as each speech frame is assumed to be generated from an HMM state corresponding to only one of the S speakers. The transition probabilities can be used to discourage too frequent transitions between speakers in order to reflect speaker turn durations of a natural conversation. More details on setting and learning the transition probabilities can be found in Section 2.2.

Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_T\}$ be the sequence of observed x-vectors and $\mathbf{Z} = \{z_1, z_2, \ldots, z_T\}$ the corresponding sequence of discrete latent variables defining the hard alignment of x-vectors to HMM states. In our notation, $z_t = s$ indicates that the speaker (HMM state) s is responsible for generating observation $\mathbf{x}_t$.

To address the speaker diarization (SD) task using our model, the speaker distributions (i.e. the vectors $\mathbf{y}_s$) and the latent variables z_t are jointly estimated given an input sequence $\mathbf{X}$. The solution to the SD task is then given by the most likely sequence $\mathbf{Z}$, which encodes the alignment of speech frames to speakers.

2.2. HMM topology

The HMM topology and transition probabilities model the speaker turn durations. Fig. 1 shows an example of the HMM topology for only $S = 3$ speakers. The transition probabilities are set as follows: we transition back to the same speaker/state with probability P_{loop}. This probability is one of the tunable parameters in the model. The remaining probability $(1 - P_{loop})$ is the probability of changing speaker, which corresponds to the transition to the non-emitting node in Fig. 1. From the non-emitting node, we immediately transition to one of the speaker states with probability π_s.[2] Therefore, the probability of leaving a speaker and entering another speaker s is $(1 - P_{loop})\pi_s$. To summarize, the probability of transitioning from state s' to state s is

$$p(s|s') = (1 - P_{loop})\pi_s + \delta(s = s')P_{loop}, \tag{1}$$

where $\delta(s = s')$ equals 1 if $s = s'$ and is 0 otherwise.

The non-emitting node in Fig. 1 is also the initial state of the model. Therefore, the probabilities π_s also control the selection of the initial HMM state (i.e. the state generating the first observation). These probabilities π_s are inferred (jointly with the variables $\mathbf{y}_s$ and z_t) from the input conversation. Thanks to the automatic relevance determination principle (Bishop, 2006) stemming from our Bayesian model (see Section 2.5.3 for details), zero probabilities will be learned for the π_s corresponding to redundant speakers, which effectively drops such speakers from the HMM model. Typically, we initialize the HMM with a larger number of speakers (see Section 3.2) and we make use of this behavior to drop the redundant speakers (i.e. to estimate the number of speakers in the conversation).

2.3. Speaker-specific distributions

The speaker (HMM state) specific distributions are derived from a PLDA which is a standard model used for comparing x-vectors in speaker verification (Kenny, 2010). Here, only a simplified variant of PLDA is considered, which is often referred to as *two-covariance model* (Brummer and Villiers, 2010). This model assumes that the distribution of x-vectors specific to speaker s is Gaussian $\mathcal{N}(\hat{\mathbf{x}}_t; \hat{\mathbf{m}}_s, \boldsymbol{\Sigma}_w)$, where $\boldsymbol{\Sigma}_w$ is the within-speaker covariance matrix shared by all speaker models, and $\hat{\mathbf{m}}_s$ is the speaker-specific mean. Speaker means are further assumed to be Gaussian distributed $\mathcal{N}(\hat{\mathbf{m}}_s; \mathbf{m}, \boldsymbol{\Sigma}_b)$, where $\mathbf{m}$ is the global mean and $\boldsymbol{\Sigma}_b$ is the between-speaker covariance matrix. In general, $\boldsymbol{\Sigma}_w$ and $\boldsymbol{\Sigma}_b$ can be full covariance matrices. However, to further simplify and speed up the inference in our model, we assume that the x-vectors are linearly transformed into a space where $\boldsymbol{\Sigma}_b$ is diagonal and $\boldsymbol{\Sigma}_w$ is identity. This can be achieved as follows:

---

[2] For convenience, we allow to re-enter the same speaker as it leads to simpler update formulae.
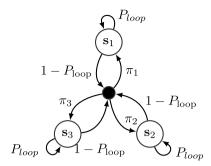
Fig. 1. HMM model for 3 speakers with 1 state per speaker, with a dummy non-emitting (initial) state.

Let $\hat{\mathbf{X}}$ be the matrix of original (untransformed) x-vectors that the parameters of the original PLDA model $\mathbf{m}$, $\boldsymbol{\Sigma}_w$ and $\boldsymbol{\Sigma}_b$ were estimated on. The x-vectors that are used as input for the diarization algorithm are obtained as

$$\mathbf{X} = (\hat{\mathbf{X}} - \mathbb{1}\mathbf{m}^T)\mathbf{E} \tag{2}$$

where $\mathbf{E}$ is the transformation matrix which transforms the x-vectors into the desired space and $\mathbb{1}$ is a vector of ones. This matrix can be obtained by solving the standard generalized eigen-value problem

$$\boldsymbol{\Sigma}_b\mathbf{E} = \boldsymbol{\Sigma}_w\mathbf{E}\boldsymbol{\Phi} \tag{3}$$

where $\mathbf{E}$ is the matrix of eigen-vectors and $\boldsymbol{\Phi}$ is the diagonal matrix of eigen-values, which is also the between-speaker covariance matrix in the transformed space. Note that the eigen-vectors $\mathbf{E}$ are, in fact, bases of linear discriminant analysis (LDA) estimated directly from the PLDA model parameters. Therefore, if we construct $\boldsymbol{\Phi}$ only using R largest eigen-values and assemble $\mathbf{E}$ only using the corresponding eigen-vectors, (2) further performs LDA dimensionality reduction of x-vectors to R-dimensional space. We use R as one of the hyper-parameters of the VBx method. In Eq. (2), we have also subtracted the global mean from the original x-vectors to have the new set of x-vectors zero-centered.

In summary, the PLDA model compatible with the new set of x-vectors $\mathbf{X}$ suggests that speaker-specific means are distributed as

$$p(\mathbf{m}_s) = \mathcal{N}(\mathbf{m}_s; \mathbf{0}, \boldsymbol{\Phi}). \tag{4}$$

For convenience and for the compatibility with the notation introduced in Diez et al. (2020), we further re-parametrize the speaker mean as

$$\mathbf{m}_s = \mathbf{V}\mathbf{y}_s, \tag{5}$$

where diagonal matrix $\mathbf{V} = \boldsymbol{\Phi}^{\frac{1}{2}}$ and $\mathbf{y}_s$ is a standard normal distributed random variable

$$p(\mathbf{y}_s) = \mathcal{N}(\mathbf{y}_s; \mathbf{0}, \mathbf{I}). \tag{6}$$

The speaker-specific distribution of x-vectors is

$$p(\mathbf{x}_t|\mathbf{y}_s) = \mathcal{N}(\mathbf{x}_t; \mathbf{V}\mathbf{y}_s, \mathbf{I}), \tag{7}$$

where $\mathbf{I}$ is identity matrix.

In our diarization model, we use (7) to model the speaker (HMM state) distributions. This distribution is fully defined only in terms of the speaker vector $\mathbf{y}_s$ (and the pre-trained matrix $\mathbf{V}$ shared by all the speakers). The speaker vector $\mathbf{y}_s$ is treated as a latent variable with standard normal prior (6), which is why the BHMM model is called *Bayesian*.[3] This way, the full PLDA model is incorporated into the BHMM in order to properly model between- and across-speaker variability. Therefore, the model is capable of discriminating between speakers just like a PLDA model when used for speaker verification.

2.4. Bayesian HMM

To summarize, our complete model for SD is a Bayesian HMM, which is defined in terms of the state-specific distributions (or so-called output probabilities)

$$p(\mathbf{x}_t|z_t = s) = p(\mathbf{x}_t|s) = p(\mathbf{x}_t|\mathbf{y}_s) \tag{8}$$

---

[3] However, unlike other "Fully Bayesian" HMM implementations (Fox et al., 2007; Beal, 2003), we do not impose any prior on the transition probabilities.

described in Section 2.3 and the transition probabilities

$$p(z_t = s | z_{t-1} = s') = p(s|s') \tag{9}$$

described in Section 2.2. By abuse of notation, $p(z_1|z_0)$ will correspond to the initial state probability $p(z_1 = s) = \pi_s$ in the following formulae.

The complete model can be also defined in terms of the joint probability of the observed and latent random variables (and their factorization) as

$$p(\mathbf{X}, \mathbf{Z}, \mathbf{Y}) = p(\mathbf{X}|\mathbf{Z}, \mathbf{Y})p(\mathbf{Z})p(\mathbf{Y})$$
$$\prod_t p(\mathbf{x}_t|z_t) \prod_t p(z_t|z_{t-1}) \prod_s p(\mathbf{y}_s), \tag{10}$$

where $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_S\}$ is the set of all the speaker-specific latent variables.

The model assumes that each x-vector sequence corresponding to an input conversation is obtained using the following generative process:

> **for** $s = 1..S$ **do**
> $\quad \mathbf{y}_s \sim \mathcal{N}(0, \mathbf{I})$
> $\quad$ **for** $t = 1..T$ **do**
> $\quad\quad z_t \sim p(z_t|z_{t-1})$
> $\quad\quad \mathbf{x}_t \sim p(\mathbf{x}_t|z_t)$

2.5. Diarization inference

The diarization problem consists in finding the assignment of frames to speakers, which is represented by the latent sequence $\mathbf{Z}$. In order to find the most likely sequence $\mathbf{Z}$, we need to infer the posterior distribution $p(\mathbf{Z}|\mathbf{X}) = \int p(\mathbf{Z}, \mathbf{Y}|\mathbf{X}) d\mathbf{Y}$. Unfortunately, the evaluation of this integral is intractable, and therefore, we will approximate it using variational Bayes (VB) inference (Bishop, 2006), where the distribution $p(\mathbf{Z}, \mathbf{Y}|\mathbf{X})$ is approximated by $q(\mathbf{Z}, \mathbf{Y})$. We use the mean-field approximation (Bishop, 2006; Kenny, 2008) assuming that the approximate posterior distribution factorizes as

$$q(\mathbf{Z}, \mathbf{Y}) = q(\mathbf{Z})q(\mathbf{Y}). \tag{11}$$

The particular form of the approximate distributions $q(\mathbf{Z})$ and $q(\mathbf{Y})$ directly follows from the optimization described below.

We search for such $q(\mathbf{Z}, \mathbf{Y})$ that minimizes the Kullback–Leibler divergence $D_{KL}(q(\mathbf{Z}, \mathbf{Y}) \parallel p(\mathbf{Z}, \mathbf{Y}|\mathbf{X}))$, which is equivalent to maximizing the standard VB objective — the evidence lower bound objective (ELBO) (Bishop, 2006)

$$\mathcal{L}(q(\mathbf{X}, \mathbf{Y})) = E_{q(\mathbf{Y}, \mathbf{Z})} \left\{ \ln \left(\frac{p(\mathbf{X}, \mathbf{Y}, \mathbf{Z})}{q(\mathbf{Y}, \mathbf{Z})} \right) \right\}. \tag{12}$$

Using the factorization (11), the ELBO can be split into three terms

$$\hat{\mathcal{L}}(q(\mathbf{X}, \mathbf{Y})) = F_A E_{q(\mathbf{Y}, \mathbf{Z})} [\ln p(\mathbf{X}|\mathbf{Y}, \mathbf{Z})] + F_B E_{q(\mathbf{Y})} \left[\ln \frac{p(\mathbf{Y})}{q(\mathbf{Y})} \right] + E_{q(\mathbf{Z})} \left[\ln \frac{p(\mathbf{Z})}{q(\mathbf{Z})} \right], \tag{13}$$

where the first term is the expected log-likelihood of the observed x-vector sequence $\mathbf{X}$ and the second and third terms are Kullback–Leibler divergences $D_{KL}(q(\mathbf{Y}) \parallel p(\mathbf{Y}))$ and $D_{KL}(q(\mathbf{Z}) \parallel p(\mathbf{Z}))$ regularizing the approximate posterior distributions $q(\mathbf{Y})$ and $q(\mathbf{Z})$ towards the priors $p(\mathbf{Y})$ and $p(\mathbf{Z})$. In (13), we modified the ELBO by scaling the first two terms by constant factors F_A and F_B.[4] The theoretically correct values for these factors leading to the original ELBO (13) are $F_A = F_B = 1$. However, choosing different values gives us finer control over the inference, which can be used to improve diarization performance. The acoustic scaling factor F_A is introduced to counter act the assumption of statistical independence between observations. The speaker regularization coefficient F_B can be seen as a regularization term penalizing the complexity of the speaker models (i.e. the posteriors of speaker latent variables should not be too far from the standard normal priors). Setting a high value of F_B results in the VB inference dropping more speakers. For further details on these scaling factors and the effect they have in the inference, we refer the reader to (Diez et al., 2020).

As described above, we search for the approximate posterior $q(\mathbf{Z}, \mathbf{Y})$ that maximizes the ELBO (13). In the case of the mean-field factorization (11), we proceed iteratively by finding the $q(\mathbf{Y})$ that maximizes the ELBO given fixed $q(\mathbf{Z})$ and vice versa. This section provides all the formulae necessary for implementing these updates or for understanding our open-source Python implementation.[5] In this section, we do not give any details on deriving the update formulae. For the readers interested in the derivations, we prepared a technical report (Diez and Burget, 2021).

---

[4] Note that similar scaling factor for the third term would be redundant as only the relative scale of the three factors is relevant for the optimization.
[5] http://speech.fit.vutbr.cz/software/vb-diarization-eigenvoice-and-hmm-priors

2.5.1. Updating $q(\mathbf{Y})$

Given a fixed $q(\mathbf{Z})$, the distribution over $\mathbf{Y}$ that maximizes the ELBO is

$$q^*(\mathbf{Y}) = \prod_s q^*(\mathbf{y}_s), \tag{14}$$

where the speaker-specific approximate posteriors

$$q^*(\mathbf{y}_s) = \mathcal{N}\left(\mathbf{y}_s | \boldsymbol{\alpha}_s, \mathbf{L}_s^{-1}\right) \tag{15}$$

are Gaussians with the mean vector and precision matrix

$$\boldsymbol{\alpha}_s = \frac{F_A}{F_B} \mathbf{L}_s^{-1} \sum_t \gamma_{ts} \boldsymbol{\rho}_t \tag{16}$$

$$\mathbf{L}_s = \mathbf{I} + \frac{F_A}{F_B} \left(\sum_t \gamma_{ts} \right) \boldsymbol{\Phi}. \tag{17}$$

where

$$\boldsymbol{\rho}_t = \mathbf{V}^T \mathbf{x}_t \tag{18}$$

In this update formula, $\gamma_{ts} = q(z_t = s)$ is the marginal approximate posterior derived from the current estimate of the distribution $q(\mathbf{Z})$ (see below), which can be interpreted as the responsibility of speaker s for generating observation $\mathbf{x}_t$ (i.e. defines a soft alignment of x-vectors to speakers). Note that while in the former frame-by-frame BHMM model zero, first and second order statistics are computed (equations (12)–(14) in Diez et al. (2020)), here only the first order statistics $\boldsymbol{\rho}_t$ are needed.

If we compare these update formulae to the corresponding ones from the more complex BHMM model in Diez et al. (2020), it can be seen that in Diez et al. (2020), $\boldsymbol{\Phi}_t$ is a frame-dependent full-matrix computationally expensive to calculate. In contrast, $\boldsymbol{\Phi}$ here does not depend on time frame t and, as pointed out in Section 2.3, it is a diagonal matrix. Therefore also matrix $\mathbf{L}_s$ is diagonal, and its inversions and application in (16) become trivial.

2.5.2. Updating $q(\mathbf{Z})$

We never need to infer the complete distribution over all the possible alignments of observations to speaker $q(\mathbf{Z})$. When updating $q(\mathbf{Y})$ using (16) and (17), we only need the marginals $\gamma_{ts} = q(z_t = s)$. Therefore, when updating $q(\mathbf{Z})$, we can directly search for the responsibilities γ_{ts} that correspond to the distribution $q^*(\mathbf{Z})$ maximizing the ELBO given a fixed $q(\mathbf{Y})$. Similar to the standard HMM training, such responsibilities can be calculated efficiently using a forward–backward algorithm as

$$\gamma_{ts} = \frac{A(t,s)B(t,s)}{\bar{p}(\mathbf{X})} \tag{19}$$

where the forward probability

$$A(t,s) = \bar{p}(\mathbf{x}_t | s) \sum_{s'} A(t-1,s') p(s|s') \tag{20}$$

is recursively evaluated by progressing forward in time for $t=1..T$ starting with $A(0,s) = \pi_s$. Similarly,

$$B(t,s) = \sum_{s'} B(t+1,s') \bar{p}(\mathbf{x}_{t+1} | s') p(s'|s) \tag{21}$$

is the backward probability evaluated using backward recursion for times $t = T..1$ starting with $B(T,s) = 1$.

$$\bar{p}(\mathbf{X}) = \sum_s A(T,s) \tag{22}$$

is the total forward probability and

$$\ln \bar{p}(\mathbf{x}_t | s) = F_A \left[\boldsymbol{\alpha}_s^T \boldsymbol{\rho}_t - \frac{1}{2} \operatorname{tr} \left(\boldsymbol{\Phi} \left[\mathbf{L}_s^{-1} + \boldsymbol{\alpha}_s \boldsymbol{\alpha}_s^T \right] \right) \right] - \frac{D}{2} \ln 2\pi - \frac{1}{2} \mathbf{x}_t^T \mathbf{x}_t = F_A \left[\boldsymbol{\alpha}_s^T \boldsymbol{\rho}_t - \frac{1}{2} \boldsymbol{\phi}^T \left[\lambda_s + \boldsymbol{\alpha}_s^2 \right] - \frac{D}{2} \ln 2\pi - \frac{1}{2} \mathbf{x}_t^T \mathbf{x}_t \right] \tag{23}$$

is the expected log-likelihood of observation $\mathbf{x}_t$ given a speaker s taking into account its uncertainty $q(\mathbf{y}_s)$. The rightmost part of (23) corresponds to an efficient evaluation of this term, where vector $\boldsymbol{\phi}$ is the diagonal of the diagonal matrix $\boldsymbol{\Phi}$, vector λ_s is the diagonal of the diagonal matrix $\mathbf{L}_s^{-1}$ and the square in $\boldsymbol{\alpha}_s^2$ is element-wise. Note also that the terms $-\frac{D}{2} \ln 2\pi - \frac{1}{2} \mathbf{x}_t^T \mathbf{x}_t$ in (23) are not only constant over VB iterations, but also constant for different speakers s. As a consequence, contribution of these terms cancels in (19) and therefore does not have to be calculated at all.

2.5.3. Updating π_s

Finally, the speaker priors π_s are updated as maximum likelihood type II estimates (Bishop, 2006): Given fixed $q(\mathbf{Y})$ and $q(\mathbf{Z})$, we search for the values of π_s that maximize the ELBO (13), which gives the following update formula

$$\pi_s \propto \gamma_{1s} + \frac{(1-P_{loop})\pi_s}{\bar{p}(\mathbf{X})} \sum_{t=2}^{T} \sum_{s'} A(t-1,s') p(\mathbf{x}_t | s) B(t,s) \tag{24}$$

with the constraint $\sum_s \pi_s = 1$. As described in Section 2.2, this update tends to drive the π_s corresponding to "redundant speakers" to zero values, which effectively drops them from the model and selects the right number of speakers in the input conversation.[6]

2.5.4. Evaluating the ELBO

The convergence of the iterative VB inference can be monitored by evaluating the ELBO objective. For the Bayesian HMM, the ELBO can be efficiently evaluated (see page 95 of Beal, 2003) as

$$\hat{\mathcal{L}} = \ln \overline{p}(\mathbf{X}) + \sum_s \frac{F_B}{2} \left(R + \ln |\mathbf{L}_s^{-1}| - \mathrm{tr}(\mathbf{L}_s^{-1}) - \boldsymbol{\alpha}_s^T \boldsymbol{\alpha}_s \right), \tag{25}$$

where R is the dimensionality of the x-vectors. Note, that since $\mathbf{L}_s$ is a diagonal matrix, $\ln |\mathbf{L}_s^{-1}|$ can be calculated just as the sum of the log of the elements in the diagonal. This way of evaluating the ELBO is very practical as the term $\overline{p}(\mathbf{X})$ from (22) is obtained as a byproduct of "updating $q(\mathbf{Z})$" using the forward–backward algorithm. On the other hand, the ELBO can be evaluated using (25) only right after the $q(\mathbf{Z})$ update. Thus, the improvements in the ELBO obtained from $q(\mathbf{Y})$ or π_s updates cannot be monitored, which might be useful for debugging purposes. Therefore, we also provide the derivation formulae for the explicit evaluation of all three ELBO terms from (13) in Diez and Burget (2021).

The complete VB inference consisting of iterative updates of $q(\mathbf{Y})$, $q(\mathbf{Z})$ and parameters π_s is summarized in the following algorithm:

Initialize all γ_{ts} as described in Section 3.2.
repeat
    Update $q(\mathbf{y}_s)$ for $s=1..S$ using (15)
    for $t = 1..T$ **do**
        Calculate $A(t, s)$ for $s=1..S$ using (20)
    for $t = T..1$ **do**
        Calculate $B(t, s)$ for $s=1..S$ using (21)
    Update γ_{ts} for $t=1..T, s=1..S$ using (19)
    Update π_s for $s=1..S$ using (24)
    Evaluate ELBO $\hat{\mathcal{L}}$ using (25)
until convergence of $\hat{\mathcal{L}}$

---

3. Experimental setup

3.1. x-vector extractor and PLDA

As described in the previous section, VBx diarization relies on a pre-trained x-vector extractor and a PLDA model. Since we report results on both 16 kHz recordings (DIHARD and AMI) and 8 kHz telephone recordings (CALLHOME), we train two x-vector extractors and the corresponding PLDA models, one for each condition. The complete PyTorch (Paszke et al., 2019) recipe for x-vector extractor and PLDA training is available at https://github.com/phonexiaresearch/VBx-training-recipe.

3.1.1. x-vector extractor architecture

Both 8 kHz and 16 kHz x-vector extractors use the same deep neural network architecture based on ResNet101 (He et al., 2016; Zeinali et al., 2019). In both cases, the neural network inputs are 64 log Mel filter bank features extracted every 10 ms using 25 ms window. The two x-vector extractors differ only in the frequency ranges spanned by the Mel filters, which are 20–7700 Hz and 20–3700 Hz for the 16 kHz and 8 kHz systems, respectively. The x-vector extractor architecture is summarized in Table 1. The first 2D convolutional layer operates on the $64 \times T$ matrix of log Mel filter bank features, where T is the number of frames in the input segments. For training, we use 4 s segments (i.e. $T = 400$). The following layers are standard ResNet blocks (He et al., 2016). As in the original x-vector architecture (Snyder et al., 2018), the statistical pooling layer is used to aggregate information over the whole speech segment (i.e. mean and standard deviation of activations is calculated over the time dimension). After the pooling layer, a linear transformation is used to reduce the dimensionality to obtain the 256-dimensional x-vectors.

The x-vector extractors are trained using stochastic gradient descent and additive angular margin loss (Deng et al., 2019) with speaker identities as class labels. We ramp-up the margin during the first two epochs (pass through the training data) and then train the neural network for another epoch with fixed margin $m = 0.2$.

---

[6] This is known as automatic relevance determination (ARD) principle (Bishop, 2006).

Table 1

The structure of the proposed ResNet101 architecture. The first dimension of the input shows the size of the filterbank and the second dimension indicates the number of frames.

Layer	Structure	Stride	Output
Input	–	–	64×T×1
Conv2D-1	$3 \times 3, 32$	1	64×T×32
ResNetBlock-1	$\begin{bmatrix} 1 \times 1, 32 \\ 3 \times 3, 32 \\ 1 \times 1, 128 \end{bmatrix} \times 3$	1	64×T×128
ResNetBlock-2	$\begin{bmatrix} 1 \times 1, 64 \\ 3 \times 3, 64 \\ 1 \times 1, 256 \end{bmatrix} \times 4$	2	32×T/2×256
ResNetBlock-3	$\begin{bmatrix} 1 \times 1, 128 \\ 3 \times 3, 128 \\ 1 \times 1, 512 \end{bmatrix} \times 23$	2	16×T/4×512
ResNetBlock-4	$\begin{bmatrix} 1 \times 1, 256 \\ 3 \times 3, 256 \\ 1 \times 1, 1024 \end{bmatrix} \times 3$	2	8×T/8×1024
Statistics Pooling	–	–	16×1024
Flatten	–	–	16384
Linear	–	–	256

Table 2

Number of speakers and total amount of training data for the 16 kHz x-vector extractor.

Set	Speakers	Hours
VoxCeleb1	1211	323
VoxCeleb2	5994	2290
CN-CELEB	973	264
Total	8178	2877

Table 3

Number of speakers and total amount of training data for the 8 kHz x-vector extractor.

Set	Speakers	Hours
VoxCeleb1	1211	323
VoxCeleb2	5994	2290
CN-CELEB	973	264
Mixer collection	4254	3805
Switchboard	2591	1170
DeepMine	1858	688
Total	16881	8540

3.1.2. x-vector extractor training data

The 16 kHz x-vector extractor is trained using data from VoxCeleb1 (Nagrani et al., 2017) (323 h of speech from 1211 speakers), VoxCeleb2 (Chung et al., 2018) (2290 h, 5994 speakers) and CN-CELEB (Fan et al., 2020) (264 h, 973 speakers) (see Table 2). The energy-based VAD from Kaldi (Povey et al., 2011) toolkit is used to remove silence frames. Speakers with less than 2 recordings are discarded. Further, we drop utterances with less than 4 s of speech. This way, about 4% of speech data is discarded. Data augmentation is performed the same way as in the SRE16 Kaldi recipe (Kaldi, 2017), resulting in four additional copies of the data with artificially added noise, music or reverberation. Training examples are randomly sampled from the training data. This way we extract about 89 million examples (original and augmented 4s segments), which cover more than 60% of the speech from the training corpora.

To train the 8 kHz x-vector extractor, the same data sets are used as in the 16 kHz case. Additionally, the following data sets were used: Mixer collection (NIST SRE 2004–2010, 3805 h, 4254 speakers), Switchboard (1170 h, 2591 speakers) and DeepMine (Zeinali et al., 2018) (688 h, 1858 speakers) (see Table 3). Any wide-band data used were downsampled to 8 kHz and passed through a telephone codec (selected randomly from AMR, G.711, G.726, GSM-FR and GSM-EFR). The same data selection and augmentation was used as for the 16 kHz case. Note that about 30% of DeepMine data were discarded as this dataset contains many utterances with less than 4 s of speech (mostly phrases for text-dependent speaker verification).

3.1.3. PLDA training

The 8 kHz and 16 kHz PLDA models are trained on the same data as the corresponding x-vector extractors. For this purpose, one x-vector is extracted from each individual recording (e.g. one cut from a YouTube video in the case of the VoxCeleb data). The length of such recordings can range from 4 s to several minutes. Note that the PLDA trained on such x-vectors is later used in VBx to operate on x-vectors extracted from much shorter 1.5 s segments. This mismatch, however, does not seem to negatively affect diarization performance.

3.2. Diarization pipeline

To perform the diarization, each input recording is first split into speech segments according to the oracle VAD and the segments shorter than 0.1 s are discarded. From these segments, x-vectors are extracted every 0.25 s from overlapping sub-segments of 1.5 s (or less than 1.5 s for the last sub-segments or shorter segments). The x-vectors are centered, whitened and length normalized (Garcia-Romero and Espy-Wilson, 2011) (which is also done for the PLDA training data).

As described in Section 2.2, VBx diarization needs an initial assignment of x-vectors to speakers. For this purpose, the x-vectors are pre-clustered using AHC to obtain the initial speaker labels. The only input to the AHC is the matrix of cosine similarities between all pairs of x-vectors. The threshold used as the stopping criterion for AHC is tuned to under-cluster so that the following VBx has more freedom to search for the optimal results and converge to the right number of speaker models.[7] Nevertheless, the same threshold is used for all our results on all the datasets.

In the final step, the x-vectors are further clustered using the VBx model and the inference described in Section 2. For this step, the x-vector dimensionality is further reduced to 128 dimensions (see parameter R in Section 2.3). Note that unlike in our previous works (Landini et al., 2020; Diez et al., 2020), we do not perform any adaptation of PLDA models to the target data as this paper aims to present a simple diarization system which performs well for different datasets. Nevertheless, we tune the VBx parameters: F_A, F_B, P_{loop} on the respective DIHARD II, AMI and CALLHOME development sets.

In order to demonstrate the effectiveness of the VBx method, we also report results for baseline systems where only AHC is used to cluster x-vectors. In this case, the stopping threshold is tuned to obtain the best performance on the respective development set.

3.3. Evaluation datasets

3.3.1. CALLHOME

The 2000 NIST Speaker Recognition Evaluation (LDC2001S97[8]) dataset, usually referred as "CALLHOME", NIST (2000) has been the standard dataset for diarization in the last decade (Shum et al., 2013; Senoussaoui et al., 2014; Garcia-Romero et al., 2017). In its full form,[9] it consists of 499[10] recordings of conversational telephone speech in Arabic, English, German, Japanese, Mandarin and Spanish. The number of speakers per recording ranges between 2 and 7, although 87% of the files contain only 2 or 3 speakers. It amounts to around 15 h of speech after VAD.

Since a development-evaluation split for CALLHOME is not available, we split the dataset into two halves as defined in the Kaldi recipe for CALLHOME.[11] We use this split to perform cross-validation to tune parameters i.e. F_A, F_B and P_{loop}. For the results we present, in order to simplify the setup, we simply chose one combination of parameters that was close to optimal on both partitions and report results with it on the whole set.

3.3.2. AMI corpus

When trying to cover the most standard datasets for speaker diarization, we could not leave AMI out (Carletta et al., 2006). The AMI meeting corpus is a multi-modal data collection of 100 h of meeting recordings. This corpus was recorded using both close-talking and far-field microphones. It consists of 171 meetings recorded at the University of Edinburgh (U.K.), Idiap (Switzerland), and the TNO Human Factors Research Institute (The Netherlands). The dataset comes with annotations for automatic speech recognition (ASR). AMI has been widely used by the community for diarization purposes. Still, somewhat surprisingly, no standard evaluation protocol is used in works that report results on this dataset.

The description of the different evaluation protocols reported in the literature as well as our proposed new protocol for evaluation on AMI can be found in Section 4.

3.3.3. DIHARD II

DIHARD II is one of the newest datasets designed for diarization. This dataset was created as an extension of the first DIHARD dataset for the second DIHARD challenge (Ryant et al., 2019), the second of a yearly series of challenges designed to foster research on diarization in hard conditions. One of the main features of this dataset, is that it contains audios from several sources (YouTube, court rooms, meetings, etc.) covering a wide range of numbers of speakers per recording (1 to 10) and large variety of channels and audio conditions. The corpus consists of 192 development and 194 evaluation recordings, containing around 18 h and 17 h of speech, respectively.

---

[7] Note that the inference in BHMM cannot converge to higher number of speakers than what is suggested by the AHC-based initialization.

[8] https://catalog.ldc.upenn.edu/LDC2001S97

[9] Not only the English partition, nor the partition limited to 2 speaker audios sometimes used.

[10] One audio is commonly excluded because its references have formatting errors.

[11] https://github.com/kaldi-asr/kaldi/blob/master/egs/callhome_diarization/v2/run.sh

3.4. Evaluation protocol

Diarization performance is evaluated in terms of diarization error rate (DER) as defined by NIST (NIST, 2002):

$$DER = \frac{SER + FA + Miss}{Total\_speech} \qquad (26)$$

where:

- SER stands for speaker error, the amount of time that speech is attributed to incorrect speakers
- FA is false alarm, the amount of time that non-speech regions are incorrectly attributed to a speaker (or time when overlapped speech is found in single speaker speech regions)
- $Miss$ stands for missed speech, the amount of time that speech is not attributed to any speaker
- Total_speech is the total amount of speech, accounting also for speaker overlaps.

Note that, as we use oracle VAD in all our experiments, FA error is zero, and the $Miss$ (due only to non-handled overlapped speech regions) can be directly calculated as $DER - SER$ as we also report SER. We also evaluate the system in terms of Jaccard error rate (JER), which has been established as a secondary metric in the latest diarization challenges (Ryant et al., 2019; Chung et al., 2020). JER is similar to DER, although it weighs every speaker equally, regardless of the amount of speech they produced. All experiments are evaluated using the dscore tool.[12]

For a more thorough analysis of results, for CALLHOME and AMI we consider three setups for evaluation: First, a *forgiving* one in which a 0.25 s collar is considered for DER estimation and no overlap is evaluated. This is the standard configuration used for these datasets and allows a comparison of results with previously published works. Second, we consider a similar evaluation using a 0.25 s collar but accounting for overlapped speech, as in Chung et al. (2020). This *fair* setup covers a pragmatic scenario where all speech is evaluated, while being flexible on the speaker change points, as no realistic human annotation can achieve frame-precision. Finally, the *full* one, in which no collar is used and overlapped speech is evaluated, which is in line with the setup used in latest diarization challenges (Ryant et al., 2019; Watanabe et al., 2020). We consider that looking into the numbers for the *fair* and *full* setups is truly relevant on these datasets: very low DER values are already being achieved on these sets with the *forgiving* setup, which suggest that future evaluations could shift into more challenging setups. Also, the hot-topic end-to-end approaches are likely to surpass the performance of current systems on overlapped speech regions, and suitable baselines need to be established.

For the more recent DIHARD II dataset, only the *fair* and *full* evaluation setups are considered.

Note that JER considers no collar and evaluates overlap regions by definition, so it is not affected by these configurations. To avoid confusion (and repetition), we only report this value on the *full* setup. It should be noted that if we evaluate our systems with JER, the only coherent setup for the DER is the *full*. From our point of view, the *full* setup is the best choice for the evaluation. Even if human annotations have errors it does not mean that speech that could be evaluated should always be discarded around each speaker change point. The only thing that the annotation errors imply is that diarization systems have a non-zero lower bound on the diarization error. Besides, the *full* setup avoids introducing the collar length parameter that the evaluation metric depends on.

4. AMI evaluation protocols

4.1. Evaluation protocols found in previously published works

As pointed out before, when it comes to evaluation on AMI, there is not a defined evaluation protocol. Different authors evaluate their systems on different audio types (recordings from Mix-Headset microphones, microphone arrays, etc.). Besides, authors use different data partitions (train/dev/eval sets) and use different references. This makes it practically impossible to compare results between sites. We would like to highlight that during our search for baselines in the literature, we found that most works are unaware of this inconsistency of evaluation protocols for AMI, which frequently leads to unfair comparisons.

Based on our literature review we replicated some evaluation protocols from works presenting remarkable performance. We use these protocols to evaluate our approach and fairly compare it with the respective works.

There are two major publicly available recipes for diarization on AMI that are included in Pyannote[13] (Bredin et al., 2020) and Kaldi[14] (Povey et al., 2011) toolkits. These two recipes evaluate only on one audio type, which is the independent headset microphone mixed audio (Mix-Headset in AMI). Each recipe also derives their references in different ways from the official ASR transcriptions. Finally, they both use their own partitioning of the data, Pyannote uses the Full-corpus AMI partition and Kaldi claims to use the official Full-corpus-ASR AMI partition.[15] Nevertheless, the Kaldi partition differs from the Full-corpus-ASR one, as it includes one meeting in the training set (IB4005) which causes speaker overlap between train and dev sets. This meeting is explicitly excluded in the Full-corpus-ASR partition.

All the works that we compare with use different combinations of partitions and references taken mostly from these two recipes. The first six columns in Table 5 summarize this mixture of protocols. Some works use Pyannote partition (Bredin et al., 2020; Bullock

---

et al., 2020), while others use the Kaldi one (Sun et al., 2020; Maciejewski et al., 2018; Raj et al., 2020a,b), or even a modified version of the Kaldi one, excluding TNO meetings (Pal et al., 2020; Sun et al., 2019). As for references, AMI corpus comes only with manual and automatic transcriptions for ASR training and there are different ways of deriving diarization references out of them, as will be described later. In previously published works, we find again references from Pyannote (Bredin et al., 2020; Bullock et al., 2020; Raj et al., 2020a,b) and from Kaldi (Maciejewski et al., 2018), although some works use modified versions of these based on their own ASR forced alignment (Sun et al., 2020), or simply derive their own (Pal et al., 2020). Note also the mix of evaluation setups (better described in Section 3.4) with different collars and criteria for including or excluding overlapped speech. Column 5 in the table shows the amount of speech (in seconds) evaluated with each protocol for dev and eval sets. As it can be seen, evaluation can range from around 13 000 to up to 52 000 s of speech for the eval set depending on the evaluation setup used.

We strongly believe a standard evaluation protocol should be established on AMI. Next, we introduce what we believe should be established as this new standard.

4.2. New AMI evaluation protocol

The following evaluation protocol was built after discussions with researchers from different labs, authors of official Kaldi and Pyannote recipes (BUT, CLSP JHU, IRIT).

We propose to use the official Full-corpus-ASR partition[14]. This way, we make the scoring of diarization tasks consistent with the scoring of speaker-attributed ASR. As mentioned before, such partition is very similar to the Kaldi partition (in fact, dev and eval sets are the same) but it has no speaker overlap between sets, which makes it suitable for diarization tasks.

Regarding the references, authors of other works do not make clear how their references were created. Our diarization references are directly derived from the AMI manual annotations, version 1.6.2.[16] These annotations are human transcriptions of all the meetings, containing words, vocal sounds and punctuation marks. To generate the references:

- All words are considered as speech and included in the references.
- Sounds of very different nature were annotated as vocal sounds. Some examples are Dutch speech, whistling, yawn, laughter, cough, clicking with tongue, raspberry noise, blowing nose, clapping, etc. Some of these are clearly speech[17] (Dutch speech), while others are clearly noises (blowing nose, clapping, etc.). For sounds such as laughter or whistling it is simply unclear if they should be considered for diarization purposes, as this would depend on the particular application of the system. Besides, we found that several of these vocal sounds are labeled without time annotations, which makes it impossible to add them to the references. We therefore decided to take a well defined, consistent and conservative approach in which all vocal sounds are discarded. This way, only the words that could be recognized by an ASR system are considered in our references. This is also more consistent with the task of speaker-attributed ASR.
- Speaker turns respect precisely the annotations, but adjacent speech segments (words) of the same speaker are merged not to create false "break" points as shown in Fig. 2.

  If adjacent speech segments from the same speaker are not merged, it truly affects the diarization evaluation when collars are considered, as the collar is applied over all VAD borders and these "break points" are considered as one of these borders. These "break points" between adjacent speech segments are common in the references derived with the Kaldi recipe. With our processing, the above transcription results in one speech segment from 0.86 to 1.40 (merged "I like apples") and another from 1.45 to 2.0 (merged "but not bananas").

  On the other hand, consecutive speech segments from the same speaker separated by pauses (silence) are not merged in any case. Using the above example again, we could think that the pause is too short and maybe it should be discarded and the two segments should be merged into one. But this kind of processing would require some heuristic to determine which is the required pause for considering separate speech segments. We prefer to follow a clean approach keeping the original pauses.

  In the case of using no collar (which we believe is the best choice for evaluation) merging the adjacent speech segments has no effect at all. Still, we establish it to support collar use if desired.

In later experiments, we use the same partition and references to evaluate our system with two audio types: AMI Mix-Headset audios and the beamformed microphone array N1, where BeamformIt (Anguera et al., 2007) is applied using the specific setup provided for AMI.

The partition, references and audios are shared in our repository.[18] This evaluation protocol will also be adopted in the latest Kaldi and pyannote recipes. Additionally, we also generated an extra version of the references including all (time-labeled) vocal sounds. As mentioned before, we consider these references not well defined, but we understand that some researchers might find them useful.

---

[16] http://groups.inf.ed.ac.uk/ami/download/
[17] Nevertheless, these controversial cases account for a negligible amount of speech with regard to the total duration of AMI dataset.
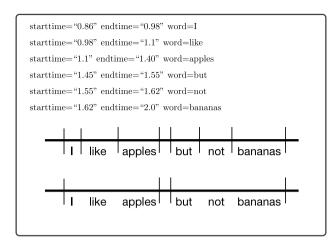[18] https://github.com/BUTSpeechFIT/AMI-diarization-setup

Fig. 2. Example of an ASR transcription with speech from a single speaker which, as the AMI one, is composed of short segments of speech for each word.

Table 4
Diarization performance on CALLHOME.

Evaluation setup			System	SER	DER	JER
Name	Collar	Overlap				
Forgiving	0.25	No	Kaldi (Sell et al., 2018)	6.48		–
			Zhang et al. (2019)	7.60		–
			McCree et al. (2019)	7.10		–
			Lin et al. (2019)	6.63		–
			Pal et al. (2020)	6.76		–
			Aronowitz et al. (2020)	5.10		–
			AHC	8.10		–
			VBx	**4.42**		–
Fair	0.25	Yes	AHC	7.53	17.64	–
			VBx	4.10	14.21	–
Full	0	Yes	AHC	11.06	25.61	35.48
			VBx	7.22	21.77	34.02

5. Results

5.1. CALLHOME

We first present in Table 4 the results of our diarization model on CALLHOME data. As a baseline, we also provide the result of a standalone AHC clustering of x-vectors, where its threshold is tuned for optimal performance. We provide these results to illustrate the gain achieved specifically by VBx diarization, which uses AHC as initialization (see Section 3.2 for details). Our VBx system achieves 4.42% DER on the *forgiving* evaluation setup, outperforming all systems from previously published works. With the *fair* evaluation setup, considering also the overlapped speech (which our system does not handle), performance drops to 14.21% DER. We would like to mention that Horiguchi et al. (2020) has shown great performance on the CALLHOME set while using a system VAD (15.29 DER on a subset of CALLHOME, using *fair* setup). Since our analysis focuses on oracle VAD, we cannot directly compare our results to theirs. Finally, with zero collar (*full* evaluation setup) increases the error to a total 21.77% DER. We believe that future research works should report results using this more challenging evaluation setup.

5.2. AMI

We first report in Table 5 results for our system using the different evaluation protocols found in the literature for AMI corpus. As explained in Section 4.1, the mix of protocols between sites called for running experiments with 3 different data partitions (train/dev/eval sets), 5 sets of references, 2 different types of audio and considering different evaluation setups to be able to compare with all works. We would like to highlight again that the different protocols differ largely in the amount of speech used for evaluation: from only 13 309 s of speech considered in Pal et al. (2020) up to 52 317 s considered in Bredin et al. (2020). When analyzing the results, it can be seen that our VBx system attains the best results on all evaluation protocols, with two exceptions. The first exception is Raj et al. (2020b), which uses Kaldi partition, Pyannote references and *Full* evaluation setup. This work presents a fusion of 3 different diarization systems dealing with overlapped speech, as compared to our single system with no overlap handling. The same

Table 5
Diarization performance of the proposed model compared to baselines from the literature.

Partition	References	Audio type	Evaluation setup			Scored speech dev/eval (s)	System	development		evaluation	
			Name	Collar	Overlap			SER	DER	SER	DER
Pyannote	Pyannote	Mix-Headset	Forgiving	0.25	No	29 200/29 609	Bredin et al. (2020)		–		4.6
							VBx		2.14		2.17
			Full	0	Yes	54 051/52 317	Bredin et al. (2020)	–	–	–	24.8
							Bullock et al. (2020)	–	–	7.2	23.8
							VBx	3.33	22.95	3.86	22.85
Kaldi	Force Aligned	Beamformed mic-array	Forgiving	0.25	No	15 053/14 080	Sun et al. (2020)		16.4		15.4
							VBx		1.32		1.84
			Fair	0.25	Yes	16 241/14 886	Sun et al. (2020)	–	19.4	–	17.8
							VBx	1.26	4.96	1.92	4.67
	Kaldi	Mix-Headset	Forgiving	0.25	No	18 743/18 219	Maciejewski et al. (2018)		–		–/(4.8[a])
							VBx		2.14		3.02/(2.58[a])
	Pyannote	Mix-Headset	Full	0	Yes	35 495/33 953	Raj et al. (2020a)	–	–	10.1	23.6
							Raj et al. (2020b)	–	21.6	–	20.5
							VBx	3.12	22.63	3.56	23.47
			–	0	No	22 812/21 911	Raj et al. (2020b)		7.7		5.2
							VBx		4.08		3.80
Kaldi no TNO	Work specific	Beamformed mic-array	Forgiving	0.25	No	14 545/13 309	Pal et al. (2020)		5.02		4.92
									6.21		2.87
							VBx		4.27		4.58

[a]Denotes dev and eval pooled results.

system evaluated only on non overlap regions obtains 7.7% and 5.2% DER on dev and eval respectively, while VBx obtains only 4.08% on dev and 3.8% on eval. The second exception is the eval result from Pal et al. (2020), which uses Kaldi "no TNO" partition. While this system has significantly better performance than our system (only) on the eval set (2.87 vs. 4.58 DER), when analyzing the results from the paper, it seems to us that this number is an outlier inconsistent with the other results presented in the work: all systems presented in the paper have consistently similar performance on dev and eval sets, this particular system reduces the error more than a 50% (6.21% DER on dev vs 2.87% on eval). Also, this would not be the system of choice if selected according to the performance on the dev set, for which the system was tuned (also shown in the table, with 5.02% DER on the dev set and 4.92% DER on eval). Out of all the partitions used, this is the one with the smallest eval set, which might result in noisier results.

In Table 5, we have demonstrated that VBx method has superior performance as compared to other published works on AMI dataset. However, to offer a fair comparison, we had to deal with too many protocols. This led us to the proposal of the new evaluation protocol, described in Section 4.2 which is, already being adopted by other research labs. In Table 6, we report results obtained with the VBx system with the proposed evaluation protocol. This results can serve as a reference for future works on this corpus. Once again, results are also provided for the baseline standalone AHC when it is tuned for optimal performance (see Section 3.2 for details). For the sake of completeness, we provide results of our system when evaluated on beamformed mic-array audios as well as on Mix-Headset audios. The system is evaluated on all evaluation setups presented in 3.4. Performance on Beamformed audios gets as low as 3.9% DER on the eval set when using the forgiving evaluation. When considering overlapped speech, DER increases to 14.23%. Finally, without any collar, the system achieves 20.84% DER and 26.92% JER. For the Mix-Headset audios, results are consistently better, as expected, achieving 2.10%, 12.53% and 18.99% DER for forgiving, fair and full evaluation setups, respectively.

5.3. DIHARD II

Diarization results obtained on the DIHARD II dataset are presented in Table 7. Our VBx system obtains 18.19% DER on the development set and 18.55% DER on the evaluation set. When analyzing the table, it can be seen that, in fact, these are not the best numbers ever published on DIHARD II eval as the system does not overcome the results from Lin et al. (2020b), nor the ones we attained on the challenge (Landini et al., 2020) which are, as far as we know, still the best in the literature. However, note that the best results on DIHARD II (Landini et al., 2020) are obtained with the same VBx method as described in this paper, but including additional adaptation to the DIHARD data and additional steps (overlapped speech handling and resegmentation). Similarly, in Lin et al. (2020b) the diarization system is adapted to the DIHARD dev data. As mentioned in Section 3.2, in this paper we are aiming to use a generic approach to all datasets, without using the development sets for training or adaptation purposes. If we compare our system to the results of Landini et al. (2020) and Lin et al. (2020b) when not performing adaptation on the dev set, which are shown in brackets, our current VBx outperforms both systems.

When evaluating the system with the *fair* setup, DER improves around a 30% on both sets, resulting in 11.75% and 12.41% on dev and eval, respectively.

Table 6
Diarization performance of the proposed model on AMI with the proposed AMI protocol.

Audio type	Evaluation setup			System	development			evaluation		
	Name	Collar	Overlap		SER	DER	JER	SER	DER	JER
Beamformed	*Forgiving*	0.25	No	AHC	6.32	–		7.65	–	
				VBx	2.80	–		3.90	–	
	Fair	0.25	Yes	AHC	6.43	14.68	–	8.82	18.36	–
				VBx	2.57	10.81	–	4.69	14.23	–
	Full	0	Yes	AHC	8.68	22.14	25.29	10.93	25.48	29.85
				VBx	4.20	17.66	22.26	6.28	20.84	26.92
Mix-Headset	*Forgiving*	0.25	No	AHC	3.90	–		3.96	–	
				VBx	1.56	–		2.10	–	
	Fair	0.25	Yes	AHC	4.06	12.31	–	5.05	14.60	–
				VBx	1.43	9.68	–	2.98	12.53	–
	Full	0	Yes	AHC	6.16	19.61	23.90	6.87	21.43	25.50
				VBx	2.88	16.33	20.57	4.43	18.99	24.57

Table 7
Diarization performance of the proposed model on DIHARD II.

Evaluation setup			System	development			evaluation		
Name	Collar	Overlap		SER	DER	JER	SER	DER	JER
Full	0	Yes	Landini et al. (2020)[a]	–	17.90 (18.34)	–	–	18.21 (19.14)	–
			Lin et al. (2020a)	–	21.36	–	–	18.84	–
			Lin et al. (2020b)	–	18.76	–	–	18.44 (19.46)	–
			AHC	10.89	21.68	42.28	13.89	23.59	43.93
			VBx	7.41	18.19	42.53	8.85	18.55	43.91
Fair	0.25	Yes	AHC	8.22	14.91	–	10.94	16.67	–
			VBx	5.53	12.23	–	6.55	12.29	–

[a]Denotes our own previous results. Results in brackets were obtained when not using the dev set for training nor adaptation purposes, thus they are comparable to our results.

It is to highlight that, while VBx significantly outperforms AHC in terms of DER, both systems obtain similar JER. In general DER and JER correlate, but not always. We know that for JER it is important not to miss speakers with little speech. Therefore, if the diarization system is tuned to generate more speaker clusters, JER can benefit from it. Hence, tuning for the best DER (as we did in the paper) does not always lead to optimal JER.

6. Conclusion

This paper presents a diarization system based on a Bayesian HMM model for clustering x-vectors, also known as VBx. Our VBx diarization achieves state-of-the-art results on CALLHOME, AMI and DIHARD II without performing specific model adaptation to any of the datasets.

Most of the papers with which we have compared our system present clustering of new sets of embeddings/x-vectors. All these approaches are complementary with VBx: the VBx clustering can be combined with most of these new embeddings, which shows the further potential of the method.

In the case of AMI dataset, we have presented, for the first time, a fully fair comparison of our system with several works from the literature. One of the major contributions of the paper is the proposed evaluation protocol for AMI, which will be adopted also in future Kaldi and Pyannote recipes, and which we hope will become a new standard.

The analysis of results for CALLHOME and AMI datasets reveals that systems are reaching very low diarization error rates when evaluating with the standard 0.25 s collar and without considering overlapped speech regions. We believe that, as in latest diarization challenges, systems should be tested using more challenging evaluation setups considering also overlapped speech and no collar.

Future work will focus on combining our system with real VAD instead of the oracle labels, to approach the setup of nowadays largely popular end-to-end techniques.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgments

References

Anguera, X., Wooters, C., Hernando, J., 2007. Acoustic beamforming for speaker diarization of meetings. Audio Speech Lang. Process. IEEE Trans. 15, 2011–2022. http://dx.doi.org/10.1109/TASL.2007.902460.

Aronowitz, H., Zhu, W., Suzuki, M., Kurata, G., Hoory, R., 2020. New advances in speaker diarization. In: Interspeech 2020, 21st Annual Conference of the International Speech Communication Association, Virtual Event. Shanghai, China, 25–29 October 2020, ISCA, pp. 279–283. http://dx.doi.org/10.21437/Interspeech.2020-1879.

Beal, M., 2003. Variational algorithms for approximate Bayesian inference. (Ph.D. thesis). Gatsby Unit, University College London.

Bishop, C.M., 2006. Pattern Recognition and Machine Learning. Springer-Verlag New York, Inc., Secaucus, NJ, USA.

Bredin, H., Yin, R., Coria, J.M., Gelly, G., Korshunov, P., Lavechin, M., Fustes, D., Titeux, H., Bouaziz, W., Gill, M.-P., 2020. pyannote.audio: neural building blocks for speaker diarization. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2020, Barcelona, Spain.

Brummer, N., Villiers, E., 2010. The speaker partitioning problem. In: Proc. of Odyssey 2010.

Bullock, L., Bredin, H., Garcia-Perera, L.P., 2020. Overlap-aware diarization: resegmentation using neural end-to-end overlapped speech detection. In: IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP 2020, Barcelona, Spain.

Carletta, J., Ashby, S., Bourban, S., Flynn, M., Guillemot, M., Hain, T., Kadlec, J., Karaiskos, V., Kraaij, W., Kronenthal, M., et al., 2006. The AMI meeting corpus: A pre-announcement. In: International Workshop on Machine Learning for Multimodal Interaction. Springer, pp. 28–39.

Chung, J.S., Huh, J., Nagrani, A., Afouras, T., Zisserman, A., 2020. Spot the conversation: speaker diarisation in the wild. In: Interspeech 2020. ISCA, http://dx.doi.org/10.21437/Interspeech.2020-2337.

Chung, J.S., Nagrani, A., Zisserman, A., 2018. VoxCeleb2: Deep speaker recognition. In: INTERSPEECH.

Dehak, N., et al., 2011. Front-end factor analysis for speaker verification. IEEE Trans. Audio Speech Lang. Process. 19 (4), 788–798. http://dx.doi.org/10.1109/TASL.2010.2064307.

Deng, J., Guo, J., Xue, N., Zafeiriou, S., 2019. Arcface: Additive angular margin loss for deep face recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 4690–4699.

Diez, M., Burget, L., 2021. CSL VBx Derivations. Technical Report, Brno University of Technology, http://www.fit.vutbr.cz/~mireia/CSL_VBHMM_tech_report.pdf.

Diez, M., Burget, L., Landini, F., Černocký, J., 2020. Analysis of speaker diarization based on Bayesian HMM with eigenvoice priors. IEEE/ACM Trans. Audio Speech Lang. Process. 28, 355–368.

Diez, M., Burget, L., Landini, F., Wang, S., Černocký, H., 2020. Optimizing Bayesian HMM based x-vector clustering for the second DIHARD speech diarization challenge. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 6519–6523.

Diez, M., s Burget, L., Wang, S., Rohdin, J., Černocký, H., 2019. Bayesian HMM Based x-vector clustering for speaker diarization. In: Proc. Interspeech 2019. pp. 346–350. http://dx.doi.org/10.21437/Interspeech.2019-2813.

Fan, Y., Kang, J., Li, L., Li, K., Chen, H., Cheng, S., Zhang, P., Zhou, Z., Cai, Y., Wang, D., 2020. CN-CELEB: a challenging chinese speaker recognition dataset. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 7604–7608.

Fathullah, Y., Zhang, C., Woodland, P., 2020. Improved large-margin softmax loss for speaker diarisation. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, pp. 7104–7108. http://dx.doi.org/10.1109/ICASSP40776.2020.9053373.

Fox, E., B. Sudderth, E., Jordan, M., S. Willsky, A., 2007. The Sticky HDP-Hmm: bayesian Nonparametric Hidden Markov Models with Persistent States. Technical Report P-2777, MIT LIDS.

Fujita, Y., Watanabe, S., Horiguchi, S., Xue, Y., Nagamatsu, K., 2020. End-to-end neural diarization: Reformulating speaker diarization as simple multi-label classification. arXiv:2003.02966.

Garcia-Romero, D., Espy-Wilson, C., 2011. Analysis of i-vector Length Normalization in Speaker Recognition Systems. In: Proc. Interspeech, pp. 249–252.

Garcia-Romero, D., et al., 2017. Speaker diarization using deep neural network embeddings. In: IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, pp. 4930–4934. http://dx.doi.org/10.1109/ICASSP.2017.7953094.

He, K., Zhang, X., Ren, S., Sun, J., 2016. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pp. 770–778.

Horiguchi, S., Fujita, Y., Watanabe, S., Xue, Y., Nagamatsu, K., 2020. End-to-end speaker diarization for an unknown number of speakers with encoder-decoder based attractors. In: Interspeech 2020. ISCA.

Huang, Z., Watanabe, S., Fujita, Y., García, P., Shao, Y., Povey, D., Khudanpur, S., 2020. Speaker diarization with region proposal network. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 6514–6518.

Kaldi, 2017. SRE16 v2. https://github.com/kaldi-asr/kaldi/tree/master/egs/sre16/v2 [Downloaded: 2017-12].

Kenny, P., 2008. Bayesian Analysis of Speaker Diarization with Eigenvoice Priors. Tech. Rep., CRIM, Montreal.

Kenny, P., 2010. Bayesian speaker verification with heavy-tailed priors. In: Proc. Odyssey-10, Brno, Czech Republic.

Kinoshita, K., Delcroix, M., Tawara, N., 2020. Integrating end-to-end neural and clustering-based diarization: Getting the best of both worlds. arXiv:2010.13366.

Landini, F., Wang, S., Diez, M., Burget, L., Matějka, P., Žmolíková, K., Mošner, L., Silnova, A., Plchot, O., Novotný, O., et al., 2020. BUT system for the second DIHARD speech diarization challenge. In: ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 6529–6533.

Lin, Q., Cai, W., Yang, L., Wang, J., Zhang, J., Li, M., 2020a. DIHARD II Is still hard: Experimental results and discussions from the DKU-LENOVO team. In: Proc. Odyssey 2020 the Speaker and Language Recognition Workshop. pp. 102–109. http://dx.doi.org/10.21437/Odyssey.2020-15, http://dx.doi.org/10.21437/Odyssey.2020-15.

Lin, Q., Hou, Y., Li, M., 2020b. Self-attentive similarity measurement strategies in speaker diarization. In: Proc. Interspeech 2020. pp. 284–288. http://dx.doi.org/10.21437/Interspeech.2020-1908, http://dx.doi.org/10.21437/Interspeech.2020-1908.

Lin, Q., Yin, R., Li, M., Bredin, H., Barras, C., 2019. LSTM based similarity measurement with spectral clustering for speaker diarization.

Maciejewski, M., Snyder, D., Manohar, V., Dehak, N., Khudanpur, S., 2018. Characterizing performance of speaker diarization systems on far-field speech using standard methods. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, pp. 5244–5248. http://dx.doi.org/10.1109/ICASSP.2018.8461546.

Martin, A.F., Przybocki, M.A., 2001. Stream-based speaker segmentation using speaker factors and eigenvoices. In: 7th European Conference on Speech Communication and Technology, Eurospeech, Vol. 7, No. 2, pp. 787–790.

McCree, A., Sell, G., Garcia-Romero, D., Speaker diarization using leave-one-out Gaussian PLDA clustering of DNN embeddings. In: Interspeech, pp. 381–385.

Medennikov, I., Korenevsky, M., Prisyach, T., Khokhlov, Y., Korenevskaya, M., Sorokin, I., Timofeeva, T., Mitrofanov, A., Andrusenko, A., Podluzhny, I., et al., 2020. Target-speaker voice activity detection: A novel approach for multi-speaker diarization in a dinner party scenario. In: Interspeech 2020. ISCA, http://dx.doi.org/10.21437/interspeech.2020-1602.

Nagrani, A., Chung, J.S., Zisserman, A., 2017. VoxCeleb: a large-scale speaker identification dataset. In: INTERSPEECH.

NIST, 2000. NIST SRE 2000 Evaluation Plan. https://www.nist.gov/sites/default/files/documents/2017/09/26/spk-2000-plan-v1.0.htm_.pdf.

NIST, 2002. NIST Rich Transcription Evaluations. https://www.nist.gov/itl/iad/mig/rich-transcription-evaluation, version: md-eval-v22.pl.

Pal, M., Kumar, M., Peri, R., Park, T.J., Kim, S.H., Lord, C., Bishop, S., Narayanan, S., 2020. Meta-learning with latent space clustering in generative adversarial network for speaker diarization. arXiv:2007.09635.

Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., et al., 2019. Pytorch: An imperative style, high-performance deep learning library. In: Advances in Neural Information Processing Systems. pp. 8026–8037.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., et al., 2011. The kaldi speech recognition toolkit. In: IEEE 2011 Workshop on Automatic Speech Recognition and Understanding. IEEE Signal Processing Society.

Raj, D., Garcia-Perera, L.P., Huang, Z., Watanabe, S., Povey, D., Stolcke, A., Khudanpur, S., 2020b. DOVER-lap: A method for combining overlap-aware diarization outputs.

Raj, D., Huang, Z., Khudanpur, S., 2020a. Multi-class spectral clustering with overlaps for speaker diarization. arXiv:2011.02900.

Ryant, N., et al., 2018. DIHARD Corpus. Linguistic data consortium.

Ryant, N., et al., 2019. The Second DIHARD Diarization Challenge: Dataset, task, and baselines. In: Proceedings of Interspeech.

Sell, G., et al., 2018. Diarization is hard: Some experiences and lessons learned for the JHU team in the inaugural DIHARD challenge. In: Proc. Interspeech. pp. 2808–2812. http://dx.doi.org/10.21437/Interspeech.2018-1893, URL http://www.danielpovey.com/files/2018_interspeech_dihard.pdf.

Senoussaoui, M., Kenny, P., Stafylakis, T., Dumouchel, P., 2014. A study of the cosine distance-based mean shift for telephone speech diarization. IEEE/ACM Trans. Audio Speech Lang. Process. 22 (1), 217–227. http://dx.doi.org/10.1109/TASLP.2013.2285474.

Shum, S.H., Dehak, N., Dehak, R., Glass, J.R., 2013. Unsupervised methods for speaker diarization: An integrated and iterative approach. IEEE Trans. Audio Speech Lang. Process. 21 (10), 2015–2028. http://dx.doi.org/10.1109/TASL.2013.2264673.

Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., Khudanpur, S., 2018. X-vectors: Robust DNN embeddings for speaker recognition. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 5329–5333.

Sun, G., Zhang, C., Woodland, P.C., 2019. Speaker diarisation using 2D self-attentive combination of embeddings. CoRR abs/1902.03190. arXiv:1902.03190. URL http://arxiv.org/abs/1902.03190.

Sun, G., Zhang, C., Woodland, P., 2020. Combination of deep speaker embeddings for diarisation. arXiv:2010.12025.

Watanabe, S., Mandel, M., Barker, J., Vincent, E., Arora, A., Chang, X., Khudanpur, S., Manohar, V., Povey, D., Raj, D., Snyder, D., Subramanian, A.S., Trmal, J., Yair, B.B., Boeddeker, C., Ni, Z., Fujita, Y., Horiguchi, S., Kanda, N., Yoshioka, T., Ryant, N., 2020. CHiME-6 challenge: Tackling multispeaker speech recognition for unsegmented recordings. arXiv:2004.09249.

Xiao, X., Kanda, N., Chen, Z., Zhou, T., Yoshioka, T., Chen, S., Zhao, Y., Liu, G., Wu, Y., Wu, J., Liu, S., Li, J., Gong, Y., 2020. Microsoft speaker diarization system for the voxceleb speaker recognition challenge 2020. arXiv:2010.11458.

Zeinali, H., Sameti, H., Stafylakis, T., 2018. DeepMine speech processing database: Text-dependent and independent speaker verification and speech recognition in Persian and English. In: Odyssey, pp. 386–392.

Zeinali, H., Wang, S., Silnova, A., Matějka, P., Plchot, O., 2019. BUT system description to voxceleb speaker recognition challenge 2019. arXiv preprint arXiv:1910.12592.

Zhang, A., Wang, Q., Zhu, Z., Paisley, J., Wang, C., 2019. Fully supervised speaker diarization. In: ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 6301–6305.