Review

Speaker Diarization: A Review of Objectives and Methods

Douglas O'Shaughnessy *

Posted Date: 27 December 2024

doi: 10.20944/preprints202412.2368.v1

Keywords: speech analysis; spectral continuity; Gaussian mixture models; neural networks; intonation; Mel spectrograms

 doi:10.20944/preprints202412.2368.v1

Review

Speaker Diarization: A Review of Objectives and Methods

Douglas O'Shaughnessy

Institut National de la Recherche Scientifique, Canada; douglas.oshaughnessy@inrs.ca

Abstract: Recorded audio often contains speech from multiple people in conversation. It is useful to label such signals with speaker turns, noting when each speaker is talking and identifying each speaker. This paper discusses how to process speech signals to do such speaker diarization (SD). We examine the nature of speech signals, to identify the possible acoustical features that could assist this clustering task. Traditional speech analysis techniques are reviewed, as well as measures of spectral similarity and clustering. Speech activity detection requires separating speech from background noise in general audio signals. SD may use stochastic models (hidden Markov and Gaussian mixture) and embeddings such as x-vectors. Modern neural machine learning methods are examined in detail. Suggestions are made for future improvements.

Keywords: speech analysis; spectral continuity; Gaussian mixture models; neural networks; intonation; Mel spectrograms

---

1. Introduction

Processing of speech signals has had widespread application in recent years: *automatic speech recognition* (ASR - translating spoken audio into text), *speaker verification* (SV - identifying who is talking), language identification (which language is spoken), emotion recognition, and coding (e.g., *linear predictive coding* (LPC) in cellular telephony) are the most common uses. Given the prominence of multi-speaker conversations in podcasts, broadcasts, entertainment, and meetings, another valuable speech application is *Speaker Diarization* (SD), where a computer algorithm automatically estimates "who spoke when?" from a recorded audio signal [1]. As many recent papers have noted, SD still needs much research [2,3], as performance remains far from ideal.

One application for SD is to annotate a transcription derived from ASR, which would normally output an estimated text based on input speech, but one would not know who had said this text. Another motivation is to assist the ASR process, to allow application of specific speaker-dependent models to input speech for each portion labeled as spoken by a given person. SD can provide speaker identities for conversational artificial intelligence and help to retrieve audio/video recordings from desired target speakers. Applications for SD also include generating meeting/interview transcripts, medical notes, automated subtitling and dubbing, and downstream speaker analytics [4].

The SD field may include rich transcription, textual content retrieval, and indexing from speech; these applications utilize aspects beyond the scope of basic SD, e.g., determining syntactic boundaries and disfluencies (speech errors such as restarts and hesitations). Speech also often contains so-called *vocalizations* (e.g., back-channels, filled pauses, singing, infant babbling, laughter, coughs, breaths, lip-smacks, and humming) that are not directly relevant for ASR. This paper will concentrate on acoustic analysis of speech, rather than on textual matters, and we include vocalizations as forms of speech. Incorporating textual information into SD would render it dependent on language, and general SD, as discussed here, can be developed in a language-independent fashion.

Unlike most other speech applications, where objectives are usually straightforward, the task of SD can be phrased in different ways, depending on a-priori knowledge of who may be participating in a multi-speaker audio signal. In comparison to SD, the objective for speech coding is to produce

 doi:10.20944/preprints202412.2368.v1

a reconstructed audio signal that is natural and intelligible with low transmission rate and latency. For ASR, the aim is the text intended by a speaker. For speaker, language, or emotion recognition, the desired output is simply the respective identity. However, for SD, the difficulty of the diarization task is greatly influenced by the turn-taking behavior of speakers, in particular, how much people's speech is separated by non-speech or overlap in their speech [5].

Given an audio signal that contains speech, SD estimates an ordered set of times, called *time stamps,* that correspond to places in the audio each time that a speaker starts and stops talking. In addition, SD labels each time portion (*speaker turn*) by who is talking. For some applications, e.g., closed meetings, all participants are known to the system via prior training. However, when applying SD to arbitrary audio or broadcasts, the number and identities of speakers are often unknown. In these latter cases, SD still outputs time stamps, but notes speech portions (i.e., between each pair of successive time stamps) with labels of Speaker 1, 2, 3, etc (including portions with multiple speakers).

Audio often has overlapping speech, where two or more speakers talk simultaneously [6,7]. Indeed, in the so-called *cocktail party* situation [8–10], a microphone may record the speech of many people talking at the same time. Applying SD to such cases may not seem useful, but annotating ordinary conversations is of interest, where people often speak over each other. Usually, such interruptions, which result in portions of audio with multiple speakers, are brief, e.g., less than a few seconds and with only two simultaneous speakers (i.e., the original speaker usually ceases talking once the interruption is noticed). Accurate SD would note all such cases. The SD task is similar to other sorts of auditory grouping, such as auditory scene analysis [11].

This paper reviews relevant ideas for SD, including a description of the nature of speech, methods to process (analyze) speech, how to estimate the times of speaker change, and methods to identify speakers. Major focus is on modern machine learning methods, which now dominate most applications of artificial intelligence, but earlier techniques are also discussed. The paper will not examine the related problem of *continuous speech separation*, which isolates a set of non-overlapped speech signals from an audio stream that contains multiple utterances that are partially overlapped [12,13]. As this paper concentrates on classification (time stamps and speaker identities), it will not address peripheral tasks such as speech enhancement (de-noising and de-reverberation) [14], nor discuss joint SD and ASR [15,16].

The work updates earlier reviews of SD [17–19] that were written prior to the recent wave of neural systems. It is similar to a recent detailed review [20], but is more aimed toward explanations of methods and description of speaker aspects to exploit. We assume one microphone as source, as multi-microphone situations typically use beam-forming, which is beyond the scope of this paper [21,22].

2. Description of the Acoustics of Speech Signals

In pattern recognition applications, it is not always necessary to analyze input data before doing formal classification; some systems process input data directly, based on models derived from automatic training using suitable examples. Indeed, much of modern machine learning takes this latter approach. However, given the cost and complexity of such methods and large models, it is useful to note characteristics of objects to recognize, in efforts to simplify classification methods. Thus, we discuss here well-understood features of speech signals. These may help estimate points in time where audio shifts from one speaker to another, and help identify these speakers.

Natural speech is generated by speaker forcing air from the lungs through the passages between glottis and mouth, called a *vocal tract* (VT). There are two major classes of speech sounds: voiced and unvoiced. *Voiced* sounds derive from vibrations of the speaker's vocal cords, which create quasi-periodic puffs of air that excite the VT, which acts as a filter to shape the spectrum of the sound. Waves of air pressure exiting one's VT appear as repeated *pitch periods*, but these deviate slightly from true periodicity, as muscles do not move with exact repetition. *Unvoiced* speech, on the other hand, is random noise generated by passing air through a constricted portion of the VT. The vast

 doi:10.20944/preprints202412.2368.v1

preponderance of sounds in the world's languages fit into these two voiced and unvoiced classes (some languages also have clicking and sucking sounds, but these are far less common).

2.1. Aspects of Speech to Help Estimate Time Stamps

In terms of the temporal waveform of speech, the signal consists mostly of voiced sounds, which appear as patterns that roughly repeat with every closure of the vocal cords; this abrupt action generates a wide spectral band of energy that acts as a rough "impulse" to excite the VT filter. As the VT has a mostly tubular structure that is much more narrow (e.g., 1-2 cm diameter) than its length (average of 17 cm for men), the spectrum of the VT output is that of a series of resonances. For a 17-cm VT, these resonances, called *formants,* are spaced roughly at averages of 500, 1500, 2500, 3500, .. Hz. These values derive directly from the speed of sound (about 340 m/s), in a quarter-wavelength resonator, which is an approximate model of a VT closed at the glottis and open at the lips [23]. In strong voiced sounds called *sonorants*, which dominate most of speech, the intense lower formants slowly rise and fall with VT shape changes. These resonance movement contours, which last from 60 ms up to a few seconds (mostly 100-300 ms), are distinctive for both ASR and SD. However, few modern classification algorithms exploit such analysis, as attempts to track formants was largely discarded from consideration decades ago (with some exceptions: [24]).

Voiced non-noisy sections of speech, thus, have distinctive, regular patterns for individual speakers that can be exploited when judging continuity in audio signals. Both resonances and the vocal cord vibration rate (the latter called the *fundamental frequency*, F0) change relatively slowly, as speakers utter a sequence of linguistic units called *phonemes,* which are the constituents of words. Typically, phonemes average about 80 ms in duration, while F0 in men averages about 100 Hz. (F0 depends on vocal cord size, and can be up to 1000 Hz in small infants.) Thus, a typical voiced phoneme has about eight pitch periods, which slowly vary as F0 changes and as the VT moves from phoneme to phoneme (motion called *coarticulation*).

For SD, abrupt audio changes are often relevant. Except when the VT excitation changes between voiced and unvoiced (which occurs, on average, 0-2 times per second), the speech spectrum is often fairly stable, with incremental changes from period to period. The other source of abrupt spectral change in speech occurs for nasal consonants, e.g., when the velum lowers and there is a complete closure in the oral portion of the VT (or the reverse change at the end of such consonants). These abrupt changes of speech within an individual's voice many not occur for long stretches of speech (e.g., in a famous example used in testing speech coders decades ago: "May we all learn a yellow lion roar"). On the other hand, some unusual utterances (e.g., rapidly repeating a syllable such as /ta/) could have several abrupt changes per second.

In conversations having multiple speakers, each person would have a pattern of harmonics (spaced at multiples of F0, which changes slowly), modulated by formants that also mostly change slowly, characterized by their VT shape. One could exploit the resulting general continuity of both F0 and formants (within one speaker's voice) to help track individual voices [25]. A commonly used (older) alternative SD approach is to estimate sets of speech features every 10 ms (at a very common 100 Hz *frame* rate), and then cluster frames with related features into groups of frames that potentially represent a speaker turn. We discuss the advantages and disadvantages of these different approaches below.

2.2. Speech Aspects to Help Distinguish Among Speakers

Discriminating among people by their speech (i.e., SV) could exploit both their physical and behavioral aspects. People have varied VTs (physical) and use them differently (behavioral). Much of SV has preferred to focus on analysis of the physical via use of average VT spectra, for reasons of simplicity as well as the fact that impostors can imitate dynamic VT movements more easily than inherent VT shapes. However, *spoofing* (copying or synthesizing speech to claim false identity) appears to be a much rarer risk for SD than for SV, as SD is almost always applied to reliable data

 doi:10.20944/preprints202412.2368.v1

(e.g., there is much more motivation to falsify SV than SD). Thus, SD should be able to apply physical differences in VTs as well as how they are used.

Average resonance frequencies vary indirectly with a speaker's VT length, and average F0 varies indirectly with one's vocal cord size. However, these only provide two scalar values, which are far less than needed to distinguish among large numbers of speakers. Such features can help to discriminate among speakers, but these provide nowhere near the biometric precision of fingerprints or retinal scans.

Modern SV and SD use average speaker representations based on Mel spectra, which largely ignore speech dynamics, and also obscure useful F0 effects. Yet, speakers demonstrate significant variability in F0 and in the timing of VT movements, which directly affect resonance dynamics.

It is possible also that the phase of speech may be speaker-specific, but SV and SD research has yet to demonstrate that. Anecdotal evidence for speaker-specific phase could be that people can often be distinguished by a single short sung vowel, which shows little dynamic behavior, and whose spectral amplitude is well modelled with as few as ten LPC parameters. Such few modelling parameters that amplitude spectra provide are insufficient to distinguish among many speakers.

3. Analysis of Speech Signals

Applications for audio signals obtain useful information from the input by use of some form of analysis or processing. The focus for analysis of speech is generally both F0 and the amplitude spectral envelope of the VT transfer function. This can be a version of parameters of formants; these are efficiently described by their center-frequencies and bandwidths. Speech input, as continuous air pressure waves exiting from a VT, is converted into a digital sequence by analog-to-digital conversion, to allow processing by digital devices. Input speech signals are sampled periodically at an appropriate rate (i.e., exceeding the *Nyquist rate*, twice the signal's highest frequency), in order to preserve information over a frequency range that is relevant for an application (e.g., 200-3200 Hz for telephony speech) [26].

Despite the wide range of objectives that various speech applications have, speech analysis has used similar methods to convert high-dimensional speech data (e.g., 64 kilobits/s for telephony speech: 8000 samples/s at 8 bits/sample) to a lower-dimensional representation (e.g., tens of spectral parameters every 10 ms). This generality applies to all of: ASR, SV, language identification, emotion recognition, and SD.

Some recent neural-network SD systems (called *end-to-end:* E2E) avoid direct speech analysis on the assumption that any processing that is not directly linked to the final task objective may possibly discard relevant information. However, as much of SD is not E2E and uses some analysis, we discuss analysis methods below. As SD relies both on local cues (e.g., at speaker changes) and long-range comparisons of speaker characteristics, considerations of analysis should examine both long- and short-range scopes.

3.1. Basic Spectral Estimates of Speech Signals

To simplify processing, the time waveform of speech may be used directly as input to SD, although the utility of temporal speech samples is limited. There is a high level of encoding in speech; thus, interpretation from individual samples is useful mostly for basic energy and periodicity only. Speech temporal waveforms depend greatly on spectral phase variations, which have little impact on information in the speech message. This has led to most speech classification applications, including SD, to use analysis that is focussed on amplitude spectra.

The *discrete Fourier transform* (DFT) displays energy as a detailed function of frequency [26]. It converts N successive time samples (using a time window that is shifted periodically to handle the non-stationarity of speech) into N spectral samples (N is typically 256-512). The full DFT has too much detail to be useful for most classification problems; thus, further processing (e.g., data reduction and/or pattern extraction) is helpful in most cases.

 doi:10.20944/preprints202412.2368.v1

A common variant of DFT for speech classification applications is *filter bank energies* (FBEs); they smooth DFT amplitudes over limited frequency ranges, as in bandpass filters. To better model human audition, the nonlinear Mel scale is often applied to frequency [27], using filters of equal bandwidths below 1000 Hz and logarithmic spacing above. This scale corresponds to the lower resolution of the human audition at higher frequencies (and corresponds to the physical tapering of the basilar membrane in the inner ear). ASR often uses 20-100 of such filters, which typically reduces data from a 256- or 512-sample DFT, while retaining sufficient spectral information for speech classification.

3.2. Advanced Spectral Measures

A common approach for speech spectral analysis is the *Mel-frequency cepstral coefficients* (MFCCs) [27]. They use similar filtering as BFEs (as well as the Mel scale), but reduce to a smaller set of parameters, typically 13 for telephone speech. This small number is sufficient to discriminate formants to within 100 Hz (which is approximately the smallest amount used by speakers to distinguish among vowels in most languages). For the broad spectral purposes of SD (which need not identify individual sounds), precision of analysis can be lower, thus allowing fewer parameters, if cost reduction is important.

Another common spectral analysis method, *linear predictive coding* (LPC), uses an all-pole model of the speech spectrum [28], to directly focus on the resonances of the VT. (Spectral poles, or intense resonances, are far more important perceptually than other frequency aspects.). Standard LPC is common in cellular speech telephony, but is rarely used for SD, as the other speech measures we discussed suffice for speech continuity decisions for SD and for SV.

3.3. Time Windows

As is evident for the application of SD, speech is non-stationary; i.e., changes with time, as speakers utter a dynamic sequence of phonemes and words. For most speech analysis, the signal is regularly segmented into successive short sections; within each section, one assumes local stationarity [17]. Typical smooth (lowpass) windows will yield a representation that is an average of local events. In the case of rapid transient speech sounds, such as plosives that have acoustic events as brief as 1 ms, normally used windows often smear out phonetic effects.

Durations of speech segments vary greatly in their informational purposes, e.g., sometimes, spectral changes owing to significant VT movements are rapid, while, other times, speakers articulate long vowels with little VT (and, consequently, spectral) movement. As a suitable and simple compromise, most of speech processing calculates measures over brief time windows, updated every 10 ms. Acoustic features are typically calculated over a frame window of approximately 25 ms, shifted uniformly every 10 ms. A de facto standard of 100 sets/second is used in many speech applications, as a compromise that tracks VT coarticulation and minimizes computation.

3.4. Intonation Features

To help listeners make decisions (e.g., for text understanding and identifying who is talking), speakers vary durations and F0 in their speech [25]. Despite this useful intonational information present in speech, most recent speech classification (including SD) does not exploit intonation. Automatic estimation of durations of phonetic sequences is difficult, and often requires (unreliable) ASR. There are many F0 estimation algorithms [29], but integrating F0 into speech classification is difficult, as intonational patterns extend over many speech frames, whereas VT resonance behavior is relevant over shorter durations.

Nonetheless, intonation has been used for SD in the past [30–32]. The range and average value of F0 in speech are both speaker-specific, and F0 contours are generally slowly changing, which help with both speaker identity and continuity. In general, the features (both for resonances and intonation) chosen for use in SD have been statistical (e.g., means, ranges, and standard deviations),

 doi:10.20944/preprints202412.2368.v1

rather than dynamic (e.g., F0 contours over syllables). It may be insufficient for SD to ignore more global, dynamic feature behavior, which humans use a lot, simply because it is inconvenient to integrate into modern SD tools. Much of what is known about human speech production and perception has been ignored in modern neural systems for speech applications, because this information has been difficult to exploit in the traditional frame-based approaches.

4. Use of Speech Features for Speaker Diarization

The speech features described above (primarily MFCCs and FBEs, or in a form called a Mel spectrogram) are used in most speech classification applications. For SD, they serve both for time stamp decisions, based on spectral continuity, and for labelling speech by speaker between each successive pair of time stamps. We now discuss how to do this for localized speaker diarization.

One can distinguish two cases for SD: 1) an input conversation with a small set of subscribed speakers (known to the system via prior training; e.g., a common scenario for meetings), and 2) arbitrary unknown audio with multiple speakers (where the system has no knowledge about speakers). SD systems may assume that the number of speakers is known a priori or is at least bounded.

In the first case, for each segmented speech portion, the SD system chooses the speaker (from among the subscribed set) that is the closest match to the data. If no match is found (e.g., a spectral distance that the system estimates from the input speech for each speaker model exceeds a pre-set threshold), the system may allow an outcome of "no match." In both cases, spectral distance can be measured against a general speaker model called a UBM (Section 5.2). We discuss the case where a portion of speech has multiple speakers later.

Binary classification decisions such as speaker labelling have two types of errors: *false rejections* (FR) and *false acceptances* (FA). FR (also known as a *miss*) occurs when a classifier incorrectly rejects a true choice (or wrongly claims that there is "no match"). FA instead occurs when there is incorrect identification. In such cases, a common evaluation criterion is *equal-error rate*, which balances FA and FR at the same (low) percentage. Many classifiers decide relative to a *threshold*, accepting or rejecting, respectively, when a distance or similarity measure is above/below the threshold. As a result, FAs decrease and FRs increase as a function of thresholds.

5. Models to Apply for Speaker Diarization

The results of speech analysis can be applied in various ways for SD. The classical methods used stochastic models (section 5.1) and simple long-term average models (section 5.2). Recent SV research has led to the embedding approaches of sections 5.3-5.4. The recent surge of neural systems for machine learning is discussed in section 5.5.

5.1. Hidden Markov Models (HMMs) and Gaussian Mixture Models (GMMs)

Between the late 1970s and the early 2010s , the most common methods used for ASR were *Hidden Markov models* (HMMs) [33]. Using a *maximum likelihood* approach, HMMs sought max *Prob(T|S)*, i.e., most likely candidate text *T* for input speech utterance *S*, where *T* is a sequence of words and *S* a temporal sequence of spectral vectors at the frame rate. For efficiency (as *P(T|S)* is highly complex), with Bayes' theorem, one instead maximizes *P(S|T) P(T)*, with *P(T)* being a language model and *P(S|T)* an acoustic model. An HMM can represent a phoneme, word, or longer speech unit. It consists of an ordered sequence of states, each representing a spectral pattern of speech (corresponding to a position of the VT), with timing transitions between states to accommodate variable durations for successive frames. Using off-line training of models with much speech, HMM transitions are assigned probabilities (to model durations) and a probability density function (pdf; e.g., mixture of Gaussians) is estimated for each state's spectrum. While ASR requires frame-based decisions, SV only needs one binary speaker decision for an entire utterance. As a result, SV usually reduces the HMM approach to use only a (single-state) *Gaussian Mixture Model* (GMM).

 doi:10.20944/preprints202412.2368.v1

Gaussians are commonly used, as they need only a mean and variance to specify (their relevance for the central Limit theorem is not the reason for their choice).

As SD needs an identification for each speech portion, HMMs are sometimes used in a post-processing stage, to estimate the most likely sequence of speaker activities; e.g., utilizing subspace models or clustering via Variational Bayes VB-HMM [34]. Other forms of post-processing include *system fusion*, where multiple SD models combine results [5].

5.2. Universal Background Model

The related application to SD of SV typically uses a set of background or *cohort* speakers, to enhance the robustness of the recognizer. Such an average *universal background model* (UBM) represents a typical speaker, against which spectral measures are tested [35]. A common UBM may be a full-covariance or diagonal-covariance GMM with thousands of components, and hundreds of *eigenvoices* (diagonal covariance is commonly used to simplify computation, although that assumes uncorrelated vector elements). Posterior scaling can be applied to discourage frequent speaker transitions in the model [36]. A UBM represents data from all speakers and conditions, and is used to normalize SV scores.

5.3. Embeddings

Labelling each speech turn with an estimated speaker identity requires only a choice among a set of known subscribed speakers (case 1 in section 5), or deciding whether the turn matches a speaker that was identified earlier in the data (case 2). When comparing a representation of a known (trained) speaker model with that of a speaker turn in SD, one could calculate a suitable distance, and compare this result with a threshold. As representation spaces are typically complex for many pattern recognition tasks (including SD), simple distances such as Euclidean may not be ideal, even after suitable normalization of dimensions (e.g., mean, variance, and L2 length-normalized).

A common method today to classify a set of data of variable duration (e.g., a speaker turn) is to: 1) extract a *speaker embedding* (a mapping from data of any duration to a fixed-length vector) with a model [37], and then 2) use a multi-class discriminative training stochastic classifier such as *probabilistic linear discriminant analysis* (PLDA) [38]. Extraction of speaker embeddings is typically d-vectors or x-vectors, which are based on i-vectors (as noted in section 5.4), and derived from inner layers of a neural network trained on speaker classification. Speaker embeddings assume that only one speaker is present in each segment, and thus cannot handle overlapped speech.

5.4. Joint Factor Analysis

Both SD and SV have used some general approaches for high-dimensional pattern recognition, such as *Support Vector Machines* [39], *Joint Factor Analysis* (JFA), and *i-vectors* [40]. To model a speaker, JFA obtains a set of parameters of reduced dimension, called the i-vector, for use as a compact embedding. The means of speaker-dependent GMMs are used to form an aggregate *supervector.* The JFA model may be thought of as Gaussian pdfs on speaker- and session-dependent HMM supervectors, where most variance is exploited by few hidden variables called speaker and channel factors. A speaker supervector consists of speaker-independent, speaker-dependent (*eigenvoice*), channel-dependent (*eigenchannel*), and residual (remaining data) components. Such high-dimensional supervectors have tens of thousands of dimensions, stack first-order statistics for each mixture component, and are projected into a low-dimensional fixed-length representation (200-600 dimensions of the i-vector). The i-vector models a speaker by removing the UBM mean supervector and projecting into a low-dimensional space called a *total variability* space.

 doi:10.20944/preprints202412.2368.v1

5.5. Neural (Machine Learning) Methods

For many diverse fields, applications that process data now use *artificial neural networks* (ANNs) mostly as their main technique, because ANNs can accommodate wide ranges of data [41]. Their non-linear processing allows them to learn highly complex patterns. However, they are very difficult to interpret, and usually require huge numbers of parameters and calculation, and this often exceeds the capacity of low-resource devices (e.g., cell phones). ANNs can either transform data or recognize patterns, using automatic training from examples. An ANN maps an input sequence (e.g., speech waveform samples) non-linearly to an output sequence (e.g., a speaker-labelled set of time stamps).

5.5.1. Basics of ANNs

The fundamental element (called a *node*) of an ANN performs a function similar to the action of biological neurons. The simplest ANN is a *multi-layer perceptron* (MLP), consisting of *layers* (sets) of nodes, each sending their outputs to the nodes of the next successive layer, without feedback. For a classification task such as SD, training a basic ANN may be viewed as optimizing the combined locations of hyperplane boundaries of complex regions in a data representation space. The initial ANN layer receives a vector of speech samples from an utterance. Intermediate layers may provide various features for classification; their numbers of nodes vary greatly and are chosen empirically, and the resulting features are often difficult to evaluate. Reducing their numbers can result in so-called *bottleneck* features [42], which are sometimes interpretable.

In natural neurons, dendrites provide potentials to an axon, which yields a binary output: a brief pulse if the weighted sum of its inputs exceeds a *bias* threshold. For an algorithmic ANN node, output is specified by an *activation function* $y = \varphi(w \cdot x + b)$, where w is a N-dimensional weighting vector, b is a scalar bias value, x is a set of N input values, and $\varphi(.)$ is a scalar nonlinear function. The parameters of each model node (i.e., w and b) specify the location of a hyperplane in a data representation space; the node's binary output corresponds to either side of this hyperplane. To provide accurate solutions to practical data problems, ANNs generally require many millions of parameters to get high performance.

The general *fully-connected feedforward* MLP structure is costly in size and computation, as all nodes in each layer connect to all nodes in the next layer. For speech and many other signals, useful information is spread non-uniformly; thus, a fully-connected network is inefficient. ANNs having more than three layers are called deep (DNNs); these detailed architectures assist to handle complex tasks. Most applications use different combinations of the network components described in sections 5.5.3-5.5.5.

5.5.2. ANN Training: Loss Functions and Steepest Gradient Descent

For training of network parameters, initial estimates of node weights and biases may be specified at random, or pre-trained on unlabeled data. The ANN learning procedure is iterative, incrementally updating the parameters, based on minimizing a *loss* (cost) function. Direct minimization of network accuracy is not feasible, because ANNs need a differentiable loss function, to allow a product chain of derivatives to show which direction to alter parameters, via *steepest gradient descent*, which guides the model towards local loss minima.

Loss functions approximate a penalty for classification errors. Losses are often chosen empirically and vary greatly among applications [43], but a common one is *cross-entropy*, usually with *softmax* output units or its variants (Angular Softmax) [44]. To interpret a probability distribution, Softmax has a normalized exponential function.

All ANNs are trained to minimize a loss function, which is usually chosen to be related to optimizing the task performance. As direct error minimization is often difficult to use (e.g., not differentiable, as needed for gradient descent training), there are many possible losses, which are often chosen based on empirical performance, rather than elements related to the data.

 doi:10.20944/preprints202412.2368.v1

SD typically uses two losses: attractor existence loss and diarization loss. The first loss optimizes the estimation of each speaker's presence (for cases with an unknown number of speakers) and the diarization loss optimizes the active probabilities with a *permutation-invariant t*raining method [45]. The two losses are summed, using a *hyper-parameter* weighting (hyper-parameters are model elements that are often selected empirically a priori). The number of speakers may be estimated with the maximum "eigengap" criterion [46].

5.5.3. Convolutional Neural Networks (CNNs)

In many problems of classification, relevant information is located mostly in local ranges of data (e.g., for speech, in the center frequencies of high-energy formants). To take advantage of this, ANNs often have *convolutional* layers of nodes, which process data locally within small regions. For input of a 2-dimensional representation (e.g., time versus frequency for speech), data are multiplied by a small square weight matrix (e.g., 3x3), and then summed (*pooling*) spatially [47]. One type of CNN applies 1-D convolution (multiplication) in time and is called *time-delay NN* (TDNN); these models have no recurrence (feedback) [48]. *Conformers* have four stacked modules: an initial *feed-forward network* (FFN) module, a multi-head self-attention module (see below), a convolution module, and a second FFN module [49].

5.5.4. Recurrent Neural Networks (RNNs)

To exploit the uneven distribution of information in speech in both time and frequency, one needs a mechanism to be selective for nodes. CNNs exploit local data correlations, but recurrence helps for correlations over longer ranges. With feedback, RNNs have hidden states that store information about past input. Using specialized network gates (input, output, forget), they control the data flow among layers. Examples of these RNNs are *long short-term memory* (LSTM) and *residual network* (ResNet) [50,51].

5.5.5. Attention

The field of machine learning has recently been dominated by a model technique called *attention,* which focuses ANNs to emphasize certain portions of data, as a way to exploit non-uniform distributions of data information. Attention is calculated as a *correlation* that is found in data sequences, using matrix operations that combine *queries* (inputs), *keys* (features), and *values* (desired outputs). Such attention may be applied over time, over frequency, and over network layers [52,53].

Neural systems that use attention are often called *Transformers*. These recent *end-to-end* (E2E) methods do not use recurrence and rely only on attention. In recurrent networks such as LSTM, input is processed sequentially. On the other hand, a Transformer has no timing information, and instead uses positional encodings with a separate embedding table. Transformers typically require more computation than other ANNs. E2E modelling methods (taking data input directly without analysis, and using a unified, simple architecture) for SD have demonstrated superiority in handling overlapped speech compared to conventional pipelines based on speaker embedding clustering. However, E2E systems usually require much carefully annotated training data, which may require up to 10 minutes to manually describe each minute of training data.

5.5.6. Neural Variants of Speaker Embeddings

The activations from final hidden layers in ANNs trained by data from a given speaker can be aggregated to make a compact representation for that speaker [54], and is called a *d-vector* ("deep vector"), which uses an ANN to process each frame and its context to get a frame-level embedding. Such a low-dimensional speaker embedding model may be the average of activations from the last ANN hidden layer, which is considered as a bottleneck layer, and contains an efficient characterization of the speaker. Such embedding d-vectors represent utterances in a fixed-dimensional space, which facilitates simple classification.

 doi:10.20944/preprints202412.2368.v1

Another variant of speaker embedding is called an *x-vector*, which uses a sliding window of frames as input and a TDNN [55]. It has sub-networks at the frame and segment levels, and is connected by a *statistics pooling* layer, to locate means and variances of frame-level embeddings. This layer thus projects variable-length input into a fixed-length representation, by accumulating statistics of hidden node activations across time. Then, pooling over frame-level representations concatenates means and standard deviations for the data of each speaker. The x-vector usually employs PLDA to calculate scores, while d-vectors use cosine similarity [56]. Clustering of x-vectors may also use variational Bayesian re-segmentation [17,37], which refines the positioning of time stamps that may have been roughly estimated during clustering.

6. Speech Activity Detection (SAD)

A major task in SD is to determine which portions of an input audio signal contain speech, versus having silence or non-speech sounds. This is called *speech activity detection* (SAD) or *voice activity detection* (VAD) [57]. For SAD, classical approaches have used a combination of energy estimation and spectral analysis. When audio contains speech with only a small amount of background or channel noise, SAD could be as simple as applying an energy threshold, i.e., judge any strong signal as speech. In many practical circumstances, however, a microphone could receive many different sound sources. Since speech has several acoustic features that distinguish it from other sounds (Section 2), energy should be augmented by applying other useful data features, to accomplish more accurate SAD.

SD research often reports results for two tasks: 1) with "oracle SAD," which indicates that the classification system knows a-priori the actual time stamps (*ground truth*), or 2) the system uses SAD. A classical approach to speaker-turn segmentation does hypothesis testing on acoustic segments with sliding (possibly overlapping) successive windows. Such speaker-change detection often requires a threshold that is empirically tuned for major changes in audio, e.g., movement of the sound sources or the microphone.

As noted in Section 2, speech can be voiced or unvoiced. Unvoiced sounds are weaker and more easily confused with other audio, especially background noise, as such speech derives from noise generated at a VT constriction, but shaped by the upper VT (forward of the constriction). This shortness of the oral cavity of the VT for obstruent phonemes generally renders the speech noise as high-pass in nature, versus the low-pass nature of most environmental noise.

In the majority of speech that is voiced, the resonances of the VT have a formant structure in the spectrum (e.g., 1 kHz apart, on average, for the typical 17-cm-long VT of men). In addition, the quasi-periodic glottal excitation of the VT imposes an approximate line spectrum, spaced at intervals of F0. Both of these acoustic features (resonances and periodic) make voiced speech unlike many other non-speech sounds. Most SD exploits such features very indirectly, via Mel spectrograms, rather than any direct feature extraction. Most modern SD, based on ANNs, does not explicitly utilize the speech production features noted in this paper, but requires the network models to learn distinctions via approximate automatic training. The general assumption for machine learning is that it is best to leave processing to the automatic training, but training with simple gradient descent on generic losses imposes a heavy burden on such methods.

7. Clustering Frames by Speaker Turns

In portions of speech, as perhaps estimated by SAD, SD must determine: 1) whether one or multiple speakers are talking and 2) at what times speakers start and cease talking. One common approach is to seek times of pertinent change in speech features and/or group (cluster) related speech frames into coherent speaker segments [43,58–61]. Section 2 described features of speech that may be exploited to seek speech continuity. However, given the complexity of speech, most approaches to SD have used more general distance measures. Other earlier methods also used HMM-ASR to

 doi:10.20944/preprints202412.2368.v1

assist [58]. ASR can assist in modern neural SD as well [62], as ASR can provide textual clues to speaker turns.

As a way to measure acoustic distances between speech segments, two common methods are used for SD: *generalized likelihood ratio* (GLR) [63] and *Bayesian information criterion* (BIC) [64]. BIC searches for change points in a time window, and uses a penalized likelihood ratio test for whether data are better modelled by one or more distributions. A likelihood-based metric, GLR is the ratio between two hypotheses (i.e., whether successive segments derive from the same speaker). The *Kullback-Leibler* divergence, which estimates the distance between two random distributions [65], is also commonly used.

Other popular techniques include *spectral clustering* (SC) [51], *k-means clustering* [66–68], *mean shift clustering* [56], and *agglomerative hierarchical clustering* (AHC) [69], to aggregate the regions of each speaker into separate clusters. These tend to use a bottom-up approach that trains multiple clusters or models and successively merges them, reducing the number of clusters until one remains per speaker. SC mostly uses a cosine distance, rather than the basic Euclidean.

However, many clustering methods can only estimate one speaker per segment, and thus not handle speech of overlapping speakers. Also, standard k-means clustering uses squared error as a criterion, which generally does not well handle non-Gaussian data nor data containing imbalanced classes, which are common for speech. Indeed, many SD change-point detections are based on metric-based approaches that assume the distribution of speech features follow a Gaussian distribution, which is often not the case.

SC is common unsupervised clustering technique based on graph theory [70]. It requires a pairwise similarity score between all pairs of speaker embeddings, resulting in a similarity graph. SC is sensitive to noise in the affinity matrix.

AHC constructs a hierarchy based on distances between features and forms a group. AHC is a common unsupervised clustering technique, which uses pairwise distances between all speaker embeddings, yielding a distance matrix. A cluster hierarchy is generated (a tree diagram called a dendrogram). Each input sample starts in its own cluster, and pairs of clusters are then merged.

Variations on clustering include adding constraints [71,72] and affinity propagation [73], as well as X-means and online re-clustering [66,74]. There is also Viterbi realignment, which re-segments an audio stream is based on a current clustering hypothesis, with re-training on an alternative segmentation [75,76]. Most clustering in SD is unsupervised [77]. Some SD studies have used mean-shift clustering, which assigns data points to clusters iteratively via modes in a non-parametric distribution [78].

In *deep embedded clustering* [79], input features (speaker embeddings) are transformed to be more separable. For differentiability (needed to train ANNs), each embedding has a probability for available speaker clusters, which are iteratively refined using bottleneck features estimated by an autoencoder [80]. Some SD approaches combine segmentation and clustering into one trainable model [77].

8. Summary of Speaker Diarization Methods

We distinguish two different approaches to SD: classical modular and neural.

8.1. Modular SD

Traditional SD systems have used a module sequence: 1) a speaker embedding extractor for each segment, 2) clustering merges segments based on speaker embedding similarity and assigns speaker labels to all segments, 3) optionally, a compensation algorithm such as Variational-Bayesian refinement can calibrate the clustering results. In such modular SD, several components must be optimized with separate criteria and it may be difficult to handle overlapped speech, as the unsupervised clustering often assumes that each segment can only be assigned to one speaker.

8.2. Recent ANNs for SD

 doi:10.20944/preprints202412.2368.v1

E2E ANN-based SD (EEND) needs labeled data to handle the joint speech actions of each speaker at each time frame [81–83]. Such is often trained on simulated data, and then adapted to target conditions by using labeled data. Modern systems include: Encoder-Decoder Attractor architecture EDA-EEND [84], x-vector clustering [55], the self-attentive EEND (SA-EEND) model [52], and Region-Proposal Network based SD [48]. Some of these accommodate the problem of determining how many speakers are in a conversation. One limitation of EEND is its fixed number of output heads. EDA-EEND employs an LSTM encoder-decoder to predict an *attractor* for each speaker (in order to aggregate frames). Re-segmentation is an optional step that further refines the diarization prediction [34]. EEND has difficulty with audio-conferences having more than 3 speakers and audio exceeding 10 minutes. Basic EEND functions by batch processing, which has excessive computer memory with long recordings.

SD DNNs often have several recurrent layers and develop them with the *permutation invariant training* method [45]. However, they only show good performance in constrained settings and do not generalize well to real-world conditions.

9. Ways to Evaluate Speaker Diarization

Measuring success for SD is more complicated than for other speech applications. For example, speech coders and synthesizers seek high intelligibility and naturalness. ASR and SV seek minimal word and speaker identity errors, respectively. (All, of course, wish to minimize cost and latency.) However, SD involves both timing estimation and speaker recognition; it is not immediately clear which is more important. A common measure of SD success is the *Diarization Error Rate* (DER). DER treats three error types with equal weight: *false alarm* (FA: section of audio labelled incorrectly as speech), *miss* (duration of speech not so labelled), and *confusion* (portion of speech with mislabelled speakers). Thus, DER = (FAs + Misses + Confusions) / Total Duration of audio.

DER does not explicitly judge whether time stamps are appropriate, i.e., penalize errors in locating times where a speaker stops/starts; it examines the average of labelling accuracy over all frames. Indeed, some SD challenges allowed error deviations in time stamps via a *forgiveness collar*, which meant ignoring small timing errors (e.g., under 250 ms). This "score collar" around every boundary of reference segments was intended to mitigate effects of inconsistency in reference transcriptions.

Other SD measures include WDER, which is a version of the DER measure based on words, rather than audio durations. The *Jaccard error rate* (JER), as in the DIHARD II set, is based on the Jaccard index, a similarity measure used to evaluate the output of image segmentation systems. JER = (FAs + Misses) / Total; so, it is DER, not counting confusions. DER values typically exceed 10% on most standard datasets (see the next Section), and can range much higher on some.

10. Datasets for Speaker Diarization

A disadvantage of SD approaches based on models is their reliance on external data for the training, which makes them less robust in acoustic conditions not seen in training. So, data are important. To evaluate SD performance, one typically uses annotated datasets, i.e., sets of audio containing multiple people talking, where time stamps are labelled manually. Training can be done on any audio, but supervised training would use labelled datasets, which could be from natural sources (e.g., YouTube, public broadcasts). Training sometimes uses "synthetic" datasets, where individual speakers' speech (i.e., from monologues, not conversations) are superimposed artificially. Such simulated mixtures may also have added noise and reverberation with simulated room impulse responses. SD performance varies with how well the simulated data used in training matches real (target domain) conversations.

Early SD datasets were the ICSI meeting corpus [85] and the NIST Rich Transcription sets from meeting conference audio [86,87]. An earlier dataset still widely used for challenging ASR is the

 doi:10.20944/preprints202412.2368.v1

CallHome dataset, which has 500 sessions of multilingual telephone speech. Each session has 2-6 speakers, but most conversations (2-5 minutes) each have two dominant speakers.

Other datasets are DiHARD [88], Augmented Multiparty Interaction (AMI) Meeting corpus [89], the CHIME series [90–92], Fearless Steps [93], Iberspeech RTV [94], LibriCSS [12], M2Met [95], and the French REPERE [96]. DISPLACE 2023 has conversations using different languages, high levels of noise, reverberations, and overlapping speech.

Freely available datasets include: AISHELL-4 [97], Albayzin/RTVE [98], AliMeeting [95], VoxConverse [99], Ego4D [100], and This American Life [101].

The AMI corpus includes 100 hours of meeting conversations. DIHARD I and II development sets have 19 and 23 hours of data. The VoxConverse datase has approximately 20 hours of conversations from YouTube. The VoxCeleb datasets, widely used in SV, have speech segments of more than ten seconds (on average). An open-source toolkit for SD is written in Python [102].

11. Discussion

This paper has reviewed the technical problems and methods associated with speaker diarization. It first discussed the acoustical nature of speech signals, from the point of view of which aspects or features may be employed to distinguish speakers. Notably, most of speech is voiced (quasi-periodic) with resonances and F0 that mostly change slowly. Approximately eight times per second, speech transitions between phonemes, which result in spectral changes that are sometimes abrupt, sometimes smooth. Otherwise, i.e., during approximately 85-90% of speech frames, there is significant spectral continuity in the acoustic signal of a speaker.

Individual phonemes have patterns of two types: (voiced) sonorants or noisy consonants. The former have slowly changing resonances (on average, one per kHz in typical 17-cm vocal tracts), with slowly changing harmonics. The latter have mostly brief high-pass noise patterns. As the second category has much less speaker information the first category, SD should likely pay far more attention to sonorants.

However, SD rarely exploits any of this knowledge based on phonology and phonetics, and treats speech as a general acoustic signal. Modern SD methods do not explicitly take advantage of the smooth movements of formants and F0 in individual speakers' speech. The widely-used Mel spectrogram patterns contain this data, but in highly-coded form, which is not evident to exploit in neural networks, especially when using general loss functions. In addition, SD generally assumes access to entire recordings before processing, which hinders streaming, on-line use.

12. Conclusions

This paper has noted how recent and modern SD has functioned, explaining the basics of spectral analysis as applied to speech. It has suggested areas that might be profitably explored for improvements. In cases where there are few sections of overlapped speech, segmentation by speaker has been successful, and speaker labelling shows performance similar to that found in automatic speaker verification. However, common meeting data and ordinary conversations have multiple interruptions, which presents additional difficulty, often leading to significant error rates.

References

1. K. Han and D. Wang, "A classification based approach to speech segregation." *The Journal of the Acoustical Society of America*, 132(5), 3475-3483, 2012.
2. J. LeRoux, S. Wisdom, H. Erdogan, and J.R. Hershey, "SDR–half-baked or well done?" *ICASSP*, 626-630, 2019.
3. 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." *Interspeech*, 2808-2812, 2018.

 doi:10.20944/preprints202412.2368.v1

4. T. Von Neumann, K. Kinoshita, M. Delcroix, S. Araki, T. Nakatani, and R. Haeb-Umbach, "All-neural online source separation, counting, and diarization for meeting analysis." *ICASSP*, 91-95, 2019.
5. D. Raj, L.P. Garcia-Perera, Z. Huang, S. Watanabe, D. Povey, A. Stolcke, A. and S. Khudanpur, "DOVER-lap: A method for combining overlap-aware diarization outputs." *IEEE Spoken Language Technology Workshop*, 881-888, 2021.
6. N. Kanda, X. Xiao, Y. Gaur, X. Wang, Z. Meng, Z. Chen, and T. Yoshioka, "Transcribe-to-diarize: Neural speaker diarization for unlimited number of speakers using end-to-end speaker-attributed ASR." *ICASSP*, 8082-8086, 2022.
7. N. Kanda, Y. Gaur, X. Wang, Z. Meng, Z. Chen, T. Zhou, and T. Yoshioka, "Joint speaker counting, speech recognition, and speaker identification for overlapped speech of any number of speakers." *Interspeech*, 36–40, 2020.
8. S. Haykin and Z. Chen, "The cocktail party problem." *Neural computation*, 17(9), 1875-1902, 2005.
9. B. Arons, "A review of the cocktail party effect." *Journal of the American Voice I/O Society*, 12(7), 35-50, 1992.
10. A.W. Bronkhorst, "The cocktail party phenomenon: A review of research on speech intelligibility in multiple-talker conditions." *Acta Acustica united with Acustica*, *86*(1), 117-128, 2000.
11. A.S. Bregman, Auditory Scene Analysis, 1990.
12. Z. Chen, T. Yoshioka, L. Lu, T. Zhou, A. Meng, Y. Luo, J. Wu, X. Xiao, and J. Li, "Continuous speech separation: Dataset and analysis." *ICASSP*, 7284-7288, 2020.
13. Y. Luo and N. Mesgarani, "Conv-Tasnet: Surpassing ideal time–frequency magnitude masking for speech separation." *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(8), 1256-1266, 2019.
14. K. Kinoshita, M. Delcroix, S. Araki, and T. Nakatani, "Tackling real noisy reverberant meetings with all-neural source separation, counting, and diarization system." *ICASSP*, 381-385, 2020.
15. L.E. Shafey, H. Soltau, and I. Shafran, "Joint speech recognition and speaker diarization via sequence transduction." *Interspeech*, 396–400, 2019.
16. X. Xiao, N. Kanda, Z. Chen, T. Zhou, T. Yoshioka, S. Chen, Y. Zhao, G. Liu, Y.Wu, J. Wu, and S. Liu, "Microsoft speaker diarization system for the VoxCeleb speaker recognition challenge 2020." *ICASSP*, 5824-5828, 2021.
17. X. Anguera, 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*, 20(2), 356-370, 2012.
18. S.E. Tranter and D.A. Reynolds, "An overview of automatic speaker diarization systems." *IEEE Transactions on Audio, Speech, and Language Processing*, 14(5), 1557-1565, 2006.
19. M.H. Moattar and M.M. Homayounpour, "A review on speaker diarization systems and approaches." Speech Communication, 54(10), 1065-1103, 2012.
20. T.J. Park, N. Kanda, D. Dimitriadis, K.J. Han, S. Watanabe, and S. Narayanan, "A review of speaker diarization: Recent advances with deep learning." *Computer Speech & Language*, 72, 101317, 2022.
21. X. Chang, W. Zhang, Y. Qian, J. Le Roux, and S. Watanabe, "MIMO-Speech: End-to-end multi-channel multi-speaker speech recognition." *IEEE Automatic Speech Recognition and Understanding Workshop*, 237-244, 2019.
22. T. Yoshioka, T., H. Erdogan, H., Z. Chen, Z. and F. Alleva, "Multi-microphone neural speech separation for far-field multi-talker speech recognition." *ICASSP*, 5739-5743, 2018.
23. D. O'Shaughnessy, Speech Communication: Human and Machine, IEEE Press, 2000.
24. Y. Dissen, J. Goldberger, and J. Keshet, "Formant estimation and tracking: A deep learning approach," *J. Acoust. Soc. Am.*, 145 (2), 642-653, 2019.
25. G. Hu and D. Wang, "Monaural speech segregation based on pitch tracking and amplitude modulation." *IEEE Transactions on Neural Networks*, 15(5), 1135-1150, 2004.
26. A.S. Spanias, "Speech coding: A tutorial review." *Proceedings of the IEEE*, *82*(10), 1541-1582, 1994.
27. S.B. Davis and P. Mermelstein, "Comparison of parametric representations for monosyllabic word recognition in continuously spoken sentences," *IEEE Trans. ASSP*, vol. 28, 357–366, 1980.
28. J. Makhoul, "Linear prediction: A tutorial review." *Proceedings of the IEEE*, 63(4), 561-580, 1975.

29. L. Rabiner, M. Cheng, A. Rosenberg, and C. McGonegal, "A comparative performance study of several pitch detection algorithms." *IEEE Transactions on Acoustics, Speech, and Signal Processing*, *24*(5), 399-418, 1976.
30. A.G. Adami, R. Mihaescu, D.A. Reynolds, and J.J. Godfrey, "Modeling prosodic dynamics for speaker recognition." *ICASSP*, IV-788-791, 2003.
31. M. Zelenák and J. Hernando, "The detection of overlapping speech with prosodic features for speaker diarization." ICSLP, 2011.
32. G. Friedland, O. Vinyals, Y. Huang, and C. Muller, "Prosodic and other long-term features for speaker diarization." *IEEE Transactions on Audio, Speech, and Language Processing*, 17(5), 985-993, 2009.
33. L.R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition." *Proceedings of the IEEE*, *77*(2), 257-286, 1989.
34. M. Diez, L. Burget, S. Wang, J. Rohdin, and J. Cernocký, "Bayesian HMM Based x-Vector Clustering for Speaker Diarization." *Interspeech*, 346-350, 2019.
35. J. Geiger, F. Wallhoff, and G. Rigoll, "GMM-UBM based open-set online speaker diarization." *Interspeech*. 2330-2333, 2010.
36. P. Singh, H. Vardhan, S. Ganapathy, and A. Kanagasundaram, "LEAP diarization system for the second DIHARD challenge." *Interspeech*, 983-987, 2019.
37. D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and S. Khudanpur, "Deep neural network-based speaker embeddings for end-to-end speaker verification." *IEEE Spoken Language Technology Workshop*, 165-170, 2016.
38. G. SelL and D. Garcia-Romero, "Speaker diarization with PLDA i-vector scoring and unsupervised calibration." *IEEE Spoken Language Technology Workshop*, 413-417, 2014.
39. B. Fergani, M. Davy, and A. Houacine, "Speaker diarization using one-class support vector machines." *Speech Communication*, 50(5), 355-365, 2008.
40. P. Kenny, T. Stafylakis, P. Ouellet, and H.J. Alam, "JFA-based front ends for speaker recognition." *ICASSP*, 1705-1709, 2014.
41. Y. LeCun, Y. Bengio, G. Hinton, "Deep learning," Nature, 521, 436–444, 2015.
42. S.H. Yella, A. Stolcke, A. and M. Slaney. "Artificial neural network features for speaker diarization." *IEEE Spoken Language Technology Workshop*, 402-406, 2014.
43. E. Fini and A. Brutti, "Supervised online diarization with sample mean loss for multi-domain data." *ICASSP*, 7134-7138, 2020.
44. J. Deng, J. Guo, N. Xue, and S. Zafeiriou, "Arcface: Additive angular margin loss for deep face recognition." *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4690-4699, 2019.
45. M. Kolbæk, D. Yu, Z.H. Tan, and J. Jensen, "Multi-talker speech separation with utterance-level permutation invariant training of deep recurrent neural networks." *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, *25*(10), 1901-1913, 2017.
46. T.J. Park, K.J. Han, M. Kumar, and S. Narayanan, "Auto-tuning spectral clustering for speaker diarization using normalized maximum eigengap." *IEEE Signal Processing Letters*, 27, 381-385, 2019.
47. Z. Huang, S. Watanabe, Y. Fujita, P. García, Y. Shao, D. Povey, and S. Khudanpur, "Speaker diarization with region proposal network." *ICASSP*, 6514-6518, 2020.
48. Y.C. Liu, E. Han, C. Lee, and A. Stolcke, "End-to-end neural diarization: From transformer to conformer." *arXiv*: 2106.07167, 2021.
49. R. Zazo, A. Lozano-Diez, J. Gonzalez-Dominguez, D. Toledano, and J. Gonzalez-Rodriguez, "Language identification in short utterances using long short-term memory (LSTM) recurrent neural networks." *PloS one*, 11(1), e0146917, 2016.
50. Q. Wang, C. Downey, L. Wan, P.A. Mansfield, and I.L. Moreno, "Speaker diarization with LSTM." *ICASSP*, 5239-5243, 2018.
51. Y. Fujita, N. Kanda, S. Horiguchi, Y. Xue, K. Nagamatsu, and S. Watanabe, "End-to-end neural speaker diarization with self-attention." *ASRU*, 296-303, 2019.
52. C. Subakan, M. Ravanelli, S. Cornell, M. Bronzi, and J. Zhong, "Attention is all you need in speech separation." *ICASSP*, 21-25, 2021.

53. D. Garcia-Romero, D. Snyder, G. Sell, D. Povey, and A. McCree, "Speaker diarization using deep neural network embeddings." *ICASSP*, 4930-4934, 2017.
54. F. Landini, J. Profant, M. Diez, and L. Burget, "Bayesian HMM clustering of x-vector sequences (vbx) in speaker diarization: theory, implementation and analysis on standard tasks." *Computer Speech & Language*, 71, 101254, 2022.
55. M. Senoussaoui, P. Kenny, T. Stafylakis, and P. Dumouchel, "A study of the cosine distance-based mean shift for telephone speech diarization." *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(1), 217-227, 2013.
56. I. Medennikov, M. Korenevsky, T. Prisyach, Y. Khokhlov, M. Korenevskaya, I. Sorokin, T. Timofeeva, A. Mitrofanov, A. Andrusenko, I. Podluzhny, and A. Laptev, "Target-speaker voice activity detection: a novel approach for multi-speaker diarization in a dinner party scenario". *Interspeech*, 274–278, 2020.
57. M. Padmanabhan, L.R. Bahl, D. Nahamoo, and M.A. Picheny, "Speaker clustering and transformation for speaker adaptation in speech recognition systems." *IEEE Transactions on Speech and Audio Processing*, 6(1), 71-77, 1998.
58. S.H. Shum, N. Dehak, R. Dehak, and J.R. Glass, "Unsupervised methods for speaker diarization: An integrated and iterative approach," *IEEE Transactions on Audio, Speech, and Language Processing*, 21(10), 2015-2028, 2013.
59. Q. Li, F.L. Kreyssig, C. Zhang, and P.C. Woodland, "Discriminative neural clustering for speaker diarisation." *IEEE Spoken Language Technology Workshop*, 574-581, 2021.
60. Q. Lin, R. Yin, M. Li, H. Bredin, and C. Barras, "LSTM based similarity measurement with spectral clustering for speaker diarization." *Interspeech*, 2019, 366–370, 2019.
61. W. Xia, H. Lu, Q. Wang, A. Tripathi, Y. Huang, I.L. Moreno, and H. Sak, "Turn-to-diarize: Online speaker diarization constrained by transformer transducer speaker turn detection." *ICASSP*, 8077-8081, 2022.
62. H. Gish, M.H. Siu, and J.R. Rohlicek, "Segregation of speakers for speech recognition and speaker identification." *ICASSP*, 873-876, 1991.
63. S. Watanabe, "A widely applicable Bayesian information criterion." *The Journal of Machine Learning Research*, 14(1), 867-897, 2013.
64. M.A. Siegler, U. Jain, B. Raj, and R.M. Stern, "Automatic segmentation, classification and clustering of broadcast news audio." *DARPA Speech Recognition Workshop*, 1997.
65. D. Dimitriadis and P. Fousek, "Developing On-Line Speaker Diarization System." *Interspeech*, 2739-2743, 2017.
66. K. Kinoshita, M. Delcroix, and N. Tawara, "Integrating end-to-end neural and clustering-based diarization: Getting the best of both worlds." *ICASSP*, 7198-7202, 2021.
67. A. Ng, M. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm." *Advances in neural information processing systems*, 14, 2001.
68. K. Han and S.S. Narayanan, "A robust stopping criterion for agglomerative hierarchical clustering in a speaker diarization system." *Interspeech*, 1853-1856, 2007.
69. H. Ning, M. Liu, H. Tang, and T.S. Huang, "A spectral clustering approach to speaker diarization." *ICSLP*, 2006.
70. I. Davidson and S.S. Ravi, "Clustering with constraints: Feasibility issues and the k-means algorithm." *SIAM International Conference on Data Mining*, 138-149, 2005.
71. C. Yu and J.H. Hansen, "Active learning based constrained clustering for speaker diarization." *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 25(11), 2188-2198, 2017.
72. R. Yin, H. Bredin, and C. Barras, "Neural speech turn segmentation and affinity propagation for speaker diarization," *Interspeech*, 2018.
73. Y. Xue, S. Horiguchi, Y. Fujita, S. Watanabe, P. García, and K. Nagamatsu, "Online end-to-end neural diarization with speaker-tracing buffer." *IEEE Spoken Language Technology Workshop*, 841-848, 2021.
74. H. Bredin and A. Laurent, "End-to-end speaker segmentation for overlap-aware resegmentation." *Interspeech*, 2021.
75. L. Bullock, H. Bredin, and L.P. Garcia-Perera, "Overlap-aware diarization: Re-segmentation using neural end-to-end overlapped speech detection." *ICASSP*, 7114-7118, 2020.

 doi:10.20944/preprints202412.2368.v1

76. A. Zhang, Q. Wang, Z. Zhu, J. Paisley, and C. Wang, "Fully supervised speaker diarization." *ICASSP*, 6301-6305, 2019.
77. D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5), 603-619, 2002.
78. J. Xie, R. Girshick, and A. Farhadi, "Unsupervised deep embedding for clustering analysis." *International conference on machine learning*, 478-487, 2016.
79. X. Guo, L. Gao, X. Liu, and J. Yin, "Improved deep embedded clustering with local structure preservation." *International Joint Conference on Artificial Intelligence*, 1753-1759, 2017.
80. Y. Fujita, N. Kanda, S. Horiguchi, K. Nagamatsu, and S. Watanabe, "End-to-end neural speaker diarization with permutation-free objectives." *Interspeech*, 4300–4304, 2019.
81. E. Han, C. Lee, and A. Stolcke, "BW-EDA-EEND: Streaming end-to-end neural speaker diarization for a variable number of speakers." *ICASSP*, 7193-7197, 2021.
82. A. Tripathi, H. Lu, and H. Sak, "End-to-end multi-talker overlapping speech recognition." *ICASSP*, 6129-6133, 2020.
83. S. Horiguchi, Y. Fujita, S. Watanabe, Y. Xue, and K. Nagamatsu, "End-to-end speaker diarization for an unknown number of speakers with encoder-decoder based attractors." *Interspeech*, 269–273, 2020.
84. D. Janin, J. Baron, J. Edwards, D. Ellis, D. Gelbart, N. Morgan, B. Peskin, T. Pfau, E. Shriberg, A. Stolcke, and C. Wooters, "The ICSI meeting corpus." *ICASSP*, vol. 1, 364-367, 2003.
85. J.G. Fiscus, J. Ajot, and J.S. Garofolo,, "The Rich Transcription 2007 meeting recognition evaluation." *International Evaluation Workshop on Rich Transcription*, 373-389, 2007.
86. M. Przybocki and A. Martin, "2000 NIST Speaker Recognition Evaluation," *Linguistic Data Consortium*, 2011.
87. N. Ryant, P. Singh, V. Krishnamohan, R. Varma, K. Church, C. Cieri, J. Du, S. Ganapathy, and M. Liberman, "The third DIHARD diarization challenge." *Interspeech*, 3570–3574, 2021.
88. J. Carletta, "Unleashing the killer corpus: experiences in creating the multi-everything AMI Meeting Corpus." *Language Resources and Evaluation*, 41, 181-190, 2007.
89. C. Boeddeker, J. Heitkaemper, J. Schmalenstroeer, L. Drude, J. Heymann, and R. Haeb-Umbach, "Front-end processing for the CHiME-5 dinner party scenario." *CHiME5 Workshop*, 2018.
90. S. Watanabe, M. Mandel, J. Barker, E. Vincent, A. Arora, X. Chang, S. Khudanpur, V. Manohar, D. Povey, D. Raj, and D. Snyder, "CHiME-6 challenge: Tackling multispeaker speech recognition for unsegmented recordings." *International Workshop on Speech Processing in Everyday Environments*, 2020.
91. S. Cornell, M. Wiesner, S. Watanabe, D. Raj, X. Chang, P. Garcia, Y. Masuyama, Z.Q. Wang, S. Squartini, and S. Khudanpur, "The CHiME-7 DASR Challenge: Distant Meeting Transcription with Multiple Devices in Diverse Scenarios." *arXiv:2306.13734*, 2023.
92. J.H. Hansen, A. Joglekar, M.C. Shekhar, V. Kothapally, C. Yu, L. Kaushik, and A. Sangwan, "The 2019 inaugural Fearless Steps challenge: A giant leap for naturalistic audio," *Interspeech*, 2019.
93. J.M. Perero-Codosero, J. Antón-Martín, D.T. Merino, E.L. Gonzalo, and L.A.H. Gómez, "Exploring Open-Source Deep Learning ASR for Speech-to-Text TV program transcription." *IberSPEECH*, 262-266, 2018.
94. F. Yu, S. Zhang, Y. Fu, L. Xie, S. Zheng, Z. Du, W. Huang, P. Guo, Z. Yan, B. Ma, and X. Xu, "M2MeT: The ICASSP 2022 multi-channel multi-party meeting transcription challenge." *ICASSP*, 6167-6171, 2022.
95. J. Kahn, O. Galibert, L. Quintard, M. Carré, A. Giraudel, and P. Joly, "A presentation of the REPERE challenge." *International Workshop on Content-Based Multimedia Indexing*, 1-6, 2012.
96. Y. Fu, L. Cheng, S. Lv, Y. Jv, Y. Kong, Z. Chen, Y. Hu, L. Xie, J. Wu, H. Bu, and X. Xu, "AISHELL-4: An open source dataset for speech enhancement, separation, recognition and speaker diarization in conference scenario." *Interspeech*, 2021.
97. E. Lleida, A. Ortega, A. Miguel, V. Bazán-Gil, C. Pérez, M. Gómez, M. and A. De Prada,"Albayzin 2018 evaluation: the Iberspeech-RTVE challenge on speech technologies for Spanish broadcast media." *Applied sciences*, *9*(24), 5412, 2019.
98. J.S. Chung, J. Huh, A. Nagrani, T. Afouras, and A. Zisserman, "Spot the conversation: speaker diarisation in the wild." *Interspeech*, 299–303, 2020.

 doi:10.20944/preprints202412.2368.v1

99. K. Grauman, A. Westbury, E. Byrne, Z. Chavis, A. Furnari, R. Girdhar, J. Hamburger, H. Jiang, M. Liu, X. Liu, and M. Martin, "Ego4d: Around the world in 3,000 hours of egocentric video." *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18995-19012, 2022.
100. H.H. Mao, S. Li, J. McAuley, and G. Cottrell, "Speech recognition and multi-speaker diarization of long conversations." *Interspeech*, 2020.
101. H. Bredin, "pyannote. audio 2.1 speaker diarization pipeline: principle, benchmark, and recipe." *Interspeech*, 1983-1987, 2023.