THE UNIVERSITY *of* EDINBURGH

Leveraging deep speaker embedding variability factors for speaker verification and diarization

Chau Van Quy Luu

Doctor of Philosophy
Institute for Language, Cognition and Computation
School of Informatics
University of Edinburgh

2023

Abstract

The tasks of speaker verification (SV, determining whether a test utterance has the same speaker as an enrolment utterance) and speaker diarization (SD, determining 'who spoke when?') both fall under the umbrella of speaker recognition tasks. Both SV and SD have become tasks with high applicability in mainstream technology. Example applications of SV and SD are a voice assistant that only activates for a specific user, and colour-coding subtitles according to the speaker that produced them, respectively.

Both SV and SD systems have found success in recent years by utilising speaker embeddings, vector representations of speaker identity extracted from segments of speech. By comparing the similarity of the embeddings extracted from different utterances, it is possible to distinguish the speaker of each utterance, and this mechanism is what many successful verification and diarization systems are based upon.

Deep speaker embeddings are the speaker embeddings extracted from the intermediate layer of a neural network. This neural network is trained in such a way as to encode speaker identity in a discriminative fashion in the desired intermediate layer. For example, often the training objective is speaker classification or a variation thereof, and while this has been shown to be a very successful strategy, various sources of information can be encoded into this embedding space. Some of these sources are speaker related, and intuitively make up part of speaker identity, such as speaker gender. However, some sources of information are not explicitly speaker related, such as the channel and recording information, but can nonetheless be captured during training.

In this work, we look at these sources of information and variability, describing them *speaker embedding variability factors* and explore how they interact with and affect the downstream tasks of SV and SD. Specifically, our work looks at the following topics: reducing channel variability, reducing speaker variability distribution mismatch, explicitly encouraging variability for speaker identity related factors for increased robustness, and investigating the contribution of certain speaker attributes to separability.

For reducing channel variability, we propose a training regime based on adversarial methods that adds an adversarial loss based on discriminating whether pairs of embeddings come from the same recording. This approach adds channel invariance to the training objective of the embedding network while being dataset agnostic, not requiring any additional labels. We show that our induced channel invariability improves verification for out-of-recording pairs of utterances, while also improving diarization

performance.

In the pursuit of encouraging speaker identity related variability, we propose a multi-task learning training framework for leveraging auxiliary labels, adding additional attribute related tasks to the overall training objective. In conjunction with the standard speaker classification task, the addition of speaker age and speaker nationality classification tasks was shown to improve verification and diarization performance, particularly when fine-tuning to new domains. We suggest the reason for this improvement is due to the robustness imparted by structuring the embedding space in a way that we already know is speaker identity related, thus decreasing the risk of fitting to other non-identity related factors.

We also investigated the contribution of speaker attributes to speaker separability using disentangled representations. Here, we combined the aforementioned multi-task framework along with adversarial methods to successfully isolate aspects of speaker identity in specific dimensions of the speaker embedding. We then ablated these dimensions and found that for different datasets, different speaker attributes were of varying importance to separability in diarization and verification tasks, with gender a particularly strong factor for in-the-wild celebrity utterances.

Furthermore, by looking at the logits of the speaker classification network that speaker embeddings are extracted from, we found that for a group unseen utterances, the predicted posterior distribution (of training set speakers) was extremely skewed. By implementing a form of iterative fine-tuning on high probability training set speakers in combination with a form of dropout on the output layer, we showed improvements to verification performance. We suggest that the cause of this improvement is due to a distribution mismatch of speakers, relating to the aforementioned speaker attributes.

Overall, we explored several different approaches to manipulating the variability factors present in deep speaker embeddings, finding that each approach had merits when applied to specific scenarios. We suggest approaches for future work that build upon the techniques outlined in this thesis, in particular for speaker attribute-related learning and disentanglement.

Acknowledgements

My deepest thanks to my supervisors Steve Renals and Peter Bell for their guidance throughout this doctorate. Without their experience, insight and patience, none of this would have been possible.

I would also like to thank my colleagues at CSTR and Edinburgh, whose presentations, conversations, and discussions always managed to help illuminate a path forward for me, even if they did not realise it: Ondrej Klejch, Joachim Fainberg, Erfan Loweimi, Ramon Sanabria, Jennifer Williams, Shucong Zhang, Mark Sinclair, Jason Fong, Johannah O'Mahony. CSTR is such a wonderful group with such an enthusiastic, wholesome and nurturing atmosphere, and so I must thank everyone there, even if not named here, for making that possible.

Thanks also to my friends and family. Ziggy Shaw, Samuel Haines, Tom Arneil, Gervaise Miller, Ryan Harris, Rebecca Brambilla, Minh Luu. Your friendship, love, and support has kept me (somewhat) sane throughout all this, and for that I am incredibly grateful.

Thank you also to Yulan Liu, Ilya Sklyar, and Anna Piunova during my time at Amazon for being such enthusiastic and helpful guides while I was there.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*Chau Van Quy Luu*)

Table of Contents

Chapter 1

Introduction

The act of identifying a person by their voice is an important aspect of human communication, be that in verifying the identity of someone in the context of a casual or professional phone call, or passively following the turns in conversation, for example in a radio show or podcast (Hansen and Hasan 2015). As such, automatic speaker recognition systems have found numerous practical applications, such as the verification of speaker identity for security purposes given a speech segment, or automatically colour-coding subtitles according to speaker in broadcast media. These two tasks fall under the general tasks of speaker verification and diarization respectively.

Speaker verification is the task of determining whether the claim that an unknown speaker matches a specific identity is true. In practice, this usually involves comparing a segment of speech from the unknown speaker to a template belonging to the specific identity that needs to be verified. Speaker diarization is the task of partitioning a stream of audio according to speaker identity, grouping segments which belong to the same speaker together, answering ”who spoke when?”.

In many modern speaker recognition systems, a primary component to both of these tasks is the use of vector representations that encode the identity of the speaker from an input segment of speech called speaker embeddings (Hansen and Hasan 2015; Bai and X.-L. Zhang 2021). Utilising these, one can perform verification by measuring the similarity between the embedding produced by the unknown speaker against that of the stored embedding belonging to the claimed speaker, and then making a decision based on some threshold. In a similar fashion, diarization can be performed by measuring the similarity between the embeddings extracted from all pairs of segments in a recording,

and then clustering based on these similarities.

With speaker embeddings being such a key component in these systems, the question of how exactly to produce them, given the raw audio of an input utterance, becomes crucial. This task falls broadly under the field of representation learning (Bengio et al. 2014), which aims to learn representations of data which make it easier to extract useful information. In Bengio et al. (2014, p. 1), this task is described as 'learn[ing] to identify and disentangle the underlying explanatory factors hidden in the observed milieu of low-level sensory data' and more broadly, to 'understand the world'. Representation learning is the task of learning to extract this information in an automatic fashion, replacing manual feature engineering, and is applicable for almost any kind of data, for example images, video, audio, or text data.

In the case of image representations, the explanatory factors in an image may be related to information about shapes, objects and structures, and image representations generally do extract this information from the original image (Kiela and Bottou 2014). A desirable property for learned image representations might be that for two images of, for example, an elephant, their representations would be similar, even if the original raw data in the original input images differed significantly due to different angles, rotation and lighting. These properties enable the representations to be used for different downstream tasks, such as image retrieval, where similar images are gathered for a query image (Barz and Denzler 2019), based on the extracted image embeddings.

In the case of learning representations speech and a speaker's voice, we also want to extract the underlying explanatory factors behind speaker identity. As humans, we may be able to intuitively distinguish certain factors that contribute to what makes up a speaker's voice. Such factors may include physically grounded properties such as pitch and timbre, which themselves have explanatory factors, such as age and gender, and the anatomical and physiological reasons for differing vocal tract lengths (Belin et al. 2004; Pernet and Belin 2012). Other identity-discriminative factors from speech may be semantical in nature, such as the differences in what a doctor might say compared to the replies from their patient (Shafey et al. 2019), or lexical, with slang and vocabulary revealing aspects of identity (Jessen 2007). While human ability to distinguish and perceive identity may be described and explained in terms of these factors, the goal of representation learning for speaker recognition is to automatically and implicitly learn a model of all the different factors that contribute to speaker identity. In principle, the downstream tasks of SV and SD require an embedding space in which speech from

the same speaker is close together, and speech from different speakers is distant, but it stands to reason that a model which *understands* what makes up speaker identity would need to incorporate and encode some of the aforementioned factors into the embedding space.

While we have outlined the overall goals of representation learning, the techniques of automatically learning representations have rapidly evolved over the past decade or so, and are often specific to the data and the downstream tasks that the representations are utilised for. For speaker embeddings, the most popular method for a number of years was the *i*-vector method (Dehak et al. 2011), based on Gaussian Mixture Modelling and Factor Analysis. This method modelled the contribution of speaker and channel information probabilistically for an utterance. However, in recent years, the most popular method for extracting speaker representations has been to extract speaker embeddings from the intermediate layer in a deep neural network, hence *deep* speaker embeddings, with the work of Snyder et al. (2017) proposing deep speaker embeddings called *x*-vectors as an alternative to *i*-vectors.

The popularity of the newer method is evidenced by the technical reports of winning teams in competitions for speaker verification and diarization, such as the VoxSRC challenge. The first VoxSRC challenge in 2019 featured only deep neural network (DNN)-based speaker embedding extraction in the winning verification systems (Chung et al. n.d.), and when diarization was added to the 2020 edition of the VoxSRC challenge, the top three teams in both verification and diarization tracks again used DNN-based speaker embeddings (Nagrani et al. 2020). This trend has continued, with all neural techniques dominating in the 2021 and 2022 editions of VoxSRC (Brown et al. 2022; Huh et al. 2023).

This move towards methods based around deep neural networks has been observed in many different fields of machine learning, with state-of-the-art performance being dramatically improved by the use of deep neural networks. These fields include automatic speech recognition (Hinton et al. 2012), natural language processing and image and video tasks (LeCun et al. 2015), with this overall trend being described as a ‘deep learning revolution’ (Sejnowski 2018). Deep neural architectures have the ability to discover intricate structure from the raw data via the back-propagation algorithm, and the multiple layers of a neural network can learn deeper abstractions based on the representations of previous layers, hence the term *deep* when describing the neural networks (LeCun et al. 2015).

For image embeddings for use in image retrieval, these embeddings are typically extracted from deep neural networks trained on image classification (Kiela and Bottou 2014), specifically the ImageNet dataset (Jia Deng et al. 2009) which has 1000 image classes, depicting different subjects, such as different animal species and various man-made objects. In combination with convolutional layers, the network is able to extract and encode visual information in subsequent layers. While trying to interpret what each layer learns is an ongoing field of research (Yu et al. 2014), there is strong evidence that information about edges, shapes, colours, and ultimately objects are learned by different layers in these networks (Mahendran and Vedaldi 2014; Rojas et al. 2020). Intuitively, it is clear to see how edges, shapes and colours would contribute to an *understanding* of what distinguishes the kind of subject depicted in an image, and from the explorations into what is learned at each layer, it seems as though deep networks are able to extract information about these factors.

For speaker embeddings, the representations are typically extracted from a network trained on classifying the speaker of an utterance (Snyder et al. 2018). The result of this is that an intermediate layer can be used to provide a speaker discriminative representation. Like image embeddings, it would be expected that the network learns to extract information about factors that contribute to the overall task, in this case, the classifying the speaker. These explanatory factors are referred to in this doctoral thesis as *speaker embedding factors*. Indeed, the idea different kinds of information are encoded in speaker embeddings is supported by a variety of work exploring the additional information encoded in deep speaker embeddings. Examples of the kind of information encoded have been channel information, emotion, sentiment and linguistic content. For example, Williams and King (2019) showed that the speaking style and emotion information is contained within x-vectors. Similarly, Raj et al. (2019) explored a broad set of categories for what kind of information is encoded in x-vectors. They probed for any additional information encoded in the embeddings by training additional separate classifiers to predict speaker gender, recording, speaker rate, utterance length, data augmentation type and other transcription related targets, such as the transcription, specific words and phonemes. They found that the embeddings contained information for nearly all categories explored, with strong indication for gender, channel and speaking rate information being present. From these works, deep neural networks have shown they able to capture different sources of information when trained to classify speakers, and this strong modelling performance is also reflected in

the state-of-the-art performance in speaker recognition tasks such as verification and diarization.

1.1 Motivation

However, despite the powerful modelling capabilities of deep neural networks and deep speaker embeddings are not without their problems. While some speaker embedding factors may align with an intuitive understanding of the explanatory factors behind speaker identity, such as gender and accent (Jessen 2007), an automatically learned representation may also capture aspects of audio and or speech that do not align with factors that humans may intuitively perceive. Channel information, relating to the acoustic recording conditions in a segment of audio, is an example of such a factor that is often found in speaker embeddings. If we consider a catalogue of broadcast media recordings, it may be that for a presenter of a specific radio show, who only ever appears on that show, the recording conditions of said programme are a speaker-distinguishing factor amongst that catalogue. However, there are scenarios in which this channel information cannot be leveraged usefully, and may not be helpful for discriminating between speakers, even if it was discriminative during training. In the realistic case of diarizing a single microphone recording of unseen speakers (determining who spoke when), the speech segments that are compared for speaker identity always belong to the same recording, thus sharing channel or session information. In this instance, a model trained to leverage channel information to discriminate between speakers may suffer in discriminative performance when tested on utterances from the same recording, or perform poorly on comparing utterances from a single speaker in different recording conditions. Similarly, for a hypothetical dataset of utterances collected from a smart speaker in different households, extracting the specific acoustic conditions of each household may be helpful when classifying between speakers in this dataset, but will not be as useful if later we wish to distinguish between members within the same household.

In general, representations learned from classification tasks may be biased towards the characteristics of that classification task, and the aforementioned leveraging channel information is an example of this. It is an example of a factor which can become a detrimental source of variability for downstream tasks. One reason for this is a mismatch between training conditions and the application in the evaluation phase, where

channel information no longer becomes a discriminative aspect. This mismatch can also apply to factors other than channel information. If we consider other speaker embedding variability factors, it is possible to conceive of train and evaluation situations in which certain factors no longer become discriminative. As an overly simplified example, an evaluation set which contains only North American-accented speakers will not be able to leverage any variability that has been potentially learned based on international English accents. This kind of mismatch, where the distribution of data used during the modelling stage differs from the data distribution seen at inference, has been referred to in the literature as a distribution or domain mismatch (L. Zhang et al. 2017). Techniques for mitigation of this problem generally fall under the field of domain adaptation, and ideas from this field could be applied to speaker representation learning to mitigate any potential mismatch.

As mentioned previously, one of the goals of representation learning is to extract and encode some of the explanatory factors behind the input data. For speaker identity, these factors could include things like gender, age, accent and nationality. In fact, many of these factors are outlined by human experts of speaker recognition tasks in the field of forensic speaker identification (Jessen 2007), where these attributes are often used as evidence for determining speaker identity in law enforcement proceedings. Given an unseen caucasian female American speaker in their thirties, one might expect that the embeddings extracted from this speaker have closer similarity in the embedding space to other speakers who share some of the same demographics; the reason for this being that these attributes contribute to the generative factors that make up a speaker's voice.

An embedding space that can encode speaker-specific attributes is one that we would generally welcome, since we understand that these are strongly related to identity, and if these factors are captured, it should lead to better robustness and generalisability. Indeed, this is preferable to embedding variability based on channel information, which may not be discriminative in many scenarios, such as comparing segments of speech coming from the same recording (diarization), or performing speaker verification between utterances recorded in similar conditions. A potential avenue for encouraging this favourable structure of the embedding space is to use multi-task learning (MTL) (Caruana 1998), whereby learned representations can have improved robustness when they are used to solve different, but related, problems. For example, Parveen and Green (2003) found improvements in ASR by performing speech enhancement and

phonetic classification using the same hidden representation. Similarly, Y. Liu et al. (2019) found that simultaneously learning to classify the phonetic information in an utterance improved speaker embedding performance.

While capturing the underlying factors behind speaker identity and encoding this into a descriptive and discriminative representation is the overall goal for SV and SD, certain applications of speaker embeddings would ideally have these factors be completely interpretable and separable. An example of that would be for data privacy, in which it may be desirable to easily obscure certain sensitive attributes of a speaker, such as their age or gender (Tomashenko et al. 2022). The desirable property which satisfies this is for a representation to be *disentangled*, or for the desired attributes to be separable by dimension within the representation. This raises the question of how to perform this, and manipulate the speaker embedding factors such that they become disentangled.

1.2 Our contribution

This dissertation aims to address some of the issues and areas identified above relating to speaker embedding variability factors. Firstly, in Chapter 2 a detailed technical description of how deep speaker embeddings are trained and extracted is given, in addition with descriptions of how these fit into modern speaker verification and diarization systems. Chapter 3 will then outline the datasets that will be used in future chapters.

In Chapter 4, we propose a method for tackling the problem of channel information in deep speaker embeddings. Using adversarial training, we propose a method that discourages the learned representations from encoding channel information, with the purposes of making these representations more robust to scenarios in which channel information cannot be leveraged.

Chapter 7 details our approach to the distribution mismatch problem by proposing a method that trains an embedding extractor to be robust to different speaker distributions. Furthermore, we propose a method for adapting a network to a target speaker distribution, given the data but no labels for this target data.

In Chapter 5, we utilise multi-task learning and add auxiliary attribute classifiers to speaker embedding network training, with the intent of structuring the learned embedding space to reflect our knowledge of certain specific attributes, namely speaker nationality and speaker age. By doing so, we hope to encourage reliance on speaker

related information, thus improving robustness and generalisability.

Chapter 6 look at disentangling certain speaker attributes within learned representations. Specifically, gender, age and nationality are targeted to be isolated within specific dimensions of a speaker embedding. We propose a method that combines the techniques of Chapters 4 and 5, providing a training framework utilising both multi-task learning and adversarial techniques that can disentangle specific attributes in a supervised manner.

Finally, in Chapter 8 we summarise the findings of the previous chapters, contextualising these contributions within the field, and providing suggestion for how this work could be further expanded and explored.

1.2.1 Corresponding Publications

Chapters 4, 5, 6 and 7 were adapted from the following conference publications respectively:

- Chau Luu et al. (May 2020a). "Channel Adversarial Training for Speaker Verification and Diarization". In: *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7094–7098. DOI: 10.1109/ICASSP40776.2020.9053323
- Chau Luu et al. (Aug. 30, 2021). "Leveraging Speaker Attribute Information Using Multi Task Learning for Speaker Verification and Diarization". In: *Interspeech 2021*. Interspeech 2021. ISCA, pp. 491–495. DOI: 10.21437/Interspeech.2021-622. URL: https://www.isca-speech.org/archive/interspeech_2021/luu21_interspeech.html
- Chau Luu et al. (Sept. 18, 2022). "Investigating the Contribution of Speaker Attributes to Speaker Separability Using Disentangled Speaker Representations". In: *Interspeech 2022*. Interspeech 2022. ISCA, pp. 610–614. DOI: 10.21437/Interspeech.2022-10643. URL: https://www.isca-speech.org/archive/interspeech_2022/luu22_interspeech.html
- Chau Luu et al. (Nov. 1, 2020b). "Dropping Classes for Deep Speaker Representation Learning". In: *The Speaker and Language Recognition Workshop (Odyssey 2020)*. The Speaker and Language Recognition Workshop (Odyssey

2020). ISCA, pp. 357–364. DOI: 10.21437/Odyssey.2020-50. URL: https://www.isca-speech.org/archive/odyssey_2020/luu20_odyssey.html (visited on 10/18/2022)

Chapter 2

Speaker Recognition using Deep Speaker Embeddings

This chapter will describe the relevant methods and work that this dissertation is built upon, providing appropriate context to the work and experiments performed. Specifically, we will cover the mechanisms by which speaker embeddings are extracted, touching on both historically successful methods based on probabilistic generative models, along with the method that this work focuses most on, speaker embeddings extracted with a deep neural network trained on discriminative tasks. After establishing the methods for speaker embedding training and inference, we will also describe some of the downstream tasks that these embeddings are used for. We will cover speaker verification and diarization, and how deep speaker embeddings are typically used in modern systems.

Speaker embeddings are vector representations that encode speaker identity in a discriminative manner given an input utterance of arbitrary length. As touched on in the introduction, the ability to produce a speaker-discriminative vector with a fixed number of dimensions given a segment of speech enables a variety of downstream speaker recognition tasks for unseen speakers, such as verification and diarization.

2.1 Speaker Embeddings: Generative Models

For many years, the most successful method of extracting speaker embeddings was the *i*-vector method, proposed by Dehak et al. (2011). This method built upon a large

body of work that employed factor analysis on top of probabilistic generative models to extract the contribution from speaker- and channel-dependent information in utterances. These relied on Gaussian Mixture Models (GMMs) and the use of the *Universal Background Model* (UBM) method, which we will now describe. For more detail on generative methods, we suggest the reader to refer to the review by Hansen and Hasan (2015).

2.1.1 Gaussian Mixture Models

Gaussian Mixture Models are mixtures of Gaussian probability density functions (PDFs) parameterized by their mean vectors and covariance matrices, along with the weights of each of the mixture components. The overall GMM model is represented as a weighted sum of the individual Gaussian PDFs. For a random vector $\boldsymbol{x}_n$ modelled by M Gaussian PDFs, each with mean vectors $\boldsymbol{\mu}_g$, covariance matrices Σ_g, where $g = 1, 2, \ldots, M$ are the indices for each of the M Gaussian components, the overall PDF of $\boldsymbol{x}_n$ is given by

$$f(\boldsymbol{x}_n|\lambda) = \sum_{g=1}^{M} \pi_g N(\boldsymbol{x}_n|\boldsymbol{\mu}_g, \Sigma_g), \quad \lambda = \{\pi_g, \boldsymbol{\mu}_g, \Sigma_g | g = 1 \ldots M\} \tag{2.1}$$

where π_g is the weight of the g^{th} mixture component, and λ denotes the parameters of the GMM. This PDF can be used to model a sequence of T acoustic feature frames, such as filterbank energies $\mathcal{X} = \{\boldsymbol{x}_n | n \in 1 \ldots T\}$. From the GMM, it is possible to calculate the probability of observing the sequence of acoustic feature frames $\mathcal{X}$:

$$p(\mathcal{X}|\lambda) = \prod_{n=1}^{T} p(\boldsymbol{x}_n|\lambda) \tag{2.2}$$

The parameters λ are trained using the expectation-maximisation (EM) algorithm (Dempster et al. 1977). Note that this assumes acoustic feature vectors to be independent.

2.1.2 Universal Background Model

Gaussian mixture models were first employed for a speaker identification task (speaker classification) in which each speaker in a dataset was modelled using a different GMM, and the likelihood for a test utterance belonging to each modelled speaker was calculated using Equation 2.2. By choosing the most likely speaker to have produced

the test utterance, one could obtain the classification result (D. Reynolds and Rose 1995). However, for verification, which involves determining whether or not a test utterance belongs to a specific enrolment speaker, an *alternate speaker model*, representing speakers other than the enrolment speaker, is required. This need for an alternate speaker model motivated the introduction of a *background* or *world* model to represent all speakers in general. Using a background model, the likelihood of a test utterance belonging to a speaker specific model or the background model can be compared to provide a verification decision. Practically speaking, producing this background model, known as the Universal Background Model (UBM), means training a GMM on a corpus of many speakers.

While the UBM as an alternate speaker model was introduced by D. A. Reynolds (1997), it became popular to use the UBM as the initial model for modelling all speakers, including enrollment speakers. For each enrollment speaker, the UBM parameters could be updated via a Maximum A Posteriori (MAP) adaptation (D. A. Reynolds et al. 2000), which re-estimated the parameters based on observing the enrollment speaker.

2.1.3 GMM Supervectors

While the above methods provided mechanisms for calculating likelihoods for a test utterance based on specific speakers and a likelihood of the *alternate* or *background* speaker, there were still efforts to obtain fixed-dimensional representations for utterances of variable length, since this enabled the use of additional machine learning techniques and classifiers for more downstream tasks (Markel et al. 1977). A method that was proposed for producing a fixed length representation using the GMM-UBM models was to make a *supervector* consisting of the GMM parameters. Specifically, the mean vectors $\boldsymbol{\mu}_g$ of all the mixture components (after performing MAP adaptation) could be concatenated together to provide a fixed dimensional representation of a given utterance (P. Kenny et al. 2003). It was later found that further modelling and estimating latent variables via factor anlaysis within this GMM-UBM supervector space improved speaker recognition performance. There were several works which employed factor analysis in the supervector space, notably

- the Eigenvoice and Eigenchannel formulation (P. Kenny et al. 2003),
- Joint Factor Analysis (JFA) (Patrick Kenny et al. 2007),
- i-vectors (Dehak et al. 2011).

Each had a slightly different formulation of the latent variables which contributed to the observed supervector, but only the *i*-vector method will be covered here. The interested reader can refer to the original publications, or the review by Hansen and Hasan (2015) for more detail.

2.1.4 *i*-vectors

In the *i*-vector factor analysis formulation, an utterance is represented by a *supervector* M that is broken into additive components. The additive formulation of M is as follows

$$\boldsymbol{M} = \boldsymbol{m} + T\boldsymbol{w} \tag{2.3}$$

where $\boldsymbol{m}$ is the speaker- and channel-independent super-vector, taken as the (un-adapted) UBM supervector, and T is a rectangular matrix of low rank and $\boldsymbol{w}$ a random vector having a standard normal distribution $N(\mathbf{0}, I)$. Thus, the observed utterance $\boldsymbol{M}$ is assumed to be normally distributed with mean vector m and covariance matrix $TT^{\intercal}$.

Unlike previous work on Joint Factor Analysis (JFA), the *i*-vector model considered both speaker and channel information jointly with T, meaning that during training, all recordings and utterances were considered separately, and the *total variability* coming from speaker and channel contributed to the observed utterance $\boldsymbol{M}$. This model is essentially a simple factor analysis that enables an utterance to be projected onto the low-dimensional variability space coming from speaker and channel variability, like Principal Component Analysis (PCA) (Pearson 1901). This is particularly valuable since the dimensions of the supervector space for $\boldsymbol{M}$ and $\boldsymbol{m}$ can be very large (if we recall that $\boldsymbol{m}$ is the concatenated mean vectors of each Gaussian component in the UBM). Using a training set of utterances, T can be learned via the EM algorithm, and thus the hidden variable $\boldsymbol{w}$ can be estimated for new utterances by its posterior expectation. The estimates of $\boldsymbol{w}$, also referred to as the *total factors*, are known as the *identity-vectors* or *i-vectors*.

One advantage of the *i*-vector model compared to other factor analysis approaches, such as JFA (Patrick Kenny et al. 2007), which modelled channel and speaker variability separately, was that *i*-vector extractor parameters could be trained using unlabeled data, since the formulation treated channel and speaker variability as a whole, and thus the training utterances needed neither speaker or recording information (Hansen

and Hasan 2015). However, to account for this modelling of both speaker and channel when only speaker variability was needed, channel compensation techniques were often employed on top of *i*-vectors. Examples of such techniques were Linear Discriminant Analysis (LDA), and Probabilistic Linear Discriminant Analysis (PLDA) (Ioffe 2006; Prince and Elder 2007), first used for speaker recognition in combination with *i*-vectors by Patrick Kenny (2010). PLDA is a probablistic version of LDA, which is a supervised method of identifying the linear features that maximise the between-class separation of data and minimise the within-class scatter (Ioffe 2006). The probabilistic version of LDA uses a continuous prior to model this variation, and thus can be employed for unseen classes (for more detail, see Ioffe (2006)).

The result of training a PLDA model is the ability to perform inference about the likelihood of data points coming from the same class (speaker), with the likelihood based on a model which decomposes the total variability of the *i*-vector space into within- and between-class variation. This likelihood is the basis of PLDA scoring.

2.2 Deep speaker embeddings

While *i*-vector-based methods showed state-of-the-art performance for both speaker verification (Torres-Carrasquillo et al. 2017) and diarization (Sell and Garcia-Romero 2014) in the past, in recent years the best performing *single-system* speaker embedding extractors have been based on deep neural networks (Villalba et al. 2020), with *single-system* meaning ensemble methods or system combinations are excluded from consideration.

Early works utilising deep neural networks to learn speaker discriminative features included Konig et al. (1998), who trained a multi layer perceptron (MLP), a feed-forward neural network, to predict speaker class for a fixed length of acoustic feature frames. After training, the classifier portion of the network was discarded, and a GMM was trained on top of the frame-level hidden representations. Another example of early work in this vein was from Heck et al. (2000), who also combined frame-level neural network extracted features with GMMs. Another philosophy for using neural networks for speaker recognition tasks employed Siamese style architectures, training neural networks to classify whether pairs of utterances have the same speaker (K. Chen and Salman 2011; Salman 2012). However, the methods presented in these works were not favoured in comparison to the *i*-vector method, as *i*-vectors at the time still provided

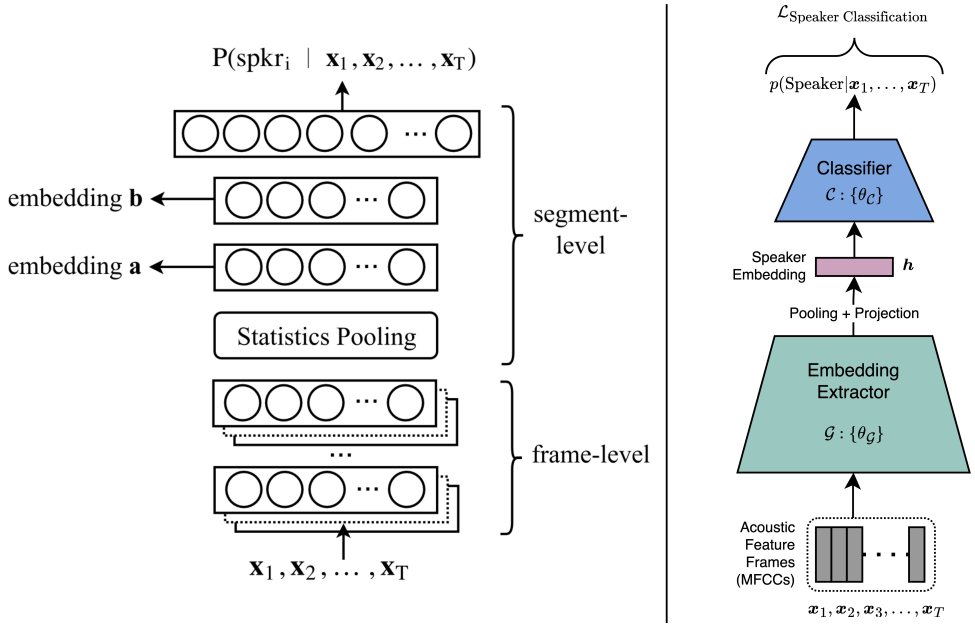

Figure 2.1: Left: Taken from Snyder et al. (2017), the *x*-vector architecture, extracting a fixed dimensional embedding from frame-level input features. Right: the generalised form of how a speaker embedding network is trained, with the classifer $\mathcal{C}$ being discarded during inference.

the best performance for speaker recognition tasks (Snyder et al. 2017).

One of the first works to show better performance than *i*-vectors for speaker recognition, specifically for text-dependent speaker verification, was that of Variani et al. (2014), who trained a DNN to classify speakers per acoustic frame for the phrase 'Okay Google', and then would average the penultimate layer hidden unit activations for each frame in the utterance to produce a speaker discriminative vector representation. Later, Heigold et al. (2016) would use a Siamese architecture on the same 'Okay Google' task, providing a verification result end-to-end, also outperforming the *i*-vector method.

However, one of the most popular works which managed to demonstrate the strength of deep speaker embeddings as a drop-in replacement for *i*-vectors was that of Snyder et al. (2017) and Snyder et al. (2018), who developed a deep neural network based architecture for speaker embedding extraction that could outperform traditional *i*-vectors for text-independent speaker verification. The resulting *x*-vectors, as they were named, were generated by using a neural network architecture composed of several stacked Time Delay Neural Networks (TDNNs) in a pyramid structure (to incorporate a greater

Layer	**Layer Context**	**Total Context**	**Input $\times$ Output**
frame1	$[t-2, t+2]$	5	120×512
frame2	$\{t-2, t, t+2\}$	9	1536×512
frame3	$\{t-3, t, t+3\}$	15	1536×512
frame4	$\{t\}$	15	512×512
frame5	$\{t\}$	15	512×1500
stats pooling	$[0, T]$	T	1500T $\times$ 3000
segment6	$\{0\}$	T	3000×512
segment7	$\{0\}$	T	512×512
softmax	$\{0\}$	T	$512 \times N$

Table 2.1: The *x*-vector architecture from Snyder et al. (2018). The *frame1-6* layers are TDNN layers and *segment6* - *softmax* are fully connected layers. The *x*-vectors are extracted from *segment6*, before the nonlinearity. N refers to the number of training speakers in the classification task. T refers to the total number of acoustic feature frames in an utterance.

temporal context), followed by a *statistics pooling* layer, which took the mean and standard deviation of each of the hidden units in the final TDNN layer across the whole time dimension, and concatenated them together to produce a fixed length representation of an arbitrary length input. With two bottleneck dimension reduction layers in between, the network was trained on a speaker classification task, using cross-entropy loss. This was one of the first works that used speaker classification of the whole utterance as the training objective, as opposed to performing this at the frame level. Using a neural architecture that could pool over a variable length utterance in combination with using an utterance-level classification strategy has been adopted by numerous subsequent work on speaker embeddings (Villalba et al. 2020). The overall *x*-vector architecture can be seen in Figure 2.1[Left] and Table 2.1, where the *x*-vector is extracted from the *segment6* bottleneck layer. Here, the *Layer Context* indicates which frames are concatenated together in the TDNN inputs - this can also be described by the kernel width and stride of a 1-dimensional convolution.

In Snyder et al. (2018), they showed *x*-vectors outperforming i-vectors in speaker verification, particularly in scenarios with larger amounts of training data. With the increasing size of speaker recognition datasets (in terms of the number of speakers and number of utterances), such as VoxCeleb (Nagrani et al. 2017; Chung et al. 2018)(see

also Chapter 3, Section 3.1), extracting features from deep discriminative models has become a particularly prominent approach. It should be noted that henceforth in this work, the term *x*-vectors will be used to describe any deep speaker embedding, and if there is greater similarity with the original *x*-vector formulation, this will be explicitly mentioned.

There are some advantages that *x*-vectors have over their shallower *i*-vector counterparts. One clear advantage is the incorporation of temporal context via the TDNNs, and this can be seen in the increasing temporal context that each subsequent layer possesses in the *x*-vector architecture (Table 2.1). As mentioned previously, the GMM-UBM method, and thus also the *i*-vector method, treats each acoustic frame independently, while the *x*-vector architecture allows the network to capture some temporal dependencies across up to 15 frames. There also is the impressive representational capacity possessed by neural networks (Hinton et al. 2012), which also gives them an advantage over the probabilistic generative models.

2.2.1 Generalised Architecture

While other works have presented variations on the *x*-vector technique and architecture, they all share some basic similarities, which we will outline. From input audio features of some length T, $\mathcal{X} = [\boldsymbol{x}_n, \ldots, \boldsymbol{x}_T]$, the overall output of the network is $p(\text{Speaker}|\mathcal{X})$, used to predict the speaker of the utterance in a multi-class classification task with S classes, where S is the number of speakers seen in the training dataset. The network architecture can be broken down into two main sections:

- Embedding extractor network $\mathcal{G}$, parameterised by θ_G processes the frame-level features, pools this to a segment-level representation, and finally projects the input features $\mathcal{X}$ into a fixed length vector $\boldsymbol{h} = \mathcal{G}(\mathcal{X})$.
- A classification portion of the overall network $\mathcal{C}$, parameterized by θ_C, predicts the speaker $p(\text{Speaker}|\boldsymbol{h})$.

A visualisation of the can be seen in Figure 2.1[right]. Both $\mathcal{C}$ and $\mathcal{G}$ are trained as a whole, but embeddings for unknown speakers are extracted from $\mathcal{G}$ alone, meaning that $\mathcal{C}$ is often discarded after training time. The result of the discriminative training objective is that the embedding vector $\boldsymbol{h}$ is a speaker-discriminative representation for any given input utterance. Comparing this general formulation with the *x*-vector formulation in Table 2.1, the frame-level processing is performed by stacked TDNNs

(*frames1-6*), and the pooling method is the statistics pooling method, concatenating the average and standard deviation of the frame-level features. Since an *x*-vector is extracted from the *segment6* layer, everything from *frame1* to *segment6* is considered part of $\mathcal{G}$, and [*segment6,softmax*] are $\mathcal{C}$.

2.2.2 Loss Functions

The objective function that the original *x*-vector architecture was trained on was the standard classification cross-entropy (softmax) loss. For a batch size N with S classes, this loss is defined as:

$$L_{\text{softmax}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{W_{y_i}^{\intercal}h_i+b_{y_i}}}{\sum_{j=1}^{S}e^{W_j^{\intercal}h_i+b_j}} \tag{2.4}$$

where $\boldsymbol{h}_i \in \mathbb{R}^d$ represents the d-dimensional deep feature or embedding, such as the input to the final projection layer of a feed-forward network, of the i-th sample. This $\boldsymbol{h}_i$ belongs to the class y_i. $W_j \in \mathbb{R}^d$ is the j-th column of the weight matrix $W \in \mathbb{R}^{d\times S}$ with $b_j \in \mathbb{R}^S$ the bias term. This loss encourages a high softmax probability for the correct class, pushing that towards 1, whilst pushing all others towards 0.

However, this does not explicitly encourage the deep feature to be similar to other members of its class, or increase the dissimilarity between it and other classes. As such, this motivated work to modify the traditional softmax loss to decrease intra-class spread and increase inter-class distance. The work of W. Liu et al. (2017), H. Wang et al. (2018), and Jiankang Deng et al. (2018) introduce a formulation of the softmax loss based on an interpretation of the hidden representation $\boldsymbol{h}$ and the final weight matrix W as having an angular decision boundary for predicting the target class. This family of losses are often referred as *angular penalty or angular margin losses*.

In the angular margin formulation, Equation 2.4 is modified such that the bias term is removed ($b_j = 0$) and both the weight and deep features are l_2 normalized and scaled such that $||W_j|| = s$. By rearranging the formula for cosine similarity, the following equation is obtained:

$$W_j^{\intercal}\boldsymbol{x}_i = ||W_j||||W_j||\cos\phi_j \tag{2.5}$$

where ϕ_j is the angle between W_j and $\boldsymbol{x}_i$. Replacing equation 2.5 into 2.4, along with

the normalized and scaled length s provides the following loss:

$$L_{\text{sphere}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s\cos\phi_{y_i}}}{e^{s\cos\phi_{y_i}} + \sum_{j=1, j\neq y_i}^{S} e^{s\cos\phi_j}} \tag{2.6}$$

which is the softmax loss but with all embeddings projected onto a hypersphere of length s. Here an additive angular margin penalty m can be added such that the inter-class discrepancy is large, which also enhances the intra-class compactness:

$$L_{\text{Arcface}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s\cos(\phi_{y_i}+m)}}{e^{s\cos(\phi_{y_i}+m)} + \sum_{j=1, j\neq y_i}^{S} e^{s\cos\phi_j}} \tag{2.7}$$

One can see that in comparison to the 'vanilla' softmax (equation 2.4), the target 'probability' of the true class, indicated by the numerator, has an additional strictness imposed on it with the addition of m. This loss is very closely related to the Additive Margin softmax (F. Wang et al. 2018) and SphereFace (W. Liu et al. 2017), which have similar properties:

$$L_{\text{AMSoftmax, CosFace}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s(\cos(\phi_{y_i})-m)}}{e^{s(\cos(\phi_{y_i})-m)} + \sum_{j=1, j\neq y_i}^{S} e^{s\cos\phi_j}} \tag{2.8}$$

$$L_{\text{SphereFace}} = -\frac{1}{N}\sum_{i=1}^{N}\log\frac{e^{s\cos(m\phi_{y_i})}}{e^{s\cos(m\phi_{y_i})} + \sum_{j=1, j\neq y_i}^{S} e^{s\cos\phi_j}} \tag{2.9}$$

2.2.3 Notable Embedding Architectures

While the basics of the *x*-vector architecture have been detailed above , these are not the only methods and architectures that have been explored in literature for deep speaker embeddings. For example, as a result of its success, there are several papers which are conceptually similar to that of *x*-vectors, with some being direct and clear variations of the original architecture. An example of an *x*-vector architecture variation is in the work of Zhu et al. (2018) and Okabe et al. (2018), which attempt to improve upon the statistics pooling layer of the *x*-vector architecture by adding self-attention to the pooling. In Zhu et al. (2018), an attention vector is learned across the sample, based on the input to the pooling layer. With this attention vector, a weighted average of those features is taken, which can be considered to be selecting more discriminative frames of speech. This can be referred to as the attentively weighted average. Okabe et al.

(2018) have extended this idea further by employing the use of the attention vector to further calculate an attentively weighted standard deviation, which is also incorporated into the pooling layer, fully reconstructing an attentive version of the original statistics pooling method in (Snyder et al. 2018; Snyder et al. 2017). Both works found improvement in speaker verification using their methods over the original *x*-vector, with Zhu et al. (2018) finding particular gains when testing on longer utterances (>30s). The reasoning given for this improvement was the ability of the attention mechanism focus only on important frames, which may be more necessary with longer utterances.

Of course, the *x*-vector method (generally characterized by a pyramid of stacked TDNNs), is not the only proposed architecture available for obtaining a fixed dimensional representation from a variable length input. Wan et al. (2020) and Quan Wang et al. (2018) employ a 3-layer Long Short Term Memory (Hochreiter and Schmidhuber 1997) (LSTM) with projection of the context vector (Sak et al. 2014) to generate their embeddings.

Xie et al. (2019) adopts a computer vision inspired approach to speaker verification, by using a network architecture more commonly used in image recognition to a sliding window spectrogram of the input utterance. This network architecture begins with a modification of ResNet (He et al. 2016), essentially a number of stacked convolutional neural network blocks (with residual connections), followed by a pooling method known as Ghost/NetVLAD (Arandjelović et al. 2016) (Vector of Locally Aggregated Descriptors), which was first introduced for the task of place recognition (recognising a location from a query image). This pooling layer stores the sum of softly assigned residuals of each input feature to a set of learn-able cluster centres. The authors argue the pooling layer can automatically learn to aggregate discriminative information across an input.

The use of larger networks with residual connections has been increasingly common in recent work (Villalba et al. 2020) and speaker recognition competitions[1].

While the slight variations to softmax (F. Wang et al. 2018; W. Liu et al. 2017; Jiankang Deng et al. 2018; H. Wang et al. 2018) have been mentioned above, some works propose methods that train on the embeddings themselves using contrastive losses such as triplet loss (Song et al. 2018), or a generalised version of triplet loss (Wan et al. 2020), but the general concept is the same, in that the network is trained to discriminate be-

[1] `https://sdsvc.github.io/mydescriptions/`

tween all speakers in the training set.

2.3 Speaker Recognition

Now that we have outlined speaker embeddings and the means of training deep speaker embedding extractors, it is necessary to also outline the downstream speaker recognition tasks that these embeddings are used for, and why those tasks are useful. In this work, the tasks of speaker verification (SV) and speaker diarization (SD) are explored.

2.4 Speaker Verification

Speaker Verification (SV) is the task of determining whether a test utterance was spoken by a hypothesised speaker, given their pre-recorded utterance(s). This task has several practical uses in modern technology. One common example is that of biometric authentication, in which a person's identity must be confirmed based on their distinctive physical characteristics (Hansen and Hasan 2015).

Most, if not all modern smartphones will have some method of biometric authentication, such as a fingerprint reader or a means of performing reliable facial recognition (Shabeer and Suganthi 2007). These examples of biometric authentication have become integral to phone usage, enabling greater security and privacy. A person's voice is also an aspect which is distinctive and shaped by physical characteristics, and thus is also considered to be a candidate for performing biometric authentication (A. Jain et al. 2004). Voice as a biometric however does present more challenges in comparison to the aforementioned fingerprint or facial biometrics, as while the distinctive physiological aspects of the vocal apparatus may be relatively consistent, there are additional volatile factors that can affect speech, such as speaker emotion and effort, medical conditions (like a common cold), in addition to challenging background acoustic conditions[2] (A. Jain et al. 2004, p.7). Nonetheless, it has been shown that multimodal biometric systems that combine evidence from multiple sources of information are more robust than unimodal systems, since they are less affected by noise in sensed data, in addition to being more resistant to spoofing attacks (Ross and A. K. Jain 2004). As a result, voice can contribute evidence to a multimodal biometric authenti-

[2]However, facial recognition from images must also deal with environmental factors such as variable pose and lighting conditions (A. Jain et al. 2004, p.7)

cation system, even if authenticating based on voice alone may be more challenging and potentially less secure than other biometrics.

Furthermore, with the increased usage and capabilities of smart voice assistants, the necessity for voice authentication has also grown. For example, dictating and sending text messages, or purchasing goods through an online marketplace are both actions that can be performed via voice assistants, but also pose the potential for bad outcomes if the authorized user is not the one performing them. While identity verification could be supported by other biometrics in this situation, authenticating based on the utterance itself is valuable from a user experience and convenience perspective (Renz et al. 2022). As such, the need for robust and reliable speaker verification systems is clear.

The overall structure of a speaker verification system is shown in Figure 2.2. This is composed of several components, which we will define here.

- **Test utterance**: this is the utterance under investigation, for which we are evaluating the claim that it was uttered by the target speaker.
- **Enrolment utterance(s)**: The pre-recorded utterance(s) belonging to the target/enrolment speaker.
- **Verification model**: The verification model, which has already been trained. (Typically, the enrolment speaker should not have been seen during training).
- **Score**: Scalar output of the verification model that compares the test utterances to the enrolment speaker.
- **Scoring threshold**: A decision threshold by which to *accept* the test utterances as belonging to the enrolment speaker, or *reject* it as being a different speaker.

In speaker verification literature, the task is also often split into two cases:

- **Text-dependent SV**: This is speaker verification in which the test and enrolment utterances are a specific phrase or set of word(s). This case is useful in certain domains, such as the person-specific activation of voice assistants using a specific key word.
- **Text-independent SV**: This is the unrestricted case, in which the test and enrolment utterances can be any word or phrase.

In this work, we only explored the text-independent SV task. While the increased variability of the observed speech in this task does mean that lexical factors can play

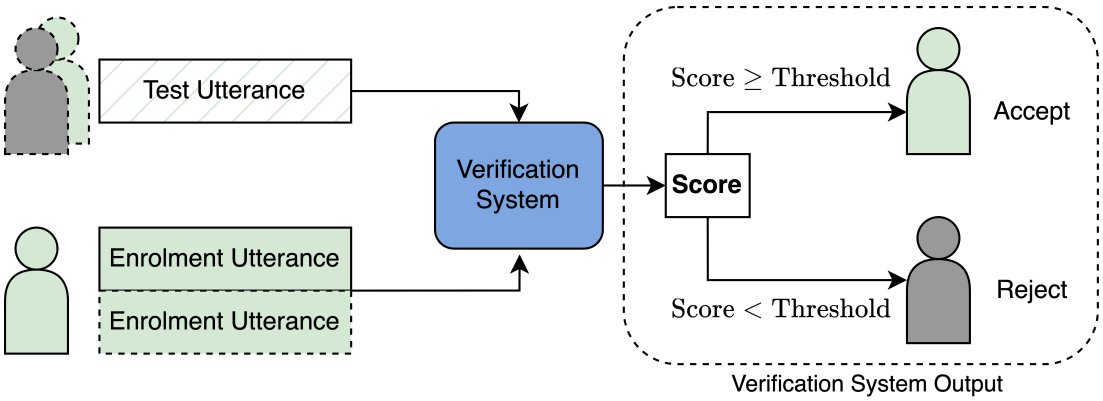

Figure 2.2: The process of speaker verification.

a role in the recognition, the text-independent SV task typically has much more data available, and the models used for this task match up better with diarization, which is always text-independent.

2.4.1 Metrics

Speaker verification is a task with a binary decision, *accept* or *reject*. As such, it uses metrics which are common in other binary classification tasks. Here we will describe the following terms/metrics:

- Receiver Operating Characterstic (ROC) curve
- Area under [ROC] Curve (AUC)
- Equal Error Rate (EER)

2.4.1.1 Receiver Operating Characteristic (ROC) and AUC

The Receiver Operating Characteristic (ROC) curve shows the performance of a binary classifier as a decision threshold is varied. This is achieved by plotting the true positive rate (TPR) against the false positive rate (FPR) at a range of threshold values. The equations for TPR and FPR are as follows:

$$\mathrm{TPR} = \frac{\mathrm{TP}}{\mathrm{TP} + \mathrm{FN}} \tag{2.10}$$

$$\mathrm{FPR} = \frac{\mathrm{FP}}{\mathrm{TN} + \mathrm{FP}} \tag{2.11}$$

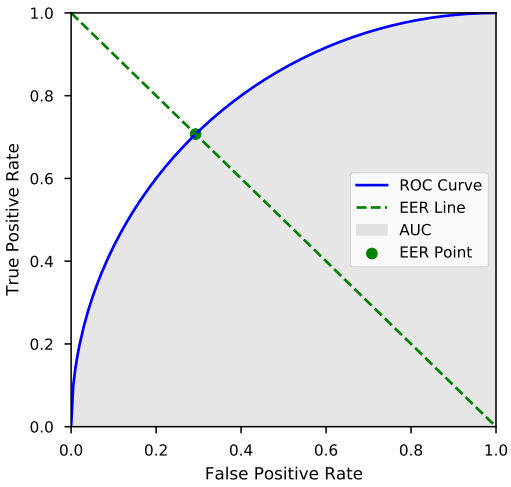

Figure 2.3: Demonstration of Equal Error Rate with respect to a ROC Curve. The EER point is when $FPR = 1 - TPR$. In this figure the EER is equal to $\sim 29.3\%$.

where TP is the number of True Positives identified by a classifier, FN the number of False Negatives, FP the number of False Positives and TN the number of True Negatives. An example of a ROC curve can be seen in Figure 2.3. Binary classifiers are also often evaluated based on the Area Under the (ROC) Curve, or the AUC. Here, a larger AUC indicates a binary classifier with better performance.

2.4.1.2 Equal Error Rate (EER)

The Equal Error Rate is an important metric in verification tasks. The equal error rate is the value at which the false acceptance rate and false rejection rate are equal. The equal error rate occurs when FPR is equal to $1 - TPR = FNR$. This corresponds to finding the intersection point between the ROC curve and the line $y = 1 - x$. This is also demonstrated in Figure 2.3.

2.4.2 Verification using embeddings

As indicated in Figure 2.4, verification with speaker embeddings is fairly straightforward, in that embeddings are extracted for test and enrollment utterances (after feature extraction and speech activity detection), and scoring is performed using these vectors. The method of scoring can vary from simply taking the cosine or Euclidean distance, or training a separate model to discriminatively score the embeddings. More detail will be given in sub-section 2.6.3, and this is also explored in Chapter 4.

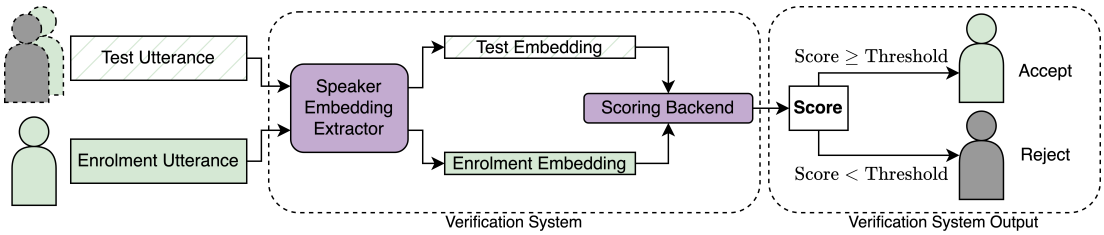

Figure 2.4: The process of speaker verification using speaker embeddings.

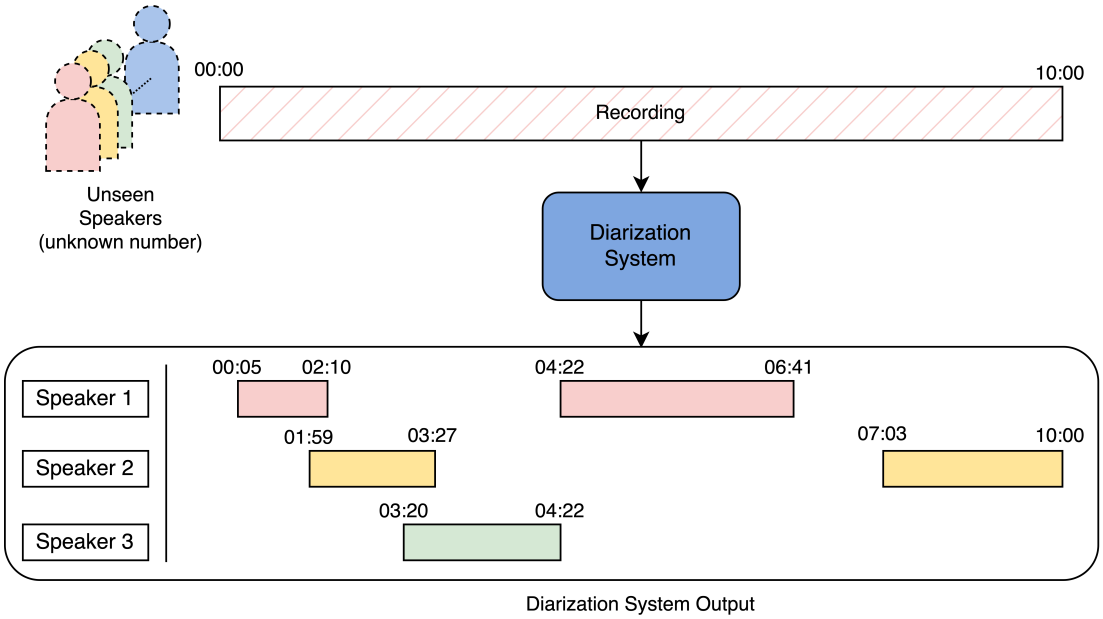

Figure 2.5: The process of speaker diarization.

It should be mentioned that not all verification systems use this pipeline, in that SAD is not always performed (Xie et al. 2019), and other architectures have been presented which also perform scoring, combining several modules in the above pipeline, most notably in terms of incorporating the scoring backend into the network architecture (Heigold et al. 2016; Ramoji et al. 2020; Ramoji et al. 2020).

2.5 Speaker Diarization

Speaker Diarization (SD) is the task of determining 'who spoke when?'. What this means is to be able to segment speech within a recording of unknown speakers and group these segments according to the speaker responsible for each speech segment. An overview of the task can be seen in Figure 2.5. As we can see, the output of the diarization system has highlighted segments of the recording which are speech uttered by the same speaker. Notice also that the system has left blank regions in which it

has hypothesized that no speech is present. There are several sub-tasks that diarization systems must perform. For example, only the regions where speakers are active should be highlighted (speech activity detection). In addition, a diarization system is typically evaluated on recordings with an unknown number of speakers, and so the number of speakers must also be ascertained accurately (speaker counting).

Diarization has several practical applications in modern technology. An example would be to use diarization in conjunction with an Automatic Speech Recognition (ASR) system to attribute words spoken to distinct speakers. Examples of this application might be to automatically colour-code subtitles in broadcast media according to speaker, or to produce speaker-attributed transcripts of interviews, business meetings, law proceedings, or parliamentary discussion.

2.5.1 Metrics

The output of a diarization system is primarily evaluated by the Diarization Error Rate (DER). This is a sum of three terms:

$$\text{DER} = \frac{\text{Speaker Confusion} + \text{False Alarm} + \text{Missed}}{\text{Total Duration}} \tag{2.12}$$

where

- **Speaker Confusion** refers to the duration assigned by the system to an incorrect speaker. This is also referred to in literature as the speaker error time.
- **Missed** refers to the duration of that speech is present, but has not labeled as speech by the diarization system's hypothesis. In the traditional diarization pipeline, this error is caused by incorrect labelling in the speech segmentation (SAD) step.
- **False Alarm** refers to the duration that a speaker has been labeled by the system, but no speech is present. Like Missed Speaker Time, this step usually occurs due to incorrect speech activity detection.

Calculating the Speaker Confusion can be difficult, as it is not known which speakers hypothesized by a given diarization system correspond to the speakers in a reference. This is a problem when the number of speakers hypothesized is different to the number of speakers in the reference. Thus, an *optimal* one-to-one assignment of each hypothesis speaker to a reference speaker is needed before calculating the DER (Sinclair

2016). This optimal mapping is usually calculated using the Hungarian algorithm, but in cases where there are a very large number of reference and hypothesis speakers and computational time may become an issue, a greedy mapping can be performed, which maps reference and hypothesis speakers iteratively based on decreasing value of the co-occurrence duration.

To further measure the performance of a diarization system, the *Speaker Counting Accuracy* can also be used:

$$\text{Speaker Counting Accuracy} = \frac{\text{num. of recordings with correct num of speakers predicted}}{\text{num. of recordings}} \tag{2.13}$$

which gives an indication of the speaker counting performance of the system.

2.5.2 Diarization using embeddings

The modular speaker diarization pipeline using embeddings is displayed in Figure 2.6. Here, one can see how each system is fed sequentially into each other to produce a diarization hypothesis. It is also clear to see how similar this approach is to the verification pipeline, the main conceptual difference being that verification is effectively performed on all pairs of segments, using clustering to produce the final output.

While this modular clustering approach does produce very strong results, with the best performing system at the third Diarization is Hard (DIHARD) challenge (Ryant et al. 2021) using a clustering based system (Y. Wang et al. 2021), the approach does have some conceptual drawbacks. Firstly, the temporal nature of the recording, including any dynamics relating to speaker turns, is not considered when clustering. Furthermore, the lexical information that could inform things like speaker change is not considered. These are drawbacks which motivate end-to-end neural architectures for diarization and the closely related task of speaker attributed ASR (Y. C. Liu et al. 2021; Kanda et al. 2021; Horiguchi et al. 2020; Horiguchi et al. 2021), where ASR is performed jointly with speaker prediction, thus combining both lexical and speaker information when making predictions about both.

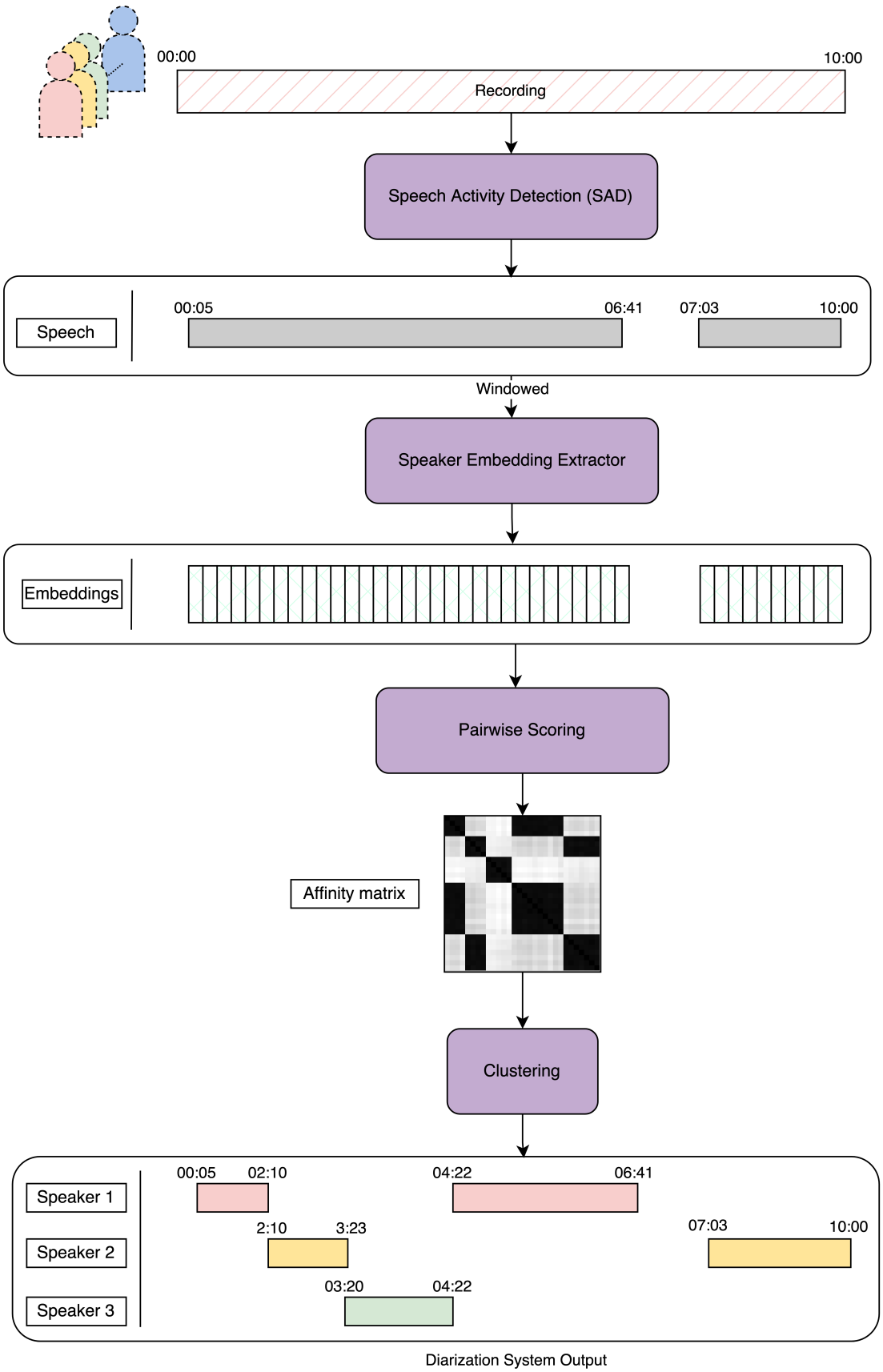

Figure 2.6: The process of speaker diarization using speaker embeddings.

2.6 Components in speaker recognition systems

This work primarily focuses on speaker embeddings, which have become a key component in many verification and diarization systems. By using speaker embeddings, verification and diarization systems take on a modular approach to tackling their respective problems, using speaker embeddings as a means to distinguish between segments of audio. While there are surrounding components (or modules) that go around the speaker embeddings, these modular systems rely heavily on having discriminative embeddings. Here, we will describe the some of the surrounding components in modular verification and diarization system.

2.6.1 Acoustic Feature Frame Extraction

Common to most speech processing tasks is an acoustic feature extraction step, in which the raw waveform audio is converted into some other more compact and informative features. This processing step is often referred to as the feature extraction frontend. Examples of these features are filterbank energies or Mel Feature Cepstral Coefficients (MFCCs) (C.-h. Chen 1976).

For filterbank based features, the discrete Fourier transform (DFT) by means of the fast Fourier transform (FFT) is taken for overlapping windows of the audio, after which a series of triangular filters is applied. These filters are distributed along the frequency scale logarithmically, which is motivated by the characteristics of the human auditory system. One such scale, which is popularly used, is the Mel scale. Here, the energy from these filter-banks can be taken directly, or further processed into MFCCs. These features have the advantage of being inexpensive to compute, along with being highly compact, representing speech adequately for downstream tasks in far fewer parameters than the raw waveform.

Other feature extraction methods have also achieved some popularity within speaker recognition literature, such as modelling directly from the raw waveform using Sincnet convolutions (Ravanelli and Bengio 2019; Jung et al. 2019; Jung et al. 2020), which are windowed filterbank extractions, implemented via the convolution operation, wherein the breadth and position of the filter is learnable via back-propagation. This is almost equivalent to filterbank energies (and often the learnable parameters are initialised similarly to the Mel scale), with the added advantage being that the filterbank scale can be learned to best suit any downstream tasks. Compared to the filterbank methods how-

ever, the addition of more learnable parameters and computation steps results in longer computation time along with higher memory costs.

2.6.2 Speech Activity Detection

Speech Activity Detection (SAD), also referred to as Voice Activity Detection (VAD), is the task of segmenting a recording according to whether or not there was speech present. This is an extremely important step in diarization, as it contributes directly to missed speaker time and false alarm time. For verification, this step is also often used in order to remove noise frames such that only speech is present for embedding extraction. For both tasks, SAD is the preparatory step before feature embedding extraction.

An extremely simple way of performing SAD is via energy-based methods, which rely on measuring the spectral energy within a window of the audio data, and making a decision based on some predefined threshold as to whether or not speech is present. This relies on the assumption that regions with speech have higher spectral energy than ordinary background noise.

While this method has advantages, being simple to implement in addition to being computationally inexpensive, it may break down in more challenging scenarios. For example, the previously stated assumption about the spectral energy of speech regions may not be true for distant microphone recordings, or recordings with loud background noise, such as the noise coming from an air conditioner. In addition, energy based methods do not incorporate any longer term temporal information outside of the energy in the window they are considering, meaning they lack the ability to make more sophisticated decisions based on context.

The current state of the art with regards to SAD, like many other fields, is also achieved by using deep neural networks (Veysov 2022) which can make more complex decisions using longer term context. However, in this work SAD was not a primary focus, and so only simple energy based SAD was employed during experiments. Furthermore, for diarization tasks, often the ground truth speech segmentation was used, in order to focus only on the contribution of speaker embedding performance.

2.6.3 Scoring

If we assume the existence of speaker-discriminative embeddings from an arbitrary segment of audio, the next step is to score the similarity between these embeddings.

This component is often referred to as a scoring backend.

The most simple and straightforward means of scoring is a non-parametric distance metric, such as the euclidean distance between two vectors $\boldsymbol{h}_a, \boldsymbol{h}_b$ or the cosine distance, which is equivalent to one minus the cosine similarity:

$$d(\boldsymbol{h}_a, \boldsymbol{h}_b) = 1 - \frac{\boldsymbol{h}_a \cdot \boldsymbol{h}_b}{||\boldsymbol{h}_a|| ||\boldsymbol{h}_b||}, \quad \text{Cosine distance} \tag{2.14}$$

While there are many possible options for non-parametric distance metrics, cosine distance/similarity is seen very frequently in verification literature (Xie et al. 2019; Villalba et al. 2020; Qiongqiong Wang et al. 2022).

The most common parametric scoring backend used in speaker recognition is Probabilistic Linear Discriminant Analysis (Ioffe 2006; Prince and Elder 2007) (PLDA). PLDA is a probabilistic version of Linear Discriminant Analysis (LDA), which is a supervised method of identifying the linear features that maximise the between-class separation of data and minimise the within-class scatter (Ioffe 2006). The probabilistic version of LDA uses a continuous prior to model this variation, and thus can be employed for unseen classes (for more detail, see Ioffe 2006). The PLDA formulation models the *i*-vector space, separating out class-dependent and session-dependent variabilities, with both lying in lower-dimensional subspaces. For an *i*-vector $\boldsymbol{w}_{i,j}$ from a session j of the speaker i:

$$\boldsymbol{w}_{i,j} = \boldsymbol{w}_0 + V\boldsymbol{\beta}_s + \boldsymbol{\varepsilon}_{i,j} \tag{2.15}$$

where $\boldsymbol{w}_0$ is the speaker-independent mean *i*-vector, V the low-rank matrix representing the speaker-dependent variability, $\boldsymbol{\beta}_i$ is the speaker factor vector, with $\boldsymbol{\varepsilon}_{i,j}$ representing the residual noise that accounts for the variability between different sessions of the same speaker. The speaker factor $\boldsymbol{beta}_i$ containing the speaker information is assumed a priori to be standard normal distributed. The prior distribution of $\varepsilon_{i,j}$ is modeled with a full covariance Gaussian:

$$\boldsymbol{\varepsilon}_{i,j} \sim \mathcal{N}(\boldsymbol{\varepsilon}_{i,j} | \boldsymbol{0}, S_c) \tag{2.16}$$

where S_c is the within-class covariance. The between class covariance is computed as $S_b = VV^\intercal$. PLDA scoring is performed by calculating the ratio between the likelihood

of the trial *i*-vector, given the target hypothesis and the corresponding likelihood given the non-target hypothesis.

If the speaker of the test and enrollment utterances are the same (hypothesis $\mathcal{M}_0$), they share the same speaker factor vector $\boldsymbol{\beta}$, whereas if they are different speakers, then the speaker factor vectors are different for both (hypothesis $\mathcal{M}_1$). Thus the likelihood ratio of $\mathcal{M}_0$ and $\mathcal{M}_1$ is:

$$\begin{aligned} R(w_1, w_2) &= \frac{P(w_1, w_2|\mathcal{M}_1)}{P(w_1, w_2|\mathcal{M}_0)} \\ &= \frac{\int P(w_1, w_2|\beta_1) P(\beta_1) d\beta_1}{\int P(w_1|\beta_1) d\beta_1 \int P(w_2|\beta_2) d\beta_2} \end{aligned} \tag{2.17}$$

where the speaker factors are integrated out, meaning the likelihood ratio is computed taking into account the uncertainty about the value of β. The PLDA model parameters are trained via the EM algorithm (Ioffe 2006).

However, while PLDA and variants of PLDA are still used by some in conjunction with scoring deep speaker embeddings (Ramoji et al. 2020; Ramoji et al. 2020; Qiongqiong Wang et al. 2022), it has been noted that a gradual shift away from parametric backends like PLDA has occurred in the literature (Qiongqiong Wang et al. 2022). With this shift, it has been suggested that the strength of some deep learning based embeddings render the decomposition of within- and between-speaker variability unnecessary.

Score normalization is often performed in order to calibrate modern speaker verification systems (Matějka et al. 2017). Without normalization, different distributions for the target and impostor scores can be obtained for different enrolled speakers. This means that setting a global decision/detection threshold is very difficult. Score normalization attempts to fix this by shifting the distributions for individual enrolment speakers, such that a single decision threshold can be used for all trials. There is a wealth of literature showing that score normalization provides a significant improvement to verification systems (Matějka et al. 2017; Shum et al. n.d.). By and large, these normalisation methods work by utilising a *cohort* of utterances from speakers assumed to be different to the enrolment and test speakers (usually training set speakers and utterances). By calculating the scores of the enrolment (and/or test) utterances with the cohort, the distribution of scores can be normalised. Since score normalisation was not employed for any experiments in this thesis, this description has been kept brief, but the interested reader should refer to Matějka et al. (2017).

2.6.4 Clustering

For diarization, in order to go from being able to score pairs of embeddings to grouping these pairs into distinct speakers, clustering is required. Specifically, the score of every pair of segments in a recording is calculated to populate an affinity matrix, upon which a clustering algorithm can be employed.

Agglomerative Hierarchical Clustering (AHC) (McQuitty 1957) is a common clustering method used for diarization. This is a hierarchical clustering method that takes the 'bottom-up' approach. What is meant by this is that the clustering process is initialized with each point belonging to its own cluster. Clusters are then merged based on their similarity. If the number of clusters/speakers is not known, which in diarization they typically are not, a pre-defined threshold will determine when to stop merging points into clusters.

Another popular method used in diarization systems is Spectral clustering (Quan Wang et al. 2018; Raj et al. 2020; Park et al. 2020; Lin et al. 2019), which uses the largest eigenvalues of the affinity matrix to perform dimensionality reduction, before then clustering using k-means in these fewer dimensions. In the case of an unknown number of clusters, an eigenvalue threshold is chosen to select k in k-means.

For both AHC and spectral clustering, hyper-parameter thresholds are learned by sweeping over a range of parameters to obtain the best fit to training or development data.

2.7 Conclusion

This chapter has provided an overview of the speaker recognition techniques used in subsequent chapters, also describing the relevant work that form the foundation that these chapters build upon.

Additional techniques will be outlined in future chapters closer to the relevant material, particularly adversarial training, covered in Chapters 4 and 6, and Multi-Task Learning (MTL) in Chapters 5 and 6.

Chapter 3

Datasets

This chapter will describe the datasets and corpora used as part of this thesis. As certain datasets are re-used across multiple chapters, this chapter will provide a singular source to refer to in order to avoid the duplication of information.

3.1 VoxCeleb

VoxCeleb, by Nagrani et al. (2017), and VoxCeleb 2, by Chung et al. (2018) are two large-scale audio-visual speaker recognition datasets collected via open-source media, specifically YouTube videos. Both employ a fully automated pipeline in order to collect videos from celebrity speakers, hence the term *Celeb*.

By searching for celebrity names on YouTube and downloading related videos, their automated pipeline combines face detection, face tracking, active speaker verification and face verification to obtain utterances for a given celebrity speaker from the videos. The thresholds for active speaker verification and face verification are all chosen conservatively as to ensure that the automatically obtained utterances belong to the person of interest.

Due to the fact that VoxCeleb was collected using an automated pipeline, it was able to provide much more data for speaker recognition tasks that previous publicly available datasets (Nagrani et al. 2017). This was reflected in a large number of different speakers, in addition to the large number of utterances in both VoxCeleb 1 and 2 (see Table 3.1).

Dataset	**VoxCeleb1**	**VoxCeleb2**
# of POIs	1,251	6,112
# of male POIs	690	3,761
# of videos	22,496	150,480
# of hours	352	2,442
# of utterances	153,516	1,128,246
Avg # of videos per POI	18	25
Avg # of utterances per POI	116	185
Avg length of utterances (s)	8.2	7.8

Table 3.1: (Taken from Chung et al. (2018)) Dataset statistics of VoxCeleb 1 and VoxCeleb 2. POI: Person of Interest

		Dev	**Test**	**Total**
VoxCeleb 2	# of POIs	5,994	118	6,112
	# of videos	145,569	4,911	150,480
	# of utterances	1,092,009	36,237	1,128,246
VoxCeleb 1	# of POIs	1,211	40	1,251
	# of videos	21,819	677	22,496
	# of utterances	148,642	4,874	153,516

Table 3.2: (Taken from Nagrani et al. (2017)Chung et al. (2018)) The dev and test splits of VoxCeleb 1 and 2 (for speaker verification).

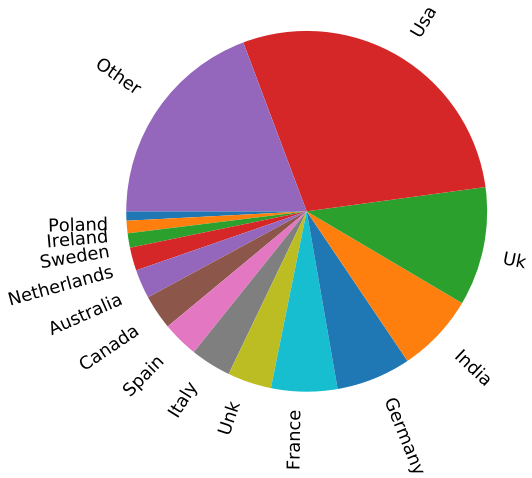

Figure 3.1: The web-scraped nationality for speakers in the VoxCeleb 2 training, only nationalities with 50 or more speakers are shown, with nationalities with less than this grouped as 'Other'.

Furthermore, the multimedia nature of the source data (YouTube videos) meant that it also provided much more varied acoustic and recording conditions than previous datasets, which had typically been collected from clean speech, telephone conversations or meeting room recordings using manual human annotation. The varied nature of this kind of data has been referred to as being collected *in the wild*.

The VoxCeleb 1 creators provide labels for both gender and nationality of the speaker, whereas VoxCeleb 2 only provides gender labels. Due to the fact that each speaker in VoxCeleb is a public figure or celebrity, the missing nationality labels for VoxCeleb 2 can be scraped from the web in a fairly rudimentary fashion, using Wikipedia summaries for example. This results in potentially noisy labels of a total of 125 nationalities, including 236 out of 5994 speakers being marked as having an 'unknown' nationality. The full distribution can be seen in Figure 3.1, with 'Unk' (unknown) being the 6th most common nationality.

3.2 SCOTUS

The Supreme Court of the United States (SCOTUS) is the highest level court in the United States, with many historic cases being argued in its over 200 year history (established 1789). Since 1955, audio recordings have been made of the oral arguments

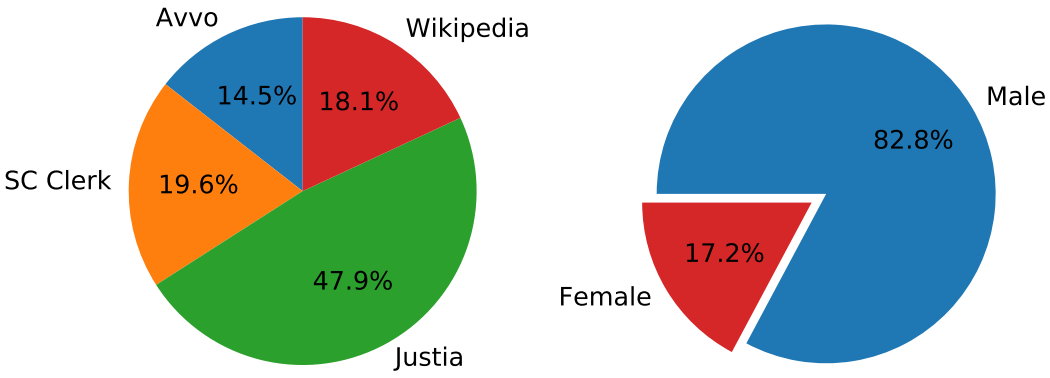

Figure 3.2: The web-scraped source of the age labels, along with the gender distributions of the speakers in the SCOTUS

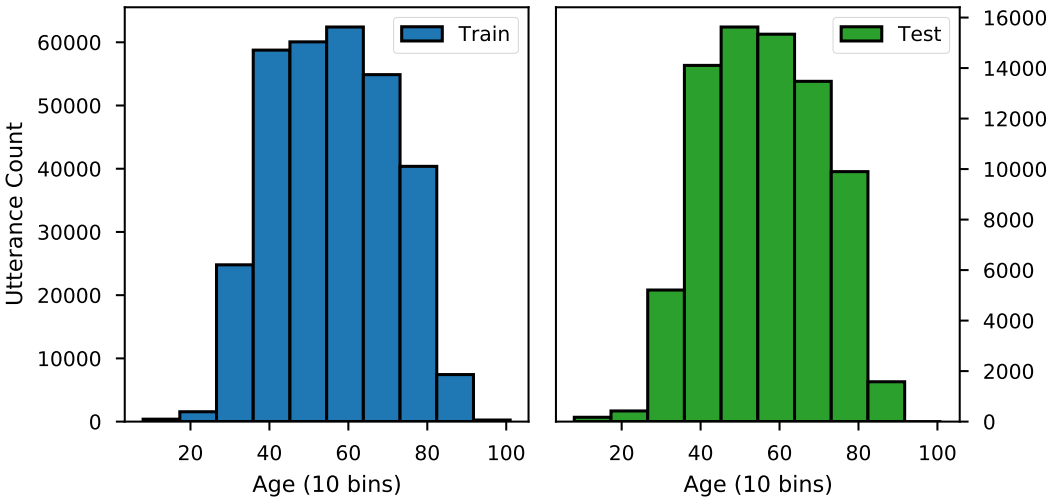

Figure 3.3: The speaker age distribution across 10 bins of utterances in the SCOTUS corpus (October 2005 onwards), split into train and test sets.

for the cases brought before the Supreme Court, along with full transcriptions. These recordings and their transcriptions have been made available via the Oyez project[1]. From October 2005, these recordings switched to a digital recording system, as opposed to reel-to-reel taping system. Due to a number of issues that the tape recordings exhibited[2], only the digital recordings were considered for experiments in this chapter. At the time of writing, this consisted of 1022 recordings, with 913 unique speakers across this set of recordings. These recordings amount to 1032 hours of audio.

The SCOTUS corpus is somewhat unique in that it features certain speakers across many years. These speakers are the justices (judges) of the Supreme Court, who serve on the court until their retirement. The average length of a supreme court justice's tenure at time of writing is approximately 17 years. This makes age a speaker attribute that is both possible and interesting to explore for this dataset, as it is uncommon to have the exact age of the speakers at the time of recordings in other datasets. However, of the digitally recorded oral arguments however, only 15 of the 913 speakers are the supreme court justices, and thus have well-known exact dates of birth. Many of the rest of the speakers are the advocates on either side of the case who present their arguments in front of the justices. Whilst some are high profile enough figures to have well known dates of birth (obtainable via Wikipedia for example), many do not. These are often lawyers or law professionals, and thus often will have their date of registration to the Bar association, a requirement to practice law, public. Using this year of registration, one can use this to approximate the age of a lawyer by subtracting 25, the usual time taken to complete the necessary qualifications. This graduation/Bar association registration information was obtained by scraping through two lawyer directory websites[3]. Another significant category of speakers is that of clerks to the supreme court justices. These clerkships are typically given to law students who have recently graduated, typically top of the class, and thus their graduation information is scrape-able from the public domain[4]. The breakdown of the sources for these web-scraped speaker ages is seen in Figure 3.2 (left). For all dates of birth that were obtained without specific month or day information, the middle day of the year (July 2nd) was taken to be their birthday, and this was used to calculate the age of the speaker in years based on the date of the recording.

[1] `https://www.oyez.org/about`

[2] `https://www.oyez.org/about-audio`, see 'sticky shed syndrome'

[3] `https://www.avvo.com`, `https://www.JUSTIA.com`

[4] `https://w.wiki/62tM`

For the training and evaluation on SCOTUS, utterances were split into train and test sets by recording, at an approximate 80% train proportion, and ensuring that the train and test distributions for age using 10 uniformly spaced bins were approximately similar. Splitting the recordings into utterances, one can see the distribution of ages in Figure 3.3, with an average age of around 60. In order to evaluate the SCOTUS data as a verification task, 15 positive and negative trials were selected for each speaker in the test recordings, excluding any speakers seen in the training set from these trials, leading to a total 3810 trials for the verification task.

3.3 CALLHOME

The CALLHOME dataset is an American English speech corpus consisting of 120 unscripted 30 minute telephone conversations between native English speakers. All calls originated in North America, with 90 of the 120 calls being placed to locations outside of North America. The majority of calls were between family or close friends (Canavan et al. 1997). This corpus contains 500 unique speakers.

More information about this dataset can be found at `https://catalog.ldc.upenn.edu/LDC97S42`

3.4 Speakers In The Wild (SITW)

The Speakers in the Wild (SITW) dataset is a human-annotated corpus from open-source media (McLaren et al. 2016a). This corpus consists of 299 speakers, with an average of 8 recordings per speaker. They also provide standardised sets of verification trials. Various metadata was also collected, including speaker gender, microphone type, observed artifacts and recording environment.

Chapter 4

Channel Information

4.1 Introduction

Channel information can be an unwanted source of variability for representations used for speaker recognition tasks. For example, a speaker-discriminative representation should be robust to different acoustic and recording conditions, and variability due to channel information would contradict this goal. While there may be cases in which channel information is also speaker discriminative, such as distinguishing between recordings of a podcast host and a live on-location sports commentator (who both usually appear in those recording conditions), there are also many realistic scenarios in which this channel information cannot be leveraged, and may not be helpful for discriminating between speakers. For example, when diarizing a single microphone recording, the speech segments that are compared for speaker identity always belong to the same recording, thus sharing channel or session information. In this instance, a model trained to leverage channel information to discriminate between speakers may suffer in discriminative performance.

Despite the fact that deep speaker embeddings are often trained in an explicitly speaker-discriminative fashion (Snyder et al. 2018), it has been shown in Raj et al. (2019) that channel information is still present in *x*-vectors (deep speaker embeddings), where meta information about the recording conditions can be successfully extracted from *x*-vectors by a separate classifier.

Disentangling channel information from speaker representations has typically been performed using Probabilistic Linear Discriminant Analysis (PLDA) for *i*-vectors. For

i-vectors, the usage of PLDA is well motivated, as *i*-vectors model both channel and speaker variability. However, in part due to the channel variability found in the deep representations, PLDA has also been used successfully in conjunction with *x*-vectors by Snyder et al. (2018) and Sell et al. (2018). With the motivation for PLDA usage with deep speaker embeddings being less clear than for *i*-vectors, it raises the question of whether it is possible to reduce the effects of channel variability during the embedding training stage.

The goal of obtaining channel invariant features is closely related to the work of producing domain-invariant features, for which adversarial training has been shown to be a powerful approach in tackling it (Ganin et al. 2016; Shen et al. 2018). In previous work, domain adversarial training has been applied to speaker embeddings by Meng et al. (2019) and Tu et al. (2019). Both took the approach of encouraging domain and therefore channel invariance by having an adversary classify the dataset or labelled environment to which generated features belong. However, this is a coarse modelling of the domains over which generated features are encouraged to be invariant.

Here, we propose an adversarial training strategy that encourages invariance at the channel or recording level, without the need for labelled recording information, by training an adversary to predict whether pairs of same-speaker embeddings belong to the same recording. Since this recording-level adversarial penalty affects channel-related information, the approach encourages channel-invariant embeddings.

4.2 Related Work

Handling channel variability for speech applications has been an area of interest for many years, with 'approaches [...] divided broadly into three classes: model adaptation, channel adaptation and robust features' (Ponting 1999, p.1). The adversarial method taken in this work falls under the approach of producing robust features.

There have been several works which have utilised adversarial training to encourage channel (or domain) invariance for speaker recognition tasks, such as the aforementioned work of Meng et al. (2019) and Tu et al. (2019). Also notable is the approach of Z. Chen et al. (2020), who also added an adversary to penalise the inclusion of channel information in the embedding layer of an *x*-vector network. Here, they had the adversary predict either the device from which the recordings were collected, or used labels collected for the type of environment that the utterance was recorded in (quiet,

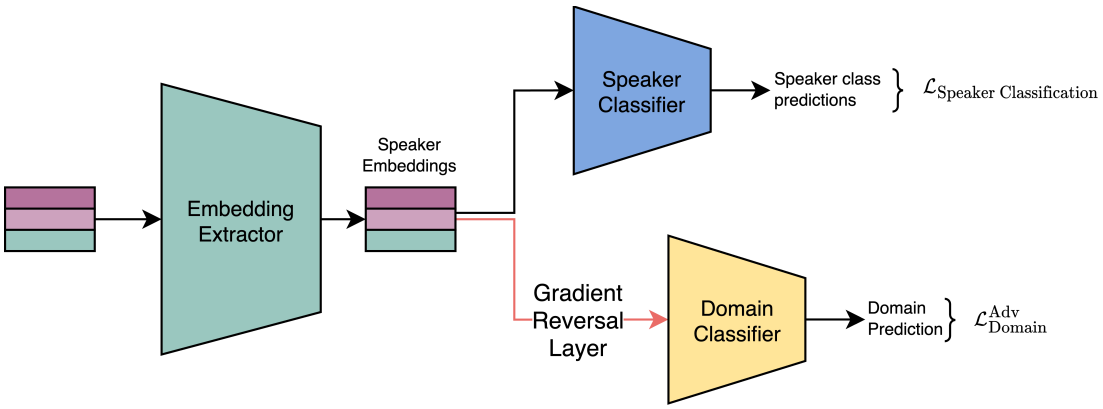

Figure 4.1: The domain adversarial training regime, in which the extracted features are penalised for containing domain information via the adversarial domain classification loss and gradient reversal layer.

office, car etc.). They found that using either device or environment type labels as an adversarial task successfully improved speaker verification performance, suggesting that encouraging channel-invariance resulted in a more robust speaker embedding extractor.

However, both Meng et al. (2019) and Z. Chen et al. (2020) used proprietary datasets which contained either device identification information or the environment labels (or both), making their proposed methods unsuitable for many other public speaker recognition datasets, such as the widely used VoxCeleb (Section 3.1) dataset. Our proposed approach however can function with any dataset. Furthermore, our method targets channel invariance at a very fine-grained level, at the level of the recording, which is a more stringent a requirement than a broad environmental label.

4.3 Domain Adversarial Training

Domain adversarial training describes the family of techniques that train an adversary with a domain discriminating objective on top of a feature extractor (Ganin et al. 2016). This process is more clearly shown through Figure 4.1, in which the adversary, labelled the discriminator, predicts the domain from the embeddings, while the classifier predicts the class as it would do without the adversary. Using a gradient reversal layer, which multiplies the back-propagated gradients by a negative scalar, the feature extractor is incentivised to maximize the loss of the discriminator. The addition of this layer means that the generated features are encouraged to be invariant under the

domain, which is the adversarial classification objective.

Extending the notation defined in section 2.2, where a speaker embedding extractor $\mathcal{G}$, parameterised by $\theta_{\mathcal{G}}$ and classifier $\mathcal{C}$ parameterised by $\theta_{\mathcal{C}}$ predict the probability of a training speaker given an input utterance, a discriminator $\mathcal{D}$, parameterized by θ_D, is trained to ascertain the domain of the generated features, and the classification loss is added as an adversarial penalty to the overall loss function of a domain adversarial neural network (DANN) (Ganin et al. 2016; Shinohara 2016):

$$\mathcal{L}_{\text{DANN}}(\theta_g, \theta_c, \theta_d) = \mathcal{L}_C(\theta_c, \theta_g) - \lambda \mathcal{L}_D(\theta_d, \theta_g), \tag{4.1}$$

where λ is a controllable parameter to determine the weighting of this loss term. Allowing the adversary to act against the classifier is implemented via a gradient reversal layer between the generator and the discriminator.

However, there still remains a choice of what the adversarial objective is. Utilising domain adversarial training for speaker recognition for example could be done as it was in the work of Tu et al. (2019), who had a discriminator predict the dataset to which generated features belong to ensure robustness across distinct domains. On the other hand, Meng et al. (2019) had training data with room condition labels which the discriminator was trained to predict.

For the dataset-adversarial approach (Tu et al. 2019), the difference in domains between the different datasets that make up training data is assumed to be meaningful enough that adversarial predicting this attribute will result in more robust and invariant speaker representations. While this assumption may be accurate, especially with distinct dataset choices, this criteria does not consider sources of channel-variability that are present within a dataset, meaning dataset-adversarial training may be too broad a criterion. Furthermore, the use of recording environment labels (Meng et al. 2019; Z. Chen et al. 2020) is uncommon in many datasets used for speaker recognition tasks, and thus is not a viable approach in many scenarios.

4.4 Channel-Adversarial Training

In this work, we propose a method that attempts to solve the issues in finding a suitable adversarial objective for encouraging channel invariance, with the objective needing to be specific enough to target the channel information, without the use of specific labels

for recording conditions. This is achieved in our method by having the adversary classify *pairs* of embeddings as being within-recording or not, thus penalising the inclusion of channel information in the embeddings. This is implemented by attaching a discriminator with a gradient reversal layer that takes concatenated pairs of embeddings as input. This discriminator outputs a binary prediction for being within-recording or not.

However, there must be some consideration as to what utterances are paired together to train the discriminator. Naively, one might select pairs randomly with a 1:1 within-recording out-of-recording ratio. However, this may lead to undesirable outcomes that suppress the wrong sources of variability. For example, for datasets which do not have any recording information, or contain only recordings where a single speaker is present, it is important that the discriminator only receives pairs of embeddings which belong to the same speaker. The reason for this is that if no different-speaker same-recording pairs exist, the discriminator would be able to ascertain the within-recording label based on the identities of the speakers. This in turn encourages the embeddings to suppress speaker information, which is the opposite of the overall training objective. Thus, the pair selection strategy used in this work utilises only same-speaker pairs, to ensure that speaker information cannot be leveraged in the adversarial task. For some datasets, specifically ones with many recordings with multiple speakers, it may be possible to intelligently select different-speaker pairs in a way that makes sure that the speaker information cannot be used reliably by the discriminator to predict the channel information, but such an approach would not be applicable to many datasets, and thus only same-speaker pairs were considered.

The pair selection strategy is enacted as follows, with Figure 4.2 displaying this visually:

1. Given a speaker k, randomly select two recordings that speaker k is present in. We will refer to these as recording *foo* and recording *bar*.
2. From recording *foo*, randomly sample two segments in which speaker k is active, ideally not overlapping.
3. From recording *bar*, randomly sample one segment in which speaker k is active.
4. From one of the segments taken from *foo*, pair this with both:
    - the other segment from *foo*, forming a within-recording pair

- the segment from *bar*, forming an out-of-recording pair

The result of this pair selection is that we now have a triplet of 3 segments, along with 2 pairs combined from these 3 segments. From these 3 segments, we can train the speaker embedding network as in Figure 4.3. The 3 segments pass through the embedding extractor and the speaker classifier portions of the network in the same way that they would do in the *x*-vector method, with the output of the speaker classifier predicting the training speaker class, for example using the standard cross entropy classification loss $\mathcal{L}_{\text{Speaker Classification}}$. However, we also concatenate the speaker embeddings for the corresponding segments in accordance to the pair selection process we performed before, providing our within-recording and out-of-recording pairs. These pairs are passed through a gradient reversal layer before the Channel Pair Discriminator, where the discriminator predicts whether or not the pair was a same-recording pair or an out-of-recording pair. This loss $\mathcal{L}^{\text{adv}}_{\text{Channel Pair Classification}}$ is a binary cross entropy loss. All components are trained jointly by summing the losses like so

$$\mathcal{L}_{\text{Total}} = \mathcal{L}_{\text{Speaker Classification}} - \lambda \mathcal{L}^{\text{adv}}_{\text{Channel Pair Classification}} \tag{4.2}$$

where λ is the weighting for the adversarial loss. In practice, we populate a mini-batch during training with multiple same-speaker triplets, sampling N speakers per batch, resulting in a total batch size of $3N$ for the embedding extractor and speaker classifier, along with an input batch size of $2N$ for the discriminator.

Our approach encourages channel invariance at a fine-grained level without requiring any additional labels, and imposing few restrictions on the kind of dataset that can be used. The only notable requirement is that speakers in the dataset should ideally appear in multiple different recordings, such that channel information can be suppressed relative to the speaker. However, this is common for many speaker recognition datasets. Furthermore, we suggest two ways of circumventing this requirement for speakers that do not have multiple recordings:

- Choose segments that are temporally close together as being within-recording, and choose a segment further away from this pair as being the out-of-recording segment. This assumes that more channel information is shared between temporally adjacent segments.
- Use data augmentation to create different channel information for the same

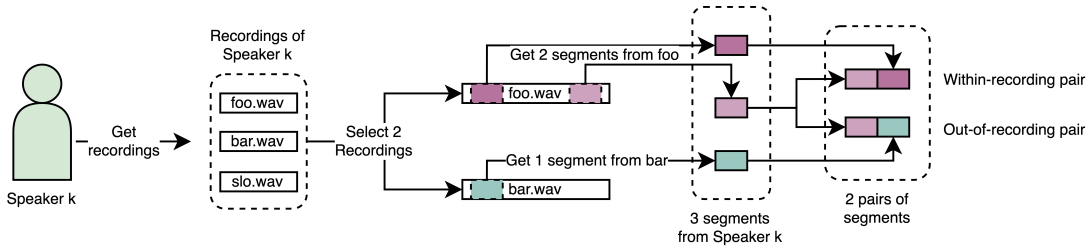

Figure 4.2: The pair selection strategy behind the proposed method. For a Speaker k, 3 segments are sampled from 2 recordings. From those 3 recordings, 2 pairs are chosen, one within-recording pair and one out-of-recording pair.

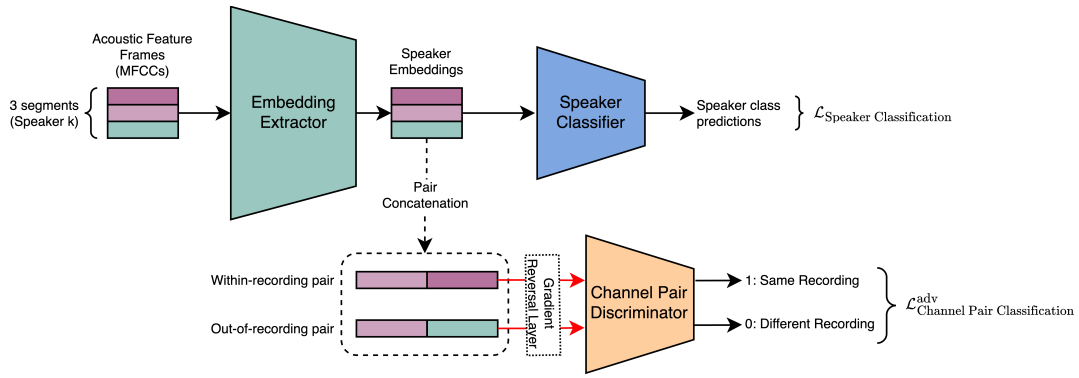

Figure 4.3: The proposed architecture, using the pair selection strategy in Figure 4.2. The speaker classifier is trained to predict the training speaker class, and the discriminator is trained to classify whether or not a pair of utterances are from the same recording.

> recording, treating each individual random augmentation and/or augmentation type (background music, noise, room impulse response etc.) as being different recordings of the speaker.

In addition, it is also possible to simply exclude the single-recording speakers from being used as input to the discriminator, while still being used for the classifier. This would be reasonable if only a minority of speakers were in only a single recording. However, the option involving data augmentation effectively allows this method to be used for any dataset.

4.5 Experimental Setup

4.5.1 Data

The VoxCeleb 1 (Nagrani et al. 2017) evaluation set and the CALLHOME corpus[1] were used in the evaluation tasks. CALLHOME is typically used for speaker diarization, and therefore non-speaker-overlapping segments were extracted with minimum duration 0.5s according to the ground truth segmentation. From these sub-segments, trial pairs occurring within the same recording were selected for evaluating speaker verification performance.

The training data used was the same as in the Kaldi[2] recipes for VoxCeleb and CALLHOME. For training the VoxCeleb system, the VoxCeleb 2 (Chung et al. 2018) corpus was augmented with additive background noise from the MUsic Speech And Noise (MUSAN) corpus (Snyder et al. 2015) and was also convolved using room impulse responses from the Room Impulse Response Noise Database (RIRs) (Ko et al. 2017).

Each RIR was used to create a convolved version of each training utterance, utilising only the 'smallroom' and 'mediumroom' categories of impulse responses. As for additive noise, the MUSAN corpus contains three different noise categories: noise, music, babble. Each of these categories had individual settings in order to produce augmented versions of every utterance with variation in noise augmentation:

- 'noise SNR range: $[0, 5, 10, 15]$
- 'music' SNR range: $[5, 8, 10, 15]$
- 'babble' SNR range: $[13, 15, 17, 20]$
- 'noise' number of additive noises: 1
- 'music' number of additive noises: 1
- 'babble' number of additional noises: $[3, 4, 5, 6]$

VoxCeleb utterances of the same speaker originating from the same original video were considered to be from the same recording, unless augmentation was applied, in which case we deemed the channel information to be sufficiently different to label such segments as being different recordings.

[1] `https://catalog.ldc.upenn.edu/LDC97S42`
[2] `http://kaldi-asr.org`

For CALLHOME, the training data used was the same as the Kaldi recipe, using a combination of the NIST SRE 2004-2008 corpora, along with Switchboard 1, 2 and Cellular, all telephony data. These recordings were augmented with background noises and room impulse responses as above. Augmented versions were considered as different recordings.

4.5.2 Baselines

The network architecture for the generator closely follows Snyder et al. (2018), the original *x*-vector architecture, utilising the same widths of temporal context at each layer, along with the choices for the number of hidden units at each layer. Leaky ReLU and Batch Normalization were applied at each layer.

The reader can refer to Table 2.1 in Chapter 2 for a detailed description of the temporal context at each convolutional/TDNN layer.

Instead of using the stats pooling that the original architecture used, we used attentive stats pooling (Okabe et al. 2018), with 128 hidden units in the single attention head for the VoxCeleb system, and 64 for the CALLHOME system. After pooling, the VoxCeleb system was projected to an embedding of size 512, and CALLHOME to a size of 128.

The classifier network was a single hidden layer feed forward network with 512 hidden units for all models, projecting to the number of classes for each dataset. The classifier was trained using an additive margin softmax loss (H. Wang et al. 2018) using the recommended hyper-parameter of $m = 0.35$ (see equation 2.8 in Chapter 2). All layers had a dropout schedule applied that started at 0, rose to 0.2 in the middle and dropped off to 0 thereafter, similar to the Kaldi recipe.

Networks were trained on batches of utterances between $2-4$s in duration with 400 speakers sampled per batch. Speakers were cycled in each batch to ensure a uniform distribution of speakers across training. The VoxCeleb system was trained for $100,000$ batches and the CALLHOME for $25,000$. SGD was used with learning rate 0.4 and momentum 0.5, with the learning rate halving at 60% of the way through training, and halving for every 10% thereafter.

For both VoxCeleb and CALLHOME there exist pre-trained models in Kaldi, which were also used for benchmarking. Note that the Kaldi VoxCeleb model is trained using

the VoxCeleb 1 training portion in addition to VoxCeleb 2, meaning the Kaldi model is trained on around 20% more speakers, and 13% more utterances.

4.5.3 Acoustic features

For all experiments, 30-dimensional MFCCs were extracted, with the standard 25ms window and 10ms step. Cepstral mean and variance normalization was applied to each utterance before training and only voiced frames were selected, judged by a simple energy based VAD system.

4.5.4 Similarity scoring

For both verification and diarization, either a cosine similarity or PLDA backend was used, utilising length normalization for both. The PLDA model was trained on only the training data for that task, meaning either VoxCeleb 2 or the SRE-Switchboard combination. This differs particularly from some works using CALLHOME, which trained on some folds of the CALLHOME data, using the unseen folds for evaluation (Lin et al. 2019; A. Zhang et al. 2019). The CALLHOME dataset was completely held out for evaluation in this work.

4.5.5 Diarization

The diarization pipeline was as follows. From oracle speech activity marks, 1.5s sub-segments were extracted with a 0.75s overlap. Speaker embeddings were extracted from each sub-segment, normalised, and agglomerative hierarchical clustering was performed on the cosine similarity matrix. Cluster label overlaps were resolved by taking the mid-point of the overlap. Final diarization error rate was computed using `md-eval.pl`[3] with a forgiveness collar of 0.25s.

4.5.6 Adversarial Experiments

To establish a baseline for other domain adversarial techniques, the CALLHOME model was also trained with a dataset-predicting adversary. The training data was split into three domain labels according to the dataset: SRE, Switchboard Cellular, or Switchboard. This adversarial discriminator was trained on the 3-class classification task on all embeddings in a batch using a cross-entropy loss. This baseline was not

[3] `https://github.com/nryant/dscore/blob/master/scorelib/md-eval-22.pl`

	EER	
	Cosine	PLDA
Baseline (Kaldi)	9.77%	3.10%
Baseline (ours)	5.94%	3.87%
Matched-Epoch	5.83%	3.92%
Channel-Adversarial	**4.21%**	**2.98%**

Table 4.1: EER values for the VoxCeleb 1 test set using cosine similarity or PLDA backend.

	EER			
	All pairs		Within-rec	
	Cosine	PLDA	Cosine	PLDA
BL (Kaldi)	29.29%	19.06%	30.05%	23.16%
BL (ours)	**19.09%**	16.19%	28.51%	20.47%
Matched-Epoch	20.32%	17.75%	29.55%	22.43%
Dataset-Adv	19.45%	16.30%	26.71%	20.55%
Channel-Adv	21.11%	**15.65%**	**26.30%**	**19.01%**

Table 4.2: EER values for utterances from the CALLHOME dataset using cosine similarity or PLDA backend.

possible with VoxCeleb due to the lack of domain label candidates, as only one dataset was used for training.

The discriminator in all experiments was a simple feed-forward network which had one hidden layer with 512 units, outputting a single value for the within-recording prediction. For the channel-adversarial model, the size of the input was twice that of an embedding, so 1024 for the VoxCeleb system and 256 for the CALLHOME system. The gradient reversal layer λ value was set to 1.

4.6 Results and Discussion

Table 4.1 shows speaker verification results on VoxCeleb for each model. When all components were trained from a random initialization, the channel-adversarial model did not converge. However, when the discriminator was added as an additional head

	DER
Baseline (Kaldi)	11.69%
Baseline (ours)	11.21%
Dataset-Adv	10.97%
Channel-Adv	**10.01%**

Table 4.3: Diarization error rate on CALLHOME using a cosine similarity back-end.

to an already trained speaker embedding network (trained in the standard fashion with only a speaker classification objective), the technique showed a marked improvement in performance, listed as *Channel-Adversarial* in the table. For clarity, during finetuning, both speaker classification and the adversarial training were performed, summing losses as in equation 4.2.

Since the *Channel-Adversarial* model was trained for additional steps compared to the baseline, we also show results for a *Matched-Epoch* model, a control model that was trained (only on speaker classification) for the same additional number of training steps that the *Channel-Adversarial* required – this model never improves on the performance of the baseline.

The improvement of our baseline over the Kaldi baseline for cosine similarity is likely due to the use of attentive statistics pooling and the angular penalty softmax. The most comparable network architecture in the literature is that of Okabe et al. (2018), which achieves an EER of 3.8% on VoxCeleb. In the more recent VoxSRC[4] competition, much lower values for EER on VoxCeleb 1 were achieved ($< 2\%$), generally using much deeper and larger models with higher dimensional inputs. However, our results outperform others using small variations on the original *x*-vector architecture, in addition to outperforming some deeper models with more parameters e.g. Xie et al. (2019) and Jung et al. (2019).

Table 4.2 shows the verification performance of utterances from CALLHOME, for both within-recording pairs and all pairs of utterances. 'Within-recording' pairs are pairs of utterances where both utterances come from the same recording. The intention of this subset of pairs was to evaluate the effect of suppressing channel information.

Here, the channel-adversarial model with a PLDA backend produces the best EER in both scenarios. The adversarial models appear to perform better for within-recording

[4] `http://www.robots.ox.ac.uk/~vgg/data/voxceleb/competition.html`

pairs, with the channel-adversarial model performing the best once again, outperforming the dataset-adversarial model. Interestingly, the cosine similarity of the channel-adversarial model appears to degrade on the *all pairs* scenario, which may be an indicator that the channel-adversarial method is effective in suppressing the channel information which is additionally helpful in the *all pairs* evaluation.

Across all models, PLDA improves performance on verification, but the effectiveness of this improvement is somewhat unpredictable. The usage of PLDA in conjunction with deep speaker embeddings, and particularly with angular penalty losses is an active field of research (Qiongqiong Wang et al. 2022), and this should be explored further in future work.

Table 4.3 displays the diarization performance on CALLHOME using a cosine similarity backend, with the channel-adversarial model once again performing the best. This improvement correlates with the verification performance in the 'within recording' scenario, where suppressing the channel information leads to improved performance. This supports the hypothesis that channel variability is a nuisance/noise factor when performing diarization.

4.7 Summary

In this chapter, we proposed a method for training speaker embeddings to be channel-invariant, targeting channel variability at the recording level. Our method performs this by adding an adversarial discriminator that takes in as input *pairs* of same-speaker utterances, classifying whether this pair belong to the same recording or not. Compared to previous work utilising adversarial training for encouraging robust features, our method does not require any additional labels relating to the acoustic conditions or environment.

We found that our method successfully suppressed channel information, evidenced by a slightly decreased performance on verification between out-of-recording segment pairs on the CALLHOME dataset. However, this suppression of channel information led to increased robustness, which was reflected on the improved performance of verification on within-recording pairs, and also on diarization. Furthermore, we improved performance on VoxCeleb verification with our method, suggesting that the induced channel-invariance conveyed robustness in this domain also.

The recording-invariant adversarial approach was also tackled by later work by Chung et al. (2020), who implemented a very similar method of encouraging channel-invariance by using triplets of a within-recording pair of utterances and an out-of-recording utterance with VoxCeleb. This was later extended in the same research group by Huh et al. (2020), who cited the work of this Chapter, specifically targeting data augmentation invariance to encourage invariance to the artificial channel variability introduced by augmentation, finding this improved robustness.

For future work, it may have been interesting to explore and probe further into the sources of variability that were affected by our approach. For example, linguistic information may have also been suppressed also by this method, since segments belonging to the same recording may have more linguistic similarity than ones taken from another recording, where potentially different topics were spoken about. This is related to the findings of Raj et al. (2019), who found that keyword information could be successfully probed from *x*-vectors.

Clearly however, if this was indeed the case, this did not harm performance. Arguably, it would be preferable to have linguistically-invariant speaker embeddings (especially in the text-independent speaker verification paradigm), although work by Y. Liu et al. (2019) found that incorporating phonetic information to deep speaker embeddings *improved* performance, so this likely warrants further investigation.

While this Chapter explored variability relating to channel information, inspecting the sources of variability related to speaker identity is something that is explored in Chapters 5 and 6.

Chapter 5

Speaker Attributes

5.1 Introduction

It may be useful to take a step back in examining the task of speaker identification, in that it is a task which has existed and been performed by humans before the prevalence of machine learning and particularly the emergence of deep learning. One such example is in the field of *forensic phonetics and acoustics* (Jessen 2007; Hansen and Hasan 2015), in which speaker classification is a common goal in identifying suspects in criminal cases. This is typically employed in the practical tasks called *voice analysis/profiling* and *voice comparison*, of which the latter is equivalent to speaker verification. For both of these tasks, forensic speaker classification experts typically profile speakers according to a number of attributes or characteristics that allow their voices and speech to be categorised and qualitatively compared. For example, these attributes could be gender, age, dialect, accent, medical conditions and sociolect (education-level - which affects things like lexicon, syntax and stylistics)(Jessen 2007). These attributes are all elements which contribute to the concept of speaker identity, at least in practical terms for some who work in the field of forensic speaker identification.

It is exploring these attributes in the context of deep speaker embeddings that this chapter wishes to explore. When training a speaker embedding extractor, the goal is to encode speech in the embedding space in such a way as to have different speakers be far apart and similar/same speakers closer together. When viewed through the lens of speaker attributes, one might imagine female speakers occupying a separable neighbourhood in the embedding space compared to male speakers. This intuition carries

over for other attributes such as accent or age, although it should be mentioned that this may occur naturally, which is noticeable particularly with the successes of deep learning with capturing surprisingly complex relationships in the data despite having only been given *simple* learning objectives (Mikolov et al. 2013).

However, these attributes play a significant role for some forensic speaker identification experts, and it can be argued that there may be benefit in somehow leveraging these attributes to aid in training a speaker embedding extractor. For example, it may be the case that encouraging embeddings to additionally adhere to an attribute classification task may result in a more robust embedding space, since this reflects our understanding of what the explanatory factors are behind speaker identity. Furthermore, encouraging the embedding space to be richer and more descriptive of the nuances that make up speaker identity may be useful for other downstream tasks. The concept which these ideas draw upon is that of Multi-Task Learning (Caruana 1998), which posits that machine learning models applied to different but related tasks can benefit from using the same underlying representations. In this chapter, we propose using multi-task learning to encourage deep speaker embeddings to capture speaker related attributes.

5.2 Related Work

The concept of multi-task learning (MTL) (Caruana 1998) revolves around the idea that machine learning models that may be used to solve different problems using the same data can benefit from sharing a common representation. In the work of Parveen and Green (2003), they found improvement in increasing the robustness of hybrid RNN/HMM ASR system by performing speech enhancement as an additional task to classification, with both tasks relying on the same hidden representation. Similarly, relating to ASR, the work of Bell et al. (2017) found that simultaneously training on additional tasks to predict both context-dependent and context-independent targets regularly improved performance in an ASR setting.

For speaker recognition related tasks, MTL was implemented by Dey et al. (2018), where the content of the utterance (the word spoken) was used as an additional task to train a deep speaker embedding extractor. This additional task was shown to improve overall speaker verification performance. Likewise, Y. Liu et al. (2019) found that learning to classify the phonetic information in a speaker embedding also improved performance. However, both Dey et al. (2018) and Y. Liu et al. (2019) used

linguistically related additional tasks, and these are not necessarily related to discerning speaker identity. Furthermore, You et al. (2019) incorporated an additional task of predicting the first and higher order statistical information about the input utterance during the speaker embedding training process. The higher order statistics of an input utterance have been used in estimating signal quality for speaker verification (Richiardi and Drygajlo 2008), and potentially the speaker embedding space benefits from encoding information about this.

There may be several explanations for why multi-task learning may improve performance for speaker recognition. Ideally, the embedding space provided by the extractor should be able to describe multiple properties of the data, such as the speaker attributes mentioned above (gender, age, accent etc.). By additionally and explicitly training towards an embedding space that can describe these attributes, it follows that this may lead to a more robust, more descriptive and discriminative embedding space. Encouraging encoding these speaker attributes is preferable for example to encoding channel information, which may be a nuisance factor in many scenarios (see Chapter 4).

5.3 Attributes

If we are to explicitly encourage the speaker embedding space to capture certain attributes, it raises the question as to which ones would be preferable to use. While some forensic phonetics and acoustic literature have listed attributes such as gender, accent and age as being useful when distinguishing speakers (Jessen 2007), forensic speaker identification can be ‘a very complex procedure, [varying] among practicioners’, with there being ‘no standard set of procedures every practitioner agrees upon’ (Hansen and Hasan 2015, p.7). As such, we will discuss the viability of potential attributes to explore within the MTL framework.

5.3.1 Gender

Gender may be an obvious and simple factor when discriminating between voices due to the large discrepancy in the average fundamental frequency between male and female voices (Pernet and Belin 2012; Jessen 2007; Kovacić and Balaban 2009). However, with such a broad categorisation of speakers, it is unclear how much this will add in a multi-task learning setting.

5.3.2 Accent

Accent and dialect (which are separate but will be considered as equivalent in this work), are crucial aspects which forensic speaker identification experts may examine when distinguishing between speakers (Jessen 2007). Region-specific pronunciations of phones and words are clearly discriminative if the speakers to be compared do not share the same pronunciation. Although the concept of phones is not necessary as it is for many automatic speech recognition systems, this information may be useful in highlighting differences in accents.

In the work of Viñals et al. (2019), they utilised networks which were trained to classify phones in order to extract a phone embedding. These were then provided as additional inputs at an intermediate layer to the speaker embedding network. Specifically, this was inserted before the attention mechanism in their network to pool the relevant information into the final speaker embedding. They found improved performance through incorporating this phonetic information as opposed to leaving it out, suggesting this additional task was helpful in highlighting speaker discriminative aspects of the data. In combination with an attention mechanism, the intuition may be that the phonetic information can inform the network which segments of the utterance are particularly discriminative, and this is particularly relevant in the case of accent.

5.3.3 Age

The phenomenon of how a human voice can change due to the effects of ages is fairly well known (Mueller 1997), with degradation in the material making up the vocal folds sometimes leading to changes in the fundamental frequency of the voice, and other qualities in vocal delivery, such as jitter, shimmer, volume and overall quality. In severe cases this can lead to severe changes in the intelligibility of the voice and may lead to diagnoses of vocal disorders such as dysphonia (Rapoport et al. 2018). The term *presbyphonia* has been used to describe the changes in the voice due to age. As such, this attribute may be an interesting task to explore with regards to what makes up speaker identity.

Age is a particularly interesting attribute in that the age of a speaker is not strictly attached to the speaker in the manner that accent and gender are. More concretely, accent and gender do not provide additional information compared to the speaker labels, and are instead broader groupings of the speaker labelling. Age on the other hand captures

a parallel task which is not necessarily constant within speaker, yet still related to the aspects that make up speaker identity and one's 'voice'. The ability of the data to produce a different age label for the same speaker may encourage the speaker embeddings to capture more subtle aspects of what makes up speaker identity.

5.3.4 Linguistic Content

The role of linguistic content to indicate speaker is intuitively clear in speaker recognition tasks with very clearly defined roles, such as diarizing doctor-patient conversations (Shafey et al. 2019). However, this notion of linguistic content informing the speaker identity can extend beyond such a restricted example. In certain cases, the usage of certain words may indicate a certain level of education, described as sociolect in (Jessen 2007). In addition to the lexicon, the education level may also affect the syntax and stylistics, which falls under the general topic of sociolinguistics. While this may not be relevant in all scenarios, such as diarizing performative recordings, such as plays, movies or audiobooks, linguistic content is certainly an aspect which informs speaker identity.

For example, in the work of Raj et al. (2019), they found they were able to accurately predict certain keywords based on *x*-vector speaker embeddings, suggesting even that even with only a training with the objective of classifying speakers, some linguistic information remains in this task, presumably because it is also informative of speaker.

In the work of Shafey et al. (2019) a fairly novel method of diarization is employed, by jointly performing ASR along with diarization. This is performed by modifying an end-to-end ASR model which uses a recurrent neural network transducer (RNN-T). This sequence to sequence model, which for ASR is usually trained to output a sequence of morphemes, is trained to also output speaker change tokens within the sequence, indicating a different speaker is responsible for the proceeding morpheme tokens. Although this method in its exact implementation is only applicable to a two-speaker, two-role scenario, this is a particularly novel way of performing diarization, compared to the more standard practice of clustering based on the embedding similarity matrix. Instead of producing any explicit speaker representations at all, only the speaker turns are modeled in context of the words transcribed. In addition, the authors argue the joint optimization of ASR and speaker turn detection may allow this method to leverage *both* linguistic and acoustic cues in inferring speakers.

Park and Georgiou (2018) attempted to learn from lexical information as well as acoustic by attempting to predict speaker turn points from a concatenation of MFCCs with one-hot word vectors, put through an encoder-decoder style architecture. Like the work of Shafey et al. (2019) however, this method is restricted to two person datasets.

5.3.5 Other

Outside of the mentioned speaker attributes, there are additional factors which result in identifying markers in a person's speech. One such example is any voice or speech related medical impairments. Using simple rule based algorithms based on the fundamental frequency, jitter, shimmer, and harmonic to noise ratio, the work of Cesari et al. (2018) demonstrated success in predicting if a person had speech impairments. It follows that this too is an identifying factor, as this is also an attribute used in forensic speaker identification, with certain impairments such as stutters or lisps being particularly insightful traits (Jessen 2007).

5.3.6 Non-speaker Attributes

Although the benefits of multi-task learning has been framed thus far as a means of categorising that directly contribute to a speaker's voice or the content of their speech, there may also be value in exploring certain aspects which are not so obviously related to speaker identity. In Chapter 4, we discussed that channel information can in certain cases be helpfully speaker discriminative, and that the inclusion of this information is a consequence of the training process. With some datasets including additional metadata around speakers or even the recordings, it may be worth exploring what effect adding classifying these has on performance. On example might be for instance classifying the genre of a broadcast media corpus, as this is a broad task parallel to speaker classification that has numerous influences from many different sources, such as linguistic cues and channel information.

5.4 Multi-task Learning

The MTL framework we propose builds upon on the standard formulation for training a deep speaker embedding extractor, detailed in Chapter 2, section 2.2, where the

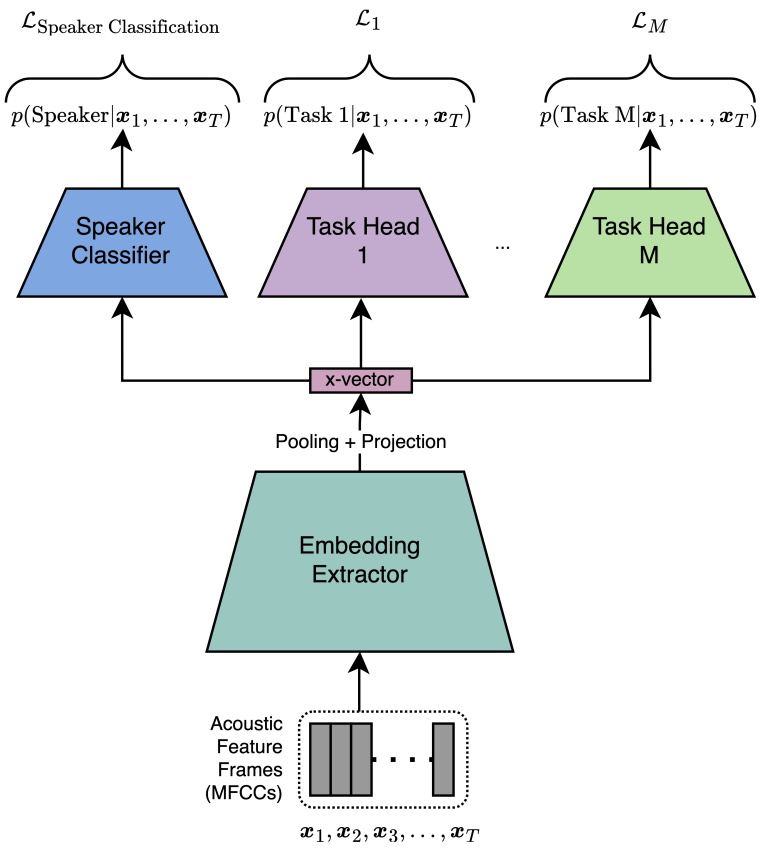

Figure 5.1: Diagram showing the generalised architecture for a speaker embedding extractor with multiple additional tasks. While the additional task heads in this diagram perform classification, they could be used to perform any task with a loss function, such as regression.

embeddings are trained primarily on classification between training set speakers, using a speaker classification head $\mathcal{C}_{\text{speaker}}$, parameterized by $\theta_{\mathcal{C}_{\text{speaker}}}$. In our framework, this speaker classification task can be supplemented with any number of additional tasks, by simply introducing more classification heads that take in as input the speaker embeddings. These classification heads can be feed forward networks, just like the speaker classifier, except they predict a chosen attribute about the speaker, such as accent, or gender. These additional classification heads are be parameterized by $\theta_{\mathcal{C}_{\text{Task}}}$, where the task can be specified. This formulation is described visually in Figure 5.1.

The losses $\mathcal{L}$ for each task are combined in the following manner,

$$\mathcal{L}_{\text{multi-task}} = \mathcal{L}_{\text{speaker}} + \sum_{i=1}^{M} \lambda_m \mathcal{L}_{\text{m}} \tag{5.1}$$

where M is the number of additional tasks supplementing speaker classification and λ_m is the weighting of the loss $\mathcal{L}_{\text{m}}$ from task m. In the instance with just additional age and gender tasks supplementing the speaker task, the loss would be as follows:

$$\mathcal{L}_{\text{multi-task}} = \mathcal{L}_{\text{speaker}} + \lambda_{\text{age}}\mathcal{L}_{\text{age}} + \lambda_{\text{gender}}\mathcal{L}_{\text{gender}} \tag{5.2}$$

5.5 Experimental Setup

5.5.1 Data

Exploring how leveraging specific speaker attributes can enhance speaker embeddings first requires data with appropriate labelling. This kind of speaker metadata is not always available in certain datasets, and thus specific datasets must be used.

5.5.1.1 VoxCeleb

As with previous chapters, the VoxCeleb 1 and 2 datasets were used for test and training data respectively. However, we additionally leveraged the nationality labels that we web-scraped for VoxCeleb 2. From the web-scraping, 125 nationalities were found, but to avoid any speakers being the only ones with their nationality class, any of these speakers were grouped into the 'unknown' class, leaving 102 nationality classes. The full distribution of nationalities can be seen in Chapter 3, Figure 3.1.

5.5.1.2 SCOTUS (Oyez)

The Supreme Court of the United States (SCOTUS) corpus, detailed in Chapter 3, Section 3.2 was used here, utilising the approximate age labels acquired for speakers in this corpus.

A diarization task on the SCOTUS test set recordings was also performed using the models trained and selected based on the verification task. The diarization data preparation differed slightly in that embeddings were extracted for 1.5 second long segments with a 0.75s shift for each recording. Due to the long duration of many of the recordings (>50 minutes), clustering was performed only for approximately 25 minute segments (this splitting up of recordings had some leeway as to not cut off a single uninterrupted speech segment). This process resulted in 2408 training sub-recordings and 595 test sub-recordings.

5.5.2 Model details

The speaker embedding extractors for both VoxCeleb and SCOTUS followed the original *x*-vector architecture, up until the embedding layer which had 256 hidden units. For classification heads that utilized the standard cross-entropy loss, these had a similar architecture to the *x*-vector network, having two hidden layers also of dimension 256 before being projected to the number of classes. The non-linearity used throughout was Leaky ReLU. For classification heads that used an angular penalty loss, these were simply a single affine matrix on top of the embedding layer that projected into the number of classes, using the CosFace (H. Wang et al. 2018) loss. Embedding extractors were trained with different configurations of classification heads, and verification and diarization performance was evaluated.

We also performed contrastive experiments with randomly shuffling the labels of the additional tasks – such as age – to eliminate the possibility any kind of positive regularization effect that the additional task may have irrespective of the information in the labels.

For all embedding extractor training setups for SCOTUS, regardless of the number of classification heads, networks were trained for 50,000 iterations on 350 frames of 30-dimensional MFCCs, with batch size 500, using a small held out set of training utterances for validation. Stochastic gradient descent was used with learning rate 0.2 and momentum 0.5.

For diarization of the SCOTUS corpus, embeddings were extracted for every 1.5s with 0.75s overlap using reference speaker activity detection segmentation, using cosine similarity for scoring and using agglomerative hierarchical clustering to the oracle number of speakers. Due to the supreme court justices appearing in both train and test recordings, the diarization error rate was evaluated for two scenarios: The first evaluation scenario was standard in that all the speech segments were scored, including the speakers which appeared in the training set. The second evaluation scenario was to only score portions of the speech in which unseen speakers were talking.

Speaker embedding extractors for VoxCeleb were trained on the VoxCeleb 2 training set, with 5994 speakers, augmented in the standard Kaldi fashion with babble, music and background noises along with reverberation as in (Snyder et al. 2018). Speakers who were the only members of their nationality in the training set were grouped into the same class as the speakers who could not have their nationality scraped, yielding

Model	VoxCeleb EER
Only Speaker (Baseline)	3.04%
Speaker + Random	4.32%
Speaker + Nationality	**2.95%**
Only Nationality	13.38%

Table 5.1: Speaker verification performance on VoxCeleb using additional tasks

102 nationality classes. Networks were trained for 100,000 iterations with the same batch size and optimization settings as the SCOTUS models.

When evaluating models trained on VoxCeleb on SCOTUS, the VoxCeleb models was were fine-tuned for 5000 iterations on the SCOTUS training set, varying whether or not the full network or only the last linear layer of the extractor was fine-tuned, along with whether or not age in addition to speaker labels were trained on during the fine-tuning. For the first 1000 iterations of fine-tuning, all embedding extractor parameters were frozen to allow the freshly initialized classification head(s) to fit to the new data.

5.6 Results and Discussion

All results are shown in Table 5.2. Results for adding a 10-class age classification task to speaker embedding extractors for the SCOTUS corpus can be seen in the first portion of the table, separated into the models trained with cross-entropy loss on the speaker classification head, and the models trained with CosFace loss on the speaker classification head. The Diarization Error Rate (DER) results are split into two scoring scenarios, 'All' indicates that all speech was scored and 'Unseen' indicates that only speech segments from unseen speakers was scored. Adding gender classification was not found to have any positive effect, and thus these results have been omitted for brevity.

For the standard cross-entropy loss, for configurations of the age loss with λ_{age}=0.5, the verification and diarization performance was improved over the baseline. The results of the parameter search for λ_{age} can be seen separately in Table 5.4.

In contrast, both the control experiment of randomly shuffled labels and the control experiment featuring completely random speaker labels yielded no improvement.

The SCOTUS trained models also feature an experiment in which the only classifica-

Training Set	Model	SCOTUS Fine-tune label set	SCOTUS		
			EER	DER	
				All	Unseen
SCOTUS	Only Speaker	-	3.14%	27.58%	19.75%
	Speaker + Random labels	-	3.78%	27.87%	19.14%
	Speaker + Age	-	2.68%	26.14%	18.02%
	Only Speaker (CosFace)	-	2.71%	26.51%	19.75%
	Speaker (CosFace) + Age	-	**2.62%**	**21.80%**	**14.08%**
	Only Age	-	3.99%	37.08%	26.18%
	Only Gender	-	19.42%	58.02%	44.78%
	Only Random	-	23.31%	69.13%	48.97%
VoxCeleb 2	Only Speaker (Baseline)	(Last Linear) Sp.	2.26%	20.09%	14.07%
	Speaker + Random	(Last Linear) Sp.	2.98%	22.18%	16.32%
	Speaker + Nationality	(Last Linear) Sp.	1.99%	18.74 %	13.54%
	Only Speaker	(LL) Sp. + Age	2.00%	17.81%	12.44%
	Speaker + Nationality	(LL) Sp. + Age	1.89%	**14.82%**	**10.45%**
	Only Speaker (Baseline)	(Full) Sp.	1.63%	29.56%	20.02%
	Speaker + Nationality	(Full) Sp.	1.57%	25.04%	16.93%
	Only Speaker	(Full) Sp. + Age	1.57%	25.98%	17.44%
	Speaker + Nationality	(Full) Sp. + Age	**1.52%**	19.77%	13.57%

Table 5.2: Verification and diarization performance on SCOTUS for various models with multi-task learning objectives.

tion head was the Age classification head, and this performs surprisingly well on the speaker verification and diarization tasks, despite only being trained to distinguish between 10 age categories. This suggests that in order to predict age, some knowledge of speaker identity is also required.

As seen in Table 5.3, the age accuracy of the non-CosFace networks trained with both speaker and age outperformed that of training on age alone, suggesting that performing tasks in combination was able to improve both tasks. This is supported by previous MTL literature, which indicates training on multiple tasks may be helpful to each individually.

It was also interesting to note from Table 5.3 that the age accuracy performance with

Model	**Age Accuracy**
Only Age	77.0%
Speaker + Age (Softmax)	**78.1%**
Speaker + Age (CosFace)	75.4%

Table 5.3: Age accuracy of different models on the SCOTUS test set. Always picking the most probable class was ∼60% accurate.

	EER		DER (all)	DER (unseen)
Dataset	VoxCeleb	SCOTUS	SCOTUS	SCOTUS
Only Speaker	3.04 %	2.71%	26.51%	19.75%
+Attribute ($\lambda = 0.1$)	2.99%	**2.47%**	25.34%	15.88%
+Attribute ($\lambda = 0.05$)	**2.95%**	2.52%	23.10%	15.94%
+Attribute ($\lambda = 0.01$)	3.00%	2.62%	**21.80%**	**14.08%**

Table 5.4: Verification and diarization performance for different settings of λ for additional Nationality and Age tasks. Performance listed here is for models trained from scratch. Attribute for SCOTUS was age and the attribute for VoxCeleb was nationality.

the CosFace speaker classification loss degraded compared to the models trained with the standard cross-entropy loss . This may suggest the effect of the angular penalty loss encouraging a tighter within-class speaker distribution negatively affects the age task, potentially making the embedding space less descriptive for these auxiliary attributes.

The addition of gender classification not improving results is unsurprising, considering male and female voices can largely be distinguished based on their fundamental frequency (F0) (Jessen 2007; Pernet and Belin 2012), and thus this does not add much discriminatory power to the primary goal of speaker recognition.

For SCOTUS trained models, it is clear from Table 5.2 that CosFace improves the speaker verification and diarization performance over the standard cross-entropy loss, with λ_{age} being changed to 0.01 to account for the change in the relative scale of $\mathcal{L}_{\text{speaker}}$. The addition of the age classification head similarly improves over only using speaker labels, producing the best results for verification and diarization for all the configurations shown, making a relative improvement of 17.8% in DER and 3.3% in EER over the 'Only Speaker' CosFace baseline.

While it may be surprising that using age improves performance, since the data is

recorded over a long time-span, and one might expect that *ignoring* age may actually be beneficial, we hypothesize that learning the related task of age classification encourages the embedding space to capture speaker identity in a richer way, which generalizes better. It is possible for instance that age would be a nuisance factor if performance was assessed on a closed set of speakers (speaker identification), since age would indeed be irrelevant, but in this case, evaluation speakers are unseen, and thus we think the network is able to use knowledge of age to create a more robust representation of unseen speakers and what constitutes a voice identity.

The performance of VoxCeleb trained models can also be seen in Tables 5.1 and 5.2. Note all speaker classification heads for VoxCeleb models utilized the CosFace loss. The verification performance of these models on VoxCeleb alone can be seen in Table 5.1, with the addition of the nationality task (λ_{nat}=0.05) yielding a 3% relative performance improvement.

As for Table 5.2, when evaluated on SCOTUS, the fine-tuning of the VoxCeleb models was performed on either the Last Linear (LL) layer of the embedding network, or the whole (Full) network. Either Speaker (Sp.) labels alone were used, or Age was added as an auxiliary fine-tuning task, with λ_{age}=0.05. Models which only use speaker labels at all stages of training (marked as 'Baseline' in the Table) are improved upon in both verification and diarization by utilising either Nationality or Age tasks during the primary training stage or the fine-tuning stage respectively. Indeed, the best performance for verification and diarization is found when auxiliary tasks are employed at both stages.

For the best baseline verification model with no additional tasks, a 6.7% relative improvement in EER is found by using nationality and age (1.63% $\rightarrow$ 1.52%), and similarly for diarization, using auxiliary tasks at both stages yields a 26.2% relative improvement in DER scoring all regions (20.09% $\rightarrow$ 14.82%).

While using both additional tasks yields the best performance, improvements are still found when using a single auxiliary task at either stage of training, suggesting that this technique is still valuable for scenarios in which speaker attribute information is limited or missing from the desired domain.

Despite this, the gains that can be found on VoxCeleb models with adding nationality as a task may suggest that providing the network with additional prior information about the underlying structure of the data can still improve performance. The reason

for this improvement may be down to the nationality task being easier than the speaker task. This is closely related to the reasoning behind curriculum learning (Bengio et al. 2009), in which performance gains can be achieved by training models on progressively more complex tasks. In this instance, the simultaneous training of the easier task may provide a more reliable pathway to convergence than the harder labels alone. As with curriculum learning, adding this easier task may lead to better generalization and overall performance.

5.7 Summary

Overall, these experiments demonstrated that training on auxiliary speaker attribute tasks in addition to speaker classification can yield more robust representations for verification and diarization. We argue that this improved performance is achieved by encouraging the speaker embedding space to be structured along known explanatory factors for speaker identity, thus generalising better to unseen data.

The metadata and data preparation collected as part of this work (VoxCeleb 2 nationality labels, SCOTUS and SCOTUS age labels) also present an opportunity to explore these attributes within speaker embeddings. For example, achieving disentangled representations in a supervised fashion is possible, and this is what Chpater 6 explores.

Chapter 6

Attribute Contributions to Separability

6.1 Introduction

Speaker embeddings are a crucial component in many speaker recognition pipelines, with extracting speaker discriminative features being a key step in speaker verification and diarization. In recent years, obtaining speaker embeddings from the intermediate layer of a neural network (*x*-vectors) has become the leading method for both tasks (Snyder et al. 2018; Sell et al. 2018), outperforming the traditionally successful i-vector technique (Dehak et al. 2011).

There are many properties of speech that convey a speaker's identity, including factors related to the physical properties of the vocal apparatus producing the speech (influenced by factors such as gender, age or medical conditions), in addition to properties relating to accent, dialect, native language and sociolect (social or professional group, which can determine lexicon, syntax, stylistics). Humans, upon hearing a new voice, can intuitively infer many of these properties. It is these properties that are sometimes used to distinguish between speakers when speaker classification is performed by human experts for criminal cases, a field of practice known as forensic phonetics and acoustics (Jessen 2007; Hansen and Hasan 2015).

What this work aims to explore is whether these speaker attributes can be disentangled in the embedding space, and if so, also to determine the contribution that each attribute has on speaker separability. The definition of disentangled representations can be somewhat unclear, but generally speaking, disentangled representation learning aims to learn representations that axis aligns with the underlying generative factors

of the data (Higgins et al. 2018a; Bengio et al. 2014; DiCarlo and Cox 2007). The exact criteria that determine what constitutes 'generative factors of the data' is under debate (Locatello et al. 2019), but in the context of speaker representations and the human voice, we suggest the factors supported by forensic phonetics literature, like gender, age and accent, are excellent candidates for generative factors that constitute speaker identity. To be explicit, this would mean specific dimensions of the speaker embedding would describe these generative factors in their entirety.

In order to achieve disentangled speaker embeddings in a supervised fashion, this work proposes an architecture that adds pairs of attribute specific task heads alongside the standard speaker classification objective to the standard speaker embedding network. Each pair consists of a predictor and an adversary, which act on complementary dimensions of the embedding, simultaneously encoding attribute information in the chosen dimensions while also removing it from the remaining dimensions.

Using these disentangled embeddings, this work also seeks to understand how information about the gender, age or nationality of a speaker contributes towards the discriminative performance of embeddings in verification and diarization applications. This is explored by evaluating on the VoxCeleb (Nagrani et al. 2017; Chung et al. 2018) dataset, along with US Supreme Court recordings.

6.2 Related Work

In the work of Williams and King (2019), speaker representations were disentangled into style and speaker factors using an dual pathway auto-encoder architecture, which used multi-task learning to encourage two auto-encoder latent spaces to separate out these two factors. This work looks at speaker embeddings with a similar approach, but focuses on speaker-specific sources of variation, in addition to incorporating adversarial training techniques to ensure disentanglement.

Both deep speaker embeddings and i-vectors have already been shown to encode a wide variety of information and meta-information about speakers and utterances, such as speaking style and emotion (Williams and King 2019; Pappagari et al. 2020), accent and language (Maiti et al. 2020) or speaker gender, channel and transcription information (Raj et al. 2019). Furthermore, in Chapter 5, we showed that explicitly encouraging the speaker embedding space to capture nationality and age using multi-task learning could lead to more robust performance on unseen speakers. While we looked

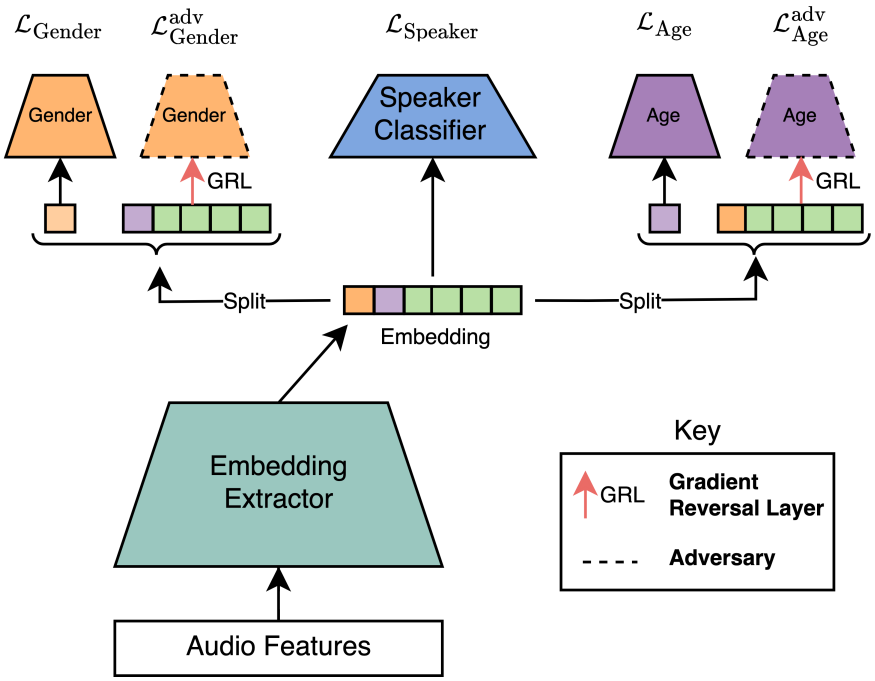

Figure 6.1: The architecture for training a speaker-attribute disentangled speaker embeddings (SplitDim-Adv).

at improving embedding performance by adding auxiliary speaker-attribute tasks, this Chapter looks to disentangle and probe these attributes by using similar techniques.

The topic of disentangled speaker representations is also closely linked with the field of voice privacy (Aloufi et al. 2020; Nautsch et al. 2020; Williams et al. 2021), wherein certain attributes are desirable to obscure in speaker embeddings to protect against malicious attackers. Notably, the work of Noé et al. (2021) used adversarial training to control the gender element of an auto-encoder architecture, seeking to be able to control that element and therefore provide gender-invariant representations. A follow up paper (Noé et al. 2022) utilised normalising flows to again obscure the gender information in speaker embeddings, finding this to be an improvement over the adversarial method.

6.3 Methodology

Building upon the techniques of previous chapters, this chapter combines MTL and adversarial techniques to achieve disentanglement of speaker attributes in the embedding space. The reader can refer to Chapters 5 and 6 for information on MTL and

adversarial training respectively.

6.3.1 Disentanglement

This work proposes a means of utilising both MTL and adversarial techniques to encourage the speaker embedding space to factorize out specific sources of speaker variation. This is achieved by having auxiliary task heads act on subsets of the full speaker embedding dimensions, supplemental to the standard speaker classification head which takes in the full embedding as input. In this system, each factorized speaker attribute would have a pair of task heads, a predictor and an adversary with a gradient reversal layer.

For example for gender, if we would like to factorize out this attribute into the first dimension of the speaker embedding, the first dimension would be used as input to the predictor, a standard classification head that predicts the gender of the speaker. Simultaneously, the remaining dimensions of the embedding would be input into the adversary that is also predicting gender. By doing so, the first dimension is encouraged to be predictive of gender, while the rest of the speaker embedding is penalized for containing this information - thereby factorising out this speaker attribute.

Importantly, all dimensions are still used as input to the speaker classification head, meaning all sources of variation can be used in performing speaker classification. Figure 6.1c displays how the proposed system could be trained to factor out Gender and Age into the first and second dimensions of a speaker embedding respectively[1].

6.4 Experimental Setup

The two datasets used in this work were VoxCeleb (Nagrani et al. 2017; Chung et al. 2018) and the Supreme Court of the United States (SCOTUS) oral arguments corpus (*Transcripts and Recordings of Oral Arguments - Supreme Court of the United States* 2022), which have web-scrapable speaker attribute information about nationality and age respectively (and both having gender labeling). More information on both datasets can be found in Chapter 3.

The architecture chosen for the speaker embedding extractor was the *x*-vector architecture, which was trained on VoxCeleb 2 for 200,000 iterations. The number of em-

[1] `https://www.github.com/cvqluu/splitdim_disentangle`

bedding dimensions was chosen at 64. For all experiments in which the embedding dimension was split up (referred to as SplitDim), the first embedding dimension was always used to capture the Gender. For VoxCeleb SplitDim experiments, dimensions 2-12 were used as input for the nationality classification task, and for SCOTUS SplitDim experiments, dims 2-12 were re-purposed for a 10-bin age classification task.

SplitDim experiments were also performed without the addition of the adversaries, denoted by Adv or No-Adv. A baseline was also trained which only had a speaker classification head (Figure 6.1a). Models evaluated on SCOTUS were fine-tuned on SCOTUS from the VoxCeleb model for 20,000 iterations. The following values were chosen for each loss weighting: $\lambda_{\text{Gender}} = 0.05$, $\lambda^{\text{adv}}_{\text{Gender}} = -20.0$, $\lambda_{\text{Nationality, Age}} = 0.05$, $\lambda^{\text{adv}}_{\text{Nationality, Age}} = -10.0$.

The adversarial loss weightings were set much higher than the other learning objectives, since lower settings failed to discourage the representations from encoding the target attributes.

To establish the effectiveness of the proposed method of disentangling speaker attributes, both qualitative and quantitative approaches were taken. Firstly, the embedding spaces were examined using t-SNE (van der Maaten and Hinton 2008), varying which dimensions to include in this visualization, and labeling points based on supposedly disentangled attributes. Furthermore, the embeddings were probed for information by training a separate feed forward neural network on 50,000 embeddings (as fixed inputs) from the training set, and then evaluating on the test set. If for example a separate classifier was able to perform gender classification successfully on the non-gender dimensions of the embedding, it would imply that the disentanglement had not been successful, as this information remained in the other dimensions. Similarly, the opposite observation would demonstrate that gender information was successfully removed and factored out into the desired dimension.

After showing a suitable level of disentanglement, the verification and diarization performance of these models were evaluated in terms of Equal Error Rate (EER) and Diarization Error Rate (DER). This was evaluated while removing certain attributes (dimensions) from the embeddings, and thus demonstrating what each attribute might contribute to the overall speaker separability. To account for the performance change from removing dimensions of the embedding alone, dimensions were removed from the baseline model to find the average new performance with a reduced number of di-

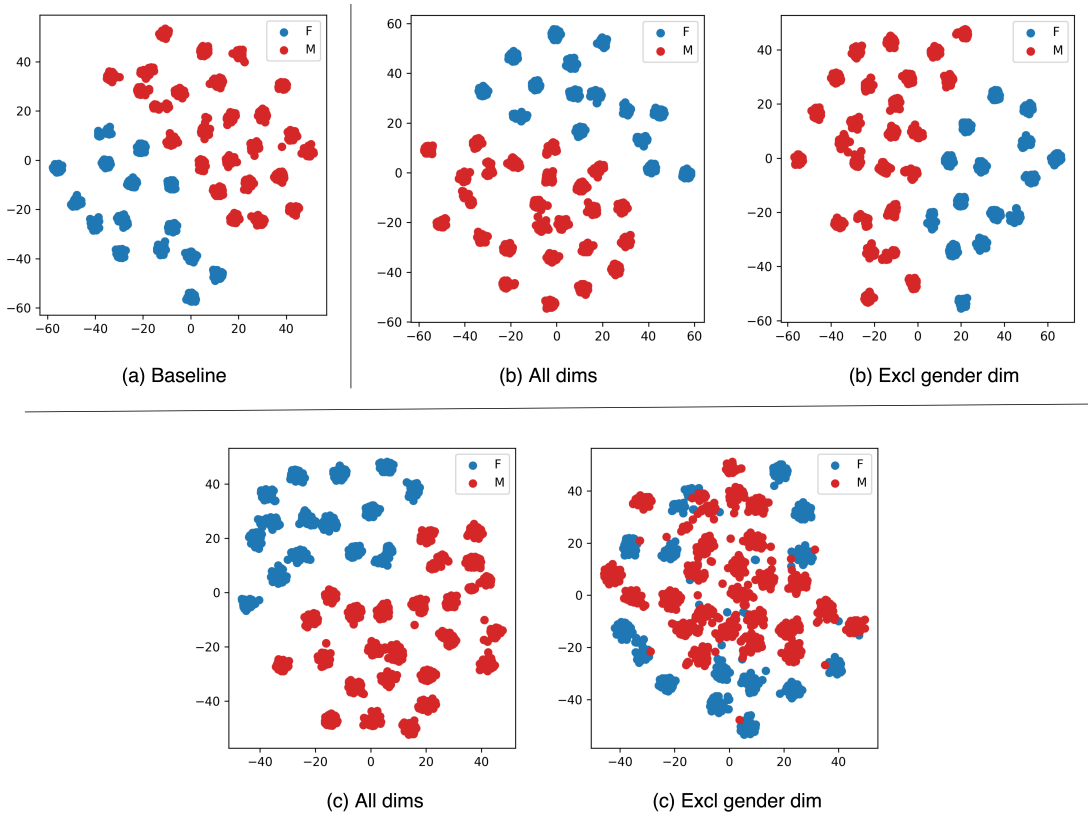

Figure 6.2: Gender color-coded t-SNE projections of the embeddings produced by: (a) baseline model, (b) SplitDim-no-Adv model, (c) SplitDim-Adv model

mensions, sampling at maximum 1000 permutations of the dropped dimensions. All embeddings were normalized and scored using cosine similarity. Diarization was performed using agglomerative hierarchical clustering with linkage threshold tuned on train-set recordings, extracting embeddings for 1.5s windows with 0.75s stride from oracle speaker activity boundaries.

6.5 Results and Discussion

The t-SNE plots of the embeddings produced by various models can be seen in Figure 6.2. Here, one can see that in almost all embedding spaces, the separation of embeddings by gender is clearly visible, and this includes Figure 6.2b, showing that the SplitDim without adversaries still encodes gender in the remaining embedding dimensions. SplitDim-Adv however (Figure 6.2c), improves in this regard, as when removing the gender dimension, shows much less clear separation between embeddings from each gender. This indicates the necessity of including the adversary to ensure such an at-

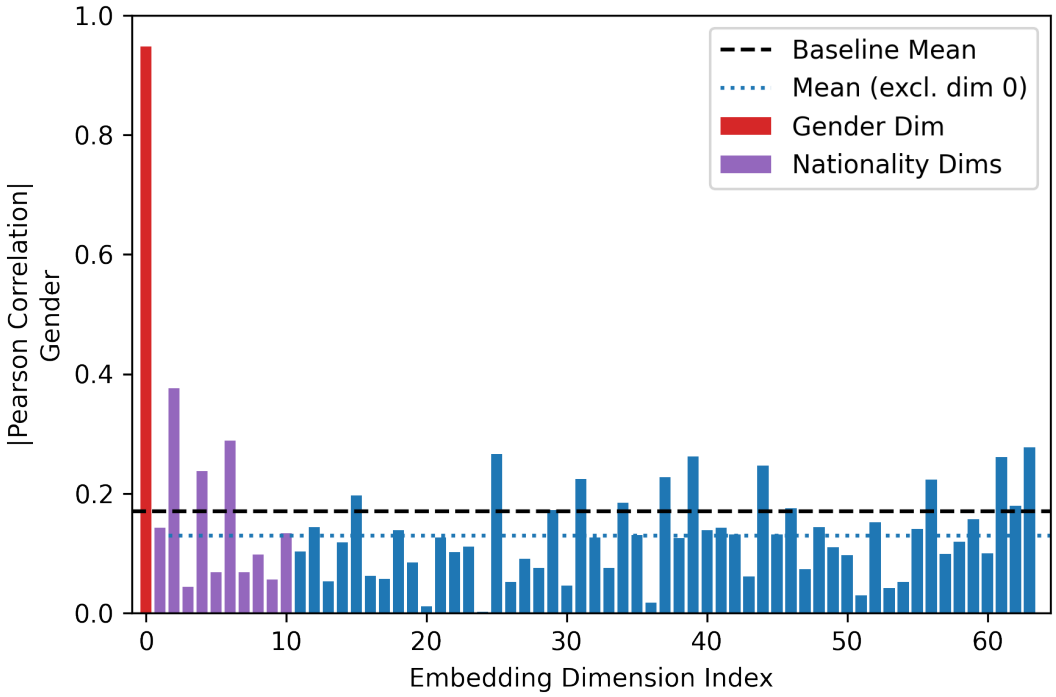

Figure 6.3: The average absolute Pearson correlation co-efficient of each embedding dimension of a SplitDim-Adv model with the speaker gender.

tribute is truly disentangled in the embedding. Similar visualizations were not done for nationality or age, since the number of classes does not make a distinction visually clear in 2 dimensions.

Figure 6.3 shows the average absolute Pearson correlation coefficient of each speaker embedding with the speaker Gender for the VoxCeleb test set for the SplitDim-Adv model. Here, we can see that the 0^{th} dimension is highly correlated, as we would expect, since the model was trained to encode gender into this dimension. As for the other dimensions, the mean correlation (excluding dimension 0) is shown in the blue dotted line, and this has decreased compared to the mean of the baseline model, which is indicated by the black dashed line. This is also an indicator that the gender information is successfully being removed from the remaining dimensions.

This idea is confirmed further with Table 6.1, in which a separate classifier was used to probe the embeddings for gender and nationality information. Here, the probing classifier was unable to achieve high accuracy on gender classification when the gender dimension was removed from the SplitDim-Adv embeddings, reducing the probed gender accuracy from 99.47% to 67.08%, which is less accurate than always predict-

		Probed Accuracy	
		Gender	Nat.
	Baseline	99.47%	75.92%
	Pick most probable class	70.77%	59.55%
No-Adv	All dims	99.36%	73.38%
	-Gender dim	98.27%	73.01%
	-Nationality dims	98.35%	70.39%
	-Nationality, Gender dims	98.48%	68.52%
Adv	All dims	99.49%	72.38%
	-Gender dim	67.08%	72.86%
	-Nationality dims	97.31%	60.04%
	-Nationality, Gender dims	64.18%	58.38%

Table 6.1: The gender and nationality accuracies on VoxCeleb when training a separate probe classifier on embedding features, removing dimensions.

ing the most probable test set gender (70.77%). For context, the probing network was trained on 50,000 examples from VoxCeleb 2, which has a 62% male ratio.

For SplitDim-no-Adv, gender accuracy was still very high, even when removing the gender dimension, again suggesting that without the adversary, gender information is still present in the other embedding dimensions. These conclusions also carry over to the results with probed nationality accuracy, where the addition of the adversary (Adv) resulted in a much more significant reduction in probed accuracy when removing the relevant dimensions, compared to not (No-Adv).

On a practical note, it should be mentioned that attempts to add the task specific heads to an embedding extractor pre-trained on only speaker classification were unsuccessful, resulting in embedding spaces that could not be disentangled. SplitDim-Adv models were only successful when training from scratch or by initialising using another SplitDim-Adv model (as was the case when fine-tuning SplitDim-Adv from VoxCeleb to SCOTUS). This could explain the findings of (Noé et al. 2022; Noé et al. 2021), which found adversarial techniques to be ineffective in obscuring the gender information when using pre-trained speaker embeddings.

In Table 6.2, the speaker verification performance on VoxCeleb is shown for the baseline model, along with the SplitDim-Adv model. Firstly, we can see that disentangling

the space has incurred a reduction in performance (4.22% to 6.68% EER), which is likely due to the addition of the four extra tasks of the SplitDim-Adv model (Gender, Gender-Adversary, Nationality, Nationality-Adversary). With extra tasks, especially adversarial ones, reaching the optimal embedding space for speaker recognition may be difficult if tasks can conflict with each other (as they are designed to do in adversarial training).

This conflicting performance may also raise questions as to what degree it is possible to fully disentangle certain attributes. For example with age and gender, male and female voices may age in significantly different ways, and thus in order to capture that effectively, the dimensions reserved for predicting age may benefit from containing information about the gender also. This kind of query is very much an open question in disentangled representation learning literature (Locatello et al. 2019), and out of the scope of this paper. However, considering that conceptually these attributes are very much interlinked, it is perhaps unsurprising that we observe a performance degradation.

Table 6.2 also shows the verification performance when removing these attribute specific dimensions. As mentioned in section 6.4, there is a general performance impact to be expected from removing dimensions in general, and thus the same number of dimensions was also removed from the baseline for a fairer comparison with the removal of attribute specific dimensions. When comparing like for like, the removal of the single gender dimension is significant in comparison to removing a single dimension (1.8% versus 14.2% relative increase in EER), suggesting gender is a powerful contributor to speaker separability, at least in this test set. Likewise, removing Nationality and Nationality with Gender dimensions results in performance degradation beyond that of the baseline model, further supporting that these attributes are significant sources of speaker variation in the speaker embedding space.

For SCOTUS in Table 6.3, verification performance follows a similar trend to VoxCeleb, with gender once again being a significant factor in affecting separability, whereas the affect that removing age had on performance was more than the baseline expectation from removing 10 dims, but not as significant as nationality. However, for diarization, results are somewhat unexpected, with the SplitDim-Adv model outperforming the baseline in all cases. Also unexpectedly, removing gender with diarization produces a very similar performance decrease compared with removing a single dimension from the baseline. The most likely reason for this is the nature of the SCOTUS

	EER	Δ%
Baseline	4.22%	-
Baseline (avg. excl. 1 dim)	4.30%	1.8%
Baseline (avg. excl. 10 dim)	4.81%	14.0%
Baseline (avg. excl. 11 dim)	4.88%	15.6%
All dims	6.68%	-
-Gender dim	7.63%	14.2%
-Nationality dims	8.76%	31.1%
-Nationality, Gender dims	10.19%	52.5%

Table 6.2: Verification performance on VoxCeleb, using the SplitDim-Adv embeddings and the subset of dimensions. Also shown is the relative percentage increase in EER compared to using all dimensions. -Gender removes 1 dim and -Age removes 10 dims.

corpus, which is particularly male dominated. Although the verification trials were selected to be speaker balanced (77% male), this is not the case with diarizing the raw test set recordings, which in terms duration are >90% male. Thus when scoring all pairs of segments, the overwhelming majority of pairs cannot benefit from distinguishing by gender.

6.6 Summary

In this work, we showed that utilising multi-task learning alongside adversarial training can effectively disentangle and factorize speaker attributes in the speaker embedding space, with the use of the adversaries essential in separating out sources of variation. Using these disentangled representations, we looked at how gender, age and speaker nationality contribute toward speaker separability, finding that gender information was a significant source of information when discerning between speakers in the embedding space for verification, compared to that of nationality or age. However, this

However, general speaker embedding performance suffered in recognition tasks, and it remains an open question as to whether disenganglement in the representation space can be achieved without losing some expressibility. When it comes to vision tasks, disentangled representations (from variational auto-encoders and generative adversarial networks (GANs)) typically 'have significantly worse performance on downstream tasks such as retrieval and transfer learning' (Burns et al. 2021, p.1).

	EER	Δ%	DER	Δ%
Baseline	2.10%	-	32.19%	-
Baseline (excl. 1-d)	2.13%	1.4%	32.76%	1.77%
Baseline (excl. 10-d)	2.40%	14.1%	36.69%	14.0%
Baseline (excl. 11-d)	2.44%	16.2%	37.48%	16.4%
All dims	3.52%	-	29.74%	-
-Gender dim	3.67%	4.26%	30.26%	1.75%
-Age dims	4.41%	25.3%	35.07%	17.9%
-Age, Gender dims	4.62%	35.1%	35.69%	20.0%

Table 6.3: Verification and diarization performance on SCOTUS, using the SplitDim-Adv embeddings. -Gender removes 1 dim and -Age removes 10 dims.

It is also possible that architecturally, better avenues exist for extracting certain pieces of information. For example, speaker gender can be discerned reasonably well from only the fundamental frequency (F0) (Kovacić and Balaban 2009), and thus this information may be extracted fairly early on in the neural network, closer to the acoustic frame-level features, and thus it may be desirable to attempt to disentangle this earlier on in the network.

It may also be interesting to explore tasks which can make use of disentangled representations. For example, some speech synthesis applications, such as voice identity conversion may benefit from disentangled speaker representations (Benaroya et al. 2021), where certain speaker attributes can be manipulated. Utilising age- or nationality-disentangled representations may be an interesting application, with controllable aging or accent conversion being particularly intriguing.

Furthermore, this method required supervised labels in order to disentangle attributes. There is a wealth of literature on unsupervised disentanglement (Kim and Mnih 2019; Burns et al. 2021), utilising VAEs and GANs to find these factors in an unsupervised fashion. With the labels we have already collected for these datasets, it would be valuable to review some of these techniques to see if they align with the explanatory factors we have for a speaker's voice. In particular Higgins et al. (2018b) provide a metric for disentanglement based on known latent factors, and this should be explored in the context of this work.

Chapter 7

Speaker Distribution

7.1 Introduction

The ideal speaker embedding extractor should generalise well to unseen speakers, ideally remaining as discriminative on test speakers as it is on training speakers. This generally falls under the topic of machine learning model generalisation, which is an active field of research (C. Zhang et al. 2017; Neyshabur et al. 2017).

There are several obstacles which may lead to poor generalisation performance in the case of obtaining speaker discriminative embeddings. In Chapter 4, we highlighted a mismatch between training and test conditions regarding the leveraging of channel information as a factor for the degradation of performance in verification and particularly diarization. We also highlighted the idea of domain shift, for example the difference between identifying speakers in telephone conversations compared to broadcast speech.

Another potential obstacle to generalizing well, and the focus of this chapter, is a shift in the *speaker distribution*. What is meant by the term speaker distribution? We can use this term to describe the combination factors that contribute to the identifying attributes of a speaker. Examples for these factors may be gender, accent, age and other physical attributes, such as vocal tract length (these topics and the specific factors are discussed more in detail in Chapter 5).

An issue arises when the speaker distribution in a given held out evaluation set does not match the distribution seen during training. ~~that is, the expected distribution of known~~

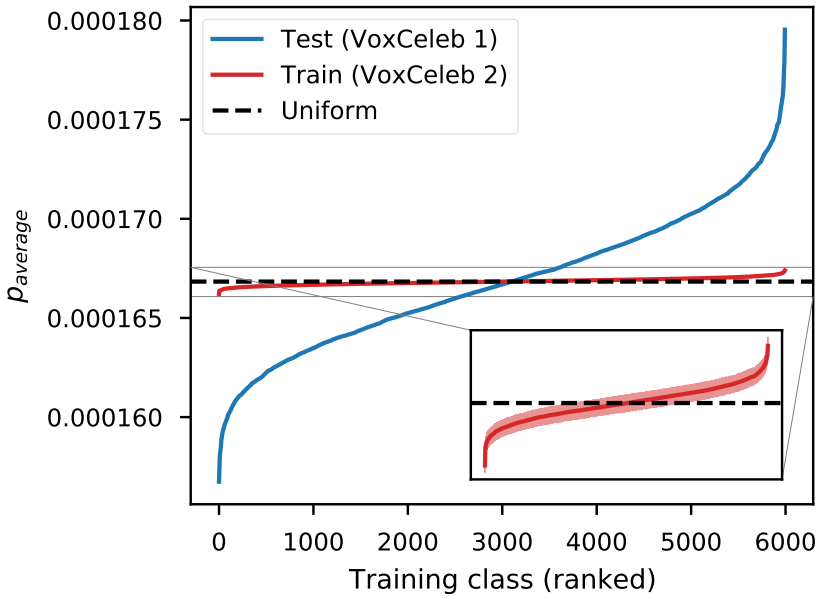

Figure 7.1: Comparison of the 5994 ranked training class probability predictions from a training set (VoxCeleb 2) and test set (VoxCeleb 1), both provided with 40 unique speakers with 42 examples each. The training set probability has uncertainty bounds for 300 bootstrap sampled variations of speakers and examples.

~~speakers is often not replicated in the evaluation set~~ A clearer explanation for this can be seen if we examine what classes are predicted by the full speaker classification network when we give as input the utterances in the test set.

Starting from a trained model, for a given dataset $\mathcal{D}$ of N examples, the average probability assigned to each class can be calculated as follows,

$$\boldsymbol{p}_{\text{average}} = \frac{1}{N}\sum_{i=1}^{N} \text{softmax}(\boldsymbol{h}_i W^T) \tag{7.1}$$

where $\boldsymbol{h}_i$ is the embedding extracted from the i^{th} utterance, and W is the final affine weight matrix. The resulting M-dimensional vector $\boldsymbol{p}_{\text{average}}$ is a representation of the mean probability that the model predicts for the presence of each speaker across the N utterances.

Given a distribution of training set speakers, one would hope that the same distribution is seen at evaluation time. However, this is often not the case, and this mismatch can be displayed using $\boldsymbol{p}_{\text{average}}$.

This effect can be seen in Figure 7.1, where a model trained on VoxCeleb 2 predicts close to the uniform distribution of classes when provided a uniform class distribution of training examples, but predicts a much more skewed $\boldsymbol{p}_{\text{average}}$ on the VoxCeleb 1 test set, with some training classes predicted to be much more likely than others. This is perhaps not a surprising result, as in the hypothetical situation of a test set with entirely one gender, a skewed $\boldsymbol{p}_{\text{average}}$ would be expected. However, this skew is often not as clearly explainable as the hypothetical one-gender test set, and may have multiple contributing factors, as described above. For context, the VoxCeleb 1 test set was selected to have a 'good balance of male and female speakers' (Nagrani et al. 2017), and the speakers in it were chosen because their names began with the letter 'E'. It should also be noted that there may be other non-speaker specific effects that contribute to this class imbalance, such as channel information (see Chapter 4). It is possible for example that a contributor to the class imbalance shown in Figure 7.1 is some matched channel information similarity for the held out speakers and their utterances. However, due to this effect being seen with 40 speakers with 42 utterances each, it seems unlikely that channel information is the sole contributor to this mismatch. We instead argue this is a result of a speaker distribution mismatch.

For example, $\sim$70% of VoxCeleb 1 test set speakers are American, whereas the VoxCeleb 2 training set is composed of only $\sim$30%. This is a large mismatch, and one of the probable sources of the speaker distribution mismatch observed. Indeed, the top 5 training set speakers by $\boldsymbol{p}_{\text{average}}$ predicted on the test set are American. While the concept of distribution shift may be more abstract for other machine learning model applications, for learning representations of voice identity, speaker attributes such as nationality (as a proxy for accent) offer explanations for such a distribution shift.

While Chapters 5 and 6 looked explicitly at speaker attributes and how the speaker embedding space can be manipulated to capture these sources of speaker variation, this chapter explores the concept of speaker distributions as a whole, and methods in which to tackle a shift or mismatch from train to test in an automatic fashion. One strategy explored is to be preventative in approach, encouraging trained models to be robust to different speaker distributions during training. This approach is covered in section 7.2. The other approach explored is to adapt the model itself to a target distribution, thus mitigating the speaker distribution shift. This adaptation, covered in section 7.3 is performed in an unsupervised fashion by considering the mismatch between the expected distribution of train and test speakers. Both of these approaches

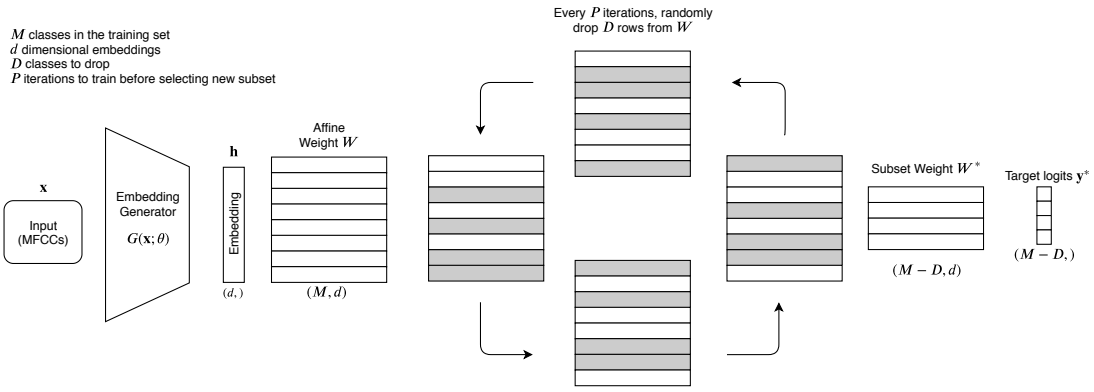

Figure 7.2: System diagram displaying the process of how classes are dropped throughout training in the proposed DropClass method.

were presented in Luu et al. (2020b), and revolve around the central idea that this distribution shift can be mitigated by dropping training classes.

7.2 Distribution Robustness: DropClass

If we recall the general framework of a speaker classification network from which a speaker embeddinng is extracted (Chapter 2, section 2.2), an embedding extractor $\mathcal{G}$ and a classifier network $\mathcal{C}$ are trained together to predict the speaker of an input utterance $\mathcal{X}$. The classifier $\mathcal{C}$ produces a prediction $\boldsymbol{y}$ given the d-dimensional embedding $\boldsymbol{h}$ it takes in as input.

$$\boldsymbol{y} = C(\boldsymbol{h};\theta_C) \tag{7.2}$$

Both $\mathcal{C}$ and $\mathcal{G}$ are trained as a whole, usually via the standard cross-entropy loss against a target one-hot vector $\hat{\boldsymbol{y}}$ which indicates which class out the set of S training classes, $\mathcal{S} : \{i,\ldots,S\}$, $\boldsymbol{x}$ belongs to.

For simplicity, the classification network $\mathcal{C}$ may be as rudimentary as an affine transform that projects the input embedding $\boldsymbol{h}$ into the correct number of dimensions, S. In this simplified case, the entirety of θ_C is a weight matrix W with the dimensions (S,d). Without a bias term, this changes Equation 7.2 to the following:

$$\boldsymbol{y} = \boldsymbol{h}W^T \tag{7.3}$$

Result: Model trained with DropClass
Given: Feature extractor $G(\boldsymbol{x};\theta_G)$, Classification affine matrix W
Set of all training classes $\mathcal{S}:\{i,\ldots,S\}$
D classes to drop per P iterations
Training dataset $\mathcal{D}_{\text{train}}$
while *not done* **do**
  Randomly sample proper subset of size $(S-D)$ from $\mathcal{S}$,
  $\mathcal{R}:\{j,\ldots,S-D\}\subset\mathcal{S}$
  $W^* \leftarrow W$[Class rows in $\mathcal{R}$]
  $\mathcal{D}_{\text{temp}} \leftarrow \mathcal{D}_{\text{train}}$[Examples from classes in $\mathcal{R}$]
  for *P iterations* **do**
    Train $G(\boldsymbol{x};\theta_G)$ and W^* using $\mathcal{D}_{\text{temp}}$
  end
end

Algorithm 1: DropClass approach to training a deep feature extractor.

where $\boldsymbol{y}$ contains the logits of the class prediction.

The proposed technique, referred to as DropClass, is detailed in Algorithm 1. When training with DropClass, every P iterations, a random subset of $\mathcal{S}$ is chosen: $\mathcal{R}\subset\mathcal{S}$ with size $S-D$ where D is a variable that determines how many classes should be dropped. P and D are configurable hyper-parameters. The set $\mathcal{R}$ defines the permitted classes in the next P iterations. The rows of the weight matrix W which correspond to the subset of classes in $\mathcal{R}$ are selected to make a new matrix, W^*, which has the dimensions $(S-D,d)$, and the output of the resulting modification of Equation 7.3, $\boldsymbol{y}*$ has dimension $(S-D)$:

$$\boldsymbol{y}^* = \boldsymbol{h}W^{*T} \tag{7.4}$$

After P iterations, the process is repeated and a new proper subset is randomly selected, with the process continually repeated until training is completed (Figure 7.2).

This proposed method can be compared with a number of existing techniques in literature, in particular Dropout (Srivastava et al. 2014). DropClass essentially drops units in the output classification layer and synchronizes this with the data provided to the model, ensuring that no dropped classes are provided while the corresponding classification units are dropped.

The effectiveness of Dropout has been justified by the technique performing a continuous sampling of an exponential number of thinned networks throughout training and then taking an average of these at test time (Baldi and Sadowski 2014; Warde-Farley et al. 2014). As a result of this model averaging, Dropout has been shown to reduce over-fitting and generally improve performance (Srivastava et al. 2014), and has seen widespread adoption in many different applications of neural networks (Dahl et al. 2013; Variani et al. 2014; Y. Wang et al. 2017). Similar in its justification, DropClass is continuously sampling from a large number of different classification tasks on which the embedding generator G must perform well, in theory making it agnostic to any one specific task or speaker distribution.

This technique also has some similarity to some techniques in the field of meta-learning for few-shot learning, specifically Model-Agnostic Meta Learning (MAML) (Finn et al. 2017) and the related technique Almost No Inner Loop (ANIL) (Raghu et al. 2020). MAML is a method for tackling few-shot learning problems by utilising two nested optimisation loops. The outer loop finds an initialisation for a network which can adapt to new tasks quickly, whilst the inner loop uses the initialisation from the outer loop and learns from a small number of examples from each desired task (referred to as the 'support set'), performing a few gradient updates.

Raghu et al. (2020) found the strength of MAML lay in the quality of the initialisation found by the outer loop, with each task specific adaptation in the inner loop mostly reusing features already learned in the outer loop step. They proposed ANIL, which reduces the inner task-specific optimisation loop to only optimize the classification layer, or 'head', of a MAML-trained network. Similar to DropClass, ANIL makes a distinction between the part of the overall classification network which generates discriminative features (referred to as the 'body'), and the classification head, which is more task specific. raghuRapidLearningFeature2020 et al also proposed the No Inner Loop (NIL) method, which uses the cosine similarity between the generated features of an unseen example to the generated features of a small number of known examples to weight the classification prediction. This use of cosine similarity to compare embeddings is extremely commonplace in speaker recognition (Hansen and Hasan 2015) and in practice, the inference step of the NIL technique is identical to a 1 to N speaker identification set up, if one considers the utterances from the N enrolment speakers to be the small number of labeled examples, the 'support set'.

This similarity of the problems of the few-shot learning and speaker recognition tasks

has influenced the proposal of DropClass, both of which aim to produce a 'body' that generates features applicable to a distribution of tasks (sub-set classification) rather than to a single task. However, DropClass does not perform the outer and inner loops found in MAML/ANIL which explicitly optimizes the network to be robust to additional gradient steps per sub-task. Instead, DropClass encourages performance on all tasks by continually randomizing the training objective, implicitly encouraging the generated features to perform well across subtasks. Despite this, exploring ANIL and MAML for speaker representation learning would be a natural extension to this work. This extension would be particularly interesting considering the experiments on the NIL method (cosine similarity scoring) from Raghu et al. (2020), specifically Table 5. They found that MAML and ANIL trained models significantly outperformed multi-class training models, where all possible classes were trained simultaneously. Considering the *multi-class training* paradigm is the most common approach to training deep speaker embedding extractors, there could well be gains to be found in adopting a meta-learning approach to training speaker embedding extractors.

7.3 Distribution Matching: DropAdapt

Returning to Figure 7.1, which displays the average class probability predicted for test set utterances compared to train, it can be seen that the test set predicted class distribution is significantly more imbalanced than the training set distribution. This observation can be interpreted in a number of ways. For example, it is well known that class imbalance is a significant impedance to performance in classification tasks (Buda et al. 2018), especially in cases in which training and inference have significantly different distributions. It is a natural extension to this that the performance of an embedding extracted from a classification network would degrade in performance in the same manner, which has been seen in the work of Huang et al. (2019) and Khan et al. (2017). In these works, they found cost sensitive training and oversampling methods to increase the performance of learned representations.

The second closely related interpretation and hypothesis is that the 'low probability' classes predicted by the model are in some way less important to the performance of the embeddings on the test set. These 'low probability' classes are suggested by the model's predictions to be less likely to be present in the test set. This might imply that distinguishing between these specific classes is not as crucial to the end task as the

other classes are.

Following from these interpretations, this technique, referred to as DropAdapt, works via dropping these low probability classes permanently to fine-tune a fully trained model, adapting the model to a test set and hopefully increasing performance. This is described in Algorithm 2. This method should be applied only to a fully or near fully trained model, as an accurate estimation of the training class occupation must be obtained first.

To ensure an accurate probability estimation of the test set throughout the fine-tuning, this ranking (and dropping) of the least probable classes can be performed periodically, meaning this technique is functionally similar to the DropClass method above, except that classes are removed permanently, and the dropping of classes is determined by the probability criterion $\boldsymbol{p}_{\text{average}}$ instead of randomly.

A slight variation of this method explored in this work is referred to DropAdapt-Combine, in which instead of permanently removing these classes, all the low probability classes are combined into a single new class such that the examples belonging to the removed classes are not completely discarded.

This method can be compared to techniques in the fields of active learning and learning from small amounts of data, such as the Facility-Location and Disparity-Min models (Kaushal et al. 2019), which put heavy emphasis on selecting the right subset of examples in order to learn efficiently. These methods are typically used to capture the whole distribution of the desired dataset in as few examples as possible, encouraging a diverse and representative subset of examples. However, it is implied by Figure 7.1 that in this speaker embedding task, even if the whole training dataset were used, this may not be representative of the distribution found at test time. DropAdapt can be seen as a means of correcting this mismatch through subset selection for fine-tuning.

Buda et al. (2018) and Huang et al. (2019) found oversampling minority classes to be an effective strategy in improving performance for neural networks on imbalanced datasets. Viewing this problem as a dataset imbalance problem, DropAdapt could also be interpreted as a corrective oversampling strategy, training additionally on those classes which are retained to better match the target distribution.

This train-test distribution mismatch is also closely linked to the field of domain adaptation and the domain-shift problem (Patel et al. 2015). However, DropAdapt is primarily proposed as a means of adapting to a class/speaker distribution mismatch, as it

Result: Model tuned with DropAdapt
Given: Trained feature extractor $\mathcal{G}(\boldsymbol{x};\theta_{\mathcal{G}})$, trained W
Training set $\mathcal{D}_{\text{train}}$
Unlabeled Test/enrolment utterances $\mathcal{D}_{\text{enrol}}$
while *not done* **do**
  Calculate all $\boldsymbol{p}_{\text{average}}$ from $\mathcal{D}_{\text{enrol}}$ (see Equation 7.1)
  Rank classes by $\boldsymbol{p}_{\text{average}}$
  Select set of higher probability classes from $\mathcal{S}$, dropping lowest probability D classes $\rightarrow \mathcal{R}$
  if *Combine* **then**
    Assign all examples not in $\mathcal{R}$ same class label in $\mathcal{D}_{\text{train}}$
    $W^* \leftarrow W$[Class rows in $\mathcal{R}$]
  else
    $\mathcal{D}_{\text{train}} \leftarrow \mathcal{D}_{\text{train}}$[Examples from classes in $\mathcal{R}$]
    $W^* \leftarrow W$[Class rows in $\mathcal{R}$]
  end
  $W \leftarrow W^*$
  $\mathcal{S} \leftarrow \mathcal{R}$
  for *P iterations* **do**
    Train $\mathcal{G}(\boldsymbol{x};\theta)$ and W using $\mathcal{D}_{\text{train}}$
  end
end

Algorithm 2: DropAdapt method for adapting a deep feature extractor to a chosen dataset

is likely that $\boldsymbol{p}_{\text{average}}$ is less informative the greater the domain mismatch. Combining domain adaptation techniques with DropAdapt could be an interesting extension to this work.

7.4 Experimental Setup

The following section details the experimental setup and the experiments performed utilising the proposed methods. All experimental code can be found online [1].

The primary task that these experiments attempted to improve performance on was that of speaker verification, specifically that on VoxCeleb 1 (Nagrani et al. 2017) and

[1] https://github.com/cvqluu/dropclass_speaker

Speakers In The Wild (SITW) core-core task (McLaren et al. 2016b). Although there exist several metrics to evaluate verification performance, which are typically chosen depending on the desired behaviour of a system, the primary metric explored here was the equal error rate (EER), as that is the primary metric for evaluation on VoxCeleb 1.

The training data used for all experiments was the VoxCeleb 2 development set (Chung et al. 2018), which features 5994 unique speakers. This was augmented in the standard Kaldi[2] fashion with noise, music, babble and reverberation. The original *x*-vector architecture was used with very little modification, using Leaky ReLU instead of ReLU, with 30-dimensional MFCC features as inputs, and 512-dimensional embeddings. The main difference between this implementation and that of Snyder et al. (2018) was the use of the CosFace (H. Wang et al. 2018) angular penalty loss function instead of a traditional cross entropy loss. This classification transform also was applied directly to the embedding layer, unlike the original, which has an additional hidden layer between the embedding layer and the classification layer. This means that the simplified notation for the classifier C following from equation 7.3 is an accurate representation of our model. All pairs of embeddings were $L2$ normalized and scored using cosine distance.

A batch size of 500 was used, with each example having 350 frames. Each batch had the same number of unique speakers as examples. Models were trained for 120,000 iterations, using SGD with a learning rate of 0.2 and momentum 0.5. The learning rate was halved at 60,000, 80,000, 90,000, and 110,000 steps. For DropAdapt fine-tuning, the learning rate was chosen to be the same as it was at the end of training the original model, and all the enrolment utterances were used to calculate $\boldsymbol{p}_{\text{average}}$.

7.5 Results and Discussion

7.5.1 DropClass Experiments

Our initial experiments investigated favourable settings of P and D for DropClass, and the results are shown in Figure 7.3, where the number of classes to drop was fixed at $D = 5000$ and the number of iterations P was varied between 50 and 4,000, the latter being slightly over 1 epoch's worth of data with the chosen batch size. It can be seen that improvements over the baseline are to be found more reliably at lower values of

[2]https://kaldi-asr.org/

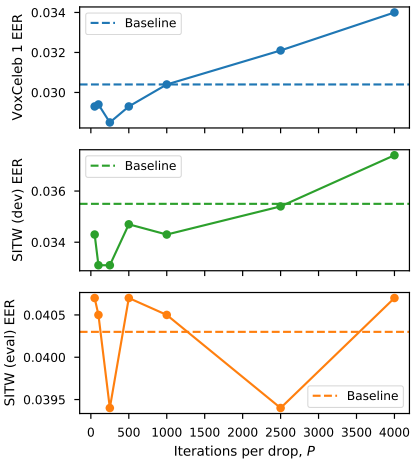

Figure 7.3: Comparison on the effect on EER of varying the number of iterations to run before re-selecting the class subset, fixed at dropping $5000/5994$ training classes each period.

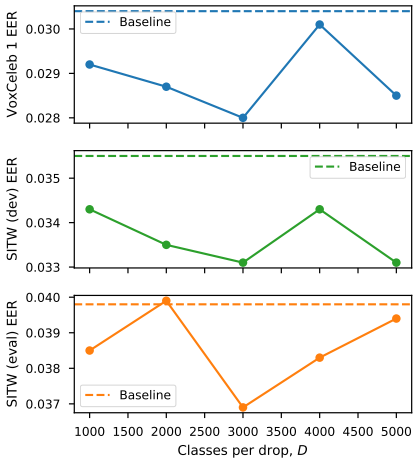

Figure 7.4: Comparison on the effect on EER of varying the number of classes to cycle out every 250 iterations.

P, with consistent performance improvements when $P < 1000$ for both VoxCeleb and SITW (dev). This is perhaps unsurprising, as a motivating factor for this technique was to train a network on many different permutations for robustness on a variety of tasks, and thus with a training budget for each model of 120,000 iterations, this is not a large number of permutations throughout training. As the value for P increases, this may also increase the risk of incurring the phenomenon of catastrophic forgetting (Ganin et al. 2016; Goodfellow et al. 2015), an issue in which networks trained on a new task begin to degrade on the task that they previously were trained on.

From the previous experiment, choosing the best performing value of $P = 250$ across each dataset, the number of classes to drop D was then varied from 1000 to 5000, shown in Figure 7.4. From this, we can see that for nearly all configurations of D, performance was improved on all datasets using DropClass over the baseline. Dropping approximately half the classes at D=3000 appeared to produce the best performance, although a more thorough exploration with different training data is likely required to ascertain if any heuristic exists for the selection of this value. However, from the previous experiments, it can be seen that for a suitably low value of P, DropClass can convey improvements over the baseline.

It may be surprising that rotating classes in this way of the output layer produces improved performance, but this performance benefit is more explainable when viewing DropClass as an extension of Dropout on the output layer, as mentioned above.

It is however important to note that a crucial component of this method is the use of the CosFace (H. Wang et al. 2018) angular penalty loss, with Table 7.1 showing a comparison of the effect that changing the loss function had on the improvement that DropClass produced on VoxCeleb. A more in-depth analysis on how each loss function changes with the permutations of each subset of classes is required.

7.5.2 DropAdapt Experiments

Table 7.2 displays the relative improvement in EER from utilising the DropAdapt and DropAdapt-Combine method, using either the enrolment speakers from VoxCeleb 1 or SITW (dev) to choose which speakers to drop. The starting point was a standard classification trained baseline. Models were trained on a budget of 30,000 iterations, and one configuration for D and P was tested. Also compared were the following control experiments: The baseline but trained additionally for the same number of iterations as DropAdapt, Drop-Random, which drops random classes permanently, ignoring the $\boldsymbol{p}_{\text{average}}$ score, and Drop Only Data, which removes the low probability classes from the training data, but does not remove the relevant rows in the final weight matrix, bypassing the use of W^* in Equation 7.4.

Compared to the baseline and control experiments, both DropAdapt and DropAdapt-Combine show strong performance gains on VoxCeleb. The 2.63% EER on VoxCeleb is particularly impressive when compared to other works which use similar or larger network architectures and more training data and achieve $> 3\%$ EER (Okabe et al. 2018; Xie et al. 2019). The improvements over the baseline on SITW however are more modest, with DropClass trained models and 'Drop Only Data' outperforming the DropAdapt models.

An interesting observation is the fact that dropping only the data improved performance on VoxCeleb, but not as much as the DropAdapt methods. As discussed in section 7.3, DropAdapt can be viewed as a form of corrective oversampling of targeted classes, with oversampling techniques having been shown to improve performance in imbalanced data scenarios (Huang et al. 2019; Khan et al. 2017). From this, we can see that for the within-domain data, some of the benefit of DropClass is gained from only

fine-tuning via oversampling, but this benefit is increased further by also dropping the classes from the output layer. Conversely, for the out-of-domain SITW dataset, dropping only the classes from the data performed the best. We hypothesize that the reduced effectiveness of DropAdapt in this case may be due to the technique having to adapt to not only a new speaker distribution, but also a new domain. Further exploration combining DropAdapt with traditional domain adaptation techniques is left for future work.

In addition, more experimentation on the configurations of P and D could be explored, as it may be possible for example that the iterative dropping of classes is not necessary, and that the initial probability estimation is suitable. Furthermore, the most obvious extension left for future work is to use both DropClass and DropAdapt in conjunction, as both have been shown to provide performance increases in parallel.

Following up on the hypothesis presented in section 7.3 that the imbalanced distribution of $\boldsymbol{p}_{\text{average}}$ on the test set may be an indicator of train-test mismatch and thus incurring performance loss, Figure 7.5 shows the EER and the KL divergence ($D_{KL}(p||U)$) from the VoxCeleb test set $\boldsymbol{p}_{\text{average}}$ to the uniform distribution as the DropAdapt-Combine model is trained. As we can see from the figure, while the EER decreases, the distribution of $\boldsymbol{p}_{\text{average}}$ also gets closer to the uniform distribution. Whilst there appears to be a correlation, this is likely not a strongly linked pair of observations, in that we can easily break this relationship by training only the final affine matrix W and freezing the embedding extractor to provide more favourable class weightings for $\boldsymbol{p}_{\text{average}}$. However, in the case of DropAdapt, the decreasing $D_{KL}(p||U)$ may indicate that a favourable change in the extracted representations is occurring. This could be useful as a stopping criterion for cases in which adaptation data has no labels at all.

7.6 Summary

In this chapter we presented the DropClass and DropAdapt methods for training and fine-tuning deep speaker embeddings. Both methods are based around the notion of dropping classes from the final classification output layer while also withholding examples belonging to those same classes. Drawing inspiration from Dropout and meta-learning, DropClass is a method that drops classes randomly and periodically throughout training such that a model is trained on a large number of different classification objectives for subsets of the training classes as opposed to classifying on the full set of

	EER (VoxCeleb)	
	Baseline	DropClass
Softmax	5.89%	6.25%
CosFace (H. Wang et al. 2018)	**3.04%**	**2.80%**
SphereFace (W. Liu et al. 2017)	3.92%	4.76%
ArcFace (Jiankang Deng et al. 2018)	3.19%	3.08%
AdaCos (X. Zhang et al. 2019)	3.24%	3.81%

Table 7.1: EER values on VoxCeleb 1 for using DropClass (P=250, D=3000) or not with different angular penalty loss functions, all with the paper recommended settings of hyper-parameters.

	EER	Rel Impr
Baseline (VoxCeleb)	3.04%	-
Baseline (More iterations)	3.06%	-0.7%
Drop Random (D=500, P=5000)	3.08%	-1.3%
Drop Only Data (D=500, P=5000)	2.86%	5.9%
DropAdapt (D=500, P=5000)	2.68%	11.8%
DropAdapt-C (D=500, P=5000)	**2.64%**	**13.2%**
Baseline (SITW)	3.55%	-
Baseline (More iterations)	3.61%	-1.7%
Drop-Random (D=500, P=5000)	3.73%	-5.1%
Drop Only Data (D=500, P=5000)	**3.31%**	**6.7%**
DropAdapt (D=500, P=5000)	3.47%	2.3%
DropAdapt-C (D=500, P=5000)	3.39%	4.5%

Table 7.2: Relative improvement in EER from using DropAdapt and DropAdapt-Combine (DropAdapt-C) on the VoxCeleb 1 and SITW datasets on a budget of $30,000$ iterations

classes. We argue that this can lead to an embedding space that is more robust to different speaker distributions, although further testing with different sampled train and test portions would verify this more concretely. We also show that in conjunction with the CosFace (H. Wang et al. 2018) loss function, DropClass can improve verification performance on the VoxCeleb and SITW core-core tasks.

We present the mismatch in speaker distribution between train and test as a potential

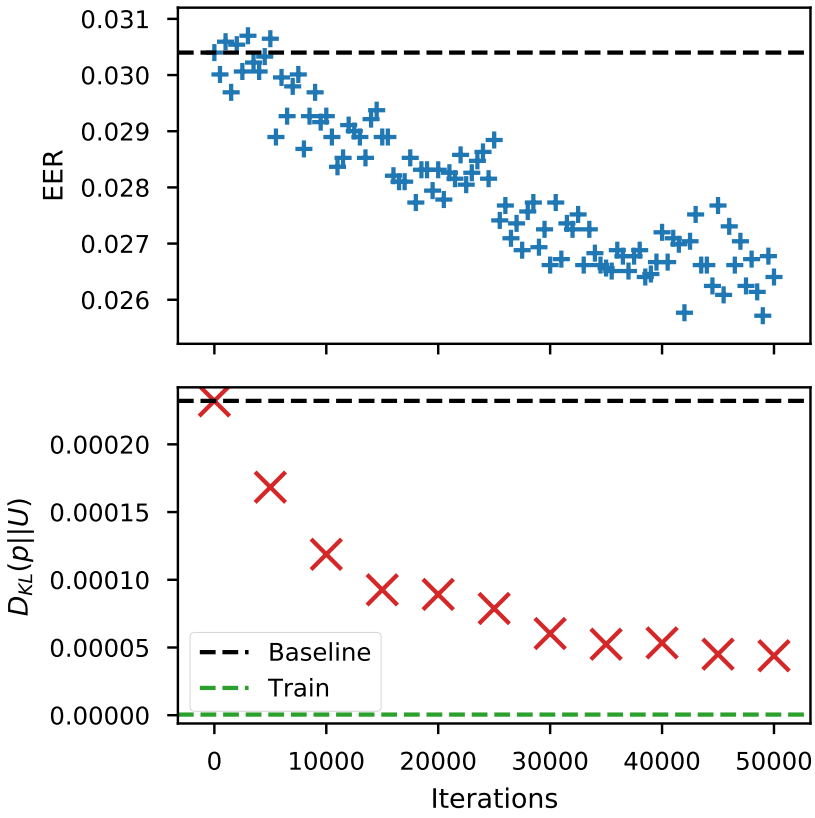

Figure 7.5: Plot of the evolving EER on VoxCeleb as classes are dropped with DropAdapt-Combine (P=5000, D=500) along with the KL divergence from $\boldsymbol{p}_{\text{average}}$ on the test set to the uniform distribution.

reason for reduced performance in verification, and propose DropAdapt as a means of alleviating this. DropAdapt is a method which can adapt a trained model to a target dataset with unknown speakers in an unsupervised manner. This is achieved by calculating the average predicted probability of each training class with the adaptation data as input. From these predictions, the model is fine-tuned by dropping the low probability classes and training for more iterations, focusing on the classes which the model has predicted to be presented in the adaptation dataset. This is not unlike traditional oversampling techniques. Applying DropAdapt to VoxCeleb leads to a large improvement over the baseline, with DropAdapt also outperforming simply oversampling the same classes, suggesting it may be an effective strategy in adapting to a different class distribution than what was seen during training. We also show empirically that as the class distribution mismatch is corrected during DropAdapt, so too does the verification performance increase.

The main downside of this method is the computational cost of training the whole network for more steps in order to adapt to data, so it may be interesting to explore if a simpler and more computationally inexpensive transformation could be learned to apply to the feature space. This would not be dissimilar to other feature space domain adaptation techniques, such as in the work of Swietojanski and Renals (2014), which, for the purpose of ASR, learns to re-weight hidden activations based on speaker-dependent parameters. It may be possible to extend this idea to re-weight activations in the embedding network based on an observed speaker distribution, as opposed to a specific speaker.

Chapter 8

Conclusions and Future Work

8.1 Summary

In this thesis, we explored how different sources of variability in deep speaker embeddings can be influenced for speaker verification and diarization.

In Chapter 4, we proposed a method for encouraging channel invariance at the recording level, utilising adversarial training. In comparison to previous work utilising adversarial techniques for robust deep speaker embeddings, our method does not require environment labels or annotations about the channel information, while also providing a finer-grained adversarial objective than approaches which use environment labels or dataset labels. Our method instead compares pairs of same-speaker utterances using the adversarial discriminator, classifying whether or not the pair belong to the same recording, thus suppressing channel information via the adversarial loss term. We showed via experiments on the VoxCeleb and CALLHOME datasets that our proposed method yielded improved performance over both a standard baseline along with a dataset-adversarial approach. This performance increase was particularly notable in scenarios in which channel information was a nuisance factor - namely diarization, and verification of pairs within a recording.

Furthermore, in Chapter 5 we used multi-task learning to explicitly encourage the deep speaker embedding extractors to encode speaker attribute-related information. Here, we supplemented the standard speaker classification task with speaker attribute classification tasks, performing nationality and age classification simultaneously with speaker classification. The motivation for doing so was to have the speaker embedding space

encode sources of variability that we know to be speaker identity related, and thus should generalise to unseen data. This is in contrast to a speaker embedding factor like channel information, which we showed before to not generalise in certain scenarios. On the VoxCeleb and SCOTUS datasets, we improved verification and diarization performance by adding the nationality and age classification tasks, with larger relative gains found when fine-tuning to new domains.

We also explored how, for the purposes of obtaining disentangled representations, speaker embedding factors can be manipulated by utilising the adversarial and multi-task learning techniques. In this Chapter (6), we showed how pairs of classifiers and discriminators acting on complementary dimensions of the speaker embedding could be used to isolate specific aspects of speaker identity in chosen dimensions of the speaker embedding. In order to validate the disentanglement, we visualised the embedding space and found that for gender, our method greatly reduced the separability of these classes when removing the appropriate dimension. We further validated disentanglement for gender and nationality information by training external classifiers on the embeddings, finding that the probe classifiers were unable to outperform always picking a specific class, thus implying that our disentanglement was successful. These disentangled embeddings were also used to explore how much gender, age and nationality contribute to speaker embedding separability. In this exploration, we found that gender was a significant attribute in embedding separability for the VoxCeleb test set, while nationality also contributed to a lesser degree. We concluded that the choice of test set affects attribute contributions to separability greatly, as the male dominated SCOTUS corpus did not degrade in performance as much as VoxCeleb when removing gender information.

Chapter 7 looked at how to mitigate potential mismatches in the speaker distribution seen at train versus at evaluation time. Here, we proposed two methods relying on dropping classes from the output layer in order to induce distribution robustness or to provide a means of adapting to a different speaker distribution in an unsupervised manner. Here, we argued that our adaptation method works by emphasising the important aspects of variability. Both methods showed improvements for verification and diarization, with the adaptation method showing large relative gains.

Overall, several areas relating to the sources of variability in deep speaker embeddings were explored, finding that manipulating these factors for a desired task was often a productive approach.

While each approach explored in each chapter has its merits, they may have particular circumstances where they should or shouldn't be applied. One interesting use of speaker embeddings is for text-to-speech (TTS) synthesis (Jia et al. 2019), where speaker embeddings are used to encapsulate speaker identity for later synthesis, potentially synthesising a new speaker from only a single utterance (and a single embedding). With this usage of speaker embeddings, it is possible to envisage use-cases where different sources of variability may indeed be desirable. For example, a user may want a TTS system to produce the same recording characteristics of a reference utterance, and thus adversarially suppressing this information would be counterproductive. On the other hand, the disentanglement of speaker attributes may be very desirable with TTS or voice conversion (Ding and Gutierrez-Osuna 2019), since transferring or modifying particular speaker attributes may enable extra functionality. Here, the decreased verification and recognition performance incurred from adversarial disentanglement may be offset by the benefits of controllable of attributes.

Even without disentanglement, it is possible that the embeddings trained with speaker attribute tasks also provide some benefit by having the embedding space explicitly able to describe age and nationality, and this is an area which warrants further investigation.

8.2 Future work

In the previous chapters, we outlined how speaker embedding factors can be manipulated in order to better serve downstream tasks. This section will outline ideas and areas which could improve our understanding of representation learning for the human voice, or lead to further improvements for recognition tasks.

8.2.1 Speaker distribution adaptation

In the case of speaker distribution adaptation (Chapter 7), we think that it would be worth exploring whether it is possible to perform a computationally cheaper form of speaker distribution adaptation, avoiding using gradient descent to adapt the model. We would look to do so by incorporating ideas from the fields of speaker adaptation for ASR (Li and Sim 2010; Klejch 2020). For example, some ASR acoustic models use additional features, such as *i*-vectors (Saon et al. 2013) or *x*-vectors (Rownicka et al. 2019) in order to inform the acoustic models about target speaker and channel characteristics (these acoustic models are trying to predict phonetic information). By

informing the models of the speaker characteristics, they are better able to contextualise phonetic information. We believe a similar approach may be possible when adapting speaker embedding extractors to different speaker distributions. As we were able to show that fine-tuning on training set speakers (which did not involve any test set labels) improved performance on the test set, perhaps it is possible to learn how to adapt our embedding extractor without the need for more gradient descent steps.

For example, if we can train a model to encode information about the target speaker distribution, perhaps this can be used to transform the embedding space, such that target speaker distribution is more discriminative in this representation. There are a number of ways we could approach this, but to start with, it may be interesting to see if we could train a separate neural network to adapt to new speaker distributions by learning a non-linear transformation of the speaker embedding space. We could do this by randomly sampling a subset of speakers, and then optimizing the adaptation network to learn a transformation that increases separability in context of that subset. This would be a meta-learner that learns to adapt the learned representation (Klejch 2020).

Another idea is to borrow from the work of Swietojanski and Renals (2014), which learns to re-weight hidden units based on speaker-dependent parameters for ASR (LHUC). Instead of using a separate adaptation model, we would instead train our model to always be appropriate for a given speaker distribution, re-weighting based on the current distribution.

8.2.2 Disentanglement

8.2.2.1 Unsupervised Disentanglement

In Chapter 6, we looked at the disentanglement of speaker attributes using a combination of multi-task learning and adversarial training. However, our method relies on the web-scraped attribute labels we acquired. In the literature for disentangled representations, a common goal is to be able to obtain disentanglement in an unsupervised manner, meaning without additional labels (Higgins et al. 2018b; Kim and Mnih 2019; Burns et al. 2021).

While we were able to show disentanglement for our method, which has the benefit of being controllable with regards to dimension and attribute, it would be interesting to pursue an unsupervised approach, instead leveraging the labels we have collected in

order to measure the disentanglement of different unsupervised approaches. Notably, Higgins et al. (2018b) proposed a disentanglement metric based on ground truth latent factors for vision and human faces. To our knowledge, the same exploration has not been performed for speaker embeddings, most likely due to the lack of labels for such attributes.

8.2.2.2 Downstream applications for disentangled representations

As mentioned previously, disentangled representations have clear advantanges and applications in cases like protecting privacy of sensitive attributes. However, disentanglement, as a property, has also shown to be beneficial for transfer learning tasks in visual embeddings, such as in the work of van Steenkiste et al. (2020), where they found that disentangled visual representations do indeed improve downstream abstract visual reasoning tasks. The downstream usage for disentangled speaker representations may be less clear, but there are indeed several applications that this could be tested on.

As mentioned above, some speaker adaptive acoustic model architectures rely on using speaker embeddings as auxiliary factors to enable them to better inform predicting phonetic information (Klejch 2020). Exploring how disentangled speaker representations and speaker adaptive ASR interact is certainly worth pursuing.

Furthermore, the task of voice conversion, changing the identity of input speech and synthesising that into a new voice, often also makes use of speaker embeddings (Doddipatla et al. 2017; Williams et al. 2020). Having controllable disentangled features, such as accent, age, or gender would be a useful application of the disentangled speaker embeddings.

Bibliography

Aloufi, Ranya, Hamed Haddadi, and David Boyle (Nov. 9, 2020). "Privacy-Preserving Voice Analysis via Disentangled Representations". In: *Proceedings of the 2020 ACM SIGSAC Conference on Cloud Computing Security Workshop*, pp. 1–14. DOI: 10.1145/3411495.3421355. arXiv: 2007.15064. URL: http://arxiv.org/abs/2007.15064 (visited on 02/02/2022).

Arandjelović, Relja, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic (May 2, 2016). *NetVLAD: CNN Architecture for Weakly Supervised Place Recognition*. DOI: 10.48550/arXiv.1511.07247. arXiv: 1511.07247 [cs]. URL: http://arxiv.org/abs/1511.07247 (visited on 11/16/2022). preprint.

Bai, Zhongxin and Xiao-Lei Zhang (Apr. 3, 2021). *Speaker Recognition Based on Deep Learning: An Overview*. DOI: 10.48550/arXiv.2012.00931. arXiv: 2012.00931 [eess]. URL: http://arxiv.org/abs/2012.00931 (visited on 10/18/2022). preprint.

Baldi, Pierre and Peter Sadowski (May 2014). "The Dropout Learning Algorithm". In: *Artificial Intelligence* 210, pp. 78–122. ISSN: 0004-3702. DOI: 10.1016/j.artint.2014.02.004. pmid: 24771879.

Barz, Björn and Joachim Denzler (Jan. 2019). "Hierarchy-Based Image Embeddings for Semantic Image Retrieval". In: *2019 IEEE Winter Conference on Applications of Computer Vision (WACV)*. 2019 IEEE Winter Conference on Applications of Computer Vision (WACV), pp. 638–647. DOI: 10.1109/WACV.2019.00073.

Belin, Pascal, Shirley Fecteau, and Catherine Bédard (Mar. 2004). "Thinking the Voice: Neural Correlates of Voice Perception". In: *Trends in Cognitive Sciences* 8.3, pp. 129–135. ISSN: 1364-6613. DOI: 10.1016/j.tics.2004.01.008. pmid: 15301753.

Bell, Peter, Pawel Swietojanski, and Steve Renals (Feb. 2017). "Multitask Learning of Context-Dependent Targets in Deep Neural Network Acoustic Models". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 25.2, pp. 238–247. ISSN:

2329-9290, 2329-9304. DOI: 10.1109/TASLP.2016.2630305. URL: http://ieeexplore.ieee.org/document/7747518/ (visited on 03/11/2021).

Benaroya, Laurent, Nicolas Obin, and Axel Roebel (July 27, 2021). *Beyond Voice Identity Conversion: Manipulating Voice Attributes by Adversarial Learning of Structured Disentangled Representations*. arXiv: 2107.12346 [cs, eess]. URL: http://arxiv.org/abs/2107.12346 (visited on 11/20/2022). preprint.

Bengio, Yoshua, Aaron Courville, and Pascal Vincent (Apr. 23, 2014). "Representation Learning: A Review and New Perspectives". arXiv: 1206.5538 [cs]. URL: http://arxiv.org/abs/1206.5538 (visited on 03/28/2022).

Bengio, Yoshua, Jérôme Louradour, Ronan Collobert, and Jason Weston (2009). "Curriculum Learning". In: *Proceedings of the 26th Annual International Conference on Machine Learning - ICML '09*. The 26th Annual International Conference. Montreal, Quebec, Canada: ACM Press, pp. 1–8. ISBN: 978-1-60558-516-1. DOI: 10.1145/1553374.1553380. URL: http://portal.acm.org/citation.cfm?doid=1553374.1553380 (visited on 03/11/2021).

Brown, Andrew, Jaesung Huh, Joon Son Chung, Arsha Nagrani, Daniel Garcia-Romero, and Andrew Zisserman (Nov. 16, 2022). *VoxSRC 2021: The Third VoxCeleb Speaker Recognition Challenge*. arXiv: 2201.04583 [cs, eess]. URL: http://arxiv.org/abs/2201.04583 (visited on 07/29/2023). preprint.

Buda, Mateusz, Atsuto Maki, and Maciej A. Mazurowski (Oct. 2018). "A Systematic Study of the Class Imbalance Problem in Convolutional Neural Networks". In: *Neural Networks* 106, pp. 249–259. ISSN: 08936080. DOI: 10.1016/j.neunet.2018.07.011. arXiv: 1710.05381 [cs, stat]. URL: http://arxiv.org/abs/1710.05381 (visited on 11/16/2022).

Burns, Andrea, Aaron Sarna, Dilip Krishnan, and Aaron Maschinot (Aug. 14, 2021). *Unsupervised Disentanglement without Autoencoding: Pitfalls and Future Directions*. arXiv: 2108.06613 [cs]. URL: http://arxiv.org/abs/2108.06613 (visited on 11/30/2022). preprint.

Canavan, Alexandra, David Graff, and George Zipperlen (1997). *CALLHOME American English Speech*. Linguistic Data Consortium. DOI: 10.35111/EXQ3-X930. URL: https://catalog.ldc.upenn.edu/LDC97S42 (visited on 11/28/2022).

Caruana, Rich (1998). "Multitask Learning". In: *Learning to Learn*. Ed. by Sebastian Thrun and Lorien Pratt. Boston, MA: Springer US, pp. 95–133. ISBN: 978-1-4615-5529-2. DOI: 10.1007/978-1-4615-5529-2_5. URL: https://doi.org/10.1007/978-1-4615-5529-2_5 (visited on 11/16/2022).

Cesari, Ugo, Giuseppe De Pietro, Elio Marciano, Ciro Niri, Giovanna Sannino, and Laura Verde (2018). "Voice Disorder Detection via an M-Health System: Design and Results of a Clinical Study to Evaluate Vox4Health". In: *BioMed Research International* 2018, p. 8193694. ISSN: 2314-6141. DOI: 10.1155/2018/8193694. pmid: 30175144.

Chen, Chi-hau (1976). *Pattern Recognition and Artificial Intelligence: Proceedings of the Joint Workshop on Pattern Recognition and Artificial Intelligence, Held at Hyannis, Massachusetts, June 1-3, 1976*. Academic Press. 640 pp. ISBN: 978-0-12-170950-1. Google Books: wW9QAAAAMAAJ.

Chen, Ke and Ahmad Salman (Nov. 2011). "Learning Speaker-Specific Characteristics With a Deep Neural Architecture". In: *IEEE Transactions on Neural Networks* 22.11, pp. 1744–1756. ISSN: 1941-0093. DOI: 10.1109/TNN.2011.2167240.

Chen, Zhengyang, Shuai Wang, Yanmin Qian, and Kai Yu (May 2020). "Channel Invariant Speaker Embedding Learning with Joint Multi-Task and Adversarial Training". In: *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6574–6578. DOI: 10.1109/ICASSP40776.2020.9053905.

Chung, Joon Son, Jaesung Huh, and Seongkyu Mun (Feb. 3, 2020). *Delving into VoxCeleb: Environment Invariant Speaker Recognition*. arXiv: 1910.11238 [cs, eess]. URL: http://arxiv.org/abs/1910.11238 (visited on 11/27/2022). preprint.

Chung, Joon Son, Arsha Nagrani, Ernesto Coto, Weidi Xie, Mitchell McLaren, Douglas A Reynolds, and Andrew Zisserman (n.d.). "VOXSRC 2019: THE FIRST VOXCELEB SPEAKER RECOGNITION CHALLENGE". In: ().

Chung, Joon Son, Arsha Nagrani, and Andrew Zisserman (Sept. 2, 2018). "VoxCeleb2: Deep Speaker Recognition". In: *Interspeech 2018*. Interspeech 2018. ISCA, pp. 1086–1090. DOI: 10.21437/Interspeech.2018-1929. URL: https://www.isca-speech.org/archive/interspeech_2018/chung18b_interspeech.html (visited on 11/16/2022).

Dahl, George E., Tara N. Sainath, and Geoffrey E. Hinton (May 2013). "Improving Deep Neural Networks for LVCSR Using Rectified Linear Units and Dropout". In: *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. 2013 IEEE International Conference on Acoustics, Speech and Signal Processing, pp. 8609–8613. DOI: 10.1109/ICASSP.2013.6639346.

Dehak, Najim, Patrick J. Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet (May 2011). "Front-End Factor Analysis for Speaker Verification". In: *IEEE Transactions on Audio, Speech, and Language Processing* 19.4, pp. 788–798. ISSN: 1558-7924. DOI: 10.1109/TASL.2010.2064307.

Dempster, A. P., N. M. Laird, and D. B. Rubin (1977). "Maximum Likelihood from Incomplete Data via the EM Algorithm". In: *Journal of the Royal Statistical Society. Series B (Methodological)* 39.1, pp. 1–38. ISSN: 0035-9246. JSTOR: 2984875. URL: https://www.jstor.org/stable/2984875 (visited on 11/23/2022).

Deng, Jia, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei (June 2009). "ImageNet: A Large-Scale Hierarchical Image Database". In: *2009 IEEE Conference on Computer Vision and Pattern Recognition*. 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp. 248–255. DOI: 10.1109/CVPR.2009.5206848.

Deng, Jiankang, Jia Guo, Jing Yang, Niannan Xue, Irene Kotsia, and Stefanos Zafeiriou (2018). "ArcFace: Additive Angular Margin Loss for Deep Face Recognition". In: *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1. ISSN: 0162-8828, 2160-9292, 1939-3539. DOI: 10.1109/TPAMI.2021.3087709. arXiv: 1801.07698 [cs]. URL: http://arxiv.org/abs/1801.07698 (visited on 11/16/2022).

Dey, Subhadeep, Takafumi Koshinaka, Petr Motlicek, and Srikanth Madikeri (Apr. 2018). "DNN Based Speaker Embedding Using Content Information for Text-Dependent Speaker Verification". In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5344–5348. DOI: 10.1109/ICASSP.2018.8461389.

DiCarlo, James J. and David D. Cox (Aug. 1, 2007). "Untangling Invariant Object Recognition". In: *Trends in Cognitive Sciences* 11.8, pp. 333–341. ISSN: 1364-6613. DOI: 10.1016/j.tics.2007.06.010. URL: https://www.sciencedirect.com/science/article/pii/S1364661307001593 (visited on 03/28/2022).

Ding, Shaojin and Ricardo Gutierrez-Osuna (Sept. 15, 2019). "Group Latent Embedding for Vector Quantized Variational Autoencoder in Non-Parallel Voice Conversion". In: *Interspeech 2019*. Interspeech 2019. ISCA, pp. 724–728. DOI: 10.21437/Interspeech.2019-1198. URL: https://www.isca-speech.org/archive/interspeech_2019/ding19_interspeech.html (visited on 08/02/2023).

Doddipatla, Rama, Norbert Braunschweiler, and Ranniery Maia (Aug. 20, 2017). "Speaker Adaptation in DNN-Based Speech Synthesis Using d-Vectors". In: *Interspeech 2017*. Interspeech 2017. ISCA, pp. 3404–3408. DOI: 10.21437/Interspeech.2017-1038. URL: https://www.isca-speech.org/archive/interspeech_2017/doddipatla17_interspeech.html (visited on 11/30/2022).

Finn, Chelsea, Pieter Abbeel, and Sergey Levine (July 18, 2017). *Model-Agnostic Meta-Learning for Fast Adaptation of Deep Networks*. DOI: 10.48550/arXiv.1703.03400. arXiv: 1703.03400 [cs]. URL: http://arxiv.org/abs/1703.03400 (visited on 11/16/2022). preprint.

Ganin, Yaroslav, Evgeniya Ustinova, Hana Ajakan, Pascal Germain, Hugo Larochelle, François Laviolette, Mario Marchand, and Victor Lempitsky (May 26, 2016). *Domain-Adversarial Training of Neural Networks*. DOI: 10.48550/arXiv.1505.07818. arXiv: 1505.07818 [cs, stat]. URL: http://arxiv.org/abs/1505.07818 (visited on 11/13/2022). preprint.

Goodfellow, Ian J., Mehdi Mirza, Da Xiao, Aaron Courville, and Yoshua Bengio (Mar. 3, 2015). *An Empirical Investigation of Catastrophic Forgetting in Gradient-Based Neural Networks*. DOI: 10.48550/arXiv.1312.6211. arXiv: 1312.6211 [cs, stat]. URL: http://arxiv.org/abs/1312.6211 (visited on 11/16/2022). preprint.

Hansen, John H.L. and Taufiq Hasan (Nov. 2015). "Speaker Recognition by Machines and Humans: A Tutorial Review". In: *IEEE Signal Processing Magazine* 32.6, pp. 74–99. ISSN: 1558-0792. DOI: 10.1109/MSP.2015.2462851.

He, Kaiming, Xiangyu Zhang, Shaoqing Ren, and Jian Sun (June 2016). "Deep Residual Learning for Image Recognition". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Las Vegas, NV, USA: IEEE, pp. 770–778. ISBN: 978-1-4673-8851-1. DOI: 10.1109/CVPR.2016.90. URL: http://ieeexplore.ieee.org/document/7780459/ (visited on 11/16/2022).

Heck, Larry P., Yochai Konig, M. Kemal Sönmez, and Mitch Weintraub (June 1, 2000). "Robustness to Telephone Handset Distortion in Speaker Recognition by Discriminative Feature Design". In: *Speech Communication* 31.2, pp. 181–192. ISSN: 0167-6393. DOI: 10.1016/S0167-6393(99)00077-1. URL: https://www.sciencedirect.com/science/article/pii/S0167639399000771 (visited on 11/24/2022).

Heigold, Georg, Ignacio Moreno, Samy Bengio, and Noam Shazeer (Mar. 2016). "End-to-End Text-Dependent Speaker Verification". In: *2016 IEEE International Con-*

ference on Acoustics, Speech and Signal Processing (ICASSP). 2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Shanghai: IEEE, pp. 5115–5119. ISBN: 978-1-4799-9988-0. DOI: 10.1109/ICASSP.2016.7472652. URL: http://ieeexplore.ieee.org/document/7472652/ (visited on 11/09/2022).

Higgins, Irina, David Amos, David Pfau, Sebastien Racaniere, Loic Matthey, Danilo Rezende, and Alexander Lerchner (Dec. 5, 2018a). "Towards a Definition of Disentangled Representations". arXiv: 1812.02230 [cs, stat]. URL: http://arxiv.org/abs/1812.02230 (visited on 03/28/2022).

Higgins, Irina, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner (2018b). "Beta-VAE: Learning Basic Visual Concepts with a Constrained Variational Framework". In: International Conference on Learning Representations. URL: https://openreview.net/forum?id=Sy2fzU9gl (visited on 11/30/2022).

Hinton, Geoffrey, Li Deng, Dong Yu, George E. Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N. Sainath, and Brian Kingsbury (Nov. 2012). "Deep Neural Networks for Acoustic Modeling in Speech Recognition: The Shared Views of Four Research Groups". In: *IEEE Signal Processing Magazine* 29.6, pp. 82–97. ISSN: 1558-0792. DOI: 10.1109/MSP.2012.2205597.

Hochreiter, Sepp and Jürgen Schmidhuber (Nov. 1, 1997). "Long Short-Term Memory". In: *Neural Computation* 9.8, pp. 1735–1780. ISSN: 0899-7667. DOI: 10.1162/neco.1997.9.8.1735. URL: https://doi.org/10.1162/neco.1997.9.8.1735 (visited on 11/16/2022).

Horiguchi, Shota, Yusuke Fujita, Shinji Watanabe, Yawen Xue, and Kenji Nagamatsu (Oct. 5, 2020). "End-to-End Speaker Diarization for an Unknown Number of Speakers with Encoder-Decoder Based Attractors". arXiv: 2005.09921 [cs, eess]. URL: http://arxiv.org/abs/2005.09921 (visited on 08/03/2021).

Horiguchi, Shota, Shinji Watanabe, Paola Garcia, Yawen Xue, Yuki Takashima, and Yohei Kawaguchi (July 4, 2021). "Towards Neural Diarization for Unlimited Numbers of Speakers Using Global and Local Attractors". arXiv: 2107.01545 [cs, eess]. URL: http://arxiv.org/abs/2107.01545 (visited on 08/05/2021).

Huang, Chen, Yining Li, Chen Change Loy, and Xiaoou Tang (Apr. 29, 2019). *Deep Imbalanced Learning for Face Recognition and Attribute Prediction*. DOI: 10.485

50/arXiv.1806.00194. arXiv: 1806.00194 [cs]. URL: http://arxiv.org/abs/1806.00194 (visited on 11/16/2022). preprint.

Huh, Jaesung, Andrew Brown, Jee-weon Jung, Joon Son Chung, Arsha Nagrani, Daniel Garcia-Romero, and Andrew Zisserman (Mar. 6, 2023). *VoxSRC 2022: The Fourth VoxCeleb Speaker Recognition Challenge*. arXiv: 2302.10248 [cs, eess]. URL: http://arxiv.org/abs/2302.10248 (visited on 07/29/2023). preprint.

Huh, Jaesung, Hee Soo Heo, Jingu Kang, Shinji Watanabe, and Joon Son Chung (Oct. 30, 2020). "Augmentation Adversarial Training for Self-Supervised Speaker Recognition". arXiv: 2007.12085 [cs, eess]. URL: http://arxiv.org/abs/2007.12085 (visited on 03/28/2021).

Ioffe, Sergey (2006). "Probabilistic Linear Discriminant Analysis". In: *Computer Vision – ECCV 2006*. Ed. by Aleš Leonardis, Horst Bischof, and Axel Pinz. Vol. 3954. Berlin, Heidelberg: Springer Berlin Heidelberg, pp. 531–542. ISBN: 978-3-540-33838-3 978-3-540-33839-0. DOI: 10.1007/11744085_41. URL: http://link.springer.com/10.1007/11744085_41 (visited on 11/13/2022).

Jain, A.K., A. Ross, and S. Prabhakar (Jan. 2004). "An Introduction to Biometric Recognition". In: *IEEE Transactions on Circuits and Systems for Video Technology* 14.1, pp. 4–20. ISSN: 1051-8215. DOI: 10.1109/TCSVT.2003.818349. URL: http://ieeexplore.ieee.org/document/1262027/ (visited on 11/26/2022).

Jessen, Michael (2007). "Speaker Classification in Forensic Phonetics and Acoustics". In: *Speaker Classification I: Fundamentals, Features, and Methods*. Ed. by Christian Müller. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, pp. 180–204. ISBN: 978-3-540-74200-5. DOI: 10.1007/978-3-540-74200-5_10. URL: https://doi.org/10.1007/978-3-540-74200-5_10 (visited on 11/16/2022).

Jia, Ye, Yu Zhang, Ron J. Weiss, Quan Wang, Jonathan Shen, Fei Ren, Zhifeng Chen, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, and Yonghui Wu (Jan. 2, 2019). *Transfer Learning from Speaker Verification to Multispeaker Text-To-Speech Synthesis*. DOI: 10.48550/arXiv.1806.04558. arXiv: 1806.04558 [cs, eess]. URL: http://arxiv.org/abs/1806.04558 (visited on 08/02/2023). preprint.

Jung, Jee-weon, Hee-Soo Heo, Ju-ho Kim, Hye-jin Shim, and Ha-Jin Yu (July 16, 2019). *RawNet: Advanced End-to-End Deep Neural Network Using Raw Waveforms for Text-Independent Speaker Verification*. DOI: 10.48550/arXiv.1904.08104. arXiv: 1904.08104 [cs, eess]. URL: http://arxiv.org/abs/1904.08104 (visited on 11/13/2022). preprint.

Jung, Jee-weon, Seung-bin Kim, Hye-jin Shim, Ju-ho Kim, and Ha-Jin Yu (May 7, 2020). *Improved RawNet with Feature Map Scaling for Text-independent Speaker Verification Using Raw Waveforms*. DOI: 10.48550/arXiv.2004.00526. arXiv: 2004.00526 [cs, eess]. URL: http://arxiv.org/abs/2004.00526 (visited on 11/13/2022). preprint.

Kanda, Naoyuki, Xiong Xiao, Jian Wu, Tianyan Zhou, Yashesh Gaur, Xiaofei Wang, Zhong Meng, Zhuo Chen, and Takuya Yoshioka (July 6, 2021). "A Comparative Study of Modular and Joint Approaches for Speaker-Attributed ASR on Monaural Long-Form Audio". arXiv: 2107.02852 [cs, eess]. URL: http://arxiv.org/abs/2107.02852 (visited on 08/02/2021).

Kaushal, Vishal, Rishabh Iyer, Suraj Kothawade, Rohan Mahadev, Khoshrav Doctor, and Ganesh Ramakrishnan (Jan. 3, 2019). *Learning From Less Data: A Unified Data Subset Selection and Active Learning Framework for Computer Vision*. DOI: 10.48550/arXiv.1901.01151. arXiv: 1901.01151 [cs]. URL: http://arxiv.org/abs/1901.01151 (visited on 11/16/2022). preprint.

Kenny, P., M. Mihoubi, and Pierre Dumouchel (Sept. 1, 2003). "New MAP Estimators for Speaker Recognition". In: *8th European Conference on Speech Communication and Technology (Eurospeech 2003)*. 8th European Conference on Speech Communication and Technology (Eurospeech 2003). ISCA, pp. 2961–2964. DOI: 10.21437/Eurospeech.2003-759. URL: https://www.isca-speech.org/archive/eurospeech_2003/kenny03_eurospeech.html (visited on 11/23/2022).

Kenny, Patrick (2010). "Bayesian Speaker Verification with Heavy-Tailed Priors". In: *Proc. The Speaker and Language Recognition Workshop (Odyssey 2010)*, paper 14.

Kenny, Patrick, Gilles Boulianne, Pierre Ouellet, and Pierre Dumouchel (May 2007). "Joint Factor Analysis Versus Eigenchannels in Speaker Recognition". In: *IEEE Transactions on Audio, Speech and Language Processing* 15.4, pp. 1435–1447. ISSN: 1558-7916. DOI: 10.1109/TASL.2006.881693. URL: http://ieeexplore.ieee.org/document/4156202/ (visited on 11/23/2022).

Khan, Salman H., Munawar Hayat, Mohammed Bennamoun, Ferdous Sohel, and Roberto Togneri (Mar. 23, 2017). *Cost Sensitive Learning of Deep Feature Representations from Imbalanced Data*. DOI: 10.48550/arXiv.1508.03422. arXiv: 1508.03422 [cs]. URL: http://arxiv.org/abs/1508.03422 (visited on 11/16/2022). preprint.

Kiela, Douwe and Léon Bottou (2014). "Learning Image Embeddings Using Convolutional Neural Networks for Improved Multi-Modal Semantics". In: *Proceedings*

of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP). Doha, Qatar: Association for Computational Linguistics, pp. 36–45. DOI: 10.3115/v1/D14-1005. URL: http://aclweb.org/anthology/D14-1005 (visited on 11/19/2022).

Kim, Hyunjik and Andriy Mnih (July 9, 2019). *Disentangling by Factorising*. DOI: 10.48550/arXiv.1802.05983. arXiv: 1802.05983 [cs, stat]. URL: http://arxiv.org/abs/1802.05983 (visited on 11/30/2022). preprint.

Klejch, Ondrej (2020). "Learning to Adapt: Meta-Learning Approaches for Speaker Adaptation". The University of Edinburgh.

Ko, Tom, Vijayaditya Peddinti, Daniel Povey, Michael L. Seltzer, and Sanjeev Khudanpur (Mar. 2017). "A Study on Data Augmentation of Reverberant Speech for Robust Speech Recognition". In: *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5220–5224. DOI: 10.1109/ICASSP.2017.7953152.

Konig, Yochai, Larry Heck, Mitchel Weintraub, Mustafa Sönmez, and R E (May 1, 1998). "Nonlinear Discriminant Feature Extraction For Robust Text-Independent Speaker Recognition". In: *Proc. RLA2C, ESCA workshop on Speaker Recog- nition and its Commercial and Forensic Applications*.

Kovacić, Damir and Evan Balaban (Sept. 1, 2009). "Voice Gender Perception by Cochlear Implantees". In: *The Journal of the Acoustical Society of America* 126, pp. 762–75. DOI: 10.1121/1.3158855.

LeCun, Yann, Yoshua Bengio, and Geoffrey Hinton (May 28, 2015). "Deep Learning". In: *Nature* 521.7553, pp. 436–444. ISSN: 0028-0836, 1476-4687. DOI: 10.1038/nature14539. URL: http://www.nature.com/articles/nature14539 (visited on 11/20/2022).

Li, Bo and Khe Chai Sim (Sept. 26, 2010). "Comparison of Discriminative Input and Output Transformations for Speaker Adaptation in the Hybrid NN/HMM Systems". In: *Interspeech 2010*. Interspeech 2010. ISCA, pp. 526–529. DOI: 10.21437/Interspeech.2010-214. URL: https://www.isca-speech.org/archive/interspeech_2010/li10_interspeech.html (visited on 11/30/2022).

Lin, Qingjian, Ruiqing Yin, Ming Li, Hervé Bredin, and Claude Barras (Sept. 15, 2019). "LSTM Based Similarity Measurement with Spectral Clustering for Speaker Diarization". In: *Interspeech 2019*, pp. 366–370. DOI: 10.21437/Interspeech.

2019-1388. arXiv: 1907.10393 [cs, eess, stat]. URL: http://arxiv.org/abs/1907.10393 (visited on 11/09/2022).

Liu, Weiyang, Yandong Wen, Zhiding Yu, Ming Li, Bhiksha Raj, and Le Song (July 2017). "SphereFace: Deep Hypersphere Embedding for Face Recognition". In: *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Honolulu, HI: IEEE, pp. 6738–6746. ISBN: 978-1-5386-0457-1. DOI: 10.1109/CVPR.2017.713. URL: http://ieeexplore.ieee.org/document/8100196/ (visited on 11/16/2022).

Liu, Yi, Liang He, Jia Liu, and Michael T. Johnson (Dec. 5, 2019). "Introducing Phonetic Information to Speaker Embedding for Speaker Verification". In: *EURASIP Journal on Audio, Speech, and Music Processing* 2019.1, p. 19. ISSN: 1687-4722. DOI: 10.1186/s13636-019-0166-8. URL: https://doi.org/10.1186/s13636-019-0166-8 (visited on 11/16/2022).

Liu, Yi Chieh, Eunjung Han, Chul Lee, and Andreas Stolcke (June 14, 2021). "End-to-End Neural Diarization: From Transformer to Conformer". arXiv: 2106.07167 [cs, eess]. URL: http://arxiv.org/abs/2106.07167 (visited on 08/05/2021).

Locatello, Francesco, Stefan Bauer, Mario Lucic, Gunnar Rätsch, Sylvain Gelly, Bernhard Schölkopf, and Olivier Bachem (June 18, 2019). "Challenging Common Assumptions in the Unsupervised Learning of Disentangled Representations". arXiv: 1811.12359 [cs, stat]. URL: http://arxiv.org/abs/1811.12359 (visited on 03/28/2022).

Luu, Chau, Peter Bell, and Steve Renals (May 2020a). "Channel Adversarial Training for Speaker Verification and Diarization". In: *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7094–7098. DOI: 10.1109/ICASSP40776.2020.9053323.

– (Nov. 1, 2020b). "Dropping Classes for Deep Speaker Representation Learning". In: *The Speaker and Language Recognition Workshop (Odyssey 2020)*. The Speaker and Language Recognition Workshop (Odyssey 2020). ISCA, pp. 357–364. DOI: 10.21437/Odyssey.2020-50. URL: https://www.isca-speech.org/archive/odyssey_2020/luu20_odyssey.html (visited on 10/18/2022).

– (Aug. 30, 2021). "Leveraging Speaker Attribute Information Using Multi Task Learning for Speaker Verification and Diarization". In: *Interspeech 2021*. Interspeech 2021. ISCA, pp. 491–495. DOI: 10.21437/Interspeech.2021-622. URL: https:

//www.isca-speech.org/archive/interspeech_2021/luu21_interspeech.html.

Luu, Chau, Steve Renals, and Peter Bell (Sept. 18, 2022). "Investigating the Contribution of Speaker Attributes to Speaker Separability Using Disentangled Speaker Representations". In: *Interspeech 2022*. Interspeech 2022. ISCA, pp. 610–614. DOI: 10.21437/Interspeech.2022-10643. URL: https://www.isca-speech.org/archive/interspeech_2022/luu22_interspeech.html.

Mahendran, Aravindh and Andrea Vedaldi (Nov. 26, 2014). *Understanding Deep Image Representations by Inverting Them*. DOI: 10.48550/arXiv.1412.0035. arXiv: 1412.0035 [cs]. URL: http://arxiv.org/abs/1412.0035 (visited on 11/20/2022). preprint.

Maiti, Soumi, Erik Marchi, and Alistair Conkie (May 2020). "Generating Multilingual Voices Using Speaker Space Translation Based on Bilingual Speaker Data". In: *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7624–7628. DOI: 10.1109/ICASSP40776.2020.9054305.

Markel, J., B. Oshika, and A. Gray (Aug. 1977). "Long-Term Feature Averaging for Speaker Recognition". In: *IEEE Transactions on Acoustics, Speech, and Signal Processing* 25.4, pp. 330–337. ISSN: 0096-3518. DOI: 10.1109/TASSP.1977.1162961.

Matějka, Pavel, Ondřej Novotný, Oldřich Plchot, Lukáš Burget, Mireia Diez Sánchez, and Jan Černocký (Aug. 20, 2017). "Analysis of Score Normalization in Multilingual Speaker Recognition". In: *Interspeech 2017*. Interspeech 2017. ISCA, pp. 1567–1571. DOI: 10.21437/Interspeech.2017-803. URL: http://www.isca-speech.org/archive/Interspeech_2017/abstracts/0803.html (visited on 02/16/2021).

McLaren, Mitchell, Luciana Ferrer, Diego Castan, and Aaron Lawson (Sept. 8, 2016a). "The Speakers in the Wild (SITW) Speaker Recognition Database". In: *Interspeech 2016*. Interspeech 2016. ISCA, pp. 818–822. DOI: 10.21437/Interspeech.2016-1129. URL: https://www.isca-speech.org/archive/interspeech_2016/mclaren16_interspeech.html (visited on 11/28/2022).

– (Sept. 8, 2016b). "The Speakers in the Wild (SITW) Speaker Recognition Database". In: *Interspeech 2016*. Interspeech 2016. ISCA, pp. 818–822. DOI: 10.21437/Int

erspeech.2016-1129. URL: https://www.isca-speech.org/archive/interspeech_2016/mclaren16_interspeech.html (visited on 11/16/2022).

McQuitty, Louis L. (1957). "Elementary Linkage Analysis for Isolating Orthogonal and Oblique Types and Typal Relevancies". In: *Educational and Psychological Measurement* 17.2, pp. 207–229. DOI: 10.1177/001316445701700204. eprint: https://doi.org/10.1177/001316445701700204. URL: https://doi.org/10.1177/001316445701700204.

Meng, Zhong, Yong Zhao, Jinyu Li, and Yifan Gong (May 2019). "Adversarial Speaker Verification". In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6216–6220. DOI: 10.1109/ICASSP.2019.8682488.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg Corrado, and Jeffrey Dean (Oct. 16, 2013). *Distributed Representations of Words and Phrases and Their Compositionality*. DOI: 10.48550/arXiv.1310.4546. arXiv: 1310.4546 [cs, stat]. URL: http://arxiv.org/abs/1310.4546 (visited on 11/19/2022). preprint.

Mueller, P. B. (May 1997). "The Aging Voice". In: *Seminars in Speech and Language* 18.2, 159–168, quiz 168–169. ISSN: 0734-0478. DOI: 10.1055/s-2008-1064070. pmid: 9195688.

Nagrani, Arsha, Joon Son Chung, Jaesung Huh, Andrew Brown, Ernesto Coto, Weidi Xie, Mitchell McLaren, Douglas A. Reynolds, and Andrew Zisserman (Dec. 12, 2020). *VoxSRC 2020: The Second VoxCeleb Speaker Recognition Challenge*. arXiv: 2012.06867 [cs, eess]. URL: http://arxiv.org/abs/2012.06867 (visited on 07/29/2023). preprint.

Nagrani, Arsha, Joon Son Chung, and Andrew Zisserman (Aug. 20, 2017). "VoxCeleb: A Large-Scale Speaker Identification Dataset". In: *Interspeech 2017*. Interspeech 2017. ISCA, pp. 2616–2620. DOI: 10.21437/Interspeech.2017-950. URL: https://www.isca-speech.org/archive/interspeech_2017/nagrani17_interspeech.html (visited on 11/16/2022).

Nautsch, Andreas, Jose Patino, Natalia Tomashenko, Junichi Yamagishi, Paul-Gauthier Noe, Jean-Francois Bonastre, Massimiliano Todisco, and Nicholas Evans (Oct. 25, 2020). "The Privacy ZEBRA: Zero Evidence Biometric Recognition Assessment". In: *Interspeech 2020*, pp. 1698–1702. DOI: 10.21437/Interspeech.2020-1815. arXiv: 2005.09413. URL: http://arxiv.org/abs/2005.09413 (visited on 01/27/2022).

Neyshabur, Behnam, Srinadh Bhojanapalli, David Mcallester, and Nati Srebro (2017). "Exploring Generalization in Deep Learning". In: *Advances in Neural Information Processing Systems*. Vol. 30. Curran Associates, Inc. URL: https://papers.nips.cc/paper/2017/hash/10ce03a1ed01077e3e289f3e53c72813-Abstract.html (visited on 11/16/2022).

Noé, Paul-Gauthier, Mohammad Mohammadamini, Driss Matrouf, Titouan Parcollet, Andreas Nautsch, and Jean-François Bonastre (June 16, 2021). "Adversarial Disentanglement of Speaker Representation for Attribute-Driven Privacy Preservation". arXiv: 2012.04454 [cs, eess]. URL: http://arxiv.org/abs/2012.04454 (visited on 07/25/2021).

Noé, Paul-Gauthier, Andreas Nautsch, Driss Matrouf, Pierre-Michel Bousquet, and Jean-François Bonastre (Jan. 23, 2022). "A Bridge between Features and Evidence for Binary Attribute-Driven Perfect Privacy". arXiv: 2110.05840 [cs, eess]. URL: http://arxiv.org/abs/2110.05840 (visited on 03/14/2022).

Okabe, Koji, Takafumi Koshinaka, and Koichi Shinoda (Sept. 2, 2018). "Attentive Statistics Pooling for Deep Speaker Embedding". In: *Interspeech 2018*, pp. 2252–2256. DOI: 10.21437/Interspeech.2018-993. arXiv: 1803.10963 [cs, eess]. URL: http://arxiv.org/abs/1803.10963 (visited on 11/16/2022).

Pappagari, Raghavendra, Tianzi Wang, Jesus Villalba, Nanxin Chen, and Najim Dehak (May 2020). "X-Vectors Meet Emotions: A Study On Dependencies Between Emotion and Speaker Recognition". In: *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 7169–7173. DOI: 10.1109/ICASSP40776.2020.9054317.

Park, Tae Jin and Panayiotis Georgiou (Sept. 2, 2018). "Multimodal Speaker Segmentation and Diarization Using Lexical and Acoustic Cues via Sequence to Sequence Neural Networks". In: *Interspeech 2018*. Interspeech 2018. ISCA, pp. 1373–1377. DOI: 10.21437/Interspeech.2018-1364. URL: https://www.isca-speech.org/archive/interspeech_2018/park18b_interspeech.html (visited on 11/19/2022).

Park, Tae Jin, Kyu J. Han, Manoj Kumar, and Shrikanth Narayanan (2020). "Auto-Tuning Spectral Clustering for Speaker Diarization Using Normalized Maximum Eigengap". In: *IEEE Signal Processing Letters* 27, pp. 381–385. ISSN: 1070-9908, 1558-2361. DOI: 10.1109/LSP.2019.2961071. arXiv: 2003.02405. URL: http://arxiv.org/abs/2003.02405 (visited on 09/30/2021).

Parveen, Shahla and Phil Green (Sept. 1, 2003). "Multitask Learning in Connectionist Robust ASR Using Recurrent Neural Networks". In: *8th European Conference on Speech Communication and Technology (Eurospeech 2003)*. 8th European Conference on Speech Communication and Technology (Eurospeech 2003). ISCA, pp. 1813–1816. DOI: 10.21437/Eurospeech.2003-500. URL: https://www.isca-speech.org/archive/eurospeech_2003/parveen03_eurospeech.html (visited on 11/19/2022).

Patel, Vishal M, Raghuraman Gopalan, Ruonan Li, and Rama Chellappa (May 2015). "Visual Domain Adaptation: A Survey of Recent Advances". In: *IEEE Signal Processing Magazine* 32.3, pp. 53–69. ISSN: 1053-5888. DOI: 10.1109/MSP.2014.2347059. URL: https://ieeexplore.ieee.org/document/7078994 (visited on 11/19/2022).

Pearson, Karl (Nov. 1, 1901). "On Lines and Planes of Closest Fit to Systems of Points in Space". In: *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science* 2.11, pp. 559–572. ISSN: 1941-5982. DOI: 10.1080/14786440109462720. URL: https://doi.org/10.1080/14786440109462720 (visited on 11/23/2022).

Pernet, Cyril and Pascal Belin (2012). "The Role of Pitch and Timbre in Voice Gender Categorization". In: *Frontiers in Psychology* 3. ISSN: 1664-1078. URL: https://www.frontiersin.org/articles/10.3389/fpsyg.2012.00023 (visited on 09/11/2022).

Ponting, Keith M. (1999). "Channel Adaptation". In: *Computational Models of Speech Pattern Processing*. Ed. by Keith Ponting. NATO ASI Series. Berlin, Heidelberg: Springer, pp. 112–121. ISBN: 978-3-642-60087-6. DOI: 10.1007/978-3-642-60087-6_12. URL: https://doi.org/10.1007/978-3-642-60087-6_12 (visited on 11/27/2022).

Prince, Simon J.D. and James H. Elder (Oct. 2007). "Probabilistic Linear Discriminant Analysis for Inferences About Identity". In: *2007 IEEE 11th International Conference on Computer Vision*. 2007 IEEE 11th International Conference on Computer Vision, pp. 1–8. DOI: 10.1109/ICCV.2007.4409052.

Raghu, Aniruddh, Maithra Raghu, Samy Bengio, and Oriol Vinyals (Feb. 12, 2020). *Rapid Learning or Feature Reuse? Towards Understanding the Effectiveness of MAML*. DOI: 10.48550/arXiv.1909.09157. arXiv: 1909.09157 [cs, stat]. URL: http://arxiv.org/abs/1909.09157 (visited on 11/19/2022). preprint.

Raj, Desh, Zili Huang, and Sanjeev Khudanpur (Nov. 5, 2020). "Multi-Class Spectral Clustering with Overlaps for Speaker Diarization". arXiv: 2011.02900 [cs, eess]. URL: http://arxiv.org/abs/2011.02900 (visited on 05/27/2021).

Raj, Desh, David Snyder, Daniel Povey, and Sanjeev Khudanpur (Dec. 2019). "Probing the Information Encoded in X-Vectors". In: *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. 2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU), pp. 726–733. DOI: 10.1109/ASRU46091.2019.9003979.

Ramoji, Shreyas, Prashant Krishnan, and Sriram Ganapathy (Nov. 1, 2020). "NPLDA: A Deep Neural PLDA Model for Speaker Verification". In: *The Speaker and Language Recognition Workshop (Odyssey 2020)*, pp. 202–209. DOI: 10.21437/Odyssey.2020-29. arXiv: 2002.03562 [cs, eess]. URL: http://arxiv.org/abs/2002.03562 (visited on 11/13/2022).

Rapoport, Sarah K., Jayme Menier, and Nazaneen Grant (Aug. 2018). "Voice Changes in the Elderly". In: *Otolaryngologic Clinics of North America* 51.4, pp. 759–768. ISSN: 1557-8259. DOI: 10.1016/j.otc.2018.03.012. pmid: 29887345.

Ravanelli, Mirco and Yoshua Bengio (Aug. 9, 2019). *Speaker Recognition from Raw Waveform with SincNet*. DOI: 10.48550/arXiv.1808.00158. arXiv: 1808.00158 [cs, eess]. URL: http://arxiv.org/abs/1808.00158 (visited on 11/13/2022). preprint.

Renz, Andreas, Matthias Baldauf, Edith Maier, and Florian Alt (Sept. 15, 2022). "Alexa, It's Me! An Online Survey on the User Experience of Smart Speaker Authentication". In: *Proceedings of Mensch Und Computer 2022*. MuC '22. New York, NY, USA: Association for Computing Machinery, pp. 14–24. ISBN: 978-1-4503-9690-5. DOI: 10.1145/3543758.3543765. URL: https://doi.org/10.1145/3543758.3543765 (visited on 11/26/2022).

Reynolds, D.A. and R.C. Rose (Jan. 1995). "Robust Text-Independent Speaker Identification Using Gaussian Mixture Speaker Models". In: *IEEE Transactions on Speech and Audio Processing* 3.1, pp. 72–83. ISSN: 1558-2353. DOI: 10.1109/89.365379.

Reynolds, Douglas A. (1997). "Comparison of Background Normalization Methods for Text-Independent Speaker Verification". In: *5th European Conference on Speech Communication and Technology (Eurospeech 1997)*.

Reynolds, Douglas A., Thomas F. Quatieri, and Robert B. Dunn (Jan. 1, 2000). "Speaker Verification Using Adapted Gaussian Mixture Models". In: *Digital Signal Process-*

ing 10.1, pp. 19–41. ISSN: 1051-2004. DOI: 10.1006/dspr.1999.0361. URL: https://www.sciencedirect.com/science/article/pii/S1051200499903615 (visited on 11/23/2022).

Richiardi, Jonas and Andrzej Drygajlo (2008). "Evaluation of Speech Quality Measures for the Purpose of Speaker Verification". In: *Odyssey 2008: The Speaker and Language Recognition Workshop, Stellenbosch, South Africa, January 21-24, 2008*. ISCA, p. 5. URL: http://www.isca-speech.org/archive%5C_open/odyssey%5C_2008/od08%5C_005.html (visited on 11/27/2022).

Rojas, Junior, Bilal Alsallakh, Edward Wang, Sara Zhang, and Jonathan Reynolds (2020). "Probing Embedding Spaces in Deep Neural Networks". In: *NeurIPS 2020*, p. 2.

Ross, Arun and Anil K. Jain (Sept. 2004). "Multimodal Biometrics: An Overview". In: *2004 12th European Signal Processing Conference*. 2004 12th European Signal Processing Conference, pp. 1221–1224.

Rownicka, Joanna, Peter Bell, and Steve Renals (Sept. 30, 2019). *Embeddings for DNN Speaker Adaptive Training*. DOI: 10.48550/arXiv.1909.13537. arXiv: 1909.13537 [cs, eess]. URL: http://arxiv.org/abs/1909.13537 (visited on 11/30/2022). preprint.

Ryant, Neville, Prachi Singh, Venkat Krishnamohan, Rajat Varma, Kenneth Church, Christopher Cieri, Jun Du, Sriram Ganapathy, and Mark Liberman (Apr. 5, 2021). *The Third DIHARD Diarization Challenge*. DOI: 10.48550/arXiv.2012.01477. arXiv: 2012.01477 [cs, eess]. URL: http://arxiv.org/abs/2012.01477 (visited on 11/09/2022). preprint.

Sak, Haşim, Andrew Senior, and Françoise Beaufays (Feb. 5, 2014). *Long Short-Term Memory Based Recurrent Neural Network Architectures for Large Vocabulary Speech Recognition*. DOI: 10.48550/arXiv.1402.1128. arXiv: 1402.1128 [cs, stat]. URL: http://arxiv.org/abs/1402.1128 (visited on 11/19/2022). preprint.

Salman, Ahmad (Aug. 21, 2012). "Learning Speaker-Specific Characteristics With Deep Neural Architecture". The University of Manchester. URL: https://www.escholar.manchester.ac.uk/uk-ac-man-scw:167021 (visited on 11/24/2022).

Saon, George, Hagen Soltau, David Nahamoo, and Michael Picheny (Dec. 2013). "Speaker Adaptation of Neural Network Acoustic Models Using I-Vectors". In: *2013 IEEE Workshop on Automatic Speech Recognition and Understanding*. 2013 IEEE Workshop on Automatic Speech Recognition and Understanding, pp. 55–59. DOI: 10.1109/ASRU.2013.6707705.

Sejnowski, Terrence J. (2018). *The Deep Learning Revolution*. Cambridge, MA: MIT Press. ISBN: 978-0-262-03803-4.

Sell, Gregory and Daniel Garcia-Romero (Dec. 2014). "Speaker Diarization with Plda I-Vector Scoring and Unsupervised Calibration". In: *2014 IEEE Spoken Language Technology Workshop (SLT)*. 2014 IEEE Spoken Language Technology Workshop (SLT), pp. 413–417. DOI: 10.1109/SLT.2014.7078610.

Sell, Gregory, David Snyder, Alan McCree, Daniel Garcia-Romero, Jesús Villalba, Matthew Maciejewski, Vimal Manohar, Najim Dehak, Daniel Povey, Shinji Watanabe, and Sanjeev Khudanpur (Sept. 2, 2018). "Diarization Is Hard: Some Experiences and Lessons Learned for the JHU Team in the Inaugural DIHARD Challenge". In: *Interspeech 2018*. Interspeech 2018. ISCA, pp. 2808–2812. DOI: 10.21437/Interspeech.2018-1893. URL: https://www.isca-speech.org/archive/interspeech_2018/sell18_interspeech.html (visited on 11/16/2022).

Shabeer, H. Abdul and P. Suganthi (Dec. 2007). "Mobile Phones Security Using Biometrics". In: *International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007)*. International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007). Vol. 4, pp. 270–274. DOI: 10.1109/ICCIMA.2007.182.

Shafey, Laurent El, Hagen Soltau, and Izhak Shafran (July 8, 2019). *Joint Speech Recognition and Speaker Diarization via Sequence Transduction*. DOI: 10.48550/arXiv.1907.05337. arXiv: 1907.05337 [cs, eess]. URL: http://arxiv.org/abs/1907.05337 (visited on 11/19/2022). preprint.

Shen, Jian, Yanru Qu, Weinan Zhang, and Yong Yu (Apr. 29, 2018). "Wasserstein Distance Guided Representation Learning for Domain Adaptation". In: *Proceedings of the AAAI Conference on Artificial Intelligence* 32.1 (1). ISSN: 2374-3468. DOI: 10.1609/aaai.v32i1.11784. URL: https://ojs.aaai.org/index.php/AAAI/article/view/11784 (visited on 11/19/2022).

Shinohara, Yusuke (Sept. 8, 2016). "Adversarial Multi-Task Learning of Deep Neural Networks for Robust Speech Recognition". In: *Interspeech 2016*. Interspeech 2016. ISCA, pp. 2369–2372. DOI: 10.21437/Interspeech.2016-879. URL: https://www.isca-speech.org/archive/interspeech_2016/shinohara16b_interspeech.html (visited on 11/19/2022).

Shum, Stephen, Najim Dehak, Reda Dehak, and James R Glass (n.d.). "Unsupervised Speaker Adaptation Based on the Cosine Similarity for Text-Independent Speaker Verification". In: (), p. 7.

Sinclair, Mark (2016). “Speech Segmentation and Speaker Diarisation for Transcription and Translation”. University of Edinburgh. URL: https://www.research.ed.ac.uk/en/publications/speech-segmentation-and-speaker-diarisation-for-transcription-and (visited on 11/16/2022).

Snyder, David, Guoguo Chen, and Daniel Povey (Oct. 28, 2015). *MUSAN: A Music, Speech, and Noise Corpus*. DOI: 10.48550/arXiv.1510.08484. arXiv: 1510.08484 [cs]. URL: http://arxiv.org/abs/1510.08484 (visited on 07/31/2023). preprint.

Snyder, David, Daniel Garcia-Romero, Daniel Povey, and Sanjeev Khudanpur (Aug. 20, 2017). “Deep Neural Network Embeddings for Text-Independent Speaker Verification”. In: *Interspeech 2017*. Interspeech 2017. ISCA, pp. 999–1003. DOI: 10.21437/Interspeech.2017-620. URL: https://www.isca-speech.org/archive/interspeech_2017/snyder17_interspeech.html (visited on 11/19/2022).

Snyder, David, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur (Apr. 2018). “X-Vectors: Robust DNN Embeddings for Speaker Recognition”. In: *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. ICASSP 2018 - 2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). Calgary, AB: IEEE, pp. 5329–5333. ISBN: 978-1-5386-4658-8. DOI: 10.1109/ICASSP.2018.8461375. URL: https://ieeexplore.ieee.org/document/8461375/ (visited on 11/17/2021).

Song, Huan, Megan Willi, Jayaraman J. Thiagarajan, Visar Berisha, and Andreas Spanias (Aug. 4, 2018). *Triplet Network with Attention for Speaker Diarization*. DOI: 10.48550/arXiv.1808.01535. arXiv: 1808.01535 [cs, eess, stat]. URL: http://arxiv.org/abs/1808.01535 (visited on 11/19/2022). preprint.

Srivastava, Nitish, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov (2014). “Dropout: A Simple Way to Prevent Neural Networks from Overfitting”. In: *Journal of Machine Learning Research* 15.56, pp. 1929–1958. ISSN: 1533-7928. URL: http://jmlr.org/papers/v15/srivastava14a.html (visited on 11/19/2022).

Swietojanski, Pawel and Steve Renals (Dec. 2014). “Learning Hidden Unit Contributions for Unsupervised Speaker Adaptation of Neural Network Acoustic Models”. In: *2014 IEEE Spoken Language Technology Workshop (SLT)*. 2014 IEEE Spoken Language Technology Workshop (SLT), pp. 171–176. DOI: 10.1109/SLT.2014.7078569.

Tomashenko, Natalia, Brij Mohan Lal Srivastava, Xin Wang, Emmanuel Vincent, Andreas Nautsch, Junichi Yamagishi, Nicholas Evans, Jose Patino, Jean-François Bonastre, Paul-Gauthier Noé, and Massimiliano Todisco (May 14, 2022). *The VoicePrivacy 2020 Challenge Evaluation Plan*. DOI: 10.48550/arXiv.2205.07123. arXiv: 2205.07123 [cs, eess]. URL: http://arxiv.org/abs/2205.07123 (visited on 08/29/2022). preprint.

Torres-Carrasquillo, Pedro A., Fred Richardson, Shahan Nercessian, Douglas Sturim, William Campbell, Youngjune Gwon, Swaroop Vattam, Najim Dehak, Harish Mallidi, Phani Sankar Nidadavolu, Ruizhi Li, and Reda Dehak (Aug. 20, 2017). "The MIT-LL, JHU and LRDE NIST 2016 Speaker Recognition Evaluation System". In: *Interspeech 2017*. Interspeech 2017. ISCA, pp. 1333–1337. DOI: 10.21437/Interspeech.2017-537. URL: https://www.isca-speech.org/archive/interspeech_2017/torrescarrasquillo17_interspeech.html (visited on 11/23/2022).

Transcripts and Recordings of Oral Arguments - Supreme Court of the United States (2022). URL: https://www.supremecourt.gov/oral_arguments/availabilityoforalargumenttranscripts.aspx (visited on 03/28/2022).

Tu, Youzhi, Man-Wai Mak, and Jen-Tzung Chien (Sept. 15, 2019). "Variational Domain Adversarial Learning for Speaker Verification". In: *Interspeech 2019*. Interspeech 2019. ISCA, pp. 4315–4319. DOI: 10.21437/Interspeech.2019-2168. URL: https://www.isca-speech.org/archive/interspeech_2019/tu19_interspeech.html (visited on 11/16/2022).

Van der Maaten, Laurens and Geoffrey Hinton (2008). "Visualizing Data Using T-SNE". In: *Journal of Machine Learning Research* 9.86, pp. 2579–2605. URL: http://jmlr.org/papers/v9/vandermaaten08a.html.

Van Steenkiste, Sjoerd, Francesco Locatello, Jürgen Schmidhuber, and Olivier Bachem (Jan. 7, 2020). *Are Disentangled Representations Helpful for Abstract Visual Reasoning?* DOI: 10.48550/arXiv.1905.12506. arXiv: 1905.12506 [cs, stat]. URL: http://arxiv.org/abs/1905.12506 (visited on 11/30/2022). preprint.

Variani, Ehsan, Xin Lei, Erik McDermott, Ignacio Lopez Moreno, and Javier Gonzalez-Dominguez (May 2014). "Deep Neural Networks for Small Footprint Text-Dependent Speaker Verification". In: *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 4052–4056. DOI: 10.1109/ICASSP.2014.6854363.

Veysov, Alexander (Nov. 16, 2022). *Silero VAD*. URL: https://github.com/snakers4/silero-vad (visited on 11/16/2022).

Villalba, Jesús, Nanxin Chen, David Snyder, Daniel Garcia-Romero, Alan McCree, Gregory Sell, Jonas Borgstrom, Leibny Paola García-Perera, Fred Richardson, Réda Dehak, Pedro A. Torres-Carrasquillo, and Najim Dehak (Mar. 1, 2020). "State-of-the-Art Speaker Recognition with Neural Network Embeddings in NIST SRE18 and Speakers in the Wild Evaluations". In: *Computer Speech & Language* 60, p. 101026. ISSN: 0885-2308. DOI: 10.1016/j.csl.2019.101026. URL: https://www.sciencedirect.com/science/article/pii/S0885230819302700 (visited on 11/19/2022).

Viñals, Ignacio, Dayana Ribas, Victoria Mingote, Jorge Llombart, Pablo Gimeno, Antonio Miguel, Alfonso Ortega, and Eduardo Lleida (Sept. 15, 2019). "Phonetically-Aware Embeddings, Wide Residual Networks with Time-Delay Neural Networks and Self Attention Models for the 2018 NIST Speaker Recognition Evaluation". In: *Interspeech 2019*. Interspeech 2019. ISCA, pp. 4310–4314. DOI: 10.21437/Interspeech.2019-2417. URL: https://www.isca-speech.org/archive/interspeech_2019/vinals19b_interspeech.html (visited on 11/19/2022).

Wan, Li, Quan Wang, Alan Papir, and Ignacio Lopez Moreno (Nov. 9, 2020). "Generalized End-to-End Loss for Speaker Verification". arXiv: 1710.10467 [cs, eess, stat]. URL: http://arxiv.org/abs/1710.10467 (visited on 08/09/2021).

Wang, Feng, Weiyang Liu, Haijun Liu, and Jian Cheng (July 2018). "Additive Margin Softmax for Face Verification". In: *IEEE Signal Processing Letters* 25.7, pp. 926–930. ISSN: 1070-9908, 1558-2361. DOI: 10.1109/LSP.2018.2822810. arXiv: 1801.05599 [cs]. URL: http://arxiv.org/abs/1801.05599 (visited on 11/19/2022).

Wang, Hao, Yitong Wang, Zheng Zhou, Xing Ji, Dihong Gong, Jingchao Zhou, Zhifeng Li, and Wei Liu (June 2018). "CosFace: Large Margin Cosine Loss for Deep Face Recognition". In: *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Salt Lake City, UT: IEEE, pp. 5265–5274. ISBN: 978-1-5386-6420-9. DOI: 10.1109/CVPR.2018.00552. URL: https://ieeexplore.ieee.org/document/8578650/ (visited on 11/16/2022).

Wang, Qiongqiong, Kong Aik Lee, and Tianchi Liu (Apr. 10, 2022). *Scoring of Large-Margin Embeddings for Speaker Verification: Cosine or PLDA?* DOI: 10.48550/

arXiv.2204.03965. arXiv: 2204.03965 [cs, eess]. URL: http://arxiv.org/abs/2204.03965 (visited on 10/05/2022). preprint.

Wang, Quan, Carlton Downey, Li Wan, Philip Andrew Mansfield, and Ignacio Lopez Moreno (Dec. 14, 2018). "Speaker Diarization with LSTM". arXiv: 1710 . 10468 [cs, eess, stat]. URL: http : / / arxiv . org / abs / 1710 . 10468 (visited on 08/09/2021).

Wang, Yuxuan, Maokui He, Shutong Niu, Lei Sun, Tian Gao, Xin Fang, Jia Pan, Jun Du, and Chin-Hui Lee (Mar. 19, 2021). *USTC-NELSLIP System Description for DIHARD-III Challenge*. arXiv: 2103.10661 [cs, eess]. URL: http://arxiv.org/abs/2103.10661 (visited on 11/09/2022). preprint.

Wang, Yuxuan, R. J. Skerry-Ryan, Daisy Stanton, Yonghui Wu, Ron J. Weiss, Navdeep Jaitly, Zongheng Yang, Ying Xiao, Zhifeng Chen, Samy Bengio, Quoc Le, Yannis Agiomyrgiannakis, Rob Clark, and Rif A. Saurous (Apr. 6, 2017). *Tacotron: Towards End-to-End Speech Synthesis*. DOI: 10.48550/arXiv.1703.10135. arXiv: 1703.10135 [cs]. URL: http://arxiv.org/abs/1703.10135 (visited on 11/19/2022). preprint.

Warde-Farley, David, Ian J. Goodfellow, Aaron Courville, and Yoshua Bengio (Jan. 2, 2014). *An Empirical Analysis of Dropout in Piecewise Linear Networks*. DOI: 10 . 48550/arXiv.1312.6197. arXiv: 1312.6197 [cs, stat]. URL: http://arxiv.org/abs/1312.6197 (visited on 11/19/2022). preprint.

Williams, Jennifer and Simon King (Sept. 15, 2019). "Disentangling Style Factors from Speaker Representations". In: *Interspeech 2019*. Interspeech 2019. ISCA, pp. 3945–3949. DOI: 10.21437/Interspeech.2019-1769. URL: https://www.isca-speech.org/archive/interspeech_2019/williams19c_interspeech.html (visited on 08/29/2021).

Williams, Jennifer, Joanna Rownicka, Pilar Oplustil, and Simon King (Apr. 27, 2020). *Comparison of Speech Representations for Automatic Quality Estimation in Multi-Speaker Text-to-Speech Synthesis*. DOI: 10 . 48550 / arXiv . 2002 . 12645. arXiv: 2002.12645 [cs, eess]. URL: http://arxiv.org/abs/2002.12645 (visited on 11/30/2022). preprint.

Williams, Jennifer, Junichi Yamagishi, Paul-Gauthier Noe, Cassia Valentini Botinhao, and Jean-Francois Bonastre (Oct. 13, 2021). "Revisiting Speech Content Privacy". arXiv: 2110.06760 [eess]. URL: http://arxiv.org/abs/2110.06760 (visited on 02/01/2022).

Xie, Weidi, Arsha Nagrani, Joon Son Chung, and Andrew Zisserman (May 17, 2019). *Utterance-Level Aggregation For Speaker Recognition In The Wild*. DOI: 10.48550/arXiv.1902.10107. arXiv: 1902.10107 [cs, eess]. URL: http://arxiv.org/abs/1902.10107 (visited on 11/01/2022). preprint.

You, Lanhua, Wu Guo, Li-Rong Dai, and Jun Du (Sept. 15, 2019). "Multi-Task Learning with High-Order Statistics for x-Vector Based Text-Independent Speaker Verification". In: *Interspeech 2019*. Interspeech 2019. ISCA, pp. 1158–1162. DOI: 10.21437/Interspeech.2019-2264. URL: https://www.isca-speech.org/archive/interspeech_2019/you19_interspeech.html (visited on 11/19/2022).

Yu, Wei, Kuiyuan Yang, Yalong Bai, Hongxun Yao, and Yong Rui (Dec. 26, 2014). *Visualizing and Comparing Convolutional Neural Networks*. DOI: 10.48550/arXiv.1412.6631. arXiv: 1412.6631 [cs]. URL: http://arxiv.org/abs/1412.6631 (visited on 11/20/2022). preprint.

Zhang, Aonan, Quan Wang, Zhenyao Zhu, John Paisley, and Chong Wang (May 2019). "Fully Supervised Speaker Diarization". In: *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 6301–6305. DOI: 10.1109/ICASSP.2019.8683892.

Zhang, Chiyuan, Samy Bengio, Moritz Hardt, Benjamin Recht, and Oriol Vinyals (Feb. 26, 2017). *Understanding Deep Learning Requires Rethinking Generalization*. DOI: 10.48550/arXiv.1611.03530. arXiv: 1611.03530 [cs]. URL: http://arxiv.org/abs/1611.03530 (visited on 11/16/2022). preprint.

Zhang, Lei, Jian Yang, and David Zhang (Dec. 1, 2017). "Domain Class Consistency Based Transfer Learning for Image Classification across Domains". In: *Information Sciences* 418–419, pp. 242–257. ISSN: 0020-0255. DOI: 10.1016/j.ins.2017.08.034. URL: https://www.sciencedirect.com/science/article/pii/S0020025516313159 (visited on 11/20/2022).

Zhang, Xiao, Rui Zhao, Yu Qiao, Xiaogang Wang, and Hongsheng Li (June 2019). "AdaCos: Adaptively Scaling Cosine Logits for Effectively Learning Deep Face Representations". In: *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*. 2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR). Long Beach, CA, USA: IEEE, pp. 10815–10824. ISBN: 978-1-72813-293-8. DOI: 10.1109/CVPR.2019.01108. URL: https://ieeexplore.ieee.org/document/8953896/ (visited on 11/19/2022).

Zhu, Yingke, Tom Ko, David Snyder, Brian Mak, and Daniel Povey (Sept. 2, 2018). "Self-Attentive Speaker Embeddings for Text-Independent Speaker Verification". In: *Interspeech 2018*. Interspeech 2018. ISCA, pp. 3573–3577. DOI: 10.21437/Interspeech.2018-1158. URL: https://www.isca-speech.org/archive/interspeech_2018/zhu18_interspeech.html (visited on 11/19/2022).