TÉCNICO LISBOA

Automatic Detection of Profile Features

John Daniel Fidalgo Mendonça

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisors: Prof. Isabel Maria Martins Trancoso
Rui Pedro dos Santos Correia, Ph.D

Examination Committee

Chairperson: Prof. João Fernando Cardoso Silva Sequeira
Supervisor: Prof. Isabel Maria Martins Trancoso
Member of the Committee: Prof. António Joaquim da Silva Teixeira

December 2020

Declaration

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of Universidade de Lisboa.

Acknowledgments

The Thesis is a solitary moment for a student. Even more, when during a global pandemic. Writing this Thesis was harder than I thought and more rewarding than I could have ever imagined. It is, for a reason, the culmination of 5 years of study. None of this would have been possible without my parents, aunts and cousins for their support and encouragement over the years and without whom this project would not be possible. They were there for me during my best and worst moments, and for that I'm truly grateful.

I would also like to express my deepest gratitude to my supervisor, Professor Isabel Trancoso, for her guidance, assistance, keen insight, and ongoing support in bringing my research to life. I would also like to express my deepest thanks to my co-supervisor, Rui Correia, for all the advice, constant support and availability, without which this work would not have been possible.

I would like to extend a special thanks to João Freitas and everyone I met during my short stay in DefinedCrowd. While circumstances didn't allow me to meet everyone in person, I would like to thank them for sharing their time by providing important insights on Crowdsourcing.

I would also like to acknowledge my co-authors Professor Alberto Abad, Francisco Teixeira and all of my peers from INESC-ID HLT Lab for their insight and support which led to the participation on the INTERSPEECH 2020 ComParE Challenge and resulted in the development of Chapter 4 of this Thesis.

Last but not least, to all my friends and colleagues that helped me grow as a person and were always there for me.

To each and every one of you – Thank you.

Abstract

Speech corpora collected via crowdsourcing typically require costly validation to verify certain characteristics of speakers, or submission correctness. Moreover, this validation should also exclude recordings corresponding to multiple speakers sharing the same account or multiple accounts for the same speaker. This thesis focus on the use of speech pattern recognition techniques to perform this automatic validation. This is accomplished by training an x-vector based system in a large open-source corpus, and enrolling the first utterance from each speaker in a crowdsourcing corpora collection job which is then compared to subsequent task completions. The resulting speaker embeddings are also used to identify gender. As a proof-of-concept, we used this approach to validate different datasets in 3 languages, adopting score normalisation techniques. Results show an EER below the 4% mark on all experiments, indicating the possibility to adopt the same threshold without substantial loss of performance. This enables the validation of crowdsourced task completions immediately after submission.

This thesis also involved the participation in an international Computational Paralinguistics Challenge, where we studied the automatic prediction from conversational speech of breath signals obtained from respiratory belts. We analysed both original and predicted signals and identified the subsets of most irregular belt signals which yield the worst performance, and showed how they affect results. We proposed several variants of an end-to-end baseline system, such as BiLSTM, and AM/FM decomposition as input. We showed that these models can predict breathing patterns and clinically relevant parameters, such as breathing rate, in simulated video-conferencing sessions.

Keywords

Crowdsourcing; Paralinguistics; Speaker Verification; Gender Recognition; Breath Detection.

Resumo

Corpora de fala coletado usando crowdsourcing necessitam tipicamente de validações dispendiosas para verificar as características dos falantes, ou correta submissão. Adicionalmente, esta validação também deverá excluir gravações correspondentes a vários falantes que partilham a mesma conta, ou várias contas com o mesmo falante. Esta tese foca-se no uso de técnicas de reconhecimento de padrões de fala para realizar esta validação automática. Isto é efetuado treinando um sistema baseado no x-vector num corpus open-source e registando a primeira gravação de cada falante num trabalho de coleção de corpora, que é depois comparado com gravações subsequentes. Os embeddings resultantes são também utilizados para identificar género. Como teste, usou-se esta abordagem para validar diferentes datasets em 3 línguas, adotando técnicas de normalização de score. Os resultados mostram um EER abaixo dos 4% em todas as experiências, indicando a possibilidade de adotar o mesmo limiar sem perda substancial de performance. Isto permite a validação de tarefas de crowdsourcing imediatamente após submissão.

Esta tese também envolveu a participação num desafio internacional de computação paralinguística, onde foi estudado a predição automática através da fala de sinais de respiração obtidos através de cintos respiratórios. Analisou-se os sinais originais e preditos e identificou-se um subset de sinais irregulares que resultaram na pior performance, mostrando como estes afetam os resultados. Propôs-se várias variantes do sistema base end-to-end, como o BiLSTM e a decomposição AM/FM como input, mostrando que estes são capazes de predizer padrões respiratórios e parâmetros clinicamente relevantes, como a taxa de respiração, em sessões simuladas de videoconferência.

Palavras Chave

Crowdsourcing; Paralinguística; Verificação do Falante; Verificação de Género; Deteção da respiração.

Contents

List of Figures

List of Tables

Acronyms

AI	Artificial Intelligence
ASR	Automatic Speech Recognition
AM	Amplitude Modulation
CMVN	Cepstral Mean and Variance Normalization
CNN	Convolutional Neural Network
DET	Detection Error Tradeoff
DT	Decision Threshold
DNN	Deep Neural Network
EER	Equal Error Rate
FM	Frequency Modulation
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
HIT	Human Intelligence Task
JFA	Join Factor Analysis
LDA	Linear Discriminant Analysis
LSTM	Long Short Term Memory
MAP	*maximum a posteriori*
MFCCs	Mel-Frequency cepstral coefficients
MLP	Multi-layer Perceptron

NN	Neural Network
PLDA	Probabilistic Linear Discriminant Analysis
RNN	Recurrent Neural Network
SVM	Support Vector Machine
TDNN	Time Delay Neural Network
UBM	Universal Background Model
VAD	Voice Activity Detection
VoIP	Voice over IP
WCCN	Within-Class Covariance Normalization

1

Introduction

Contents

1.1 Motivation

Speech technology has significantly influenced the lives of everyday users, impacting the way people find, consume, and act on information. Starting with the widespread adoption of mobile devices such as smartphones, more recent technological advancements have led to a larger use of voice search and Intelligent Virtual Assistants. Recent studies indicate that over 50% of web searches will be conducted through voice by 2020, and that 55% of U.S. households will possess an intelligent virtual assistant [4]. Fuelling this sharp increase is the growing consumer demand for online self-service, self-reliance, and rapid query resolution, while at the same time helping companies enhance operational efficiency and reduce costs [5]. With speech technologies trending towards a more predominant use, the need for efficient and effective interactions with users has become increasingly important. The large amount of data collected resulting from the interaction with speech-based systems allows Artificial Intelligence (AI) to adapt and improve over time. AI enables the continuous improvement of speech systems by including collected speech data in the training of Machine Learning models that tackle common speech applications such as Automatic Speech Recognition (ASR) or Speech Synthesis. However, more and more use cases and industry applications that use speech to obtain interpretable speaker information have surged.

The human voice conveys substantial amounts of information related to the speaker. For instance, information including physical traits (age [6], gender [7]), language (nationality, nativeness) [8] [9], health (speech affecting diseases) [10] and mood [11] can be obtained from voice. Such profile information can be extracted directly from speech (using the raw-time waveform or the spectrum), or from speech derived features (intensity, voice quality features, speech rate, breathing rate) [12]. The automatic detection of profile features enables the development of smarter user interfaces and an enhanced user experience, especially when using devices or applications where this information is required. Additionally, it can assist in more sensitive applications such as identity verification, where speaker verification or facial reconstructions from voice [13] may be of value.

The motivation previously identified exposes the impact of voice profile feature information and metadata from speakers in the development and use of speech datasets. In this work, we will investigate the use of automatic detection of profile features and other metadata extracted from voice in two distinct fields: in crowdsourcing, and in medicine.

1.2 Objectives

As previously mentioned, this Thesis will focus on how information extraction from speech signals can assist in solving challenges in two different industry fields.

The first of those use cases is the collection of speech through the paradigm of crowdsourcing. In

crowdsourcing, existing quality control mechanisms in speech corpora are limited to situations where the user profile is irrelevant to the task, that is, we are only interested in the product of the work and not the user itself. In crowdsourcing in general, there has been extensive work and several metrics proposed to classify the quality of the tasks being performed by the crowd. In speech data collections in particular, validation tasks are typically set up after collection to validate certain characteristics of speakers (nativeness, gender) or submission correctness, which adds to the cost of the dataset. Furthermore, these validation techniques fail to detect users with malicious intentions, reducing the value of the datasets.

To address these issues we propose to develop (and test) a system that automatically creates a voice profile from speech, to ensure that gender and other metadata information are accurate and consistent throughout a collection. Considering the language/accent and channel rich environment of crowdsourcing in particular, it is expected that the system works in a multi-lingual scenario and is robust to noise and other channel conditions. That is, performance loss should not occur when language and channel conditions change.

A voice profiling system, such as the one described above, will allow for the detection of low quality work derived from profile variations of the same user or mismatch between the obtained user profile during registration, and the actual profile obtained from speech. This will allow for the detection of fraudulent behaviour, such as users attempting to share their account with multiple speakers. Additionally, mismatches between predictions and the submitted profile would also be detected. While many automatic systems that attempt to tackle the problem of low quality work in crowdsourcing exist, none take advantage of speaker characteristics in their decisions. In this thesis, we propose a new approach for user profile enforcement in speech data collections, combining speaker verification systems with gender identification to identify mismatches in profiles.

The second challenge tackled in this thesis is the automatic prediction of health-related parameters and features from speech, which are of substantial value in the field of medicine. Besides offering medically-relevant information for the diagnosis of diseases (e.g. jitter and shimmer), it can also provide cues regarding the intensity of physical activity (intensity, breathing rate) and stress. As such, the automatic detection of this information can bring value when developing datasets and models that leverage this information. For instance, these features can be used to enforce class distribution in health-related speech data collections. In this second use case, we will explore techniques for the automatic prediction of breathing patterns and other breathing-related extractable features from voice. It is expected that the resulting prediction model is able to accurately reproduce breathing patterns when using speech recorded under different conditions (such as video-calls) and be speaker-and-language independent.

1.3 Outline

This Thesis is organized as follows: in chapter 2, a review of the current state of the art on speech systems is conducted, with special focus on tasks that predict speaker information through speech. Additionally, fundamental concepts and terminologies related to speech pattern recognition are presented. In chapter 3, we will address the first challenge mentioned above, namely, how the analysis of the speech signal can help improve the process of collecting data through crowdsourcing. More specifically, the use of voice pattern recognition techniques is explored in the context of fraud detection in a crowdsourced speech data collection environment, such as speaker and gender verification. Furthermore, we compare these automated systems with naive human annotators. Chapter 4 focuses on the second challenge of the thesis, and provides an analysis of breathing pattern recognition from voice. Additionally, we propose a system that automatically predicts these patterns and related metrics. Chapter 5 is the final chapter, where conclusions pertaining this thesis are drawn, together with some topics for future work.

2

Background

Contents

Speech, besides being a vehicle for communication, also is also a reflection of the speaker's anatomical structure. The vocal production system of a human can be divided into two parts. The lungs and the diaphragm are only responsible for the pressure production required for speech. The upper vocal tract (which includes the nose, mouth, pharynx and larynx) is the one responsible for producing speech. The brain, while not frequently considered in vocal production, is also an important participant in this process. Most of the production of language functions lies in the area of the brain called Broca's Area.

Consequently, the current research in speech pattern recognition is threefold: Speech recognition, where the goal is to recognize and translate spoken language into text; Speaker Recognition, where the objective is to identify/validate speakers; and Computational Paralinguistics, which deals with states and traits of speakers as manifested in their speech signal's properties.

Similar to other Machine Learning applications, speech pattern recognition tasks share a common processing pipeline. The input signal, speech, is frequently subjected to external conditions such as noise and recording methods. As such, the signal is subjected to pre-processing steps before information extraction is conducted. Feature extraction techniques typically follow and range between frame-level information such as the spectrum or more general features such as handcrafted parameters. More recently, this feature extraction step has been replaced by end-to-end modelling using Deep Neural Network (DNN)s. In this case, a speech representation that better suits the task at hand is automatically learned by the network.

After feature extraction, a prediction (classification or regression) is calculated using a Machine Learning algorithm. Some pattern recognition systems require training, where the model's parameters are adjusted to minimize a cost function associated with wrong predictions on a given dataset. The model and its hyper-parameters can then be fine-tuned on a separate dataset, which is called the development stage. The model is then evaluated in the test stage, by assessing its performance on unseen data.

The focus of this thesis focuses on Speaker Recognition and Computational Paralinguistics. These belong to a sub-field of Speech Pattern Recognition, where the scope of the problem is to predict identifiable patterns from a given voice. These can be biometric information such as age, gender or even uniquely identifying a speaker, but also include other information such as breathing patterns, mood or diseases.

To support this work, it is important to look into the current state-of-the-art of speech processing, in general, and some of its sub-problems. First, in section 2.1, a small introduction to Artificial Neural Networks is provided, as the de facto building blocks for current speech system architectures. In section 2.2, the concepts and terminology used in Speaker Recognition are introduced. A brief historical context in Speaker and Gender Recognition is provided in section 2.3 [1]. Finally, a review of the state-of-the-art

[1] For readability, previous work on the topic of Automatic Prediction of Breathing Patterns is expanded in Chapter 4.

speaker embedding systems is presented in section 2.4. This section also includes an explanation of the decision-making process using embeddings, including Gender and Speaker Recognition.

2.1 Artificial Neural Networks

An artificial neural network, usually simply called Neural Network (NN), consists of a collection of artificial neurons that receive inputs and produce a single output. The resulting output can then be sent to multiple other neurons, forming a directed, weighted graph. These neurons are conceptually derived from biological neurons, with the link between them having a weight, which determines the strength of one node's influence on another. The output of a neuron, the activation, is the weighted sum of all of the inputs which is also summed to a bias term.

Neurons that share the same depth from the input form a layer. The first layer receives the input and is called the input layer, whereas the final layer outputs the prediction and is called the output layer. Layers in between these are considered to be hidden and form a Deep Neural Network (DNN). Older NN models were restricted to shallow architectures, however, with recent advancements both in terms of data availability and computational power, deep neural architectures with larger amounts of hidden layers are able to solve more complex problems and have become more common.

Several functions can be added to perform additional, known operations to the outputs of previous layers or networks. The activation function determines the output of the layer and can help the network during the training phase. These are typically non-linear, as a linear activation function would act as layer of the network by itself. Some examples of linear activations include the Rectified Linear Unit (ReLU), also known as a ramp function, and Softmax, which "squashes" a K-dimensional vector of arbitrary real values to a K-dimensional vector of real values that sum to 1. This property is useful in multi-class predictions, as they can be interpreted as probabilities.

Several types of NNs exist, and attempt to model the different behaviours and abilities that biological neurons possess.

Feed-forward In Feed-forward NNs, the information flows unidirectionally, from the input to the output. Several sub-types exists, with some examples enumerated below:

- **Multi-layer Perceptron (MLP):** consists of at least three layers of neurons: an input layer, a hidden layer and an output layer. The MLP is a universal function approximator, which means that given enough data, it is able to model any process.
- **Convolutional Neural Network (CNN):** is a regularized version of an MLP, with its hidden layers performing convolutions with a set of learnable filters in a restricted subarea of the previous layer.

This convolutional stage is paired with a dimensionality reduction stage, using pooling layers, which select an area of the input signal and perform a reduction operation (e.g. $max()$). CNNs are frequently used in Image Processing, as neurons respond to stimuli in a restricted region of space known as the receptive field, similar to the organization of the visual cortex.

- **Time Delay Neural Network (TDNN) [14]:** each layer receives as input a segment of the input layer and its delays, achieving time-shift invariance. This is especially useful in speech, as it allows the network to learn temporal relationships of acoustic and phonetic features.

Recurrent Neural Network (RNN): Unlike feed-forward networks, that only propagate information forward, RNNs also propagate information backwards, making them able to exhibit temporal dynamic behavior. This is done by having an internal state that processes variable length sequences of inputs. Additionally, bi-directional versions of RNNs exist, and perform predictions using both the past and future context of the input, by adding the outputs of two RNNs: one processing the input from left to right, the other one from right to left [15].

- **Long Short Term Memory (LSTM) [16]:** is a widely used version of RNNs, and consists of a cell and three gates: input, output and forget gate. The cell acts as the internal memory state of the network, with the gates regulating the flow of information.

Learning on NNs is conducted using mathematical optimization and involves adjusting the parameters (weights) of the networks using back-propagation algorithms. Back-propagation computes the gradient of a loss function with respect to the weights of the network. Gradient methods are then used to update the weights and minimize loss.

One of the main issues when using gradient-based learning methods and back-propagation during training is the vanishing gradient problem. During computation, it can be the case that the gradient will become vanishingly small, preventing weights from updating. Conversely, in the exploding gradient problem, large error gradients accumulate and result in large weight updates. In both cases, it hinders the training of the NNs. Typical solutions include gradient clipping and weight regularization.

Whilst the power of NNs was recognized since their initial application references in the 70s [17], it was only with the surge of computational power and the ease of access to large amounts of data of recent years that led to DNNs becoming the state-of-the-art in Machine Learning. However, memory constraints may still occur, specially on larger datasets and more complex models. In these cases, smaller batches of data are fed to the NN, called mini-batches, and gradients can be directly calculated from them. A problem that arises from calculating gradients for each mini-batch is the internal covariate shift, where there is a change in the distribution of network activations due to the change in network parameters during training [18]. This can be resolved by normalizing layer inputs using Batch Normalization, which also acts as a regularizer.

2.2 Speaker Recognition: Concepts

Speaker recognition consists in the identification of a speaker using its voice. Speaker recognition encompasses two distinct tasks:

1. **Speaker Identification**: assigns a given utterance to one of a closed set of known speakers.

2. **Speaker Verification**: verifies whether the given utterance belongs to a specific, previously enrolled, speaker or not.

Both tasks share the same initial pipeline in the learning phase. In this phase, the users are enrolled, that is, uniquely identifying information from a given user is stored in a database. In the case of speaker identification, the unknown speaker is compared against the full database of enrolled speakers, and the most similar enrolled speaker corresponds to the decision. In speaker verification, the unknown speaker is compared against a single enrolled speaker and returns a same-or-different speaker decision. Figure 2.1 illustrates the typical pipeline of a Speaker Verification system.

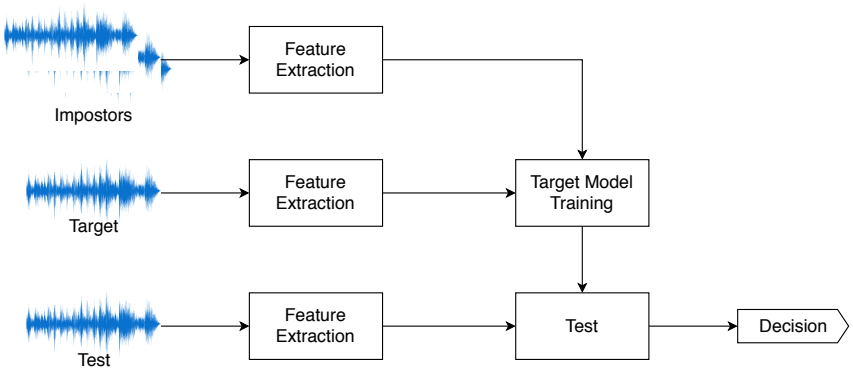

Figure 2.1: Speaker Verification system.

In the context of speaker recognition, additional considerations must be made about the architecture of the system, given the different settings and constraints that might arise from each individual task, which are enumerated below.

Open and Closed-set In closed-set speaker recognition, the speaker under test is known to belong to an enrolled speaker. This facilitates the problem, as it removes the verification dimension of the problem, as the best matching enrolled speaker is automatically chosen as prediction, regardless of how dissimilar the speakers are. In open-set speaker recognition however, the speaker under test may not belong to the pool of enrolled speakers. As such, a method similar to speaker verification must be included, where the similarity between the test speaker and the best matching enrolled speaker is checked.

Enrollment If the pool of speakers to be enrolled is known and does not change during the development and deployment of the system, then the model can be tailored to identify this pool of speakers alone. This can be done by manually adapting the system to the unique feature distribution of the enrolled speakers. However, such a model would be unfeasible when the enrolled speakers are unknown or if the need for an expanded enrollment arises. This was a limitation of preliminary speaker recognition systems that no longer exists with the introduction of automatically calculated feature sets capable of uniquely identifying speakers.

Text and Gender Independence Additional constraints and prior knowledge may be added to speaker recognition systems in order to increase its accuracy. A typical constraint is text-dependence, where the prompt read by the speaker is known. This can assist in verification by automatically rejecting speakers who provide the wrong prompt. Similarly, gender information can also be taken advantage of, as automatic gender prediction systems are very accurate for adults. Information such as gender can also be leveraged to increase speaker recognition performance, namely by modeling each gender differently.

2.3 Historical Context

The process of identifying a subject's feature from its voice is one of the basic capabilities a human being can perform using its auditory senses. Indeed, part of the human brain is exclusively dedicated to Auditory Perception, the auditory cortex. This cortex is tasked with processing the audio signal. The Wernike's area is known to perceive the output of the auditory cortex. However, the task of identifying voice features is not restricted to these areas, involving both hemispheres of the brain, each of which has different perception tasks such as rhythm or frequency detection [19]. The full process is not yet fully understood, however some simpler building blocks of this system have been investigated and important conclusions have been reached on how humans can identify different features from voice, to the point they can identify someone from it.

Most of the characteristics one can extract from a voice originates from the upper vocal tract, as indicated by the pioneering work of Gunnar Fant, in his acoustic theory of speech [20]. One of the simplest methods of identifying speaker information or even differentiating speakers is using pitch. The fundamental frequency of voice is a characteristic that contains rich information about the speaker, including age [21], gender [22], health [23] [24], mood [25], among others. An important step a human being performs while identifying speaker information is to use pitch values as part of the larger cognitive system of uniquely identifying someone, or to perform simpler tasks such as differentiating speakers or identifying someone's age or gender. The fundamental frequency of voice is able to convey information

about a speaker because it is a reflection of the speaker's own anatomical structure. Longer vocal folds produce lower pitch values, which is why male speakers often have lower pitch when compared to female speakers and children.

Consequently, the earliest speech systems took advantage of the statistical information held in the frequency domain to characterize and identify speaker information. Simple decision systems for age and gender were created using mean pitch values along utterances [26].

More complex systems can be traced back to the 1960s (US3466394A). These systems used frequency domain information to parameterize the utterance and calculate vector distance of these parameters, thus taking advantage of the research conducted in the field in the early 1960s, namely the Fant model [20] and Kersta's "Voiceprint Identification" [27]. It is important to note that early systems were all text-dependent, meaning the same phrase had to be used for enrollment and verification. Later systems were able to produce results in text-independent environments with the introduction of cepstral analysis and Linear Predictive Coding [28].

Up until the early 90s, several models for pattern recognition existed, namely the Hidden Markov Model (HMM) [29] or template matching, for speaker recognition. This changed when Gaussian Mixture Model (GMM) and Support Vector Machine (SVM)-based systems were introduced. In 1996, Schmidt applied SVM to the task of Speaker Recognition [30] (reporting 92% identification accuracy) after the successful earlier works in other fields, such as hand-writing recognition. In 1993, Reynolds successfully introduced a GMM approach for speaker verification, achieving results (96.8% identification accuracy) similar to the much more computationally complex earlier models [31].

Several age and gender recognition systems based on this new speaker recognition technique soon followed [32] [33]. Further developments in GMM, namely the introduction of Bayesian theory led to the inclusion of the Universal Background Model (UBM) [34], thus being able to introduce the concept of likelihood and its calculations for Speaker Verification/Identification tasks. This introduced the concept of *maximum a posteriori* (MAP) estimation for the adaptation of the model: MAP adaptation is used to adapt the UBMs to obtain class-specific GMMs. The GMMs are then used to calculate the likelihood of utterances to assign class labels.

Generative models such as the GMM-UBM achieved gender identification error rates of under 4% on the 2008 NIST SRE, a dataset originally intended for speaker recognition evaluation, containing 942 hours of multilingual telephone speech and English interview speech, together with speaker profile metadata.

The GMM-UBM model remained the state-of-the-art until the introduction of Factor Analysis methods in the late 00s, namely Joint Factor Analysis [35], culminating in the introduction of i-vectors [36] in 2011. The i-vector represents speaker means (depending on total variability), and contains all speaker and channel variability that can be used as a low-dimensional representation of it. This representa-

tion coupled with Probabilistic Linear Discriminant Analysis (PLDA) scoring remained the standard in Speaker Verification/Recognition until the late 2010s, with i-vectors also proving to be successful in related tasks such as gender and age recognition [37] [38]. These systems have shown to slightly improve the GMM-UBM system in distorted and multilingual datasets by an absolute gain of 25.69% in identification accuracy.

Most recently, the advent of DNNs paved the way for the introduction of models that take advantage of larger datasets. Most of these models follow a similar structure, where one part of the model is dedicated to a frame-level information extraction, followed by a prediction network that either condenses the information to produce a frame or utterance-level decision. The current de-facto state-of-the-art speaker representation is the x-vector [39] [3]. In this approach, a DNN is trained to discriminate between speakers and maps variable-length utterances to fixed-dimensional embeddings. Several end-to-end systems were proposed for pattern recognition tasks such as gender classification [40] [41]. Similar to the x-vector architecture, Convolutional Neural Networks were used to obtain speech information with temporal context. A maxpool layer was then used to discriminate information in this temporal context, which was fed to dense feed-forward layers before the last softmax layer outputs the probability per gender. No significant improvements were detected by these authors when compared to the i-vector or even GMM-UBM recognition systems when the full utterance is available.

2.4 Speaker Embeddings

2.4.1 Feature Extraction

An important step in most speech systems is feature extraction. In it, the original signal is transformed to a new representation that better exposes the information required for an accurate modelling. A typical example of a feature extraction step is to transform utterances from audio files to Mel-Frequency cepstral coefficients (MFCCs) feature vectors, which are the primary features used in speaker recognition, as they are known to represent speaker vocal tract characteristics.

MFCCs are a short-term representation of the sound spectrum, defined as the real cepstrum of the window of the signal derived from the Discrete Fourier Transform using a non-linear frequency scale. By using mel-filter banks, MFCCs mimic the perceptual scale of pitch judged by human listeners.

The MFCCs extraction of a signal, pictured in Figure 2.2, follows a short-time analysis, using sliding frames analysis along the speech signal, typically using a Hamming Window. For each frame, the signal is pre-emphasied and the Discrete Fourier Transform is calculated (using FFT, for example). The spectrum is then warped along its frequency axis into the mel-frequency scale using the filter bank, thus reflecting the human's ear perception. The logarithm of the filter bank outputs is then calculated. The MFCCs are the amplitudes of the resulting spectrum by taking the discrete cosine transform of the list

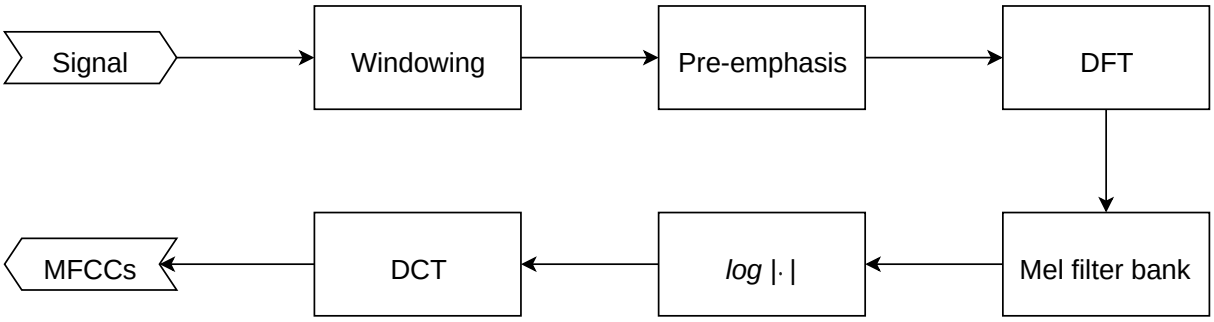

Figure 2.2: MFCCs feature block diagram

of mel log powers.

Additionally, approximations of the first and second order derivatives (also known as deltas and delta-deltas) are commonly appended to the feature vector in order to capture the dynamic temporal characteristics of speech. The first order delta coefficients are defined as follows:

$$\Delta c(t) = c(t+T) - c(t-T) \tag{2.1}$$

while the second order delta coefficients are calculated as:

$$\Delta\Delta c(t) = \Delta c(t+T) - \Delta c(t-T) \tag{2.2}$$

where T is the order of the delta computation, which is typically 2 for most applications. As such, these feature vectors, which usually lie between 13 and 40 dimensions can reach up to 120 when taking the time-differential information into account.

Cepstral Mean Subtraction (CMS) is also frequently used when utterances span on different environments by assisting in removing convolutive channel effects. However, performance degrades significantly if the training and testing is done on the same recording channel and conditions [42].

2.4.2 The Speaker Embedding

Current State-of-the-art models in pattern recognition, and speaker recognition in particular, base themselves on single fixed-dimension vectors, known as embeddings. Embeddings aim at reducing the dimensionality of the information. Earlier systems utilized Join Factor Analysis (JFA) to disambiguate between speaker and channel information modeled from a Gaussian Mixture Model (GMM). The resulting speaker-dependent vector was called the i-vector, which is explained in Section 2.5.2.A . The increased accessibility in training in deep learning allowed for the replacement of components of the i-vector pipeline (typically the Universal Background Model (UBM)) or as a stand-alone model. More recently, a embedding extraction DNN was proposed, called x-vectors, which is detailed in Section 2.5.2.B.

2.4.2.A i-vector

The i-vector system is a direct result of previous work in the field of Factor Analysis applied to the GMM model for speaker recognition. In the original model [34], a speaker's mean supervector (typically 1024/2048-dimensional) is obtained by adapting the UBM mean supervector to the speaker's frames using MAP adaptation. This is then used for classification using SVM.

Considering the size of the mean supervector, a JFA approach is applied to the GMM-UBM model [43]. The idea behind this approach is to decompose the mean supervector $\vec{M}$ into a speaker-dependent supervector $\vec{s}$ and a channel-dependent supervector $\vec{c}$:

$$\vec{M} = \vec{s} + \vec{c} \tag{2.3}$$

The supervector $\vec{s}$ can be considered normally distributed with zero mean and covariance $\vec{U}\vec{U}^T$, therefore assuming no speaker information is retained in $\vec{c}$.

$$\vec{c} = \vec{U}\vec{x} \tag{2.4}$$

This technique is referred to as eigenchannel adaptation, with the components of $\vec{x}$ called the channel factors. The speaker supervector can be further decomposed [44] into a speaker and channel independent supervector $\vec{m}$, summed with independent vectors $\vec{z}$ (common factor) and $\vec{y}$ (speaker factor):

$$\vec{s} = \vec{m} + \vec{D}\vec{z} + \vec{V}\vec{y} \tag{2.5}$$

where $\vec{D}$ is a diagonal matrix and $\vec{V}$ is a rectangular matrix of low rank.

The authors in [45] showed, however, that loss of speaker information occurs when channel factors are estimated. As a result, the i-vector system was proposed [36], using factor analysis as a feature extractor. This is done by defining a single space, called "total variability space", which includes information from both the channel and speaker. The GMM supervector is then rewritten as:

$$\vec{M} = \vec{m} + \vec{T}\vec{w} \tag{2.6}$$

where $\vec{m}$ is the speaker-and-channel-independent supervector (which plays the role of the UBM supervector), $\vec{T}$ is a rectangular matrix of low rank and $\vec{w}$ is a random vector having a standard normal distribution.

The components of the vector $\vec{w}$ are the total factors henceforth identified as identity vectors, or i-vectors. As before, $\vec{M}$ is assumed to be normally distributed with mean vector $\vec{m}$ and covariance matrix $\vec{T}\vec{T}^t$. The process of training the total variability matrix is the same as learning the eigenvoice matrix [46], except that in the former, all utterances are assumed to be produced by different speakers.

The complete extraction process of the i-vector is shown in Figure 2.3.

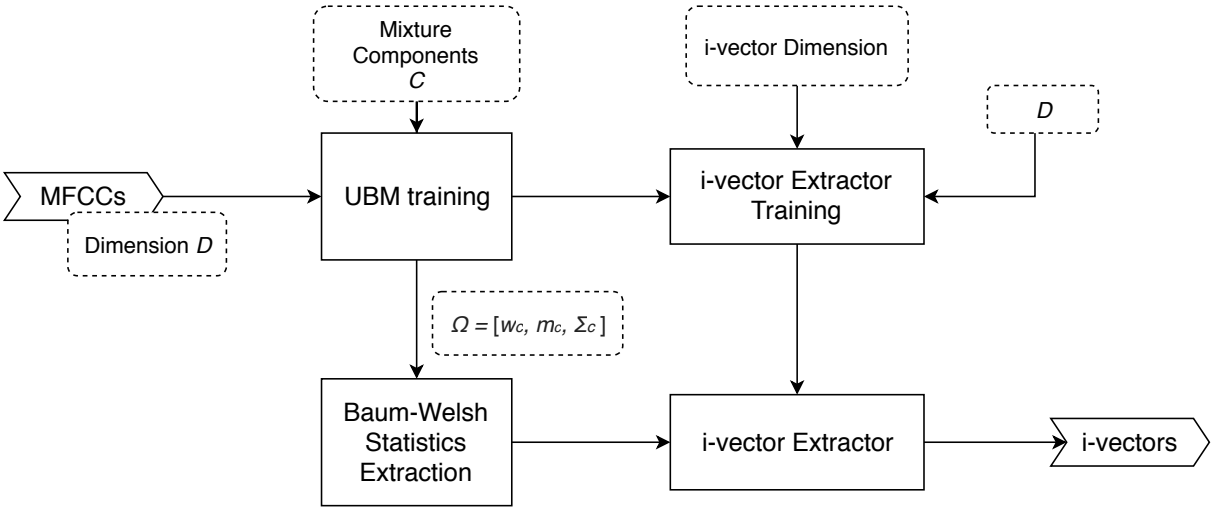

Figure 2.3: i-vector extraction process.

The total factor is a hidden variable, which can be defined by its posterior distribution conditioned to the Baum–Welch statistics for a given utterance. This posterior distribution is a Gaussian distribution [46] and it corresponds to $\vec{w}$. The Baum–Welch statistics needed to estimate the i-vector for a given speech utterance $\vec{u}$ are extracted using the UBM as follows:

$$N_c = \sum_{t=1}^{L} P(c|y_t, \Omega) \tag{2.7}$$

$$F_c = \sum_{t=1}^{L} P(c|y_t, \Omega) y_t \tag{2.8}$$

$$\tilde{F}_c = \sum_{t=1}^{L} P(c|y_t, \Omega)(y_t - m_c) \tag{2.9}$$

calculated along L frames $\{y_1, y_2, .., y_L\}$, where $\Omega = [w_c, m_c, \Sigma_c]$ is the UBM of C mixture components with gaussian index $c = 1, 2, ..., C$, weights w_c, mean m_c and covariance Σ_c , defined in some feature space of dimension F. $P(c|y_t, \Omega)$ corresponds to the posterior probability of mixture component generating the vector y_t. The i-vector can then be calculated using the following equation:

$$\vec{w} = (\vec{I} + \vec{T}^t \vec{\Sigma}^{-1} \vec{N}(u) \vec{T})^{-1} \cdot \vec{T}^t \vec{\Sigma}^{-1} \vec{\tilde{F}}(u) \tag{2.10}$$

with $\vec{N}(u)$ defined as a diagonal matrix of dimensions $CF \times CF$ with diagonal blocks $N_c I$, and $\vec{\tilde{F}}(u)$ a supervector with dimensions $CF \times 1$ constructed by concatenating all first-order Baum–Welch statistics $\tilde{F}_c$ for a given utterance u. The residual variability not captured by the total variability matrix $\vec{T}$ is modeled

in $\vec{\Sigma}$, which is a diagonal covariance matrix of dimension $CF \times CF$ estimated during factor analysis training.

Several improvements have been proposed to the baseline i-vector system. With DNN increasing performance in ASR tasks, most contributions propose a hybrid approach to the i-vector extraction pipeline by replacing the GMM-UBM by a DNN [47] [48]. In this framework, the sufficient statistics are computed by projecting the front-end features onto senones (i.e., tied-states with context dependent phones from an ASR decision tree) using the posterior probabilities estimated from a time-delay acoustic model with p-norm non-linearities. In [49], the authors proposed deriving a UBM from the Deep Bottleneck network to calculate the posterior probabilities for back-end modeling in language identification tasks (LID). Appending Bottleneck features with the spectral-based features (MFCCs) has been found to outperform the baseline MFCCs feature set in speaker identification (SID) tasks [50].

Nevertheless, the i-vector system performance deteriorates in short utterances and mismatched utterance duration between training and testing in general [51]. This is important in speaker verification systems, when it is typical to have a large utterance in the enrollment but shorter utterances in the verification phase.

2.4.2.B x-vector

The x-vector system [3] [39] is an improvement of the i-vector system, where the i-vector embedding is replaced by embeddings extracted from a feedforward DNN. Unlike the i-vector system, which uses factor analysis, the x-vector extraction is a supervised method, requiring speaker labels in its training data.

The block diagram of the x-vector approach is shown in Figure 2.4, with its configuration detailed in Table 2.1. The network is a feed-forward DNN that computes speaker embeddings from variable-length acoustic segments based on a previous end-to-end architecture [52]. This architecture can be divided into two different levels:

- The **frame level** includes the first 5 layers of the network, using a time delay architecture (TDNN) [14] that functions on speech frames, offering temporal context centered at the current frame t to the next frame (1D convolution). The last 2 layers also operate on the frame level but do not offer any added temporal context to the next frames. This architecture provides a learning method by which the initial transforms are learnt on narrow contexts and the deeper layers process the hidden activation from a wider temporal context, thus learning wider temporal relationships. During back-propagation, the lower layers of the network are updated by a gradient accumulated over all the time steps of the input temporal context, learning translation invariant feature transforms. These layers can vary between 512 and 1536, depending on the context used.

- The **segment level** is connected to the frame level using a statistics pooling layer. This pooling layer calculates the mean and standard deviation from the aggregate output from the final frame level, compiling information from the entire segment to subsequent layers. These are typically 1500 dimensional vectors, computed once for each input segment. This information is then feed-forward propagated to the next layers, ending with the softmax output layer. The output vector has as dimension K, with K being the number of speakers in the training data. The nonlinearities are all rectified linear units (ReLUs).

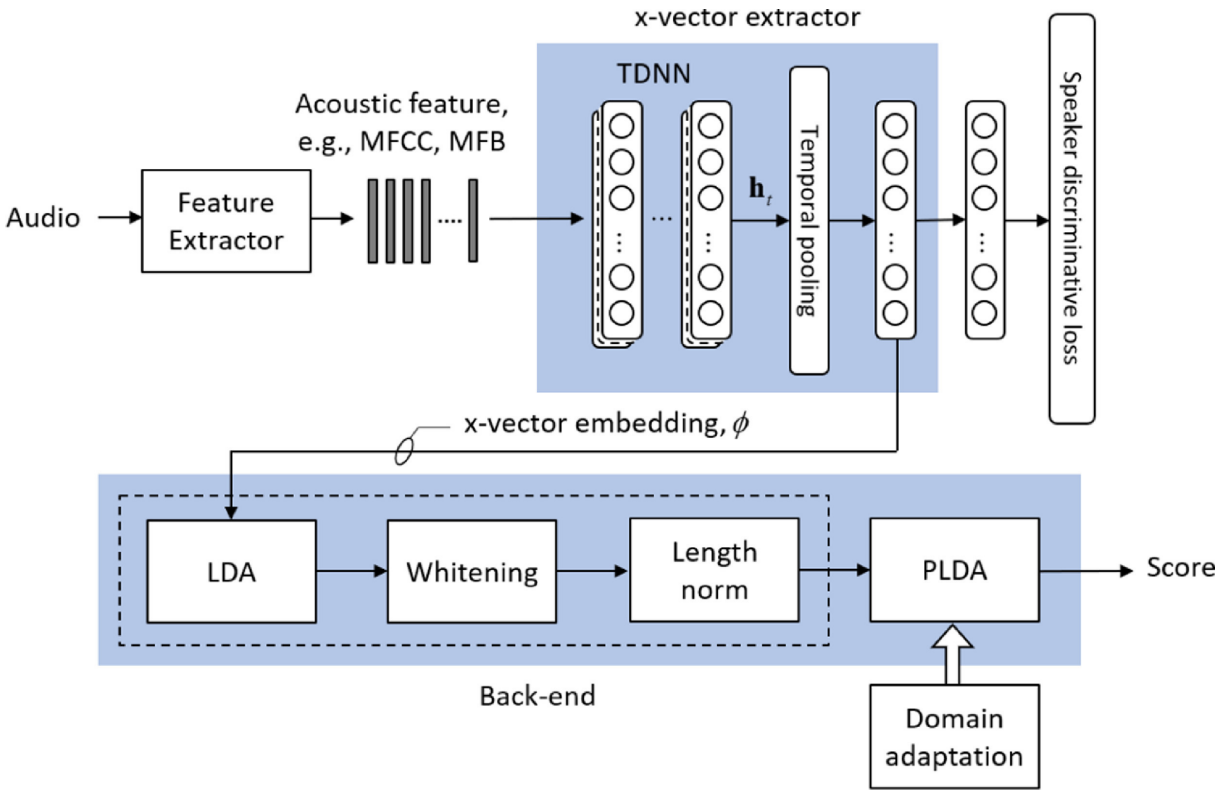

Figure 2.4: DNN architecture for x-vector extraction. Adatped from [1].

The embedding is extracted from the first segment layer. However, the embedding could also be extracted from the next segment layer: the embedding is required to contain speaker information from the entire utterance, therefore all layers after the statistics pooling layer are a valid location to extract the embedding from. The softmax output layer is not considered due to its large size and dependence on the number of speakers.

Given that the architecture is guided towards Speaker Recognition, the training of the DNN is conducted using multi-class cross entropy objective as follows:

$$E = -\sum_{n=1}^{N}\sum_{k=1}^{K} d_{nk} ln(P(spkr_k|\vec{x}_{1:T}^{(n)})) \tag{2.11}$$

where K is the number of speakers present in N training segments and d_{nk} a binary value that indicates if utterance n is from speaker k. The DNN is trained for several epochs typically using a mini-batch size of 32-64 and natural gradient stochastic gradient descent [53]. This choice of mini-batch size takes into consideration GPU memory limitations. This forces a trade-off between mini-batch size and maximum training example length. In natural-gradient SGD, the gradients are scaled by a symmetric positive definite matrix that is an approximation to the inverse of the Fisher matrix, which is estimated from all previous minibatches, using a forgetting factor to downweight minibatches that are distant in time.

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

Table 2.1: DNN architecture for x-vectors extraction [3].

Considering this system is data-hungry (requiring much more labeled data when compared to the unsupervised i-vector), data augmentation techniques such as the use of additive music, speech and noise are often employed to increase system performance [54]. Due to computational limitations, the DNN is trained using short utterance sequence length. This, conversely to the i-vector system, will cause duration mismatches with the test set, degrading performance. An extended version of the x-vector DNN architecture [55] proposes to reduce this mismatch and achieves state-of-the-art results in the Speaker In The Wild corpus [56]. In this architecture, additional dense-ReLU layers are added to the frame level of the DNN.

2.4.3 Post-Processing

Several post-processing techniques are available for channel compensation. This is important because embeddings contain speaker and channel variability information simultaneously in one space. Therefore, this compensation allows for the removal of channel effects, minimising the within-speaker variability while maximising the between-speaker discriminant information. This compensation is usually conducted using Linear Discriminant Analysis (LDA) and Within-Class Covariance Normalization (WCCN) [57] [45].

LDA offers dimensionality reduction by finding new orthogonal axes to better discriminate between different classes, maximizing between class variance (between speakers) and minimizing within-class

variance (within speakers, that is, channel effects).

The framework of the i-vector post-processing also includes WCCN [58], which is preceded by the LDA. WCCN consists in computing the within class covariance matrix in the total factor space using a set of back ground impostors:

$$W = \frac{1}{S}\sum_{s=1}^{S}\frac{1}{n_s}\sum_{i=1}^{n_s}(w_i^s - \bar{w_s})(w_i^s - \bar{w_s})^t \tag{2.12}$$

where w_s is the mean of the LDA projected total factor vector for each speaker s, S is the total number of speakers, and n_s is the number of utterances of each speaker s. The inverse of W is used to normalize the direction of the projected i-vector components. WCCN is combined with LDA by multiplying the LDA projection matrix A before w_i^s.

2.4.4 Decision Making

The main differentiator between the several pattern recognizers can be traced back to the decision making phase. Depending on the problem statement, the decision making can be done using the embedding directly as a prediction, or by making a decision based on a similarity score. The first case can be considered as classification, as we know predictions belong to a previously defined set of labels, which were used in training, and as such are known by the model. An example of this is gender recognition and closed-set speaker identification with static enrolment. The second approach is useful when we have an open-set problem. This is the case of open-set speaker recognition, namely verification, where impostors do not belong to the pool of known speakers, or/and when we wish to enroll additional speakers.

2.4.4.A Classification

When working with a classification task, predictions can be extracted directly from models. In models based on neural networks, which is the case for the x-vector and other end to end models, these predictions are available in the output layer. Conversely, extracted embeddings can become an input feature to another classification system, such as a Multi-layer Perceptron (MLP) [59].

2.4.4.B Scoring

Using this decision process, the system attributes a score to the utterance under test when comparing to the target class embedding previously enrolled, that is, a similarity metric is used to compare the utterance to the class. In an identification system, the utterance under test is compared against all enrolled speakers, with the highest scoring corresponding to the system's decision. In the case of other

paralinguistic tasks, the utterance embedding is compared against the mean-embedding of the given class.

In the wake of the development of the JFA approach, several log-likelihood scoring methods were proposed [60]. Initial state-of-the-art systems based on JFA, however, utilized cosine similarity [36] between the speaker/class model and the test utterance in order to make a decision. This is due to the fact this method was computationally less expensive. The magnitude of the cosine scoring is known to degrade system performances, therefore only the angle is used for decision. This is mostly attributed to within-speaker variabilities (channel and session).

The current state-of-the-art scoring is conducted using the Probabilistic Linear Discriminant Analysis (PLDA) classifier [61], where the embeddings are centered and projected using LDA, followed by length normalization and PLDA modelling. PLDA directly evaluates the log-likelihood ratio of the fixed-length input vectors under test belonging to the same speaker, and can be seen as a special case of JFA:

$$\vec{w} = \vec{m} + \vec{V}\vec{y} + \vec{U}\vec{x} + \epsilon \tag{2.13}$$

where $\vec{m}$ is a global offset, the columns of the speaker subspace $\vec{V}$ are eigenvoices, $\vec{y}$ is a latent vector having a standard normal prior, $\vec{U}$ is the channel subspace and ϵ is normally distributed with zero mean and a full covariance matrix. Replacing Gaussian distributions in the standard PLDA with Student's t distributions, called Heavy-Tailored PLDA (HT-PLDA), has shown to outperform the standard PLDA classifier [62]. Both PLDA approaches lead to a superior performance when compared to cosine scoring, but with the cost of needing speaker-labeled background data. Additionally, both need several samples for each background speaker spoken in different session conditions to work efficiently.

Score space normalization techniques are also used to reduce variability in the scores. This is especially important in a production setting where a reliable threshold for speaker verification is required for unseen, out-of-domain data. Furthermore, the use of score space normalization is known to improve performance and calibration in such settings.

The Z-norm (zero score normalization) addresses the problem of speaker score variability [63] by employing impostor score distribution for enrollment file. It uses a cohort list with N speakers which we assume to be different from the speakers in utterance u_e (enrollment) and u_t (test). The cohort scores S_e are formed by scoring enrollment utterance u_e with all files from the cohort. The normalized score is calculated as follows:

$$s(u_e, u_t)_{z-norm} = \frac{s(u_e, u_t) - \mu(S_e)}{\sigma(S_e)} \tag{2.14}$$

where $\mu(S_e)$ and $\sigma(S_e)$ are the mean and standard deviation of the cohort scores S_e.

The T-norm (test score normalization) [64] addresses the problem of session variability. It compen-

sates for differences between the training and testing conditions. It is similar to Z-norm with the difference that it normalizes the impostor score distribution for the test utterance (S_t). The T-norm is then:

$$s(u_e, u_t)_{t-norm} = \frac{s(u_e, u_t) - \mu(S_t)}{\sigma(S_t)} \tag{2.15}$$

where $\mu(S_t)$ and $\sigma(S_t)$ are the mean and standard deviation of the cohort scores S_t.

The combination of both norms is called ZT-normalization, and uses the Z- and T-norm in series. Scores normalized using ZT-norm are therefore normalized with respect to the enrollment and test utterances.

The S-norm (symmetric normalization) computes an average of normalized scores from Z-norm and T-norm. Unlike the ZT-norm, S-norm is symmetrical as it does not depend on the order of u_e and u_t. The S-norm can be expressed by:

$$s(u_e, u_t)_{s-norm} = \frac{1}{2} \cdot (s(u_e, u_t)_{z-norm} + s(u_e, u_t)_{t-norm}) \tag{2.16}$$

More recently scores have been normalized using adaptive normalization techniques. The AT-norm (adaptive T-norm) [65] is an evolution of T-norm, adjusting the speaker set to the target model, improving performance. This is conducted by only using part of the cohort to compute the mean and variance for normalization. The most common adaptive cohort selection proposed in the literature includes selecting the N closest files to the enrollment/test file [66]. Others have also suggested a random selection of utterances from the trial set [67].

The AS-Norm is derived from the AT-Norm, but preserves the symmetrical property of the S-Norm. With AS-norm, the score is normalized as such:

$$s(u_e, u_t)_{as-norm} = \frac{1}{2} \cdot \left[\frac{s(u_e, u_t) - \mu(S_e(\Gamma_t))}{\sigma(\Gamma_t))} + \frac{s(u_e, u_t) - \mu(S_t(\Gamma_e))}{\sigma(S_t(\Gamma_e))} \right] \tag{2.17}$$

where Γ_e is the cohort selected from the enrollment set and Γ_t the cohort from the test set.

3

Voice Profiling for Crowdsourcing

Contents

3.1 Introduction

The advent of complex models such as Deep Neural Network (DNN)s raises the need for large amounts of labelled data that is required to train these models [68]. The amount of data available is estimated to exceed 175 zeta bytes by 2025 [69], however much of this data is unstructured and unlabeled. Challenges arise, therefore, when there is a need for large amounts of labeled data to create models for specific tasks. For example, specific domain utterances are required to adapt ASR systems for automotive or medical environments; large speech corpora are required to construct accurate speaker pattern recognizers; image labels are required to train image recognition software. All of these can be easily performed by humans to produce a gold standard (a ground truth) where computers lack aptitude to do so. However, the construction of large datasets using expert annotators is both time-consuming and financially costly.

Instead of using experts to label a dataset, crowdsourcing platforms enable a more scalable labelling process by breaking down large datasets into small tasks. These well-defined micro-tasks are performed by the crowd with similar quality results [70]. This technique is often used by companies and universities by providing the required data to create accurate models at a lower cost. The required user base for a given task is obtained by awarding users for each completed task. In one hand this invites a larger pool of willing workers to complete these tasks. However, users may be encouraged to produce low quality work as it often blends in with the crowd [71]. As a result, several methods to detect low quality work have been developed, namely agreement across workers or with a gold standard [72], or more complex behavioral capturing techniques to predict outcome measures such as work quality, errors, and the likelihood of cheating [71].

Unlike other crowdsourced collections, the detection of low quality work in speech corpora collected via crowdsourcing is less straightforward. Contributors' submissions cannot be automatically probed for low quality work using agreement across workers or a gold-standard, as each submission is inherently unique. As a result, validation tasks are typically set up to verify certain characteristics of speakers (nativeness, gender) or submission correctness (prompt matching, for example), as exemplified in Figure 3.1.

Human listeners are known to excel in the task of familiar speaker identification (meaning a close acquaintance or someone famous) even in noisy conditions. Such a task can be considered pattern recognition, as humans are capable of analyzing many different aspects of a voice to identify a speaker, including spectral and prosody characteristics, gender, age range, language/accent, and speaking style [73]. As such, humans are better at recognizing people who speak a language which they are familiar with, as the phonology of the particular language is known. The findings in [74] show that familiar voice recognition and unfamiliar voice discrimination (speaker verification) are known to be separate cognitive abilities: voice discrimination involves analysis of speech features as well as the pattern-recognition

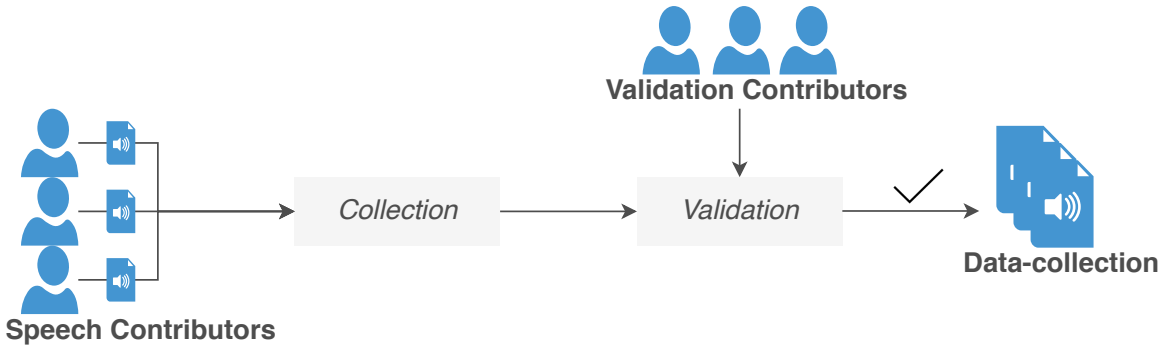

Figure 3.1: Typical dual-stage speech data collection using crowdsourcing.

ability of the brain, which is used by the familiar speaker identification as well.

The main limitations of human listeners is twofold: they may erroneously identify someone from a voice recording if they are expecting to hear that person; and are susceptible to bias such as similar recording conditions and other contextual information [75]. This may raise some questions regarding the performance of human annotators in the context of data collection validations for example, where similar voices belonging to different speakers may be validated due to confirmation bias. Additionally, uncertainty may be related to the difficulty of the task at hand or reflect the inherit ambiguity of the submission and provide useful information. An example of this is the voice of a child, which presents ambiguities in terms of gender [76]. These uncertainties may translate into diverging labels, requiring a larger pool of contributors in order to obtain the ground-truth by measuring inter-annotator agreement.

As such, a validation process that relies on human listeners adds to the costs of the dataset as multiple contributors are required for agreement. Additionally, the validation task setup for the detection of contributors sharing accounts or contributing from different accounts is less obvious. This is important because clients are not solely interested in the data, but also in the metadata that come with it, i.e., who said what. As such, the use of automatic pattern recognizers for validation may be of value.

In this chapter we will focus on speech corpora collected via crowdsourcing, i.e. speech data collections. The proposed approach in this chapter is to use speech to detect differences in extracted features between the enrolment stage, the profile of the contributor and the detection stage, all of which can be used in crowdsourcing task performance control. This is particularly relevant for DefinedCrowd, an AI startup that offers a quality-focused data platform that combines machine learning and human intelligence. For DefinedCrowd, the ability to detect mismatches in speaker profiles (e.g. gender) and fraudulent behaviour (e.g. speakers using multiple accounts) by enforcing demographic distributions is crucial, as it ensures the corpora demographic characteristics are according to clients' requirements.

In this chapter, we start by describing the datasets used in the experiments in section 3.2, which will be used for both gender and speaker verification. Section 3.3 proposes an end-to-end approach to the gender verification task in the context of crowdsourcing. In section 3.4, we investigate how speaker

verification methods can be adapted to a crowdsourcing validation setting. In particular, we explore the use of score normalisation techniques for domain adaptation. In subsection 3.4.3 in particular, we draw parallels between human speaker verification and automatic speaker verification and show how gender plays an important role in human speaker verification. As a result of this analysis, in Section 3.5, we study the use of speaker embeddings in the task of gender verification. Finally, in section 3.6, we discuss the proposed approaches and their suitability as a real world application.

3.1.1 Crowdsourcing Terminology and Workflow

Industry terminology present in crowdsourcing is dissimilar to speech research terminology. In this subsection, we present the terminology used in this field and associate it to the terminology typically seen in research literature.

Contributors (speakers) are uniquely identified and can participate in multiple crowdsourcing speech-data collections (henceforth called Jobs), being assigned a unique identifier for each Job (JobMemberID). Additionally, contributors' self-reported biometric information such as age, gender and nationality is appended to their profile. Each task within a Job, called Human Intelligence Task (HIT), indicates the prompt to be read, if it is a prompt reading task, or indication regarding the topic to be discussed, if it is a dialog or spontaneous speech. HITs can be performed by different JobMembers, and a JobMember can perform several HITs. This results in a HIT Execution (utterance), where the result is stored. A simplified database schema of a crowdsourcing environment is shown in Figure 3.2.

The typical workflow of a contributor in the context of speech data collections includes the following:

1. The contributor signs up in the crowdsourcing platform, filling out a profile form that includes age, gender and language information;

2. HITs are allocated to the contributor. These tasks are in agreement with the submitted profile: a language specific task is only given to contributors that have that language in their profile or when there is a need for more executions from that age/gender group.

3. The contributor records the utterances and submits them.

4. The completed task suffers a quality control process that includes mechanisms that discard work automatically (e.g. noise, short duration) and a screening process with additional validation tasks that check for nativeness, correct spelling, and completeness.

5. The contributor is awarded a monetary compensation only after a given number of successfully completed tasks.

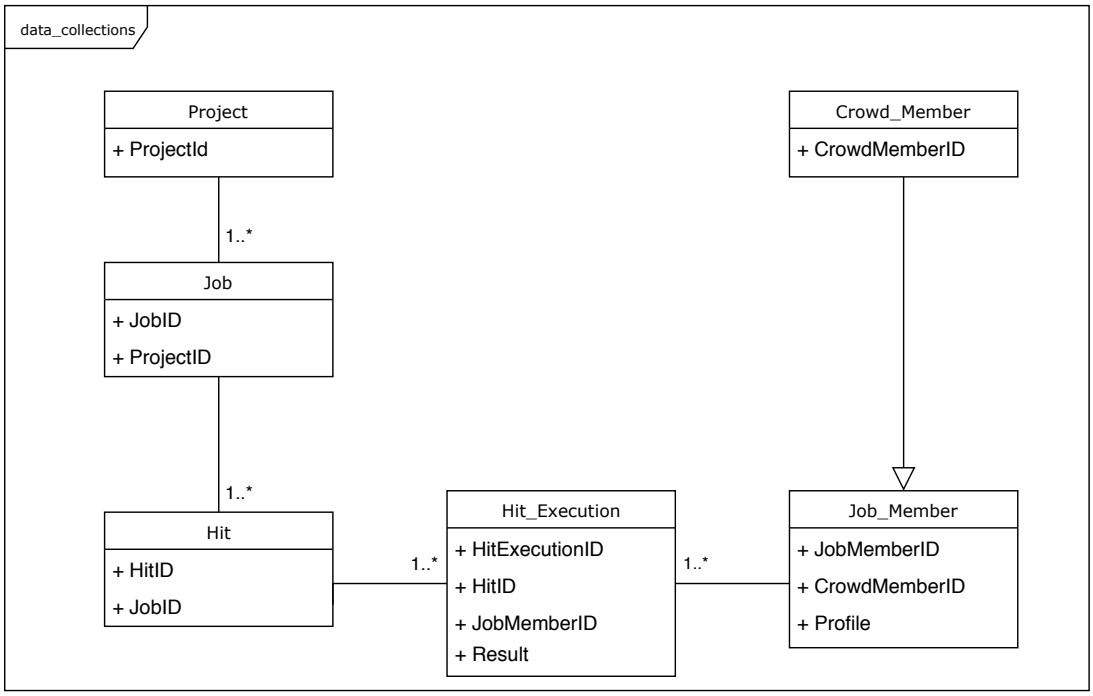

Figure 3.2: Crowdsourced Database.

3.2 Datasets

To fulfill the aforementioned goals, we will look into two main types of datasets. We begin by presenting DefinedCrowd data collections. After that, we present auxiliary datasets used in the development of this thesis, including Voxceleb and the Common Voice datasets. Details of all datasets are aggregated in Table 3.1 for convenience.

3.2.1 DefinedCrowd data collections

The speech data collections from DefinedCrowd are a multilingual collection of prompt reading tasks recorded in an application environment using a mobile phone. The database is divided according to jobs, with each job having different requirements, including language, nativeness and recording conditions. Figure 3.3 shows an instance of a HIT to be performed by a contributor, in this case reading from a prompt.

Submitted HIT executions undergo a processing step before being stored in the database. In the available datasets in particular, wave files are downsampled to 16kHz, with a bit depth of 16, in a single channel. An energy-based Voice Activity Detection (VAD) filters silence, leaving a leading silence of duration 300ms and a trailing silence of 300ms.

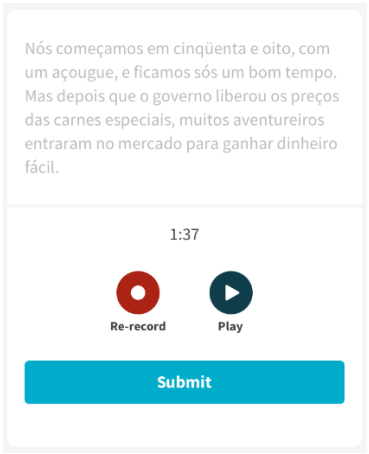

Figure 3.3: HIT Instance [2].

DefinedCrowd - US English (DC_EN) This dataset is a scripted speech data collection of American English in a silent environment. The prompts include wake-up calls of personal virtual assistants, followed by a request. A total of 493 (271 female, 222 male) adult contributors were enrolled to this job, resulting in a total of 4,188 executions. Submitted executions were validated by the crowd in a separate job: 1,455 executions were either cancelled or refused due to low quality. As such, the final dataset consists of 2,745 utterances belonging to 277 contributors. Average utterance duration is 4.98s, with a minimum duration of 0.09s and a maximum of 10.71s.

DefinedCrowd - Hebrew (DC_HE) This dataset is a scripted speech data collection of Hebrew. The prompts amount to 30 HITs. The number of enrolled contributors for this job was 223 adults, of which 115 reported to be female (108 male), producing 2,460 executions. These executions were later verified by creating a separate job, where other contributors validated the prompts. As a result of this validation, and after removing cancelled executions, the dataset amounts to 2,144 utterances and 147 contributors. The average utterance duration is 8.19s, with a minimum duration of 4.12s and a maximum duration of 18.40s.

DefinedCrowd - Mexican Spanish (DC_ES) This dataset is a scripted speech data collection of Mexican Spanish. The prompts, amounting to 9,990, consisted of wake-up calls of personal virtual assistants, followed by a song request. A total of 152 (79 female, 73 male) registered adult contributors were enrolled to this job, resulting in a total of 10,500 executions. Submitted executions were matched against the given prompt in a separate job by crowdworkers. Of these executions, 8,334 utterances belonging to 65 users form the dataset, the remainder being discarded due to user cancellations or validation failure. Average utterance duration is 4.61s, with a minimum duration of 1.47s and a maximum of 10.65s.

3.2.2 Auxiliary Datasets

3.2.2.A Voxceleb

VoxCeleb is a collection of videos from celebrities extracted from YouTube. The speakers span a wide range of different ethnicities, accents, professions and ages. The nationality and gender of each speaker (obtained from Wikipedia) are also provided. Videos included in the dataset are shot in a large number of challenging multi-speaker acoustic environments. All are degraded with real world noise, consisting of background chatter, laughter, overlapping speech, room acoustics, which is also paired with a variety of different recording conditions adding channel noise. *Voxceleb1* [77] contains over 100,000 utterances for 1,251 celebrities, with 55% of them males. *Voxceleb2* [78] contains over a million utterances from over 6,000 speakers, with 61% of the speakers being male. Utterances on both datasets range from 4 to over 20 seconds. The dataset is also multilingual, with speech from speakers of 145 different nationalities, covering a wide range of accents, ages, ethnicities and languages. Top five speaker nationalities include U.S.A, U.K, Germany, India and France.

3.2.2.B Common Voice

The Common Voice project [79] is an open-source crowdsourced speech data collection envisaged to assist in speech recognition software development. It includes almost seven thousand hours of validated recorded speech from 56 languages and dialects. The collection is conducted by untrained volunteers who read sentences from original contributions and public domain texts containing monologues from film scripts. Besides having an ID being attributed to each volunteer, metadata information such as gender and age can be disclosed by the volunteers.

The project offers several language-specific subsets for download. For this work, we used the two largest subsets available, belonging to multi-accent English and German. Given that the data collection process is similar to for-profit crowdsourcing, Common Voice (CV) datasets were also used to evaluate the performance of our validation systems. However, since there is no gratification involved in the execution of Common Voice tasks, we assumed volunteers did not submit low quality work, namely by committing fraud.

Common Voice - English (CV_EN) This corpus is a subset of the Common Voice corpus containing prompt and digit reading from speakers with different accents. The dataset is the largest subset of the Common Voice project with 1,469 validated hours of speech in .mp3 format and 61,528 speakers. The subsets used in this work resulted from a combination of the development and test subsets, from which the entries without gender information and part of the Singleword Benchmark were excluded, which resulted in 5,848 utterances from 2,467 speakers. The average utterance duration is 6.00s, with

a minimum duration of 1.22s and a maximum of 11.02s.

Common Voice - German (CV_DE) This corpus is a subset of the Common Voice corpus containing prompt reading from speakers with German and Austrian accent. The dataset is the second largest subset of the Common Voice project with 692 validated hours of speech in .mp3 format and 11,731 speakers. The subsets used in this work resulted from a combination of the development and test subsets, from which the entries without gender information were excluded. This resulted in 6,680 utterances from 1,191 speakers. Average utterance duration is 5.73s, with a minimum duration of 1.22s and a maximum duration of 11.02s.

Dataset	# Utt	# Spk
VoxCeleb1	4,878	40
VoxCeleb2	1,092,009	5,994
CV_EN	5,848	2,467
CV_DE	6,680	1,191
DC_EN	2,745	277
DC_HE	2,144	147
DC_ES	8,334	65

Table 3.1: Dataset sizes.

3.3 Gender Verification

The ability to automatically identify gender from speech has many benefits for numerous applications. Examples include enhancing Human-Computer Interaction, personalized advertising strategies and adaptive costumer service. In data collection tasks in particular, a dataset that does not follow a predetermined gender distribution, may that be due to an unbalanced gender representation or one that does not reflect the typical distribution of the task at hand, may introduce biases in machine learning classifiers. In this section, the challenge of guaranteeing gender demographics metadata correctness is addressed, using automatic gender prediction models from speech.

3.3.1 Experiments

The network used for our experiments was based on the M5 network architecture [80] and is presented in Figure 3.4. The model was implemented and trained in Python, using the PyTorch deep learning framework. The *Voxceleb2* dev subset was used for training, and the *Voxceleb2* $test$ for development. Prior to training, utterances were converted to `.wav` format and downsampled to 8 kHz using sinc inter-

polation. An energy-based Voice Activity Detection (VAD)[1] was used to filter out non-speech frames. Resulting utterances with less than 2s were padded with zeros at the end.

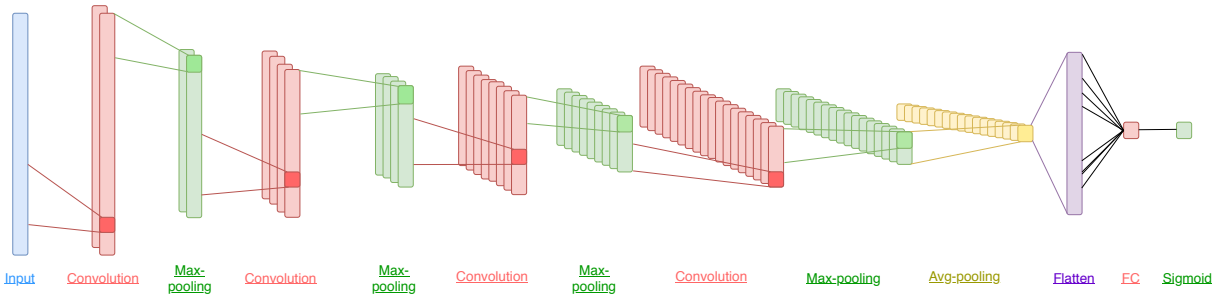

Figure 3.4: Convolutional Neural Network Architecture

The network consists of four convolutional layers, each followed by a batch normalization layer and a maxpooling layer. The first layer receptive field receives a time-domain 16000-length vector that represents a waveform of 2 seconds, at a sampling rate of 8 kHz. This layer possesses a receptive field size of 80, with 256 filters with stride 4. This offers a receptive field that covers 10ms of speech, which is comparable to window lengths of other feature extractors. The following convolutional layers have a fixed receptive field of size 3, with increasing filter length of 128-258-512. The number of feature maps doubles as temporal resolution decreases by a factor of 4 in the max pooling layers. Batch Normalization is used on the output of each convolutional layer, before applying ReLU non-linearity. This alleviates the problem of exploding and vanishing gradients. The classification step is conducted using an average pooling layer, paired with a fully connected layer of length 512, and a sigmoid layer for the output. The output provides the gender classification, with values above 0.5 indicating the speech segment was classified as 'Female'

The model was trained using the Adam [81] optimizer, with weight decay set to 0.0001. The loss function chosen for this task was binary cross entropy. At first, the learning rate was set to 0.01 and later on decreased to 0.001 during training, using a scheduler with step size of 20. The number of epochs was fixed to 200, with early stopping indicating no further improvements were detected after 35 epochs.

3.3.2 Results

This section compiles the results obtained on DefinedCrowd and Common Voice data collections, including Precision, F1-score and Recall, in Table 3.2.

The obtained results show significant performance variations in between datasets and genders. In [41], the authors reported a Recall of 98.04 and 95.05 for 'Male' and 'Female', respectively, which is similar to the performance detected on the DefinedCrowd datasets. Typically, 'Male' recall outperforms

[1] https://pytorch.org/audio/_modules/torchaudio/transforms.html#Vad

	Male			Female		
Dataset	Precision	Recall	F1 Score	Precision	Recall	F1 Score
CV_EN	0.97	0.94	0.95	0.71	0.83	0.76
CV_DE	0.98	0.96	0.97	0.80	0.86	0.83
DC_EN	0.90	0.97	0.93	0.97	0.92	0.95
DC_HE	0.99	0.97	0.98	0.97	0.99	0.98

Table 3.2: Results obtained on Crowdsourced datasets

'Female' recall, due to the fact that many speech corpora are unbalanced in terms of gender. This is also the case of *Voxceleb*, to a smaller extent, but our results do not show a consistent out-performance for 'Male' labels on the DefinedCrowd datasets. We note, however, that for the Common Voice dataset, performance metrics for 'Female' are much lower than for 'Male' (20% absolute difference in precison on CV_EN), which is beyond what is expected due to gender unbalance during training. Unlike the DefinedCrowd datasets, which were manually validated, we presented results on Common Voice datasets under the assumption that gender labels were correct. Considering these results, a manual validation step was conducted by one annotator, obtaining the true gender label of the worst performing utterances. These are characterized by having network outputs close to 0 or 1, indicating strong predictions. In CV_EN, out of the 5,847 utterances under test, 508 were miss classified, with 56 of these having strong predictions . Meanwhile in CV_DE, out of the 6,680 utterances under test, 376 were miss classified with 46 of these having strong predictions.

As a result of this manual validation, we detected that a majority of the worst performing utterances (over 80%) had in fact the wrong label. Furthermore, all of the erroneous labels were female and were attributed to male speakers. While we have no concrete explanation for the reason why a substantial amount of male speakers had 'Female' labels, we believe this is due to error during profile registration, as there is no incentive to provide 'Female' labels other than the fact the datasets themselves lack female representation.

3.4 Speaker Verification

Unlike typical speaker verification evaluation datasets, which have well defined and validated train, development and test sets, datasets collected using crowdsourcing platforms may lack these partitions and/or validated labels, making training and performance evaluation more difficult. In our experiments, we evaluate the system's performance on a job-level, meaning we have no previous information regarding a speaker, that is, there is no prior enrolment. As such, our trial setting differs from typical speaker verification evaluations because a well established enrolment set does not exist. Our implemented automatic speaker verification system takes the first completed task as enrollment, and iteratively complements

the enrolled profile with N subsequent tasks that are verified by the system. This ensures additional robustness by capturing additional intra-speaker variability not found in the first completed task.

3.4.1 Proposed Architecture

A brief overview of the speaker verification pipeline used in the experiments is represented in Figure 3.5. Its front-end (A) consists of an embedding extraction network, which condenses information related to the speaker to a fixed sized feature vector from a variable length audio signal. The back-end (B) consists of a scoring procedure and is finalised by a decision step.

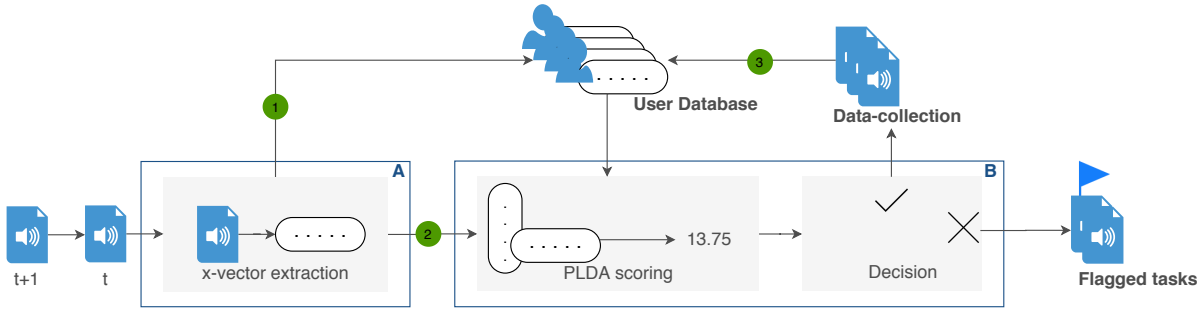

Figure 3.5: Speaker Verification pipeline.

The embedding corresponding to the first completed task of each user is computed and then enrolled to the user database for future decisions (1). Embeddings belonging to subsequent completed tasks are compared against the enrollment (2) and a same-or-different speaker decision is made. If the task is verified, that is, if the decision resulting from the scoring of the two embeddings is positive, then the user database is updated to include the new embedding (3).

Our embedding extraction and decision-making followed the Kaldi Speech Recognition Toolkit [82] recipe of *VoxCeleb*. Training is conducted on the dev set of *VoxCeleb2*.

A – Embedding extraction The speaker verification pipeline is implemented using i-vectors and x-vectors. The baseline is a traditional i-vector system. The front-end features consists of 24 MFCCs calculated every 10ms with a frame-length of 25ms that are mean-normalized over a sliding window of up to 3 seconds. Delta and delta-deltas are appended to create 72-dimension feature vectors. An energy-based VAD selects features corresponding to speech frames. The UBM is a 2048 component full-covariance GMM. The system uses a 600 dimension i-vector extractor which was trained on the 100 thousand longest utterances. This results in a reduced training time and improved performance. The i-vectors are centered, dimensionality reduced to 200 using LDA, and length normalized.

Our system proposes using x-vectors for speaker embeddings. In the proposed system, the features are 30 dimensional MFCCs obtained every 10ms with a frame-length of 25ms, mean-normalised over

a sliding window of up to 3 seconds. An energy-based VAD module filters out non speech frames. Short-term temporal context is handled by the TDNN architecture. The database was processed to maximize performance prior to DNN training. *VoxCeleb2* was augmented by combining the clean data with reverberation, noise, music, and babble from the MUSAN corpus [83]. This doubles the size of the original data. The augmented dataset is filtered by removing speakers with less than 8 utterances, and utterances with less than 5 seconds of speech. Cepstral Mean and Variance Normalization (CMVN) is applied to speech frames. Mini-batch size was set to 64. The DNN model was trained to classify training speakers using a multi-class cross entropy objective function. The network was trained for several epochs using natural gradient stochastic gradient descent. At first, the learning rate was set to 0.001, and later on decreased to 0.0001 during training using a dropout scheduler. The DNN outputs 512-dimensional embeddings which are centered, dimensionality reduced to 200 using LDA, and length normalized.

B – Scoring and Decision A score is attributed to a pair of embeddings using Gaussian-PLDA scoring [84]. An execution is considered to be verified if its embedding scores higher than a given threshold, when evaluated against the enrolled embedding. In the context of crowdsourcing, False Acceptances occur when the system validates fraudulent task completions from speakers other than the enrolled one, while False Rejections erroneously flag tasks that were completed by the enrolled speaker.

Decision thresholds are a by-product of minimising speaker recognition performance metrics. In Equal Error Rate, the threshold is chosen as to equate the False Rejection Rate (FRR) with the False Acceptance Rate (FAR). Another typical metric is the minimum normalised Detection Cost Function (minDCF), a weighted sum of False Rejection and False Acceptance probabilities, which can be used to measure performance when taking into account system calibration.

When switching to a live production environment, trials are submitted 'on-the-fly', meaning a decision threshold must be decided beforehand. If the new trials belong to unseen, out-of-domain data (with different language and channel conditions), the previously computed threshold must be adapted in order to achieve a similar performance [85]. Score space normalisation techniques can be used to tackle this problem, by reducing variability in the scores. The Adapted Symmetric Scoring normalisation [67] normalises scores according to the mean and standard deviation of impostor distributions. This normalisation subset, also called cohort, is formed by selecting the N_t closest files from the enrolment/test [86]. Other authors have also suggested a random selection of utterances [67]. Typical cohort lists have a sample size (N_c) of thousands, making them able to experiment with N_t. For instance, in [86], the authors reported a minDCF minimum by using an N_c set to between 200 and 500 comparisons, in a cohort list with N_c over 2,000 files. In an online setting where there is no prior enrolment, the use of score normalisation techniques requires a waiting period to allow for a number of utterances to be submitted

and be used in the cohort list. In our experiments, we opted, for each dataset, to select a smaller cohort list containing random utterances, and using the full list for normalisation calculation (i.e., $N_t = N_c$).

3.4.2 Experimental Set-up

The decision-making process follows a production setting that compares the initially completed tasks to all subsequent tasks from a given user, which allows for the assessment of identity as early as possible. This approach is more challenging than the iterative enrollment, however, as it reduces the enrollment exposure to the inherit acoustic variabilities in speakers, but allows for a better understanding of the system's performance in these conditions. In order to evaluate impostor rejection, all utterances belonging to other speakers are compared, generating the non-target trials (impostors).

Considering users are paid for each completed task, there is additional motivation to commit fraud, either by enrolling additional speakers on the same account (in order to expedite completed tasks) or by contributing with multiple accounts. Although several validation steps are applied to completed tasks, none include a biometric evaluation step, meaning speaker labels are not validated. The main reason for this is that the number of additional tasks required to perform speaker verification, and thus detecting fraud, is $O(N \cdot M)$ for intra-speaker fraud detection and $O(N^2)$ for inter-speaker fraud, where N is the number of contributors and M the number of HITs allocated for each contributor. Additionally, given the inherit difficulty of unknown-speaker verification tasks by humans, requiring the contributor to listen to the same audio files multiple times to reach certainty, the resulting verification may contain low-quality responses and require a larger pool of contributors in order to reach agreement. As such, a comprehensive speaker verification job would add significant costs to the speech data collection.

In order to assess the probability of occurrence of this fraudulent behaviour, the datasets from DefinedCrowd were selected for manual validation using a single annotator. Considering the size of the datasets, only a subset of trials were selected for manual validation. This manual validation step consisted of first running a speaker verification task using x-vectors on the full dataset in accordance to the trial settings previously presented. For each contributor, all flagged utterances (that failed automatic verification) were manually validated, together with the automatically verified utterance with the lowest PLDA score. If the lowest verified utterance was a false acceptance, we proceeded to the next verified utterance, up until the first true acceptance. Due to the size of the dataset, we assumed all utterances with a higher score than the first true acceptance of each contributors were also valid. Inter-speaker comparisons were used to check whether speakers were using multiple accounts. Only the utterances with the lowest PLDA score were validated, as we assumed utterances with higher scoring were correctly verified.

Although we assume tasks submitted to Common Voice are devoid of any fraud, we also conduct a, albeit smaller, manual validation step to confirm the absence of fraud. This validation is similar to the one

executed in DefinedCrowd datasets, with calculated PLDA scores filtering comparisons for subsequent inspection. Unlike previous validations, however, we only manually validate the 50 worse performing trial-comparisons of the full dataset.

3.4.2.A Results

In this subsection we present the results for the manual validation step of the crowdsourced speech data collections presented in Section 3.2 and for speaker verification tasks using the proposed automated system.

The results of the manual validation are reported in Table 3.3, where intra-speaker and inter-speaker fraud is accounted for on each dataset. Additionally, we present the size of the reduced datasets in terms of number of utterances and speakers that resulted from the removal of all fraud: all flagged utterances belonging to the same contributor were individually removed, while all utterances belonging to subsequent contributors found to have instances of inter-speaker fraud were removed, together with the flagged contributor.

	Size (reduced)		Intra-Speaker Fraud		Inter-Speaker Fraud	
Dataset	# Utt	# Spk	# Utt	# User	# Utt	# User
CV_EN	5,848	2,467	0	0	0	0
CV_DE	6,680	1,191	0	0	0	0
DC_EN	2,733	277	0	0	0	0
DC_HE	2,144	147	13	5	0	0
DC_ES	7,893	61	8	3	264	3

Table 3.3: Reduced dataset size and detected fraud.

As expected, we failed to detect any fraud on CV_EN and CV_DE. Additionally, no fraud was found on DC_EN. We note that while the amount of intra-speaker fraud detected on DC_HE and DC_ES can be considered minor, inter-speaker fraud on DC_ES in particular was substantial, considering each contributor had a large amount of tasks attributed to him/her.

Results obtained on the different reduced crowdsourced datasets are summarised in Table 3.4. We also present results for the ”baseline” *VoxCeleb1* dataset. It is possible to observe that overall Equal Error Rate (EER) (%) results on the crowdsourced datasets, with the trial settings explained in Subsection 3.4.1, are comparable with the results on *VoxCeleb1*. This is a promising result, considering the enrolment data is a single utterance per speaker. Furthermore, we note that the Decision Threshold (DT) that result in the EER have different values for each dataset: we report a mean absolute difference of 2.62 using x-vectors, which confirms the need for a normalisation step in order to use the same threshold.

Results also show that scoring using x-vectors was able to outperform i-vectors on all datasets except for DC_ES. Better results on x-vectors were expected, especially considering the i-vector extractor was

		None		AS-Norm	
Dataset	Method	EER(%)	DT	EER(%)	DT
Voxceleb1	i-vector	5.329	-1.00	-	-
	x-vector	**3.128**	-3.26	-	-
CV_EN	i-vector	3.337	4.31	2.998	0.97
	x-vector	**2.319**	7.08	**2.432**	1.94
CV_DE	i-vector	4.208	5.40	3.916	0.40
	x-vector	**2.915**	7.67	**2.860**	0.83
DC_EN	i-vector	2.737	5.10	2.982	0.86
	x-vector	**2.083**	8.09	**2.492**	0.83
DC_HE	i-vector	1.099	9.94	1.648	1.64
	x-vector	**0.599**	14.74	**1.549**	1.93
DC_ES	i-vector	**3.527**	7.51	**2.222**	-0.39
	x-vector	4.082	11.40	3.676	-0.30

Table 3.4: Results obtained on different datasets.

trained on the largest utterances of *Voxceleb2* and is therefore more susceptible to have its performance degrade due to mismatches in duration [51].

We experimented with several values of N_c for the size of the cohort list, using the obtained results without normalisation as a baseline. Considering the dataset size limitations, we opted to present average results obtained on five experiments using $N_c = 50$, which is the lowest value we obtained without suffering substantial performance losses in terms of EER. As seen in 3.4, we note overall that the EER increases slightly, except for DC_HE, where an absolute increase of around 1% was noted using x-vectors and 0.5% using i-vectors. Additionally, we noted decreases in EER on DC_ES and CV_DE. Decision thresholds on all datasets shifted to an average value of 1.17 and a mean absolute difference of 0.71, indicating a single decision threshold could be applied to these datasets in practice while maintaining the performance using unnormalised scores. The resulting score distributions can be visualised in Figure 3.6, for the DC_EN dataset.

The Detection Error Trade-off (DET) curves in Figure 3.7 show relevant differences between the AS-norm adapted and non-adapted curves, namely the progression of False Acceptance probabilities when decreasing False Rejection probability. However, False Rejection probabilities are lower on the adapted scores when minimising False Acceptance probabilities for the DC_HE and DC_EN datasets. This can be explained by a normalisation that agglomerates scores to the opposite decision region, instead of making them more separable. We hypothesise this is a consequence of the size of the cohort list. Unlike the DET curves for DC_HE and DC_EN, the curves for the DC_ES dataset do not show substantial differences, with AS-norm achieving better performance near the EER point. We find this is due to having 65 speakers in this dataset (contrasting with 147 and 277 speakers in DC_HE and DC_EN, respectively), which leads to a normalisation that reflects the original score distributions.

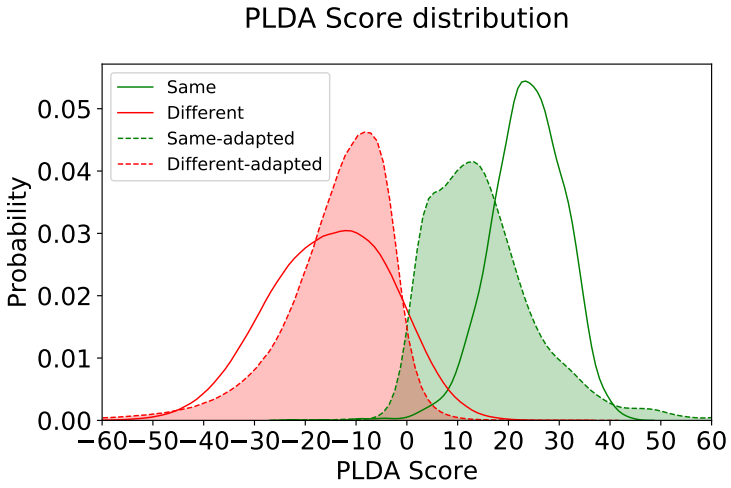

Figure 3.6: Score distributions (DC_EN) for same and different speakers with and without score normalisation.

Fraud detection results for our system are presented in Table 3.5. We note that while the number of False Acceptances (the system verified different speakers) is very low, it also presents a relatively high number of False Rejections (the system did not verify the same speaker). A higher DT would alleviate this, at the cost of a higher false acceptance rate. It is typically worthwhile to maintain a larger percentage of False Rejections (and thus lower false acceptances) in the context of crowdsourcing because the consequence of this decision is a higher number of rejected executions, but reduced fraud. When attempting to minimize false rejections, the total number of rejected executions would be lower, but it would also result in a higher presence of fraud in the dataset. Therefore, the size of the data collection can be considered one of the main driving forces in this decision, as it may not be viable to increase rejections in a small, niche, dataset, whereas in a larger dataset it may be worthwhile to reduce false acceptances further, at a cost of more rejections.

Dataset	FA(#)	FR (#)
DC_EN	-	52
DC_HE	8	8
DC_ES	0	356

Table 3.5: Fraud detection result.

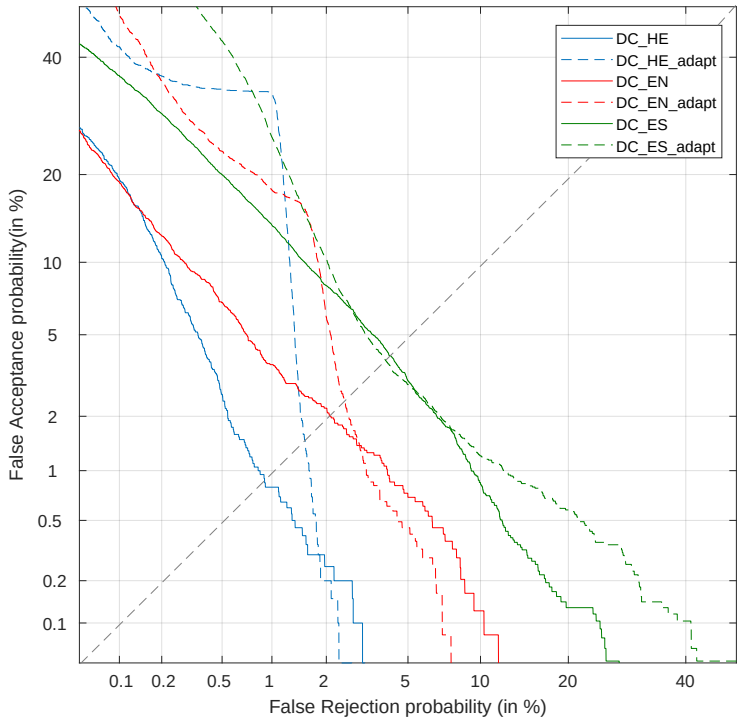

Figure 3.7: DET curve for DefinedCrowd datasets using x-vectors.

3.4.3 Comparison with Human Speaker Verification

While we wish to present an automated system that can automatically validate submitted executions with regards to speaker profile (including gender), the study of correlations (if they exist) between human uncertainty/errors pertaining to speaker validation tasks and automatic speaker verification errors may be worthwhile. This association could prove to be useful when calibrating the system, reducing its errors by introducing an uncertainty range where submissions are flagged for validation using expert annotators, for example.

In this subsection, we investigate human performance in the voice discrimination task and compare it to the automatic speaker verification system. The main motivation behind this study was to understand the correlation (if it exists) between speaker verification by humans and machines in same-or-different speaker decisions.

Previous studies on this topic limit themselves to reporting overall performances of human listeners when compared to speaker recognition systems and probing robustness to different channel conditions, such as noise. In [87], the authors studied human and machine performance using the NIST 1998 speaker evaluation data. They found that human results (using mean combining of individual responses) were similar to the best computer algorithms (at the time, GMM) in the same-handset condition and that

human results outperformed machines when the signal was degraded by background noise. A similar study was conducted in [88], where most results showed that humans outperformed machine algorithms trained to match the noise environment.

3.4.3.A Experimental Set-up

In this experiment, a subset of the DC_EN dataset was submitted for human validation using DefinedCrowd's crowdsourcing platform. The job contained 125 randomly selected trials (utterance pairs) from the x-vector speaker verification system's results (without score normalization). These profile labels of these utterances (and subsequently the same-or-different-speaker decision labels) were manually verified by the author. The PLDA scores followed a uniform distribution, translating into a balanced distribution of same and different speaker comparisons, reducing confirmation bias. Additionally, the job description did not mention the profile validation nature of the experiment and stated that different noise conditions may occur and should not be taken into account when making a decision. In this experiment, all 8 annotators were DefinedCrowd employees, therefore guaranteeing the quality of the submitted work. We also note that these annotators are considered to be naive listeners, as they lack the same expertise as forensic experts.

For each HIT, 3 randomly selected contributors were asked to listen to two utterances and respond, in a 1-5 Likert rating scale, whether they belonged to the same speaker or not. In it, choosing 1 represented being absolutely sure they were different speakers while 5 represented being absolutely sure they were the same speaker. The use of the Likert rating scale allowed for the introduction of human uncertainty. Aggregation among annotators was used to produce the final label, which is then used to compare against the profile labels and PLDA decisions. The decision follows the logic presented below:

- An audio pair is considered to have **different speakers** if the majority of responses were lower than 3 (Not Sure) in the Likert Scale and there was no more than 1 response using 3 (Not sure) or 4 (Somewhat Agree).
- An audio pair is considered to belong to the **same speaker** if the majority of responses were higher than 3 in the Likert Scale (Not Sure) and there was no more than 1 response using 2 (Somewhat Disagree) or 3 (Not sure)
- An audio pair is considered to be **ambiguous** in terms of same-or-different-speaker decision if there is more than 1 response with a Likert rating of 3, or if responses are strongly contradictory, that is, single responses using 1 (Strongly Disagree) or 5 (Strongly Agree) with opposing majority.

The speaker verification decision logic follows the one presented in Section 3.4 with the added third class "Not Sure", which models the system's uncertainty. The uncertainty interval is centered on the de-

cision threshold that results in a EER. The lower and upper bound are set to $[3, 13]$ which approximately ensures classifier confidence of 96% on the DC_EN dataset.

Additionally, the achieved agreement among annotators was measured using Krippendorff's alpha coefficient [89]. The alpha coefficient can be calculated using the following formula:

$$\alpha = 1 - \frac{D_o}{D_e} \tag{3.1}$$

where D_o is the disagreement observed and D_e is the disagreement expected by chance. An alpha value of $\alpha = 1$ indicates perfect reliability while an $\alpha = 0$ indicates the absence of reliability (statistically unrelated). An alpha valued below zero indicates disagreements are systematic and exceed what can be expected by chance.

3.4.3.B Results

We report for the ordinal data of the Likert scores submitted by the contributors an alpha value of $\alpha = 0.851$, which indicates a high degree of agreement among annotators. The decision performance of each system against the profile labels is presented in Table 3.6.

	Decision	Precision	Recall	F1 Score
Human Validation	Same-Speaker	**0.95**	0.97	**0.96**
	Different-Speaker	0.93	**1.00**	**0.96**
PLDA Validation	Same-Speaker	0.76	**1.00**	0.87
	Different-Speaker	**1.00**	0.86	0.92

Table 3.6: Performance results for same-or-different speaker decisions.

Overall, human validation results are similar to the automatic speaker verification system using PLDA scoring. Human validation outperforms PLDA on same-speaker decisions, while the PLDA yields better results on different-speaker decisions (100% precision for different-speaker decisions). However, overall, human performance is more consistent, unlike the PLDA, which reports an absolute decrease of 24% in precision on same-speaker decision when compared to different-speaker decision.

Additionally, we present a comparison of results obtained on the human validation versus speaker labels and the proposed speaker verification system of Section 3.3. These results are presented in the confusion matrix of Figure 3.8, where the top left matrix reports the confusion when comparing the PLDA system decisions with the profile labels and the top right matrix reports the confusion when comparing human decisions with the profile labels. The bottom confusion matrix reports the confusion when comparing decisions by humans to the PLDA system.

By examining the results of the confusion matrices, we conclude that human uncertainty in same-or-different speaker decisions does not match the uncertainty of the speaker verification system. We also

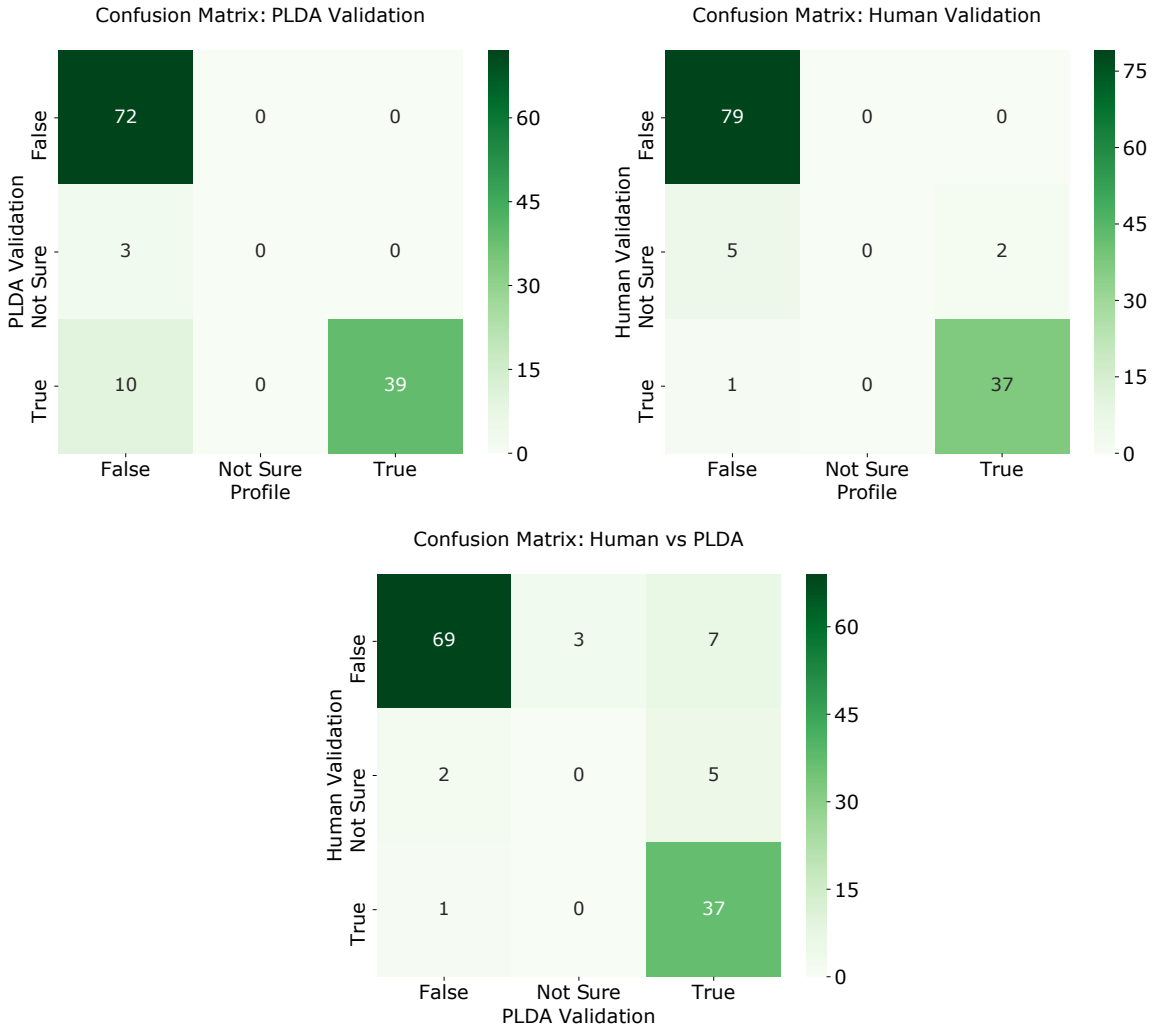

Figure 3.8: Confusion matrices for Human and PLDA speaker verifiers

note some decision mismatch of annotators when compared to the profile labels.

Additionally we present in Figure 3.9 a violin plot of individual contributors' answers versus the PLDA score, discriminating same-or-different speaker-labels. Our hypothesis was that a correlation between human and machine uncertainty in speaker verification tasks exist. This would present itself as an increasing progression of distribution's means for each Likert Score and a distribution of uncertainty (Likert Score 3) roughly centered around the Decision Threshold (DT), which in this case is $DT = 8.09$.

Observing Figure 3.9, we note that while same-speaker comparison scores are limited to values above the decision threshold, the same cannot be said about different-speaker comparisons, with PLDA scores populating the full scale. As such, no direct correlation between the PLDA Scores and Likert Scores is detected. Scores pertaining Likert Score 3 (uncertain) have mean value around 10.0, which is close to the decision threshold, but with a PLDA score minimum and maximum well within the same or different speaker score ranges. Additionally, in the cases where Likert Score responses were contrary

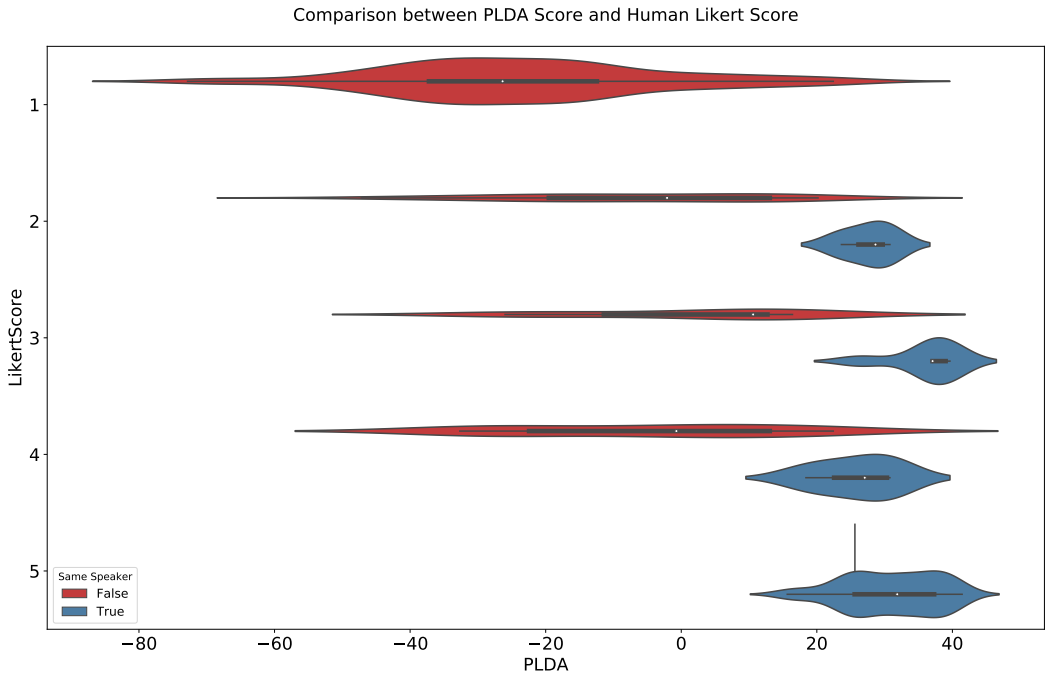

Figure 3.9: Violin plot of human vs system same-or-different speaker decisions.

to the speaker label, their PLDA scores were not centered around the DT.

Another hypothesis under study is whether different-speaker trial comparisons with a different gender would result in perfect predictions. Intuitively this would be the case, as speakers only have a single gender. Figure 3.10 confirms these results, as annotators were able to take full advantage of gender information, by accurately predicting that the trial comparison did not belong to the same speaker given that their genders were different.

Figure 3.10: Violin plot of human vs profile label.

3.5 Merging Speaker & Gender Verification

In [59], the authors explored the extracted i-vectors and x-vectors from a speaker-verification trained system to probe additional information. This information included gender, speaking rate and session related information such as word and phoneme recognition. The reported results for gender recognition indicate performances similar to state-of-the-art gender recognition systems.

As identified in previous experiments, gender plays a major role in speaker verification tasks. Indeed, as reported in Subsection 3.4.3, we found that humans were able to fully differentiate speakers given different genders. The same was concluded about the automatic speaker verification system, with PLDA score distributions for non-target trials with different genders having a much lower average score when compared to trials within the same gender.

The motivation behind this section is to confirm speaker embeddings contain sufficient information regarding gender and utilize embeddings to develop a gender validation system which could be used for crowdsourcing. This is worthwhile in the context of speaker profiling, as a sole feature vector, the embedding that is used for speaker verification, would be able to convey additional biometric information relevant for the construction and verification of said profile. This allows for reduced computational expenses, as a single feature extraction pipeline would suffice to extract all relevant information.

3.5.1 Experiments

To compare the performance of the model that predicts the gender from the speaker-trained embedding with the dedicated gender recognition model of section 3.3, experiments were conducted on several crowdsourced databases. All models were implemented and trained in Python using the PyTorch deep learning framework.

The gender extraction model from the embedding followed the architecture proposed in [59]. The model is an MLP with a single hidden layer and a ReLU activation for the first layer and a sigmoid activation for the output layer. The hidden layer size was fixed at 500. Similar to the end-to-end architecture, binary cross entropy loss was used together with Adam [81] as the optimizer, with a learning rate of 0.001. Two separate models were trained using extracted i-vectors and x-vectors using *VoxCeleb2* dev as the training dataset and *Voxceleb2* $test$ as the development set.

3.5.2 Results

This section reports the results obtained on DefinedCrowd and Common Voice speech data collections and includes Precision, F1-score and Recall for the i-vector, x-vector and end-to-end models. The best results for each metric and dataset is marked in bold, and are presented in Table 3.7.

		Male			Female		
Dataset	Architecure	Precision	Recall	F1 Score	Precision	Recall	F1 Score
CV_EN	i-vector	**0.98**	0.91	0.95	0.66	0.89	0.76
	x-vector	**0.98**	**0.94**	**0.96**	**0.72**	**0.90**	**0.80**
	end-to-end	0.97	**0.94**	0.95	0.71	0.83	0.76
CV_DE	i-vector	**0.98**	**0.96**	**0.97**	**0.80**	**0.90**	**0.85**
	x-vector	**0.98**	0.94	0.96	0.72	**0.90**	0.80
	end-to-end	**0.98**	**0.96**	**0.97**	**0.80**	0.86	0.83
DC_EN	i-vector	0.93	0.96	0.94	**0.97**	0.94	0.95
	x-vector	**0.94**	**0.97**	**0.95**	**0.97**	**0.95**	**0.96**
	end-to-end	0.90	**0.97**	0.93	**0.97**	0.92	0.95
DC_HE	i-vector	**0.99**	**0.98**	**0.99**	0.97	**0.99**	**0.98**
	x-vector	**0.99**	**0.98**	0.98	**0.98**	**0.99**	**0.98**
	end-to-end	**0.99**	0.97	0.98	0.97	**0.99**	**0.98**

Table 3.7: Results obtained on Crowdsourced datasets

It can be observed that the performance obtained using the speaker embeddings as input is comparable to the end-to-end model, with the added benefit that the model is much simpler, an MLP. In fact, the end-to-end model failed to outperform the embedding-based models on the majority of metrics, something we believe is due to the nature of the embedding extraction, which is able to convey information related to the full embedding, unlike the end-to-end model, which is restricted to exactly 2 seconds

of the utterance. This means utterances with duration lower than 2 seconds are padded with zeros before being fed to the network, and utterances longer than 2 seconds are cropped, possibly discarding relevant information pertaining gender. As exposed in Section 3.3, the Common Voice gender labels are not fully verified, which is why its results, namely on the female class are not comparable to the obtained performance on DefinedCrowd datasets.

3.6 Discussion

In this chapter we presented a profile validation task in the context of crowdsourced speech data collections quality control. Our proposed profile validation system includes a speaker verification module and a gender verification module, which are pretrained in out-of-domain open-source datasets.

A manual validation step was conducted prior to experiments in order to establish a test set that we could extract performance metrics from. Besides that, it also provided an opportunity to quantify fraud in DefinedCrowd crowdsourced speech data collections. While the percentage of fraud in our experiments may be considered minor (we reported less than 1% of utterances being fradulent), it cannot be ignored if we take into account the large volume of data generated using crowdsourcing methods. As such, these results motivate the widespread use of automatic demographic control methods in this context.

Noting the various combinations of different languages and conditions that occur during data gathering, our proposed speaker verification module is adapted to each collection automatically. This ensures that the same threshold can be applied to all collections without significantly jeopardising the system's performance. Evaluation results for the speaker verification module on crowdsourced datasets indicate an EER with or without score normalisation within the values of other speaker verification benchmarks on similar settings. Additionally, thresholds on all datasets shift to a value which facilitates the deployment of our proposed speaker verification system to other unseen datasets. Whilst EER results on the score normalised experiments do not show any significant changes, we note significant performance variations on the Detection Error Tradeoff (DET) curve. Namely, we found that with score normalisation, the DET curve suffers from large performance losses in terms of Miss probability when the False Alarm probability is minimized. We believe this is due to having a very small cohort list which we also equate to the number of comparisons ($N_c = N_t = 50$) when compared to typical cohort lists of sizes, which can be larger than 1,000 utterances, using at least 200 comparisons. This means the resulting statistics are calculated on a smaller set, that fails to represent all the variability of the target dataset. As such, larger in-domain datasets must be obtained to improve results, especially if fine-tuning of the system is needed, with regards to rejection and acceptance rates.

We also used the obtained scores from the speaker verification module to study the correlation between calculated scores and human perception (using a Likert Scale) in these tasks. Although we

confirmed humans take full advantage of gender when doing their predictions, being able to easily distinguish speakers if they had different genders, we failed to establish a correspondence between PLDA scores and human responses. We believe this result is partially due to the statistical insignificance of the sample size, making us unable to draw further conclusions regarding this subject. Another explanation for these results is the inability, despite our best efforts, to fully remove bias from naive human speakers. Human contextual and confirmation bias are known to degrade results, something that automatic speaker verification systems do not possess, and as such are unable to model using scoring.

Our gender verification module takes advantage of the information stored inside the speaker-trained embedding to also predict gender labels. With a simple MLP receiving as input the embedding, we were able to obtain results comparable to a dedicated end-to-end model. The large mismatch between the predictions and labels of the CommonVoice dataset indicated that this dataset contained incorrectly labeled genders, which was confirmed with a manual validation step.

4

Automatic Prediction of Breathing Patterns

Contents

4.1 Introduction

In Chapter 1, we discussed how the detection of speaker profile information can bring value to a myriad of applications, including Human–computer interaction, identity verification, collection of speech data and medicine. In this chapter, the second use case is addressed, medicine.

The production of speech is highly dependent on organs that are shared with the respiratory system: the lungs and the diaphragm are responsible for the pressure production required for speech; the upper vocal tract (which includes the nose, mouth, pharynx and larynx) is responsible for producing speech [90]. As such, human respiratory and speech parameters provide important cues to physicians and first-responders in determining a wide range of cardiac and respiratory diseases [91] [92] or to evaluate cognitive and neurological health [93] [94]. Furthermore, information extracted from breathing patterns during speech can be used to assist speech therapists in identifying speech impediments resulting from unfavourable respiratory planning [95]. Breathing monitoring in this context is often conducted using wearable sensors, namely, face masks and/or respiratory belts [96]. The installation of these sensors requires the presence of trained medical assistants and is frequently time-consuming, negating their usefulness in emergency situations, or when the patient cannot be physically reached. A typical example of the latter scenario occurs during medical virtual online consultations, with the patient at home, where breathing information could be of use for diagnosis or monitoring. As such, automated methods based on recorded speech alone that are able to predict breathing events and parameters such as breathing rate and tidal volume may be of substantial value.

Considering breathing patterns and related parameters provide important biomarkers regarding a speaker's health, its use may also be of value in the context of heath-related speech data collection and speaker verification in general. For instance, speech can be probed for breathing information to determine if a healthy speaker is attempting to impersonate a enrolled speaker with a medical condition, or vice versa. This can complement the work conducted in Chapter 3, by providing an additional fraud detection layer, and/or enforce class distribution in health-related speech data collections.

Previous studies on the topic of breathing detection from speech have focused mainly on automatic recognition of breathing patterns and events directly from a processed signal (e.g. [97], [98]). In [99], the authors studied the automatic detection of the breathing signal using Deep Neural Networks (DNNs). They reported a correlation coefficient between the predicted signal and the original one of 0.47, with error rates pertaining breathing rate of 4.3%.

In this chapter, the automatic prediction of breathing patterns from speech is explored in the context of the INTERSPEECH 2020 Computational Paralinguistics Challenge (ComParE): Breathing Sub-Challenge [100]. Additionally, applications in medicine, such the estimation of the breathing rate, are explored. The chapter begins with an introduction to the topic, paired with a motivation and an overview of previous methods (Section 4.1). In Section 4.3, the Amplitude Modulation (AM)/Frequency Modula-

tion (FM) decomposition method of speech signals is presented, together with a brief explanation of its use. Section 4.4 dives into the breathing pattern prediction problem, providing the experimental setup and results. Similarly, in Section 4.5, the experimental setup and results are presented for the breathing rate prediction problem. Finally, in Section 4.6, the results are discussed.

4.2 Dataset

The experiments for the Breathing Sub-challenge are conducted using a subset of the UCL Speech Breath Monitoring (UCL-SBM) database. The dataset includes speech recorded from a head-mounted condenser microphone and normalized linear voltage readings from two piezoelectric respiratory belts that respond to changes to the thoracic circumference.

All speech recordings were spontaneous, as reading tasks may introduce some bias, forcing stops that do not necessarily coincide with the breathing rhythm. The recordings were produced by native English speakers of ages ranging from 18 to 55 years old. To the best of our knowledge, all speakers were healthy. The data set contains 49 sessions, each 4 minutes in length. The corpus is split into training, development and test sets (17, 16, and 16 sessions, respectively).

An analysis of the belt signals in these datasets shows considerable variability, as illustrated in Figure 4.1: while most of the signals in the training set have quite regular breath patterns, this was not observed in almost half of the signals in the development set. This was the motivation for also experimenting with a reduced development set, $dev2$, from which 7 sessions were excluded, since the training material did not include sufficient examples of such irregular patterns (only 2 out of 17 sessions). The objective exclusion criteria was based in experimental results, as explained in the next Section.

In order to emulate the video-call consultation with a physician, the provided challenge dataset was augmented. The augmentation consists in passing the original, down-sampled speech signal by an ITU-T G.723.1 dual rate speech coder and decoder [101]. The G.723.1 audio codec, part of the ITU-T recommendation H.324, is a Code-Excited Linear Prediction Coder widely used in Voice over IP (VoIP) applications. It compresses voice audio in 30 ms frames and operates with a sampling frequency of 8 kHz/16-bit. In this implementation in particular, MPC-MLQ (Multi-pulse Coding) mode is used, operating at 6.3 kb/s. After the decoding, the signal is up-sampled back to 16 kHz and is used in training alongside with the original data. This augmentation results in the doubling of the training and development data (dev_{aug}).

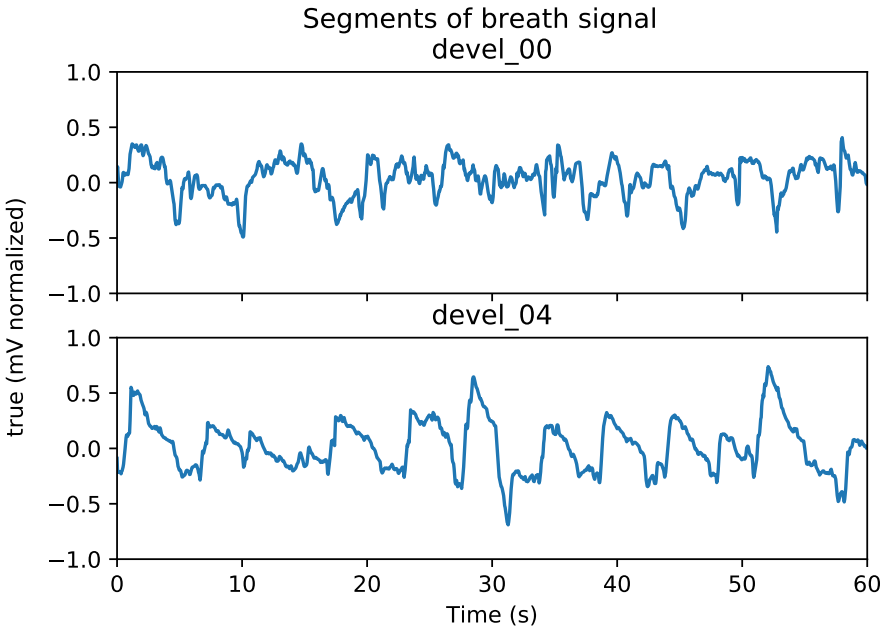

Figure 4.1: Segments of breath signals from sessions *00* and *04*.

4.3 AM-FM decomposition

The rationale behind the AM-FM decomposition is that speech is generated by a source (FM component containing speaker information), which is modulated by the vocal tract (AM component containing the message) [102]. Previous work [103] conducting AM-FM decomposition has shown only a small loss in ASR performance (4.8% WER absolute increase) when using the FM component in an HMM-GMM system. This contrasted with the WER obtained using only the FM component (43.8% absolute increase).

The spectrograms of Figure 4.2 illustrate the contents of the two components in the presence of a breathing event. The FM carrier signal clearly shows a breath signal between two words whose voicing patterns are visible. The AM signal containing the linguistic information exhibits longer pauses between the corresponding words. This was the motivation for a set of experiments on predicting breath signals from the raw time wave representation of the envelope, the carrier, or combinations of these with and without the original signal.

The AM-FM decomposition is conducted using a frequency domain linear prediction (FDLP) approach. FDLP proposes to model the speech in critical bands as a modulated signal with the AM component obtained using Hilbert envelope estimate and the FM component obtained from the Hilbert carrier. In the implementation followed [104] [1], the input speech was decomposed into 32 conventional quadrature mirror filter (QMF) bands with an analysis window of 1 second. FDLP was then applied on

[1] https://github.com/iiscleap/SignalAnalysisUsingAm-FM

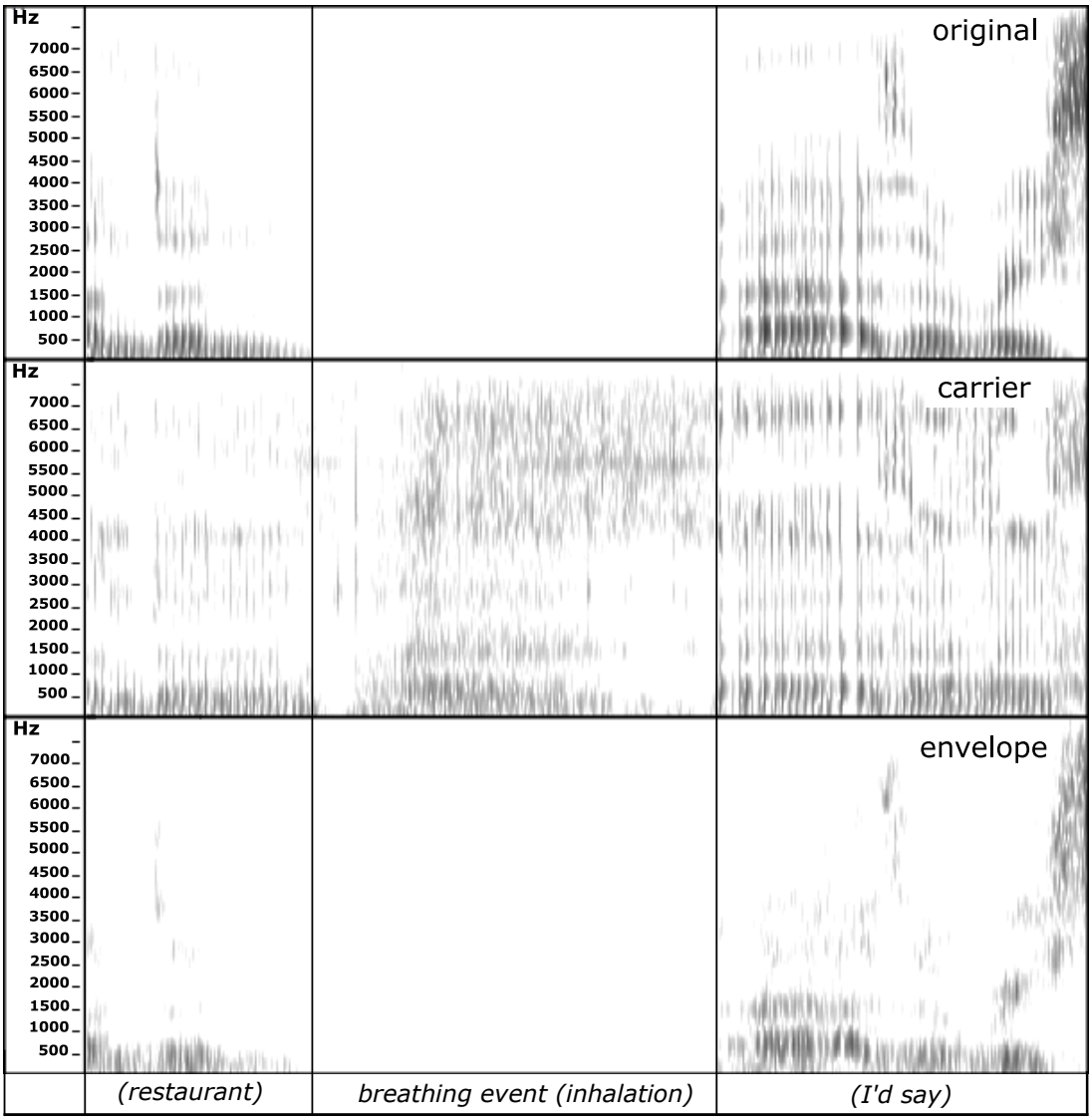

Figure 4.2: Spectrograms of speech signal showing a breathing event in between two words.

each band to model the sub-band temporal envelopes (AM components). The LP residual represents the FM in the sub-band signal. The reconstruction of the signal from the QMF bands was done by reversing the above-mentioned steps. The resulting envelope signal contains the re-synthesized signal with the intact message, but with whispered speech. With the carrier information alone, the synthesized signal sounds message-less, but with identifiable speaker cues, namely pitch and voice quality features, such as creakiness.

4.4 Breathing Pattern Prediction

4.4.1 Experimental Setup

The official baseline results indicate traditional methods such as feature extraction paired with SVM failed to produce competitive results when compared to the End-to-End Deep Sequence Modelling. As such, the official provided end-to-end baseline architecture was used as a base for experiments[2]. This architecture follows typical sequence labelling models by combining a CNN for character-level representation with an RNN (in this case an LSTM) for obtaining context. The output of these layers is then fed to a dense layer for final prediction.

Input data during training consists of 2 seconds of speech recordings sampled at 16 kHz, resulting in 32000-dimensional input vector. The network is composed of three one-dimensional convolutional layers with number of filter 64-128-256 and with a kernel size of 8-6-6. Each convolutional layer is followed by a max-pooling layer that undersamples at a stride of 10-8-8. The output is then fed to the recurrent level of the network, which consists of two stacked LSTMs with 256 hidden units. This model provides a sequence of hidden states, each of which is passed through a linear layer to provide the breath belt signal prediction. The training loss used is the Pearson correlation coefficient r, calculated between the true and predicted belt signals. For this, the true and predicted signals are flattened to calculate the training loss. All experiments were conducted using TensorFLow with a learning rate of 0.002 for the Adam optimiser, with models being trained for 100 epochs.

Several experiments were conducted replacing the self-learning character-level representation with conventional extracted features. One of the motivations behind this replacement was to understand if the low amount of training data was a limiting factor in the feature extraction step.

We trained a model using a 48-dimensional Constant-Q filter bank, calculated every 10ms with a frame-length of 25ms. The Constant-Q transform logarithmically spaces in frequency its filters, which mirrors the human auditory system, as the spectral resolution of the low frequencies is higher than the high frequencies, and the temporal resolution at higher frequencies is better than at lower frequencies. Additionally, MFCCs are extracted every 10ms with a frame-length of 25ms, mean-normalised over a sliding window of up to 3 seconds. We present results using extracted 40-dimensional MFCCs as a replacement to the CNN. We train models with the full MFCCs feature vector, and with the top and bottom 24 coefficients. This allows us to understand where the most relevant information pertaining to breath lies. Additional models are trained by replicating the frame-level of the x-vector DNN architecture, and by adding 2 convolutional layers (kernel size of 32, with a stride of (8,4)) together with max-pooling (pool filter and size of (1,3)) for obtaining contextual information from the MFCCs.

Speaker embeddings are known to encode information about the speaking rate [59]. As such, and

[2]https://github.com/glam-imperial/ComParE2020-Breathing-End2End

in an attempt to model speaker dependence, frame-level x-vectors were extracted every 25ms using Kaldi [82] and appended to the output of the last convolutional layer, before the context-level of the network (LSTM).

We also experimented training models with different losses, namely Mean and Root Mean Squared Error (MSE, RMSE) and Lin's Concordance Correlation Coefficient, using the baseline end to end architecture.

Another approach we experimented was a Bidirectional LSTM. This allows us to model respiratory planning, which by intuition takes into account not only how long was the last breath (past information), but also when we plan to stop talking in order to breath in. In this variant, the depth-concatenated forward and backward outputs are fed to the dense layer for prediction.

4.4.2 Results

Considering the fact that overall, our development set results were much lower when compared to those obtained for the training set and those that were reported in the official baseline for the test set led us to inspect the individual results of the Pearson correlation coefficient r for each session of the development set as shown in Table 4.1.

Session	00	01	02	**03**	**04**	05	**06**	**07**
r	.000	.610	.566	.768	.833	.668	.837	.781
r_{aug}	.005	.613	.569	.777	.834	.655	.845	.770
Session	08	**09**	**10**	**11**	**12**	13	**14**	15
r	.262	.753	.760	.820	.889	.291	.784	.321
r_{aug}	.262	.788	.734	.822	.887	.263	.794	.327

Table 4.1: Pearson correlation coefficient using our best reported system on the original development set.

The top line of Table 4.1 shows results on this set and the bottom line on the augmented set. The sessions showing less regular patterns corresponded to much lower values of r, and were therefore excluded from the reduced development set, $dev2$ (marked in bold). As expected, average results are considerably higher for this dataset (absolute improvement of .2). Additional models were also trained, combining $train$ with dev and $dev2$. We note that removing the 2 sessions that were reported to have irregular breathing patterns from the $train$ set did not change results. As such, all of the models for which we present results have been trained on the full $train$ set. Our best models were submitted to the challenge's platform, which contained unseen data, $test$.

4.4.2.A Results with different feature sets

Table 4.2 presents experimental results for several feature sets and losses. We note that all approaches failed to surpass the performance of the baseline on the development set. Results on the $train$ set indicate approaches using well-known handcrafted features such as the MFCCs benefit training by having a simpler model. However, this result does not extend to the Constant Q Filter-bank. This is likely due to the constant shifts in frequency as a result of variations in the spacing of the harmonics.

	r		
Architecture	$train$	dev	$dev2$
End2End Baseline	0.89	0.507	0.769
Log Mel Energy	0.87	0.383	0.577
40-dim MFCC + LSTM	0.92	0.484	-
48-dim Constant-Q Filters + LSTM	0.49	0.161	-
24-dim MFCC + LSTM	0.91	0.495	0.736
Upper 24-dim MFCC + LSTM	0.99	0.315	-
24-dim MFCC + TDNN + LSTM	0.94	0.220	-
24-dim MFCC + 2x Conv + LSTM	0.89	0.483	0.764
x-Vector Baseline	0.89	0.495	0.757
RMSE Loss Baseline	0.73	0.480	-
MSE Loss Baseline	0.73	0.480	-
CCC Loss Baseline	0.95	0.500	0.768

Table 4.2: Experimental Results for different feature sets.

Looking at the results using MFCCs and Log Mel Energy, we note that the feature set containing the full range of frequencies yielded a better performance than the approaches using only portions of the frequency spectrum. Furthermore, the better results on the lower portion of the spectrum show that more relevant information regarding breath is stored on this frequency range. Results for the TDNN point out the lack of training data needed to train such a network.

The results using x-vectors report a minor performance degradation in $dev2$, indicating that the information present in speaker embeddings does not assist in prediction.

Results using RMSE and MSE show that loss functions based on the difference between the estimated values and the actual value do not improve performance. This was expected, as the performance is evaluated using Pearson Correlation Coefficient r, which is why the CCC loss has comparable results to the baseline on $dev2$.

4.4.2.B Results with Augmentation

A summary of the results obtained for the model with the best development performance of the 100 epochs of training is presented in Table 4.3. Results on dev did not indicate any improvement of the BiLSTM approach when compared to the baseline. The results on the augmented dataset do not show consistent differences in performance when compared to the original dataset. The results on the VoIP-modified sessions are presented in Table 4.1 (bottom row), showing no notable differences either, which indicates that there is no information loss regarding breathing events when passing speech signals through the G.723.1 audio codec.

	r		
	dev	$dev2$	$test$
Baseline Approaches - Challenge dataset			
openSMILE [105]	.244	-	-
openXBOW [106]	.226	-	-
End2End	.507	.769	.731
Proposed Approaches - Challenge Dataset			
End2End FM	.442	.657	-
End2End AM	.490	.722	-
BiLSTM Original	.507	.787	.720
BiLSTM FM	.441	.696	-
BiLSTM AM	.500	.742	-
End2End Org+AM+FM	.476	.749	-
Proposed Approaches - Augmented Dataset			
	dev_{aug}	$dev2_{aug}$	$test$
End2End Original	.509	.784	-
End2End FM	.424	.621	-
End2End AM	.482	.740	-
BiLSTM Original	.514	.767	.728
BiLSTM FM	.432	.657	-
BiLSTM AM	.515	.755	-
End2End Org+AM+FM	.500	.742	-
BiLSTM Org+AM+FM	.506	.765	-
BiLSTM AM+FM	.488	.744	-

Table 4.3: Experimental Results for all systems on the Breathing Sub-challenge.

An example of the performance of the systems is illustrated in Figure 4.3, with the reference breath signal in blue and predicted signal in orange. The bottom part of Figure 4.3 illustrates the system's ability to correctly predict breathing patterns in VoIP conditions. The true breathing signal is compared with the one predicted from a signal obtained by passing a session of the UCL dataset through a real

VoIP scenario. The audio recording is transmitted over-the-air using a mobile phone and recorded using Skype platform, which uses the SILK [107] audio compression and codec. The resulting audio is down-sampled and edited to conform with the original signal's duration.

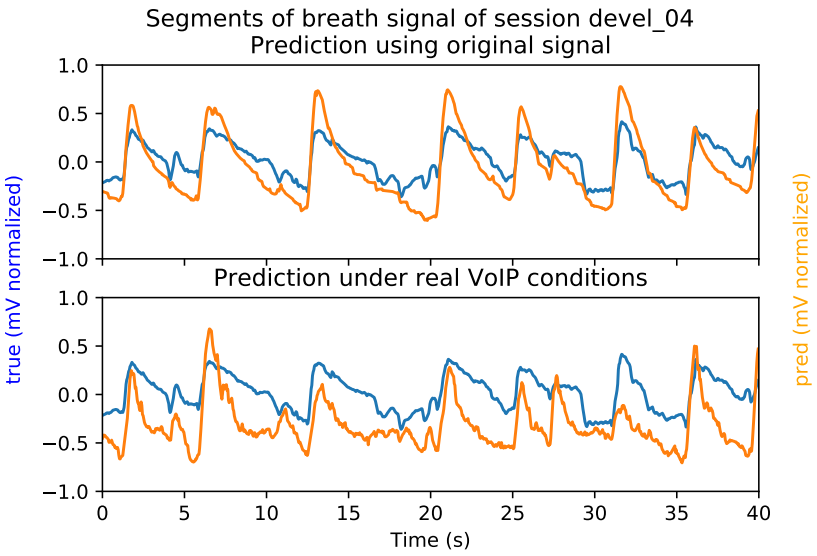

Figure 4.3: Segments of breath signals from session *devel_04*.

We note that the prediction with the VoIP signal is able to detect the time-instances of all breathing events correctly.

4.4.2.C Results with AM and FM components

Compared with the results of the original signal, as seen in Table 4.3, no improvements were detected when using only the carrier or the envelope signal (the performance gain of the BiLSTM AM model when compared to the BiLSTM Original is residual). Furthermore, all experiments indicate the performance using only the AM signal yielded the best results when compared to the FM signal. The combination of the AM and FM components, or even when including the original speech signal, failed to outperform the BiLSTM system with the original audio, and the challenge's baseline.

4.5 Breathing Rate Estimation

Breathing events are characterized in the breathing signal as a peak value (local maxima) indicating maximum intake of air during inspiration, as shown in Figure 4.4. Previous attempts to detect these events typically include the detection of zero-crossings and thresholding of the signal (using its first and second derivatives) [97] [108]. In this work, we used a slightly different approach: Considering breath is a quasi-periodic signal (the typical respiratory rate for a healthy adult at rest is 12–18 breaths

per minute [109]), the resulting cyclic characteristics of the auto-correlation will be equal to the original signal. As such, the peaks of the auto-correlation are found and the average time differences between them report the short period of the signal, which roughly corresponds to the periodicity of breath. This period will then be used as the stride of a window that will detect the local maxima of the original signal.

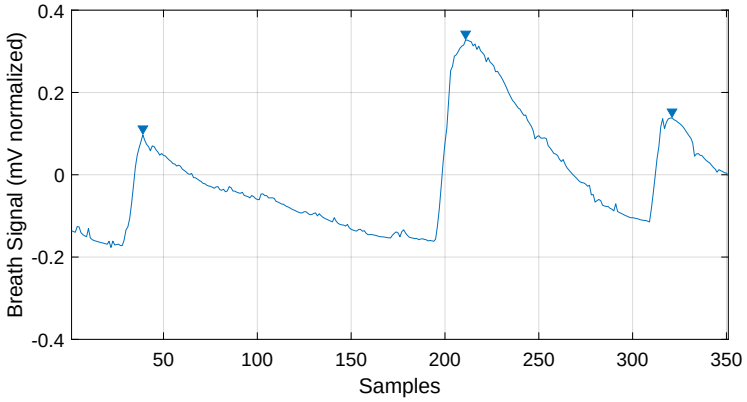

Figure 4.4: Sample of a breathing signal.

4.5.1 Experimental Setup

The *findpeaks* detection algorithm of *MATLAB ver. R2019a* was used to detect both the peaks in the auto-correlation and the breath signal. The obtained short period of the auto-correlation was then used for minimum peak separation in the breath signal. A peak detection threshold of 0.1 mV was added to filter out noise. The corresponding breathing rate is then calculated by dividing the number of detected breath events by the duration of the signal in seconds. An example of this detection is illustrated in Figure 4.4.

The behaviour of the breathing patterns of the AM and FM components was compared to a breathing event detection algorithm based on an ASR system [110]. This system was trained on the English HUB-4 dataset using Kaldi [82]. The acoustic model is a TDNN and the language model is trained on a mix of broadcast transcriptions and web news corpora. An example of the output is shown in Figure 4.5, with the ASR system in black. This segment was chosen in particular as it shows the limitations of the use of the speaker noise event detection for breathing detection. We note that by using the generic labels the system is unable to differentiate between voiced exhalation and voiced inhalation and that it does not detect unvoiced inhalation. Furthermore, the system trained with the FM component is unable to detect these voiced exhalations.

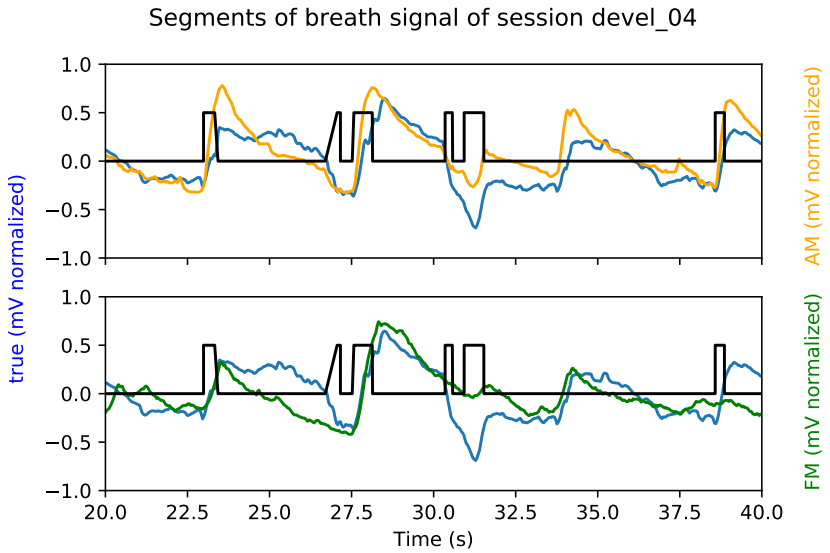

Figure 4.5: Segments of true and predicted breath signals with breathing detection algorithm using ASR.

4.5.2 Results

The breathing rate estimation results are shown in Figure 4.6. Considering no actual breathing rates were provided for each session, the results obtained using the predicted signals of our best model in $dev2_{aug}$ are compared against the breathing rate estimations of the true (label) signals. The breathing rates for the test set are also provided.

We note that the range of values of breathing rate for the labels is much higher than the ones estimated using the predicted breath signal. The presence of outliers and the overall distributions of breathing rates in the label signals indicate some of the sessions have noisy or otherwise disrupted breath signals. Indeed, the identified outliers in the label breath signals belong to the same sessions identified in Section 4.2 as having irregular breathing patterns. While this had already been shown for the development set, the data presented here shows that some sessions of the training data also share the same problem. After manually inspecting these sessions, we report similar breath patterns as the irregular ones detected in the development set.

Rates of under 0.2 were reported in [99] [108], for conversational speech, which is in agreement with the results obtained from the predicted signals. A Mean Absolute Error of 0.0664 and 0.1232 was obtained on training and dev sets, respectively.

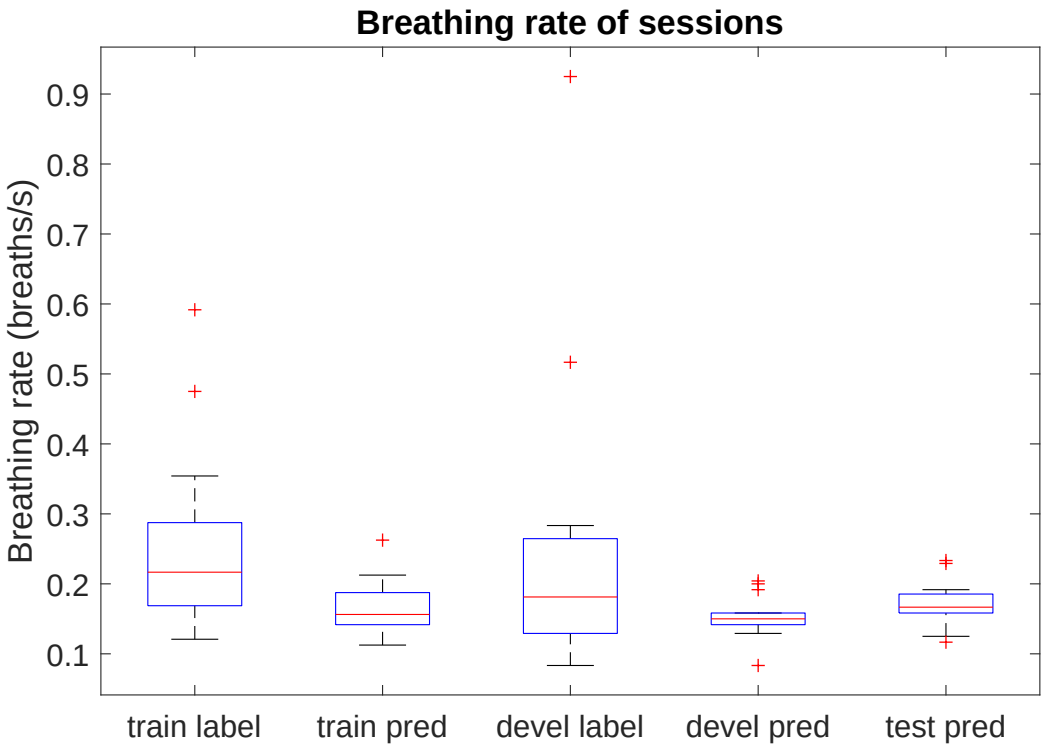

Figure 4.6: Average breathing rates (breaths per second) for the different datasets.

4.6 Discussion

In this chapter we tackled the problem of predicting the breathing signal from speech signals and adapted the original problem to also provide us with the breathing rate, which can provide a more interpretable marker for health or body conditioning.

Our exploratory analysis of the dataset showed that several of the development signals were irregular. By removing these from the evaluation, performance results improved on all experiments and exposed more promising approaches. Additionally, we augmented the challenge dataset by introducing the same files passed through a VoIP emulator. The results on the augmented dataset showed improved performance when compared to the systems trained on the challenge dataset alone. Our best approach on the development set was the BiLSTM, showing that future and past information is relevant in the context of breathing signal predictions.

The experiments conducted using the AM and FM signals show that AM signals present better results when compared to the FM signals. This can be explained by the fact that the AM component retains most of the information relevant for detecting breathing patterns, which is the message. The performance degradation on the AM component, when compared to the original signal, can be explained by the fact that relevant information is carried by the Hilbert FM carrier instead, such as voiced breathing events,

that appear on the envelope as silence. The combination of the AM/FM signals failed to show any improvements. This indicates that the availability of the various representations during training does not improve results.

5

Conclusions and Future Work

Contents

5.1 Conclusions

In this thesis we explored how extracting voice profile metadata from the speech signal can assist in the development of machine learning models. Namely, we explored two distinct use cases: one on metadata validation in a crowdsourcing setting and another in the field of medicine. This thesis demonstrates that the use of extracted speaker embeddings can provide the needed crowdsourcing submission control by providing a single-dimensional vector capable of verifying speakers and their gender. Additionally, the thesis also shows that voice pattern recognition techniques can be used to predict breathing patterns and breathing-related parameters.

In Chapter 3, we presented a speaker verification task in the context of crowdsourced speech data collections quality control. Noting the various combinations of different languages and conditions that occur during data collection, our proposed speaker verification system is pretrained on an out-of-domain dataset and adapted to each collection automatically. This ensures that the same threshold can be applied to all collections without jeopardising the system's performance. Evaluation results on crowdsourced datasets indicate an EER lower than 4% with or without score normalisation, which is on par with other speaker verification benchmarks on similar settings. Additionally, thresholds on all datasets change to values close to each other, which facilitates the deployment of our proposed speaker verification system to other unseen datasets. Whilst EER results on the score normalised experiments do not show any significant changes, we note a loss of performance when calibrating the system to reduce false rejections. When attempting to reduce false acceptances however, we noted an increase in performance on two of the datasets. Additionally, we compared the performance of this system with human same-or-different speaker decisions in a bid to understand if machine uncertainty is related to human uncertainty. Results show that there is no direct correlation between the two systems in regards to uncertainty and confusion-related errors.

Also in Chapter 3, we presented a gender recognition system based on speakers embeddings, showing that the predictions obtained using a simple MLP yielded similar results to that of a dedicated DNN gender recognizer. As such, we were able to re-utilize the pipeline already in place for speaker recognition to also provide gender-related information.

The conducted experiments on Chapter 3 showed that the proposed speaker and gender verification system is able to detect fraud and incorrect speaker labels accurately in an online setting. This allows for the the necessary corrective measures early on, reducing the costs of the dataset and flagging crowdmembers with malicious intent. While speaker fraud detection using our proposed methodology remains a proof-of-concept, our system has already been successfully used to validate gender labels in multiple datasets.

In Chapter 4, we analyzed and automatically predicted breathing patterns from speech, using signals extracted from respiratory belts as ground truth. Moreover, we studied the applicability of the AM-FM

decomposition of speech to this same task. We found that while the decomposed components did not surpass the performance of the original signal, our experiments support the hypothesis that the breathing rate is dependent on the message, since, individually, the results obtained with the AM component were able to outperform those obtained with just the FM component. In order to simulate the conditions of medical consultations over the internet, the original dataset was augmented by passing it through a VoIP coder-decoder. Overall, our experiments also indicate that future information modelled by the Bidirectional LSTM improves results.

5.2 Publications

Part of the work developed in this thesis contributed to:

Analyzing Breath Signals for the Interspeech 2020 ComParE Challenge

***Mendonça, J.**, Teixeira, F., Trancoso, I. and Abad, A. - Proceedings of Interspeech 2020, 2077-2081*

Towards Online Fraud Detection for Speech Data Collections

***Mendonça, J.**, Correia, R., Trancoso, I. and Freitas, J. - Submitted to ICASSP 2021*

5.3 Future Work

One of the main limitations of this work in regards to performance evalutation on a crowdsourcing environment was the lack of validated data. This was partially circumvented by manually validating a small number of for-profit datasets from DefinedCrowd, and datasets from the open-source Common Voice project. Our results suggest that performing a score normalisation step, on each dataset individually, resulted in the convergence of the decision threshold to a smaller range of values. As a trade-off to this threshold convergence, a perfomance loss in terms of minDCF was detected, as reported in the DET curve. This is due to our proposed normalization step using a small sample of the evaluation data as cohort set, set to 50, while typical cohort lists have sizes several times larger, something that may not be available in the earlier stages of data collecting in a production setting. A solution to this would be to take advantage of previously validated datasets and use them as cohorts for this validation. In [86], authors reported an absolute improvement of .2 in terms of minDCF when using a cohort list which matches the language and channel conditions of the evaluation data, when compared to a cohort list with different languages and different channel conditions.

For future work, one could expand experiments to include more datasets with different languages, channel conditions and task domains (e.g. free speech). Additionally, we plan on exploring additional metrics than can give insight on the behaviour of users during tasks. Metadata such as time of day can be a good indication that multiple users are sharing the same account. The use of unsupervised

clustering [111], besides also solving the problem of detecting multiple speakers using a single account, could also help detecting another unexplored fraud, which is a speaker using multiple accounts.

Another interesting future research direction is to probe additional information stored in speaker embeddings. Speaker embeddings are known to store cues related to lexical content (such as keywords present in the utterance) and meta information such as utterance length [59]. Additional biometric parameters such as age (which was not explored), paired with the previously mentioned information might add value to validation in crowdsourcing.

Our submission to the ComParE 2020 Breathing Sub-Challenge failed to surpass baseline results. As such, the main goal is to experiment augmentation and decomposition using better performing models. As an example, the winning submission comprised of an ensemble of end-to-end Deep Models using fusion of deep embeddings and decision level fusion schemes [112]. Furthermore, a future goal pertaining the breathing pattern estimation problem is to explore additional parameters that can be extracted from breathing patterns such as volumetric information (e.g. tidal volume).

Additionally, given how breathing provides important markers to several medical conditions, such as cardiac, respiratory and neurological diseases, we plan to explore speech derived breathing patterns for assisting in the automatic detection of these conditions.

Bibliography

[1] K. A. Lee, H. Yamamoto, K. Okabe, Q. Wang, L. Guo, T. Koshinaka, J. Zhang, and K. Shinoda, "NEC-TT system for mixed-bandwidth and multi-domain speaker recognition," *Computer Speech & Language*, vol. 61, p. 101033, 2020.

[2] J. Freitas, J. Ribeiro, D. Baldwijns, S. Oliveira, and D. Braga, "Machine learning powered data platform for high-quality speech and NLP workflows," in *Proc. Interspeech 2018*, 2018, pp. 1962–1963.

[3] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5329–5333.

[4] AppDynamics, "The Future of Voice Technology In the Enterprise," AppDynamics, Tech. Rep., 2018.

[5] S. G. Preeti Wadhwani, "Intelligent Virtual Assistant (IVA)," Global Market Insights, Tech. Rep., 2018.

[6] F. Metze, J. Ajmera, R. Englert, U. Bub, F. Burkhardt, J. Stegmann, C. Muller, R. Huber, B. Andrassy, J. G. Bauer, and B. Littel, "Comparison of four approaches to age and gender recognition for telephone applications," in *2007 IEEE International Conference on Acoustics, Speech and Signal Processing - ICASSP '07*, vol. 4, 2007, pp. IV–1089–IV–1092.

[7] B. Andreeva, G. Demenko, B. Möbius, F. Zimmerer, J. Jügler, and M. Oleskowicz-Popiel, "Differences of pitch profiles in germanic and slavic languages," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.

[8] J. E. Flege, "Second language speech learning: Theory, findings, and problems," *Speech perception and linguistic experience: Issues in cross-language research*, vol. 92, pp. 233–277, 1995.

[9] J. Lopes, I. Trancoso, and A. Abad, "A nativeness classifier for ted talks," in *2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2011, pp. 5672–5675.

[10] J. A. Logemann, H. B. Fisher, B. Boshes, and E. R. Blonsky, “Frequency and cooccurrence of vocal tract dysfunctions in the speech of a large sample of parkinson patients,” *Journal of Speech and hearing Disorders*, vol. 43, no. 1, pp. 47–57, 1978.

[11] S. Alghowinem, R. Goecke, M. Wagner, J. Epps, M. Breakspear, G. Parker *et al.*, “From joyous to clinically depressed: Mood detection using spontaneous speech.” in *FLAIRS Conference*. Citeseer, 2012, pp. 141–146.

[12] R. Singh, *Profiling humans from their voice*. Springer, 2019.

[13] Y. Wen, R. Singh, and B. Raj, “Reconstructing faces from voices,” 2019.

[14] V. Peddinti, D. Povey, and S. Khudanpur, “A time delay neural network architecture for efficient modeling of long temporal contexts,” in *Sixteenth Annual Conference of the International Speech Communication Association*, 2015.

[15] M. Schuster and K. K. Paliwal, “Bidirectional recurrent neural networks,” *IEEE Transactions on Signal Processing*, vol. 45, no. 11, pp. 2673–2681, Nov. 1997.

[16] S. Hochreiter and J. Schmidhuber, “Long short-term memory,” *Neural Computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[17] P. Werbos, *Beyond Regression: New Tools for Prediction and Analysis in the Behavioral Sciences*. Harvard University, 1975.

[18] S. Ioffe and C. Szegedy, “Batch normalization: Accelerating deep network training by reducing internal covariate shift,” ser. Proceedings of Machine Learning Research, F. Bach and D. Blei, Eds., vol. 37. Lille, France: PMLR, 07–09 Jul 2015, pp. 448–456.

[19] J. Pickles, *An introduction to the physiology of hearing*. Brill, 2013.

[20] G. Fant, *Acoustic theory of speech production*. Walter de Gruyter, 1970, no. 2.

[21] S. Schötz, “Acoustic analysis of adult speaker age,” in *Speaker classification I*. Springer, 2007, pp. 88–107.

[22] M. Pronobis and M. Magimai.-Doss, “Analysis of f0 and cepstral features for robust automatic gender recognition,” Idiap, Tech. Rep., 2009.

[23] J. E. Atkinson and D. Erickson, “The function of strap muscles in speech: pitch lowering or jaw opening,” *The Journal of the Acoustical Society of America*, vol. 60, no. S1, pp. S65–S65, 1976.

[24] I. Guimarães and E. Abberton, “Health and voice quality in smokers: an exploratory investigation,” *Logopedics Phoniatrics Vocology*, vol. 30, no. 3-4, pp. 185–191, 2005.

[25] G. Fairbanks and W. Pronovost, “An experimental study of the pitch characteristics of the voice during the expression of emotion,” *Speech Monographs*, vol. 6, no. 1, pp. 87–104, 1939.

[26] E. D. Mysak, “Pitch and duration characteristics of older males,” *Journal of Speech and Hearing Research*, vol. 2, no. 1, pp. 46–54, 1959.

[27] L. G. KERSTA, “Voiceprint identification,” *Nature*, vol. 196, no. 4861, pp. 1253–1257, 1962.

[28] M. Shridhar, N. Mohankrishnan, and M. Baraniecki, “Text-independent speaker recognition using orthogonal linear prediction,” in *ICASSP ’81. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 6, March 1981, pp. 197–200.

[29] N. Fakotakis, A. Tsopanoglou, and G. Kokkinakis, “Text-independent speaker recognition based on vowel spotting,” in *1991 Sixth International Conference on Digital Processing of Signals in Communications*, Sep. 1991, pp. 272–277.

[30] M. S. Schmidt, “Identifying speakers with support vector networks,” in *Proceedings of the 28th Symposium on the Interface (INTERFACE-96)*, 1996.

[31] D. A. Reynolds, “A gaussian mixture modeling approach to text-independent speaker identification.” Ph.D. dissertation, 1993.

[32] N. Minematsu, M. Sekiguchi, and K. Hirose, “Automatic estimation of one’s age with his/her speech based upon acoustic modeling techniques of speakers,” in *2002 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1. IEEE, 2002, pp. I–137.

[33] C. Müller and F. Burkhardt, “Combining short-term cepstral and long-term pitch features for automatic recognition of speaker age,” in *Eighth Annual Conference of the International Speech Communication Association*, 2007.

[34] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, “Speaker verification using adapted gaussian mixture models,” *Digit. Signal Process.*, vol. 10, no. 1, pp. 19–41, Jan. 2000.

[35] P. Kenny, “Joint factor analysis of speaker and session variability: Theory and algorithms,” 2006.

[36] N. Dehak, P. J. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, “Front-end factor analysis for speaker verification,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, May 2011.

[37] S. Ranjan, G. Liu, and J. H. L. Hansen, “An i-vector plda based gender identification approach for severely distorted and multilingual darpa rats data,” in *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2015, pp. 331–337.

[38] M. H. Bahari, M. McLaren, H. V. hamme, and D. A. van Leeuwen, “Speaker age estimation using i-vectors,” *Engineering Applications of Artificial Intelligence*, vol. 34, pp. 99 – 108, 2014.

[39] D. Snyder, D. Garcia-Romero, D. Povey, and S. Khudanpur, “Deep neural network embeddings for text-independent speaker verification,” in *Proc. Interspeech 2017*, 2017, pp. 999–1003.

[40] S. H. Kabil, H. Muckenhirn, and M. Magimai.-Doss, “On learning to identify genders from raw speech signal using cnns,” in *Proc. Interspeech 2018*, 2018, pp. 287–291.

[41] D. Doukhan, J. Carrive, F. Vallet, A. Larcher, and S. Meignier, “An open-source speaker gender detection framework for monitoring gender equality,” in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5214–5218.

[42] R. Mammone, X. Zhang, and R. Ramachandran, “Robust speaker recognition: A feature-based approach,” *Signal Processing Magazine, IEEE*, vol. 13, p. 58, 10 1996.

[43] N. Dehak, P. Kenny, R. Dehak, O. Glembek, P. Dumouchel, L. Burget, V. Hubeika, and F. Castaldo, “Support vector machines and joint factor analysis for speaker verification,” in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, April 2009, pp. 4237–4240.

[44] P. Kenny, P. Ouellet, N. Dehak, V. Gupta, and P. Dumouchel, “A study of interspeaker variability in speaker verification,” *Audio, Speech, and Language Processing, IEEE Transactions on*, vol. 16, pp. 980 – 988, 08 2008.

[45] N. Dehak, “Discriminative and generative approaches for long- and short-term speaker characteristics modeling: application to speaker verification,” Ph.D. dissertation, 2009.

[46] P. Kenny, G. Boulianne, and P. Dumouchel, “Eigenvoice modeling with sparse training data,” *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 3, pp. 345–354, May 2005.

[47] P. Kenny, V. Gupta, T. Stafylakis, P. Ouellet, and M. J. Alam, “Deep neural networks for extracting baum-welch statistics for speaker recognition,” *Proc. Odyssey Speaker Lang. Recognit*, pp. 1–8, 01 2014.

[48] Y. Lei, N. Scheffer, L. Ferrer, and M. McLaren, “A novel scheme for speaker recognition using a phonetically-aware deep neural network,” in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2014, pp. 1695–1699.

[49] Y. Song, X. Hong, B. Jiang, R. Cui, I. V. McLoughlin, and L.-R. Dai, “Deep bottleneck network based i-vector representation for language identification,” in *INTERSPEECH*, 2015.

[50] M. McLaren, Y. Lei, and L. Ferrer, “Advances in deep neural network approaches to speaker recognition,” in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2015, pp. 4814–4818.

[51] T. Hasan, R. Saeidi, J. H. Hansen, and D. A. Van Leeuwen, “Duration mismatch compensation for i-vector based speaker recognition systems,” in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2013, pp. 7663–7667.

[52] D. Snyder, P. Ghahremani, D. Povey, D. Garcia-Romero, Y. Carmiel, and S. Khudanpur, “Deep neural network-based speaker embeddings for end-to-end speaker verification,” in *2016 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2016, pp. 165–170.

[53] D. Povey, X. Zhang, and S. Khudanpur, “Parallel training of deep neural networks with natural gradient and parameter averaging,” *CoRR*, vol. abs/1410.7455, 2014.

[54] D. Snyder, G. Chen, and D. Povey, “Musan: A music, speech, and noise corpus,” *ArXiv*, vol. abs/1510.08484, 2015.

[55] D. Snyder, D. Garcia-Romero, G. Sell, A. McCree, D. Povey, and S. Khudanpur, “Speaker recognition for multi-speaker conversations using x-vectors,” in *ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2019, pp. 5796–5800.

[56] D. Garcia-Romero, D. Snyder, G. S. Sell, A. McCree, D. Povey, and S. Khudanpur, “x-vector dnn refinement with full-length recordings for speaker recognition,” in *INTERSPEECH 2019*, 2019.

[57] N. Dehak, R. Dehak, J. R. Glass, D. A. Reynolds, and P. Kenny, “Cosine similarity scoring without score normalization techniques,” in *Odyssey*, 2010.

[58] A. O. Hatch, S. S. Kajarekar, and A. Stolcke, “Within-class covariance normalization for svm-based speaker recognition,” in *INTERSPEECH*, 2006.

[59] D. Raj, D. Snyder, D. Povey, and S. Khudanpur, “Probing the information encoded in x-vectors,” in *2019 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2019, pp. 726–733.

[60] O. Glembek, L. Burget, N. Dehak, N. Brummer, and P. Kenny, “Comparison of scoring methods used in speaker recognition with joint factor analysis,” in *2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, April 2009, pp. 4057–4060.

[61] S. Ioffe, “Probabilistic linear discriminant analysis,” in *European Conference on Computer Vision*. Springer, 2006, pp. 531–542.

[62] P. Kenny, “Bayesian speaker verification with, heavy tailed priors,” *Proc. Odyssey 2010*, 2010.

[63] D. A. Reynolds, T. F. Quatieri, and R. B. Dunn, “Speaker verification using adapted gaussian mixture models,” *Digital Signal Processing*, vol. 10, no. 1, pp. 19 – 41, 2000.

[64] R. Auckenthaler, M. Carey, and H. Lloyd-Thomas, “Score normalization for text-independent speaker verification systems,” *Digital Signal Processing*, vol. 10, no. 1, pp. 42 – 54, 2000.

[65] D. E. Sturim and D. A. Reynolds, “Speaker adaptive cohort selection for Tnorm in text-independent speaker verification,” in *Proceedings. (ICASSP ’05). IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005.*, vol. 1, March 2005, pp. I/741–I/744 Vol. 1.

[66] P. Matějka, O. Novotný, O. Plchot, L. Burget, M. D. Sánchez, and J. Černocký, “Analysis of score normalization in multilingual speaker recognition,” in *Proc. Interspeech 2017*, 2017, pp. 1567–1571.

[67] S. Cumani, P. Batzu, D. Colibro, C. Vair, P. Laface, and V. Vasilakakis, “Comparison of speaker recognition approaches for real applications.” 01 2011, pp. 2365–2368.

[68] A. Halevy, P. Norvig, and F. Pereira, “The unreasonable effectiveness of data,” *IEEE Intelligent Systems*, vol. 24, no. 2, pp. 8–12, 2009.

[69] D. Reinsel, J. Gantz, and J. Rydning, “Data age 2025 - The Digitization of the World from Edge to Core,” International Data Corporation, Tech. Rep., 2018.

[70] T. S. Behrend, D. J. Sharek, A. W. Meade, and E. N. Wiebe, “The viability of crowdsourcing for survey research,” *Behavior research methods*, vol. 43, no. 3, p. 800, 2011.

[71] J. M. Rzeszotarski and A. Kittur, “Instrumenting the crowd: using implicit behavioral measures to predict task performance,” in *UIST*, 2011.

[72] R. Snow, B. O’Connor, D. Jurafsky, and A. Ng, “Cheap and fast – but is it good? evaluating non-expert annotations for natural language tasks,” in *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. Honolulu, Hawaii: Association for Computational Linguistics, Oct. 2008, pp. 254–263.

[73] J. H. L. Hansen and T. Hasan, “Speaker recognition by machines and humans: A tutorial review,” *IEEE Signal Processing Magazine*, vol. 32, no. 6, pp. 74–99, 2015.

[74] D. Van Lancker and J. Kreiman, “Voice discrimination and recognition are separate abilities,” *Neuropsychologia*, vol. 25, no. 5, pp. 829 – 834, 1987.

[75] S. M. Kassin, I. E. Dror, and J. Kukucka, “The forensic confirmation bias: Problems, perspectives, and proposed solutions,” *Journal of Applied Research in Memory and Cognition*, vol. 2, no. 1, pp. 42 – 52, 2013.

[76] P. Bhaskar Ramteke, A. A. Dixit, S. Supanekar, N. V. Dharwadkar, and S. G. Koolagudi, “Gender identification from children’s speech,” in *2018 Eleventh International Conference on Contemporary Computing (IC3)*, 2018, pp. 1–6.

[77] A. Nagrani, J. S. Chung, and A. Zisserman, “Voxceleb: A large-scale speaker identification dataset,” *Interspeech 2017*, Aug 2017.

[78] J. S. Chung, A. Nagrani, and A. Zisserman, “Voxceleb2: Deep speaker recognition,” *Interspeech 2018*, Sep 2018.

[79] R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber, “Common voice: A massively-multilingual speech corpus,” *arXiv preprint arXiv:1912.06670*, 2019.

[80] W. Dai, C. Dai, S. Qu, J. Li, and S. Das, “Very deep convolutional neural networks for raw waveforms,” in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 421–425.

[81] D. P. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *arXiv preprint arXiv:1412.6980*, 2014.

[82] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely, “The kaldi speech recognition toolkit,” in *IEEE 2011 Workshop on Automatic Speech Recognition and Understanding*. IEEE Signal Processing Society, Dec. 2011, iEEE Catalog No.: CFP11SRW-USB.

[83] D. Snyder, G. Chen, and D. Povey, “Musan: A music, speech, and noise corpus,” *arXiv preprint arXiv:1510.08484*, 2015.

[84] D. Garcia-Romero, X. Zhou, and C. Y. Espy-Wilson, “Multicondition training of gaussian plda models in i-vector space for noise and reverberation robust speaker recognition,” in *2012 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2012, pp. 4257–4260.

[85] D. A. Reynolds, “The effects of handset variability on speaker recognition performance: experiments on the switchboard corpus,” in *1996 IEEE International Conference on Acoustics, Speech, and Signal Processing Conference Proceedings*, vol. 1, 1996, pp. 113–116 vol. 1.

[86] P. Matějka, O. Novotný, O. Plchot, L. Burget, M. D. Sánchez, and J. Černocký, “Analysis of score normalization in multilingual speaker recognition,” in *Proc. Interspeech 2017*, 2017, pp. 1567–1571.

[87] A. Schmidt-Nielsen and T. H. Crystal, “Speaker verification by human listeners: Experiments comparing human and machine performance using the nist 1998 speaker evaluation data,” *Digital Signal Processing*, vol. 10, no. 1, pp. 249 – 266, 2000.

[88] S. J. Wenndt and R. L. Mitchell, “Machine recognition vs human recognition of voices,” in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 4245–4248.

[89] K. Krippendorff, “Reliability in content analysis: Some common misconceptions and recommendations,” *Human communication research*, vol. 30, no. 3, pp. 411–433, 2004.

[90] P. Lieberman, S. Fecteau, H. Théoret, R. R. Garcia, F. Aboitiz, A. MacLarnon, R. Melrose, T. Riede, I. Tattersall, and P. Lieberman, “The evolution of human speech: Its anatomical and neural bases,” *Current anthropology*, vol. 48, no. 1, pp. 39–66, 2007.

[91] C. G. Gallagher and M. Younes, “Breathing pattern during and after maximal exercise in patients with chronic obstructive lung disease, interstitial lung disease, and cardiac disease, and in normal subjects,” *American Review of Respiratory Disease*, vol. 133, no. 4, pp. 581–586, 1986.

[92] J. A. Hirsch and B. Bishop, “Respiratory sinus arrhythmia in humans: how breathing pattern modulates heart rate,” *American Journal of Physiology-Heart and Circulatory Physiology*, vol. 241, no. 4, pp. H620–H629, 1981.

[93] I. Homma and Y. Masaoka, “Breathing rhythms and emotions,” *Experimental Physiology*, vol. 93, no. 9, pp. 1011–1021, 2008.

[94] J. Hlavnička, R. Čmejla, T. Tykalová, K. Šonka, E. Růžička, and J. Rusz, “Automated analysis of connected speech reveals early biomarkers of parkinson’s disease in patients with rapid eye movement sleep behaviour disorder,” *Scientific reports*, vol. 7, no. 1, pp. 1–13, 2017.

[95] A. Rochet-Capellan and S. Fuchs, “The interplay of linguistic structure and breathing in German spontaneous speech,” in *14th Annual Conference of the International Speech Communication Association (Interspeech 2013)*, Lyon, France, Aug. 2013, p. 1228.

[96] K. Konno and J. Mead, “Measurement of the separate volume changes of rib cage and abdomen during breathing,” *Journal of applied physiology*, vol. 22, no. 3, pp. 407–422, 1967.

[97] J. Korten and G. Haddad, “Respiratory waveform pattern recognition using digital techniques,” *Computers in biology and medicine*, vol. 19, no. 4, pp. 207–217, 1989.

[98] Y. Cho, N. Bianchi-Berthouze, and S. J. Julier, “Deepbreath: Deep learning of breathing patterns for automatic stress recognition using low-cost thermal imaging in unconstrained settings,” in *2017*

Seventh International Conference on Affective Computing and Intelligent Interaction (ACII). IEEE, 2017, pp. 456–463.

[99] V. S. Nallanthighal, A. Härmä, and H. Strik, "Deep Sensing of Breathing Signal During Conversational Speech," in *Proc. Interspeech 2019*, 2019, pp. 4110–4114.

[100] B. W. Schuller, A. Batliner, C. Bergler, E.-M. Messner, A. Hamilton, S. Amiriparian, A. Baird, G. Rizos, M. Schmitt, L. Stappen, H. Baumeister, A. D. MacIntyre, and S. Hantke, "The INTERSPEECH 2020 Computational Paralinguistics Challenge: Elderly emotion, Breathing & Masks," in *Proceedings of Interspeech*, Shanghai, China, September 2020.

[101] P. Kabal, "ITU-T G. 723.1 speech coder: A MATLAB implementation," 2004.

[102] H. Dudley, "The carrier nature of speech," *Bell System Technical Journal*, vol. 19, no. 4, pp. 495–515, 1940.

[103] P. Motlicek, H. Hermansky, S. Madikeri, A. Prasad, and S. Ganapathy, "AM-FM decomposition of speech signal: Applications for speech privacy and diagnosis," Idiap, Rue Marconi 19, Idiap-RR Idiap-RR-01-2020, 1 2020.

[104] S. Ganapathy, P. Motlicek, and H. Hermansky, "Autoregressive models of amplitude modulations in audio compression," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1624–1631, 2010.

[105] F. Eyben, F. Weninger, F. Gross, and B. Schuller, "Recent developments in openSMILE, the munich open-source multimedia feature extractor," in *Proceedings of the 21st ACM International Conference on Multimedia*, ser. MM '13. New York, NY, USA: Association for Computing Machinery, 2013, p. 835–838.

[106] M. Schmitt and B. Schuller, "OpenXBOW: Introducing the passau open-source crossmodal bag-of-words toolkit," *J. Mach. Learn. Res.*, vol. 18, no. 1, p. 3370–3374, Jan. 2017.

[107] J.-M. Valin, G. Maxwell, T. B. Terriberry, and K. Vos, "High-quality, low-delay music coding in the opus codec," *arXiv preprint arXiv:1602.04845*, 2016.

[108] S. Fuchs, U. D. Reichel, and A. Rochet-Capellan, "Changes in speech and breathing rate while speaking and biking," in *ICPhS*, 2015.

[109] S. Fleming, M. Thompson, R. Stevens, C. Heneghan, A. Plüddemann, I. Maconochie, L. Tarassenko, and D. Mant, "Normal ranges of heart rate and respiratory rate in children from birth to 18 years of age: a systematic review of observational studies," *The Lancet*, vol. 377, no. 9770, pp. 1011–1018, 2011.

[110] A. Abad, P. Bell, A. Carmantini, and S. Renais, “Cross lingual transfer learning for zero-resource domain adaptation,” *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2020.

[111] D. Liu and F. Kubala, “Online speaker clustering,” in *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings. (ICASSP ’03).*, vol. 1, 2003, pp. I–I.

[112] M. Markitantov, D. Dresvyanskiy, D. Mamontov, H. Kaya, W. Minker, and A. Karpov, “Ensembling End-to-End Deep Models for Computational Paralinguistics Tasks: ComParE 2020 Mask and Breathing Sub-Challenges,” in *Proc. Interspeech 2020*, 2020, pp. 2072–2076.