TECHNISCHE UNIVERSITÄT MÜNCHEN
Lehrstuhl für Mensch-Maschine-Kommunikation

Semi-Autonomous Data Enrichment and Optimisation for Intelligent Speech Analysis

Zixing Zhang

Vollständiger Abdruck der von der Fakultät für Elektrotechnik und Informationstechnik der Technischen Universität München zur Erlangung des akademischen Grades eines

Doktor-Ingenieurs (Dr.-Ing.)

genehmigten Dissertation.

Vorsitzender:		Univ.-Prof. Dr.-Ing. habil. Dr. h.c. Alexander W. Koch
Prüfer der Dissertation:	1.	Univ.-Prof. Dr.-Ing. habil. Björn W. Schuller, Universität Passau
	2.	Univ.-Prof. Gordon Cheng, Ph.D.

Die Dissertation wurde am 30.09.2014 bei der Technischen Universität München eingereicht und durch die Fakultät für Elektrotechnik und Informationstechnik am 07.04.2015 angenommen.

Acknowledgement

First of all, I am deeply grateful to my supervisor Prof. Björn Schuller. Four years ago, I had the privilege to start work in his Machine Intelligence and Signal Processing group at Technische Universität München. Since then, Prof. Schuller spent much time on me for the guidance. I appreciate it very much, since this work would not have been possible without his excellent inspiration and support. Moreover, I am also very grateful to Prof. Gerhard Rigoll for providing a comfortable working environment at his institute, Prof. Gordon Cheng for carefully reviewing and examining this thesis, and Prof. Alexander Koch for chairing the examination board.

I would also like to thank all scientific and non-scientific colleagues for the past years, especially Jun Deng, Felix Weninger, Dr. Eduardo Coutinho, Florian Eyben, Jürgen Geiger, Dr. Martin Wöllmer, Peter Brand, Martina Römpp, Heiner Hundhammer, Dr. Daniel Willett, Dr. Joel Pinto, Christian Plahl, Raymond Brückner, Dr. Bin Dong, Dr. Hesam Sagha, Dr. Shaowei Fan, Dr. Hua Zong, Yue Zhang, Erik Marchi, Dr. Fabien Ringeval for the technical assistance and collaboration, as well as for the non-technical discussion.

Most of all, I would like to extend my sincere thanks to my parents Gengtian Zhang and Xuejiao Lin who have always supported me during all stages of my education, as well as my sister Jieqiong Zhang and my brother Yang Zhang,

Munich, September 2014 Zixing Zhang

Abstract

Intelligent Speech Analysis (ISA) plays an essential role in smart conversational agent systems that aim to enable natural, intuitive, and friendly human computer interaction. It includes not only the long-term developed Automatic Speech Recognition (ASR), but also the young field of Computational Paralinguistics, which has attracted increasing attention in recent years. In real-world applications, however, several challenging issues surrounding data quantity and quality arise. For example, predefined databases for most paralinguistic tasks are normally quite small and few in number, which are insufficient for building a robust model. A distributed structure could be useful for data collection, but original feature sets are always too large to meet the physical transmission requirements, for example, bandwidth limitation. Furthermore, in a hands-free application scenario, reverberation severely distorts speech signals, which results in performance degradation of recognisers.

To address these issues, this thesis proposes and analyses semi-autonomous data enrichment and optimisation approaches. More precisely, for the representative paralinguistic task of speech emotion recognition, both labelled and unlabelled data from heterogeneous resources are exploited by methods of data pooling, data selection, confidence-based semi-supervised learning, active learning, as well as cooperative learning. As a result, the manual work for data annotation is greatly reduced. With the advance of networks and information technologies, this thesis extends the traditional ISA system into a modern distributed paradigm, in which Split Vector Quantisation is employed for feature compression. Moreover, for distant-talk ASR, Long Short-Term Memory (LSTM) recurrent neural networks, which are known to be well-suited to context-sensitive pattern recognition, are evaluated to mitigate reverberation. The experimental results demonstrate that the proposed LSTM-based feature enhancement frameworks prevail over the current state-of-the-art methods.

Zusammenfassung

Intelligente Sprachanalyse (ISA) ist von grundlegender Bedeutung für zukünftige Sprachdialogsysteme, die auf natürliche, intuitive und benutzerfreundliche Mensch-Maschine-Interaktion abzielen. ISA beinhaltet nicht nur automatische Spracherkennung, deren Entwicklung bereits weit fortgeschritten ist, sondern auch das derzeit im Entstehen begriffene, aber zunehmend wichtige Feld der computergestützten Paralinguistik (Computational Paralinguistics). In realistischen Anwendungsszenarien stellen sich einige Herausforderungen hinsichtlich Quantität und Qualität der Daten, die von ISA-Systemen verarbeitet werden. Zunächst sind für Anwendungsgebiete im Bereich der computergestützten Paralinguistik nur wenig und meist kleine Datensätze verfügbar, was zu erheblichen Problemen bei der Robustheit der trainierten Modelle führt. Dieses Problem könnte durch Datensammlung über verteilte Architekturen gelöst werden; die Implementierung solcher Architekturen im Bereich der computergestützten Paralinguistik stellt jedoch insofern ein Problem dar, als die typischerweise verwendeten Merkmalssätze zu groß für die Übertragung über bandlimitierte Kanäle sind. Schließlich sind bei wichtigen Anwendungsszenarien von ISA wie z.B. im Freisprechmodus auch Störfaktoren wie Nachhall zu berücksichtigen, die die Erkennungsgenauigkeit signifikant beeinträchtigen können.

Als Beitrag zur Lösung dieser Probleme stellt diese Arbeit einige Ansätze zur Optimierung der Datensammlung und -übertragung vor. Zunächst werden am Beispiel der Emotionserkennung Methoden zur Datenaggregation und -selektion eingeführt, die handverschriftete und unverschriftete Daten aus verschiedenartigen Quellen mittels halbüberwachtem Lernen mit Konfidenzmaßen, aktivem und kooperativem Lernen vereinigen. Dadurch kann der Arbeitsaufwand zur manuellen Datenverschriftung erheblich reduziert werden. Anschließend wird ein Ansatz vorgestellt, in dem herkömmliche ISA-Systeme durch Split-Vektorquantisierung zur Kompression der Merkmalssätze in eine moderne verteilte Architektur überführt werden. Schließlich werden Long Short-Term Memory (LSTM) rekurrente neuronale Netze hinsichtlich ihrer Eignung zur Enthallung von Sprachmerkmalen bewertet, deren Architektur in besonderer Weise auf kontextbehaftete Musterkennungsaufgaben zugeschnitten ist. Die Ergebnisse zeigen, dass der vorgeschlagene LSTM-Ansatz zur Merkmalsverbesserung einen erheblichen Zugewinn gegenüber dem Stand der Technik bringt.

Contents

1 Introduction

1.1 Motivation

Linguistic content interpretation systems, also known as Automatic Speech Recognition (ASR) systems, have been studied for more than half a century. The aim of these systems is to enable machine–human, or even machine–machine, communication that is as natural as human–human interaction. Thanks to the advance of new techniques such as deep learning algorithms, these systems have begun to be used in real-life applications, for example, as personal voice assistants in smartphones. Nevertheless, these systems still lack any social competencies and are weak in complex environments, such as reverberant rooms. In this light, an extended research field of speech technology is emerging, termed 'Intelligent Speech Analysis' (ISA). While research in speech technology traditionally focuses on the interpretation of *verbal speech* (i.e., ASR), the new research domain of ISA is more relevant to the understanding of *non-verbal speech*, that is, *computational paralinguistics*. Paralinguistics means 'alongside linguistics'. It is defined as the discipline dealing with those phenomena that are embedded into the verbal message, for example, speakers' states, attitudes, and characteristics [1]. Computational paralinguistics indeed aims to extract these paralinguistic cues from speech by computationally intelligent algorithms provided by the machine-learning, and not just the signal-processing, community.

The advance of ISA can benefit numerous and wide-ranging potential applications in information and communication systems. For example, the improved capability of computational paralinguistics and ASR has been shown to greatly contribute to the building of more natural and friendly interactive and communicative robots. It has also been helpful for creating more sophisticated call-centre management systems, computer games, and the systems for the monitoring and surveillance of critical environments. However, research on the paralinguistic tasks in particular is still at its earliest stages. Thus, at the moment, these application systems are far from being perceived as natural, efficient, and comparable to humans in the 'wild'. This is because of the limitation of system capabilities in ASR, computational paralinguistics, and so forth.

To bridge the gap between humans and machines regarding information extraction and perception of speech, we need to overcome several major *data*-related challenges in the field of ISA:

(1) *Data scarcity for computational paralinguistics.* With respect to the analysis of paralinguistic tasks, especially emotion recognition, most previous studies based their methods on a restricted set of labelled corpora that contain a rather *small amount* of data. These data have normally been acquired from subjects representing a limited range of cultures and backgrounds. Recording environments and devices are often controlled, and the linguistic tasks asked of the subjects are often prototypical and situation-specific. Therefore, the data are easy to characterise with current pattern recognition techniques, thus leading to overestimated recognition accuracies. By contrast, realistic inter-human or human–machine interactions are more complex and diverse. Therefore, data need to be collected from different recording environments and devices, with diverse backgrounds of evaluation targets, and non-prototypical tasks (e.g., ambiguous and spontaneous emotions). Even state-of-the-art systems have difficulty in distinguishing speech patterns, owing to the narrow data set on which their development was based [2, 3, 4]. The shortage of data seriously limits the robustness of acoustic modelling and consequently restrains its real-life applications to some extent. Even though extensive research has been conducted on other pattern-recognition applications (e.g., speech recognition), insufficient efforts have been made in the computational paralinguistics area to provide any comprehensive conclusions about how to deal with this data scarcity [5, 6].

(2) *Large dimensions of data features for computational paralinguistics.* Thanks to the prevalence of networks and the advance of cloud computing technology, server- or even cloud-based recognition systems are well-suited to handling ubiquitous data processing and computing. Consequently, such a distributed structure can continuously refine classification models [7, 8]. Feature reduction has been addressed in much previous research [9, 10, 11]; however, this research does not consider the specific requirements of such a distributed computing structure. To better leverage this structure for computational paralinguistics, the data have to be optimised to meet the requirement of data transmission via physical communication channels, data storage in memory, and user-privacy protection.

(3) *Distorted data features for ASR.* Current systems of verbal speech analysis can yield a very high accuracy when recognising well articulated read speech in predefined clean acoustic conditions. Nonetheless, background noise and reverberation, which are to be expected in natural communication, can severely

downgrade recognition performance. Numerous approaches to tackle this issue have been proposed over the past decades, and have had a great success [12]. Additionally, the development of advanced algorithms makes it possible to shed new light on this issue. These algorithms could further enhance the data feature, improve recognition accuracy, and, simultaneously, do not require heavy revision of pre-existing systems.

In general, all three challenges have to do with the front stage of the processing chain of ISA systems. In addition to these challenges, there are also many others, for example, the determination of labels for subjective paralinguistic tasks, the interaction effect among multiple tasks, and the environmental noise effect for computational paralinguistics. It is worth noting that the concept of data in this thesis is defined as the pattern recognition unit before feature extraction (i.e., frame, instance, turn, example, or record), or the values after feature extraction (i.e., attribute or feature).

1.2 Aims of this Thesis

This thesis targets the aforementioned three major challenges by trying to achieve the following objectives:

(1) *Data enrichment by exploiting heterogeneous labelled data.* The most direct way to increase the data quantity is through the use of pre-existing labelled databases. However, most of these databases are prepared for specific targets with different annotations, background speakers, recording environments, etc. In this light, a unified learning mechanism is required to make better use of these heterogeneous data.

(2) *Data enrichment by exploiting vast unlabelled data, with as little human effort as possible.* In comparison with the low amount of labelled data, a vast quantity of unlabelled data are available from public sources, such as social media, and they can be acquired through the use of application systems. Conventional ways to annotate these data by human raters (labellers/annotators/coders) are extremely time-consuming and expensive [13]. Therefore, to efficiently leverage these unlabelled data, it is necessary to create a (semi-)automatic self-optimising mechanism. By doing this, existing classification models can still be used, and their parameters continuously and statistically optimised.

(3) *Data optimisation by reducing feature size.* In the case of distributed computational paralinguistics, feature optimisation methods for reducing feature dimensionality are required. These methods have the potential to benefit not only data transmission via communication line and data storage in memory, but also

privacy protection. Although distributed speech recognition systems are well-developed and even standardised, distributed computational paralinguistics lag behind [7, 8, 14]. Therefore, it is interesting to know whether the relative feature optimisation techniques used for distributed speech recognition could be transferred to the computational paralinguistics.

(4) *Data optimisation by enhancing corrupted features.* Here, the aim is to ease the impact of noise, especially reverberation, on speech recognition from the acoustic feature aspect. The advantage of this strategy is that the pre-existing acoustic models, which are trained on a clean data set, can be maintained. Recently, novel neural network structures have been employed to abstract the high-level representative features from original ones, or to create acoustic or language models for speech recognition. These networks have been demonstrated to be greatly effective for ASR [15, 16]. For feature enhancement, therefore, it is also worth investigating the effectiveness of a state-of-the-art neural network.

Generally, the central points of these challenges to be addressed and advanced are the data quantity and quality. To achieve the above objectives, this thesis is dedicated to developing (semi-)autonomous *data enrichment* and *optimisation* algorithms in the context of ISA. For computational paralinguistics (in particular for emotion recognition), a variety of methods are extensively studied to increase the amount of training data without relying on expensive human effort for labelling. These methods include semi-autonomous *pooling*, *data selection*, *self-learning* (e.g., *semi-supervised learning* and *active learning*), and their derivations. Moreover, in the case of distributed paralinguistics recognition systems, one crucial issue of feature optimisation (or rather *feature compression*) is addressed by *Split Vector Quantisation* (SVQ). For ASR, promising context-sensitive networks, namely *Long Short-Term Memory* (LSTM) neural networks [17], are employed to advance the *feature enhancement* techniques.

1.3 Structure of this Thesis

This thesis is structured into five main chapters as follows.

Chapter 2 concentrates on the theoretical framework of the ISA system. First, an overview of the workflow of the state-of-the-art ISA system is given. Then, fundamental and advanced knowledge of each crucial component is presented stage by stage: from the speech data, to the acoustic features, to the classification algorithms and the evaluation metrics.

Chapter 3 describes a set of methods that are proposed and employed in this thesis for dealing with the challenges outlined in Section 1.1. To be more specific, these methods are designed to exploit labelled data using data aggregation or data selection; or to utilise unlabelled data through various semi-supervised learning and active learning methods without involving a lot of manual effort; or to optimise feature data by reducing its feature size or alleviating the impact of channel reverberation. All these methods are successively evaluated.

Chapter 4 describes the implementations of the methods presented in Chapter 3. These applications cover speech emotion recognition, general paralinguistics recognition, and verbal speech recognition. For each task, various methods are provided to address different aspects of the challenges. Following an introduction of carefully selected databases and a description of experimental setups, the results of the corresponding methods are then reported.

Chapter 5 summaries and concludes the presented work, and suggests possible further investigations.

Additionally, Appendix A gives a short introduction of multiple databases that have been adopted to investigate the effectiveness and efficiency of the methods proceeded in this thesis.

2 General Framework of Intelligent Speech Analysis

2.1 Overview

Figure 2.1 illustrates the framework of a unified *Intelligent Speech Analysis* (ISA) system. The major objective of this system is to utilise machine learning approaches to extract interesting information from speech signals. The information generally includes speakers' verbal intentions, their personal IDs, states (e.g., anger or sleepiness), cultural background, and characteristics. Inspired by the standard structure of speech recognition [18], the framework of the ISA system can also be divided into two modules: the *front-end* and the *back-end*. The front-end module comprises several sequential processing stages that aim to maintain as much information as possible in speech input and to encode them into a collection with the fewest features. With these features, the back-end module could continuously train and update the task-oriented acoustic/language models and interpret these features as specific notations, such as words and emotions.

More specifically, in the front-end, the goal of the speech preprocessing stage – the *Signal Processing* block – is to enhance the incoming speech signals that are often distorted by various noises, such as the additive noise derived from multiple interfering speakers, environmental and recording noise, as well as the convolutional noise, such as reverberation. Many methods are employed to suppress the effect of noise, such as adaptive filtering [19], spectral normalisation and subtraction [20], Non-negative Matrix Factorisation (NMF) [19], and beamforming [12].

Voice Activity Detection (VAD) is firstly used for detecting speech signals and dropping the non-speech frame. If speech is detected, the signals will be delivered to the following processing components. Following the *Signal Processing* stage, the denoised and enhanced signals are sent to the feature extraction module. Here, *Low-Level Descriptors* (LLDs) are computed at a frequency of 100 frames per second with a typical window size of 10–30 ms. The windowing functions for the extraction of

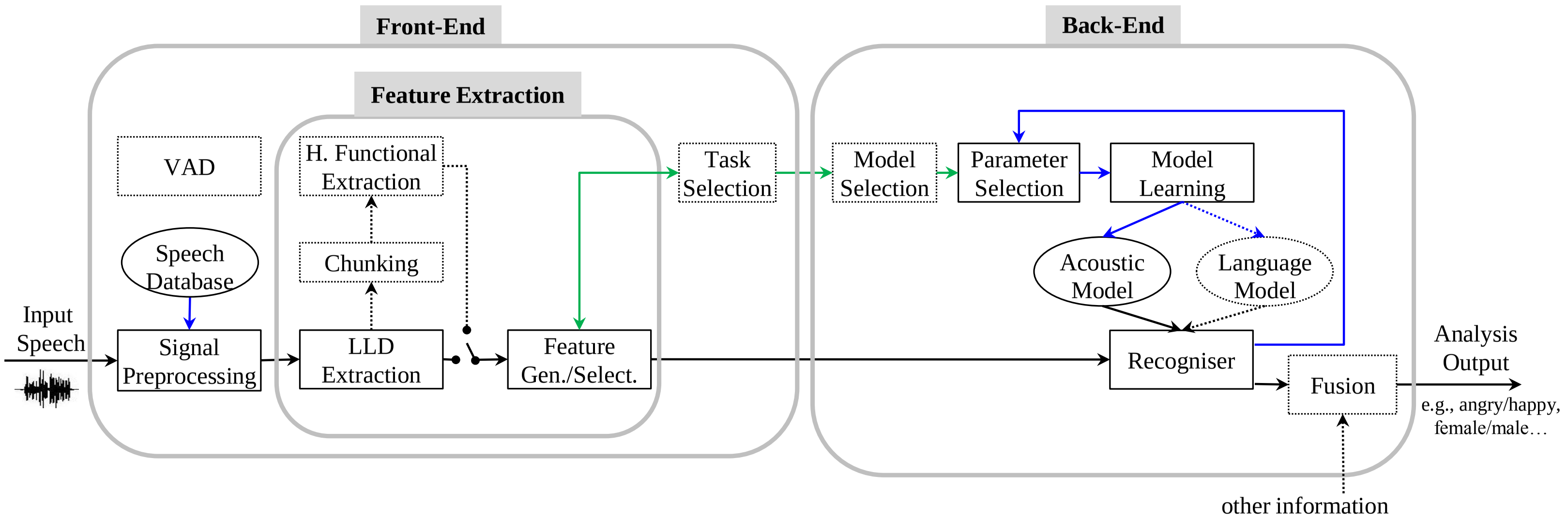


Figure 2.1: Framework of intelligent speech analysis (ISA). Dotted boxes represent optional components. Blue arrows denote steps carried out only during system training or adaptation phases. Green arrows indicate steps carried out when multiple recognition tasks need to be processed at the same time or separately.

LLDs in the time or the time-frequency domain are usually chosen as the smooth ones (Hamming or Hann) or the rectangular one [4]. Then, the LLD sequences are divided (chunked) into super-segmental turns (the *Chunking* block). The turns can be the fixed number of frames, syllables, words, acoustic chunks, sub-turns, or complete turns [21]. In the present framework, the selection of turns depends on the requirements of specific tasks. For example, emotion-related information is often reflected by short-term speech, whereas gender information can normally be accumulated over the whole speech track. After chunking the LLD sequences, *hierarchical functionals* are applied over LLDs to each turn, so as to project the multivariate time series onto a single vector of a fixed dimension that is independent of the length of the entire turn. Especially for Automatic Speech Recognition (ASR), only the process of LLD extraction is implemented to extract features, such as Mel-Frequency Cepstral Coefficients (MFCCs), due to its rapid pronunciation variation over time. To reduce the dimensionality of a feature set, a *feature selection* stage is additionally added by reducing the relevancy among attributes.

Turning now to the back-end module, the feature set is delivered to the *recogniser* for either classification, where the outputs are discrete classes (e.g., word and gender), or regression, where the outputs are continuous values (e.g., speaker's height and age). The classification/regression results from the *acoustic* and/or *language models* can also be fused with other information (the *Fusion* block), such as facial expression and motion recognition, before reaching the final decision of pattern analysis.

Moreover, before putting forth the speech analysis, pre-existing *databases* with target labels are applied for initial parameter selection and model learning. The *parameter selection* block 'fine tunes' the learning algorithms, such as the learning algorithm's topology, the type of functions, or initialisation. The *model learning* block is the actual training phase, where classifiers or regressors are built on labelled data. By this process, two models can be obtained: the acoustic model and the language model. The *acoustic model* is built on the acoustic features, whereas the language model is created on the speech content. In this thesis, the language model is only used for ASR.

Given the need to deal with diverse tasks simultaneously, and the fact that the information required for a specific task may also be relevant to other tasks, *task selection* algorithms are of significance for the execution of the ISA system. A common way is to invite the user to perform the task selection which in turn determines the feature set selection in the front-end and the *model selection* in the back-end. If no specific tasks are predefined, however, computational auditory scene analysis (CASA) could be used to automatically analyse the circumstances of speech recording (e.g., cocktail party, home, street) and to then predict the possible tasks [22].

All in all, several important challenges need to be overcome on the way towards the goal of extracting useful information from speech signals by such a unified system (cf. Section 1.1). In the ongoing, the challenges, as well as each important component in the processing chain, will be discussed in detail.

2.2 Speech Data

In the realm of pattern recognition, there is a preference to use more data to train classifiers. However, the ISA, as a fresh pattern research field (except for the speech and speaker recognition), has insufficient data with which to create a robust classification [14]. Accordingly, the widely-agreed priority is a change in the scale and quality of databases in the field of ISA, especially computational paralinguistics.

2.2.1 Databases

Before discussing the challenge of data scarcity for the ISA system, the following subsections briefly sketch and cluster the categories of speech data.

2.2.1.1 Human vs. Synthesised

From the generation point of view, speech can be primarily divided into two groups: the human and the synthesised speech. The *human* speech, as the name implies, is uttered by the vibration of human's plica vocalis. Particularly, it can be categorised as the acted one, the induced one, and the spontaneous one upon the types of speaking. On the contrary, the *synthesised* speech is created via computer systems by statistical modelling and concatenative signal generation techniques.

1) Human Speech: Acted, Induced, and Spontaneous

Due to the difficulty of controlling the power level and settings of microphones during recording, the earlier databases implemented for ISA, occurs on *acted* speech. To collect these data, the speakers (usually amateur or professional performers) are asked to pretend a predefined task, e.g., expressing various emotions like anger, disgust, fear, happiness, sadness, and surprise, by uttering mostly fixed spoken content. In fact, these databases are partly and originally not intended for analysis, but for quality measurement of speech synthesis [23]. For pattern recognition, even though they are comparably easy to obtain high accuracy with, they are quite far from realistic settings.

The *induced* group is a transitional solution to ease the recording conditions of spontaneous speech, with the aim of generating the spontaneity-similar one from the acoustic feature aspect. Generally, a Wizard-of-Oz (WOZ) is designed at the beginning (e.g., a multi-model dialogue system), then elicits the speakers to a predetermined state, giving rise to the provoking of utterances 'naturally'. Normally, the speakers are in the 'darkness' of the specific purpose in order to let them perform naturally. Hence, eliciting data in such a WOZ scenario seems to be a useful way of determining what may happen in a real-life application.

Nowadays, the actual application-oriented research work motivates to collect more *spontaneous* data from daily life. Typical scenarios are related to call centres, game-playing, communication between human and robots or automatic voice servers, and the like. As a matter of fact, the pattern recognition tasks on such natural data are much tougher than that with the acted or the induced ones. One difficulty pertains to the inherent differences between the real and the acted speeches [24, 25, 26]. These differences mainly include: speaking rate, acoustic environment, and vocal tracts variability, which result in a huge gap in acoustic features between the real and the acted speeches [9]. Another difficulty with spontaneous speech is the annotation, since the speakers are almost impossible to be known with certainty. In addition, the majority of the spontaneous speech often has nothing to do with the targets of interest. Taking the speech emotion recognition for example, the neutral speech indeed normally predominates the whole recording.

2) Synthesised Speech

By contrast, synthesised speech is fully artificially produced. The Text-To-Speech (TTS) synthesis procedure mainly consists of two stages: The first stage is text analysis and phonetic transcription. The input raw text is firstly normalised as a unified format. Then, we assign phonetic transcription to each word, and divide and mark the text into prosodic units, like phrases, clauses, and sentences. Phonetic transcriptions and prosody information together make up the symbolic linguistic representation. The second one is referred to as the synthesiser that converts the symbol linguistic representation into the waveforms by using the phonetic and prosodic information. Two primary approaches to model the synthetic speech exist: the parametric system, like articulatory and formant synthesis, and the data-based system, such as unit-selection synthesis [27].

An approach called diphone synthesis is considered as the compromise between the flexibility of parametric synthesis and the naturalness of data-based synthesis. In this thesis, the simulation of emotions is achieved by a set of parametrised rules that describe the manipulation of speech signals by the following aspects: pitch changes (including pitch level, range, variation, contour, and duration), voice quality (e.g., jitter), and articulation (substitution of centralised/decentralised vowels). For the details about the modifications, please refer to [28]. Finally, the quality of a speech synthesiser is evaluated by its similarity to the human voice or by its ability to be distinguished clearly.

2.2.1.2 Objective vs. Subjective

Upon the subjectivity of target patterns, the data can be classified as the objective and the subjective ones. The vertical axis of Figure 2.2 denotes the subjectivity of various patterns in speech.

Like traditional speech recognition and speaker identification, some other patterns in speech (deemed as *objective* patterns) can be measured (annotated/labelled) by *ground truth*, or highly correlated with the ground truth. That is, the source can be mapped one-to-one onto the variable/class: gender in female/male, age in year/day, or height in cm/in. In addition, personal state of alcoholic intoxication can be measured reasonably reliably with Blood Alcohol Concentration (BAC), the value of which could indicate the degree of intoxication. Heart rate can also be measured more or less directly by using electrocardiogram equipment.

By contrast, there are some other phenomena (namely *subjective* patterns) that can only be accessed by perceptive judgement. Such an assessment highly depends on the individuals, since each person normally has different perceptual thresholds. For music mood, for example, some would consider a musical piece sadder or happier than others, or even have opposing views due to personal associations with a song. The same holds true for other speakers' emotion-related states like stress, interest, and confidence, as well as speakers' personality-related traits of likeability, friendship, and aggressiveness. This assessment is also known as *gold standard*. In contrast to the ground truth that is for the tasks being able to be measured precisely, the gold standard is only for agreed-upon human annotation procedures. As to speech and language analysis, such a gold standard has twofold impacts: On the one side, learnt models of computer systems that process, for example, affective data are annotator-prone; on the other side, the test results might be over- or underestimated. Thus, to live up to a reliable gold standard, several annotators are required for one database.

These multiple annotations are unified into a single hierarchy as a final reference (the gold standard) by means of certain decision rules [29]. The most popular and straightforward method is the *majority voting* [30], where each rater is equally considered, and the final label is defined by the most agreement for the discrete classes or by the central tendencies (average values) over all the raters for continuous values. In practice, some annotators may lack concentration if they have to label huge amounts of data, or do not take labelling seriously at all or at any time. In this light, weighting evaluators is considered to reach the greatest consent among these with the gold standard. In [31], Grimm & Kroschel proposed the *Evaluator Weighted Estimator* (EWE) with the definition of

$$\hat{x}_n^{\mathrm{EWE},i} = \frac{1}{\sum_{m=1}^{M} r_m^i} \sum_{m=1}^{M} r_m^i \hat{x}_{n,m}^i, \tag{2.1}$$

where the indices n and m stand for the instance index $1 \leq n \leq N$ and for the

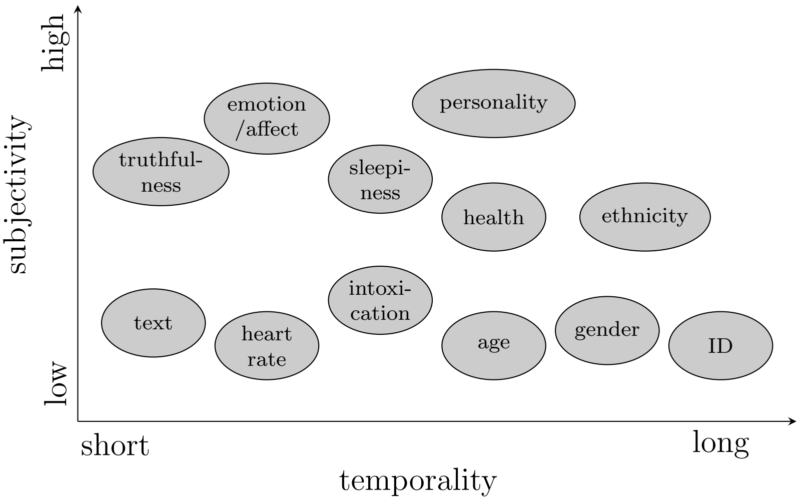


Figure 2.2: Subjectivity and temporality of various patterns in speech.

evaluator index $1 \leq m \leq M$, respectively. The $\hat{x}^i_{n,m}$ is the label of the rater m for the instance n. The evaluator-dependent weight r^i_m is estimated by calculating the correlation coefficient between the individual evaluation sequences $\{\hat{\mathbf{x}}^i_{n,m}\}_{n=1,\ldots,N}$ and the Maximum Likelihood estimation sequences $\{\hat{\mathbf{x}}^{\text{MLE},i}_n\}_{n=1,\ldots,N}$, $\{\hat{x}^{\text{MLE},i}_n\} = \frac{1}{M}\sum_{m=1}^{M}\hat{x}^i_{n,m}$, as follows

$$r^i_m = \frac{\sum_{n=1}^{N}(\hat{x}^i_{n,m} - \mu^i_m)(\hat{x}^{\text{MLE},i}_n - \mu^{\text{MLE},i})}{\sqrt{\sum_{n=1}^{N}(\hat{x}^i_{n,m} - \mu^i_m)^2}\sqrt{\sum_{n=1}^{N}(\hat{x}^{\text{MLE},i}_n - \mu^{\text{MLE},i})^2}}, \tag{2.2}$$

with

$$\mu^i_m = \frac{1}{N}\sum_{n=1}^{N}\hat{x}^i_{n,m} \tag{2.3}$$

and

$$\mu^{\text{MLE},i} = \frac{1}{N}\sum_{n=1}^{N}\hat{x}^{\text{MLE},i}_n. \tag{2.4}$$

If all evaluators have the same correlation coefficient r^i_m, the gold standard is equal to the mean of the labels of all raters $\hat{x}^i_{n,m}$. Therefore, the EWE is a weighted mean, with the weights corresponding to the 'reliability' of each rater, which is the cross-correlation of one's rating with a mean rating (over all the raters). For each rater, this cross-correlation is computed only on the block of stimuli that one rated.

2.2.1.3 Short Term vs. Long Term

In accordance with the temporal characteristic of targets, the data can be categorised as short-, medium-, and long-term tasks-related ones. The horizontal axis of Figure 2.2 denotes the temporality of various patterns in speech.

Plenty of previous research has been devoted to the ‘endpoints’ on the time scale, i.e., to spontaneous text understanding (speech recognition) and to permanent personal ID identification. Within this time axis, other speakers’ information like gender, age, personality, sleepiness, and emotion states are also crucial to the modelling and processing of human–human or human–machine interaction, giving rise to increasing attention recently.

It is widely accepted that these paralinguistic phenomena can be sorted as follows [4]:

Short-term states: They normally last for only a few microseconds, seconds, or minutes. Generally speaking, they include emotion-related states, feelings, or affects, e.g., Ekman’s six basic emotions (anger, disgust, fear, happiness, sadness, and surprise) [32], confidence, truthfulness, politeness, interest, intimacy.

Medium-term phenomena: This group generally takes at least several minutes, hours, or even days. Normally, these phenomena consist of (partly) self-induced states such as intoxication, sleepiness, and health states, as well as structural (behavioural, interactional, social) signals like consumer tendencies, interpersonal attributions, and friendship.

Long-term traits: These patterns vary in a long run (usually for months or years), or even remain permanently invariable. Generally, they are related to biological trait primitives like age, height, weight, gender; to the cultural background, race, and ethnicity; to the personal characteristics such as likeability; and to the speaker ID.

The differences among various patterns in the temporality results in different ways to handle these patterns from a processing point of view. For short-term states, taking emotion for example, it is known that the emotional state in the previous seconds influences that in the following seconds. Thus, segmenting speech signals into coherent chunks is crucial for performance analysis [21, 33]. In contrast, for medium- and long-term patterns, the method of segmentation is not so important because the state does not change quickly. Moreover, from the machine learning aspect, it is quite efficient to collect cumulative evidence for longer-term patterns [34].

2.2.2 Challenge of Data Scarcity

With the development of ASR, the data-scarcity problem has been greatly relaxed. However, it is still one of the major curses of computational paralinguistics. Even though several public databases have been designed as benchmark sets for specific

Table 2.1: Classic *paralinguistic* databases (For emotional databases, please refer to Appendix A.1). Some acronyms of tasks: *intox*ication, *personal*ities, and *likeab*ility; Languages (Lan): English (EN), French (FR), German (DE), or Dutch (NL); speech types (Y): spontaneous (S) or promoted (P); number of instances (Inst) and total speech time (T[h]); number of speakers (sk) and labellers (L); recordings (Rec) obtained by lab recording (LAB), conference recording (CON), telephone transmission (TEL), or broadcast speech (FM); and recording rate (kHz).

Task	Corpus	Lan	Y	# Inst	T[h]	# sk	# L	Rec	kHz
interest	AVIC [35]	EN	S	3 880	2.3	21	4	LAB	11
deception	CSC [36]	EN	P	1 140	7.0	32	-	LAB	16
conflict	SC [37]	FR	S	1 430	12.0	138	550	CON	16
sleepiness	SLC [38]	DE	P	9 089	21.3	99	4	LAB	16
intox.	ALC [39]	DE	P	12 360	43.8	162	-	LAB	16
heart rate	MBC [40]	EN	P	1 088	0.3	19	-	LAB	96
cognition	CLSE [41]	EN	P	2 418	1.0	26	-	LAB	16
personality	SPC [42]	FR	S	640	1.7	322	11	FM	8
likability	SLD [43]	DE	P	800	0.7	800	32	TEL	8
pathology	NCSC [44]	NL	P	2 386	2.0	55	13	LAB	16
autism	CPSD [45]	FR	P	2 542	1.0	99	4	LAB	16
ethnicity	VaB [46]	EN	S	315	175	557	-	TEL	8
age	aGender [47]	DE	P	65 364	50.6	945	-	LAB	44
gender	aGenger [47]	DE	P	65 364	50.6	945	-	LAB	44

tasks, part of which are shown in Table 2.1, they are just 'drops of water in the desert'.

The limitations of these databases can be summarised as follows:

1) *Small amount of instances.* From Table 2.1, it can be seen that most databases include only thousands of instances and several hours of recording. Compared to the state-of-the-art speech recognition systems, like Siri on iPhone, Cortana on winPhone, and Google Now on Andriod, which are trained on hundreds and thousands of hours translated recordings, the databases in Table 2.1 seem to be insufficient for building a robust model.

2) *Less spontaneity.* Most databases are recorded by a predefined recording procedure (i.e., fixed content) in laboratories. In this case, the characteristics are always exaggeratedly expressed, and the speech signals are normally quite clean. All these phenomena do not match with real-life scenarios. For real-world applications, more spontaneous speech under diverse recording environments is required.

3) *Limited number/background of speakers.* The diversity of speakers is crucial for speech analysis (cf. Section 2.3.4). Therefore, increasing the variety of speakers with different dialects, ethnicities, gender, ages, and the like, will certainly enhance the model robustness.

4) *Imbalanced class distribution.* In many cases, the examples in the training data set belonging to one class heavily outnumber that of other classes. Such an imbalanced distribution over classes leads to difficulties for the systems to learn the concept related to the minority class [48, 49, 50].

5) *Incompatibility of annotations across corpora.* Due to the lack of standardisation, each database is normally created with its own annotation method for its own goal. Let us take emotion for example – some databases are labelled by event-related emotions, such as Ekman's 'big six' emotion classes (i.e., anger, disgust, fear, happiness, sadness, and surprise) [32], but some others are marked by self-appraisal emotions (i.e., achievement, self-confidence, sociability, embarrassment, shame, guilt, and remorse) [51].

6) *High-level labelling disagreement or mislabelling.* As given in Section 2.2.1.2, the subjective tasks are accessed by annotators' perception. Such perception might be not reliable because of the annotators' mind distraction for long-time working, their reluctance to label, or their temporary variation of personal emotion. This probability results in high-level of labelling disagreement, even in mislabelling [52].

All these data-scarcity-related issues laying before the computational paralinguistics obstruct the application of intelligent speech techniques. This thesis will address such data-scarcity-related issues elaborately, and give an overview on the representative paralinguistic task of emotion.

2.3 Acoustic Feature Extraction

Speech recognition is primarily based on spectral features, such as MFCCs, whereas computational paralinguistics tends to employ a larger set of statistical features, which are extracted by applying higher hierarchical functionals over a set of LLDs. Figure 2.3 gives an overview of the acoustic feature generation steps: 1) *frame-level* LLDs extraction; 2) chunking; and 3) *super-segment-level* functionals application. The following subsections will provide more details about the LLDs and the functionals for the recognition of speech and paralinguistic tasks.

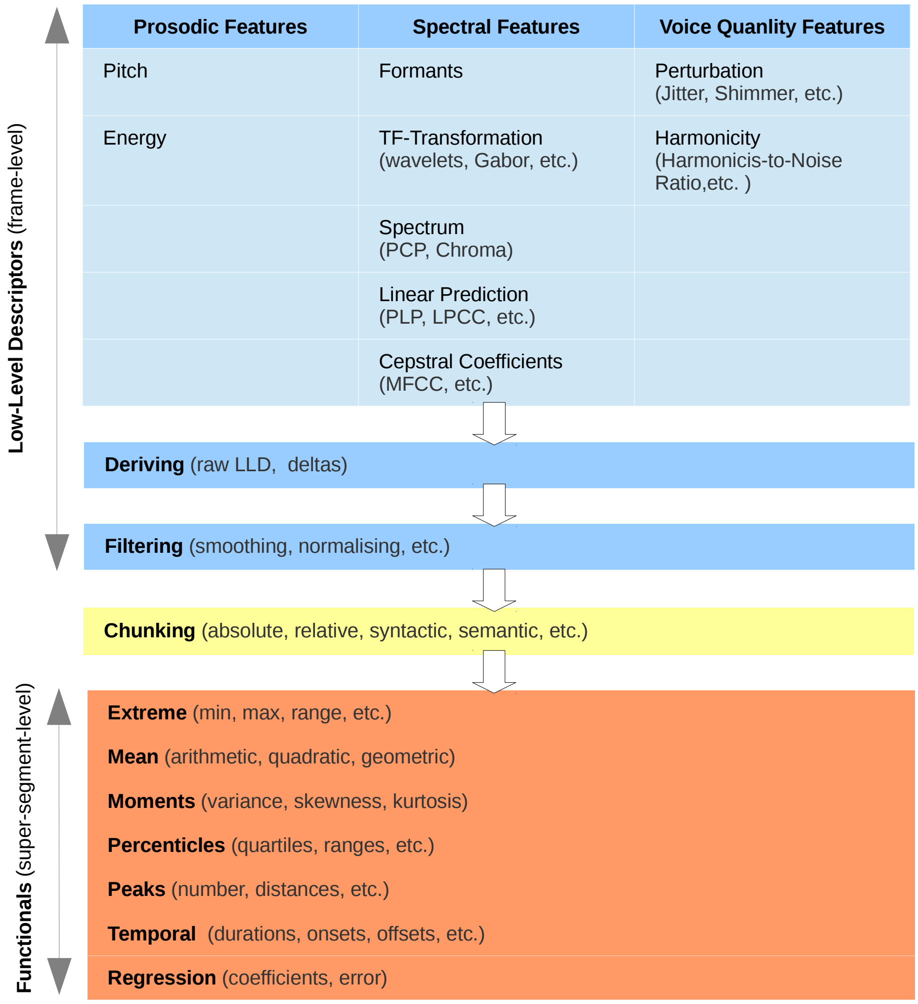

Prosodic Features
Spectral Features
Voice Quanlity Features
Pitch
Formants
Perturbation (Jitter, Shimmer, etc.)
Energy
TF-Transformation (wavelets, Gabor, etc.)
Harmonicity (Harmonicis-to-Noise Ratio,etc.)
Spectrum (PCP, Chroma)
Linear Prediction (PLP, LPCC, etc.)
Cepstral Coefficients (MFCC, etc.)
Low-Level Descriptors (frame-level)
Deriving (raw LLD, deltas)
Filtering (smoothing, normalising, etc.)
Chunking (absolute, relative, syntactic, semantic, etc.)
Functionals (super-segment-level)
Extreme (min, max, range, etc.)
Mean (arithmetic, quadratic, geometric)
Moments (variance, skewness, kurtosis)
Percenticles (quartiles, ranges, etc.)
Peaks (number, distances, etc.)
Temporal (durations, onsets, offsets, etc.)
Regression (coefficients, error)

Figure 2.3: Overview of the steps for generating acoustic features.

2.3.1 Low-Level Descriptors

Low-Level Descriptors (LLDs) are extracted at the frame-level – approximately 100 frames per second with a window size of 10-30 ms depending on the overlapping ratio. Windowing functions are usually the rectangular in the time domain, and the Hamming or the Hann in the frequency or time-frequency domain. Typical acoustic LLDs in the field can be grouped into prosodic, spectral, and voice quality features [53]. After extracting the raw LLDs, derived LLDs like deltas are often added to the feature sets. Furthermore, diverse filters (smoothing, normalising, etc.) may be applied. For the better introduction of LLDs, let us assume that $S(n)$ is a speech-signal frame after applying a window with N samples.

Prosodic Features

1) Pitch (F_0)

The fundamental frequency F_0 (or **Pitch**) plays a crucial role in the expression of paralinguistic cues via speech and thus is an important feature in the domain of computational paralinguistics. To extract the fundamental frequency, one of the most popular approaches, focusing on the signature in the time domain, uses the Auto Correlation Function (ACF) $R(k)$ [54], which is defined as

$$R(k) = \sum_{n=1}^{N-k} S(n)S(n+k), \tag{2.5}$$

where $R(k)$ depends on the time-shift k. The peak of $R(k)$ can be found at integer multiples of the period T_0 if voiced sound is uttered. Then, the fundamental frequency F_0 can be calculated as the reciprocal value of T_0 by $F_0 = \frac{1}{T_0}$.

2) Energy

The short-time **energy** of a speech-signal frame can be calculated as follows:

$$E = \log \sum_{n=1}^{N} |S(n)|^2. \tag{2.6}$$

Applying the logarithm accounts for the fact that the sensation of loudness increases logarithmically as the intensity of a stimulus grows. Usually, the short-term energy is normalised since parameters, such as the distance to the microphone, heavily influence the intensity of the recorded signals.

Spectral Features

1) Formants

Formants are defined as the spectral peaks which can be used as the distinguishing or meaningful frequency components of human speech. Essentially, it is related to the human vocal tract. Different shapes of human vocal tract will contribute to different reinforced frequency zones (resonating frequencies). From the low to high frequency, they are usually referred to as F1, F2, and F3. Among them, F1 is almost inversely correlated to the height of tongue body. The higher the tongue is, the lower the frequency of F1 is, and *vice versa*. Whereas, F2 is determined, though not entirely, how far back or how far forwards the tongue body is: The further back the tongue is, the lower the frequency of F2, and vice versa. Most often, the two formants F1 and F2 are enough to disambiguate the vowels. Compared to F1 and F2, F3 has no high relationship with the position of a tongue but with the activity of a tongue tip. To estimate the formant frequencies and bandwidths, the methods based on Linear Prediction Coding (LPC) can be used [55].

2) TF-Transformation

Rather than viewing a one-dimensional signal in the time domain, **Time-Frequency Transformation** (TF-Transformation) focuses on a two-dimensional spectrum in the time and frequency domains. One of the most basic types of TF transform is the Short-Time Fourier Transform (STFT) [56]. A typical scheme is **Gabor transform** [57]. It is used to determine the sinusoidal frequency and phase content of local sections of a signal as it changes over time. The function to be transformed is firstly multiplied by a Gaussian function, which can be regarded as a window function. The resulting function is then transformed with a Fourier transform to derive the time-frequency analysis. The window function means that the signals near the time being analysed will have higher weight.

A more sophisticated TF-Transform approach has been developed, namely **wavelet transformation** [57]. Its goal is to divide a given function or a continuous-time signal into different scale components. Usually, one can assign a frequency range to each scale component by using Heisenberg boxes. A wavelet transform is the representation of a function by wavelets. The wavelets are scaled and translated copies (or 'daughter wavelets') of a finite-length or fast-decaying oscillating waveform (or 'mother wavelet'). Wavelet transforms have the advantage over traditional Fourier transforms for representing functions that have discontinuities and sharp peaks, and for accurately deconstructing and reconstructing finite signals.

3) Spectrum

Both **Pitch Class Profile** (PCP) and **Chroma** are interesting and powerful representations of spectrum for the tonal analysis of music. PCP feature vectors represent the spectral energy distribution in the pitch classes according to the semitone band of Western music. Rather than the storing and analysing each individual musical semitone's energy (normally it includes a feature vector size of 36, 24, or 12 bands), the main idea of Chroma features is to project the entire spectrum onto 12 bins representing the 12 distinct semitones (or chroma) of the music octave.

4) Linear Prediction

Linear Prediction (LP) uses the estimated linear function of previous samples to predict the future values of discrete-time signals. A well-known derivation of LP analysis is **Linear Prediction Cepstral Coefficients** (LPCCs), which is the representation of Linear Prediction Coefficients (LPC) in the cepstrum domain. The idea of LPC is based on the speech production model. The characteristic of the vocal tract can be modelled by an all-pole filter that is equivalent to the smoothed envelope of the log spectrum of speech. LPC can be computed either by the autocorrelation or covariance methods directly from the windowed portion of speech [58]. With the obtained LPC, the LPCC can be calculated:

$$LPCC(i) = LPC(i) + \sum_{j=1}^{i-1} \frac{j-i}{i} \cdot LPC(j) \cdot LPCC(i-j), \tag{2.7}$$

where i and j are the LPCC index. LPCC has been widely used for speech recognition and has been proven to be more robust and reliable than LPC.

However, one disadvantage of LPCC is that it approximates the short-term power spectrum equally well at all frequencies of the analysis band. **Perceptual Linear Prediction** (PLP) does nevertheless take the psychophysics of human hearing into account to derive an estimate of the auditory spectrum: Beyond about 800Hz, the spectral resolution of hearing decreases with frequency. In addition, for the amplitude levels typically encountered in conversational speech, hearing is more sensitive in the middle-frequency range of the audible spectrum. Therefore, PLP analysis is more consistent with the sensitivity of human hearing to the changes of several important speech parameters [59].

5) Cepstral Coefficients

The most popular cepstral coefficients are **Mel-Frequency Cepstral Coefficient** (MFCCs). They are widely applied in ASR as they can efficiently encode spoken content while being relatively independent of speaker characteristics. They take the non-linear frequency perception of human ear into account, and use triangular overlapping filters to map the spectral powers onto the Mel scale:

$$Mel(f) = 2595 \cdot \lg\left(1 + \frac{f}{700}\right). \tag{2.8}$$

Then, they take logs with base 10 at each Mel frequency. Afterwards, MFCCs $c(i)$ are calculated from the log filterbank amplitudes $m(l)$ by applying the Discrete Cosine Transform (DCT):

$$c(i) = \sqrt{\frac{2}{N}} \sum_{j=1}^{N} \lg\left(m(j)\cos\left((j - 0.5)\frac{i\pi}{N}\right)\right), \tag{2.9}$$

where N is the number of filterbank channels.

Voice Quality Features

1) Perturbation

The most important parameters for speech signals are pitch, energy, formants, and their associated bandwidths. Also, micro-perturbations of the fundamental frequency and the intensity are sometimes still of interest. They reflect voice quality properties such as breathiness and harshness, and can be computed from pitch and energy contours, respectively. In the following, n' denotes the order of frames.

Jitter (J) denotes period-to-period fluctuations in the fundamental frequency and is calculated between successive pitch periods (T_0) of the signals:

$$J(n') = |T_0(n') - T_0(n' - 1)|, \text{for } n' > 1. \tag{2.10}$$

Shimmer (Sh) is computed as the average (over one frame) of the relative peak amplitude differences. In an analogy to jitter, the local period-to-period shimmer is expressed as follows:

$$Sh(n') = |A(n') - A(n'-1)|, \text{for } n' > 1, \tag{2.11}$$

where A is the peak to peak amplitude difference $A(n') = S_{\max(n')} - S_{\min(n')}$.

2) Harmonics-to-Noise Ratio

Harmonics-to-Noise Ratio (HNR) is another important parameter for evaluating the speech quality. It gives the energy ratio of the harmonic signal parts to the noise signal parts, and it is estimated from the short-time autocorrelation functions (60 ms window) as the ratio of the ACF amplitude $R(*)$ at F_0 and the total frame energy:

$$HNR_{acf} = \frac{RT_0}{R0 - RT_0} \tag{2.12}$$

where RT_0 is the amplitude of the autocorrelation peak at the fundamental period, and $R0$ is the 0-th ACF coefficient (equivalent to the quadratic frame energy). Normally, the ratio will be further logarithmised in dB to avoid highly negative and varying values for low-energy noise.

2.3.2 Functionals

Since ASR is a kind of short-term related task, a frame (normally 10∼30 ms) is adopted as a standard feature extraction unit. Besides, most of early studies on computational paralinguistics also directly use the frame-level LLDs for building dynamic models like Hidden Markov Models (HMMs) [60, 61]. Nevertheless, more and more recent research has begun to use static features on the supra-segmental level due to the long-term essence of paralinguistic cues. These static features are generated by projecting the time-axis-along LLD contours into a set of feature vectors with descriptively statistic functionals [53, 4, 14, 34]. In this case, static modelling, such as Support Vector Machines (SVMs), can be freely selected to analyse patterns in speech.

The super-segment is obtained by cutting the consequential LLD series into smaller parts (i.e., a fixed number of frames, acoustic chunks, voiced/unvoiced parts, phonemes, syllables, words, or sub-turns) in the sense of syntactically or semantically motivated chunks below or equal to the turn level [21]. After that, functionals are applied per LLD or spanning over LLDs. The purpose is to further reduce the feature size, by way of projecting the time series of potentially unknown length to a scalar value per applied functional. The functionals comprise (shown in Figure 2.3): extremes (minimum, maximum, ranges, etc.), mean (arithmetic, quadratic, geometric), moments (variance, skewness, kurtosis, etc.), percentiles (quantiles, ranges, etc.), peaks (number, distances, etc.), temporal variables (durations, positions, etc.), and regression (coefficients, error).

Table 2.2: Turn duration (average [avg], minimum [min], maximum [max], and standard deviation [std]) and required transmission bandwidth for three transmission strategies (i.e., *raw* coded speech, *LLDs*, and *stat*istical feature set) and four corpora (i.e., AEC, ALC, NCSC and Agender; a detailed description of the databases and respective acronyms is given in Section A.2).

	turn length (s)				bandwidth (kb/s)		
Corpus	avg	min	max	std	raw	LLDs	stat
AEC	1.7	0.1	24.5	0.8	16∼40	51.2	7.3
ALC	11.4	1.5	61.8	14.2	16∼40	188.8	12.3
NCSC	3.1	0.9	21.2	1.8	16∼40	204.8	62.4
Agender	2.6	0.3	11.3	1.2	16∼40	92.8	5.5

In this thesis, the acoustic features are extracted by the open source toolkit openSMILE [62].

2.3.3 Challenge of Large Feature Size

In order to continuously update the acoustic models, it is necessary to collect and transmit the data from the real world via the Internet, whilst guaranteeing privacy protection. Concerning such requirements, a distributed structure is proposed in Section 3.3.1.

With this system structure, data have to be transmitted from terminals to servers. Three possible types of data can be adopted: raw coded speech, LLDs features, and statistical features. One of the key points which we need to consider is the bandwidth requirement of each transmission channel in the principle of bandwidth conservation. In Table 2.2, the bandwidth requirement is calculated on the official databases of the INTERSPEECH 2009–2012 Challenges [63, 64, 65, 66] for each of the coding strategies mentioned above. Note that the ITU-G.726 protocol was considered for coding raw speech, and single precision floating point – 32 bit – was used for LLDs and statistical feature sets. In the case of statistical features, as it is assumed that the vector dimensionality is always the same per turn (and so the transmission bandwidth will vary as a function of turn duration), the bandwidth size is calculated on the average turn duration in each data set. As can be observed from Table 2.2, with the exception of the pathology task, the statistical feature set requires less bandwidth than the remaining coding strategies.

However, such a system still requires a large bandwidth if considering an application scenario that involves a large number of users/devices. Similarly, the same requirement goes for the storage size in memory. Therefore, it is necessary to further reduce the dimensionality of feature space when considering the recognition performance and privacy protection.

There are at least two general solutions that can deal with the large feature size: feature selection and feature compression. A feature selection strategy takes the features' relevance, irrelevance, and abundance into account, and aims to select a subset that can predict the output with an accuracy that is comparable to the performance of the complete feature set. Typical methods which achieve this goal include wrappers, filters, and embedded routines [67] [68]. Some algorithms, such as minimum Redundancy Maximum Relevance (mRMR) [69] and random subset feature selection [67], are now well-developed and have been successfully applied to paralinguistic tasks [10, 70]. In this thesis, nevertheless, the use of feature selection algorithms will not be analysed since they has repeatedly been explored in many studies (e.g., [71]). Instead, the feature sets which have been previously optimised for the various paralinguistic tasks will be employed in this work. There are two main reasons for this. First, this thesis intends to focus only on the essential components of the distributed system. Feature selection techniques can easily be integrated into the system as a 'plug-in' [10]. Second, in order to directly and fairly compare the performance of the distributed system with the baseline performances of embedded systems, the same features sets should be used.

Feature compression generally refers to the methods that transform a high dimensional feature space into a lower dimensional one. Typical dimensionality reduction methods include Principle Component Analysis (PCA) [72] and Linear Discriminant Analysis (LDA) [73]. These methods have been implemented, for instance, in distributed face recognition, speaker identification, and speech recognition. Another family of methods for (lossy) compression is Vector Quantisation (VQ) [74], which has been very popular in a variety of research fields such as speech coding, image and video compression, and various pattern recognition applications (e.g., face detection, texture classification). There are many variations of VQ proposed in the literature, such as distance-based VQ [75], Histogram-based Quantisation (HQ) [76], lattice VQ, and address VQ [77].

In the context of ASR, a particular VQ compression algorithm, known as Split Vector Quantisation (SVQ) [78], is selected as a standard method to address this problem on the frame-level LLDs. Whereas, how efficient this SVQ compression algorithm is on the statistical features for distributed paralinguistic recognition? In this thesis, the theoretical knowledge of SVQ, as well as its experimental evaluation on distributed paralinguistic recognition are elaborated in Section 3.3.1 and 4.2.2.

2.3.4 Challenge of Feature Corruption

It is commonly admitted that the acoustic features of speech signals represent differently in the case of different circumstances of signal production, transmission, and recording [12]. These circumstances can generally be summarised in Table 2.3, which mainly contains:

Table 2.3: Factors to distort speech signals.

Environment	**Speaker**	**Device**
• Stationary noise: – Babble noise – White noise – Music noise – Driving noise • Non-stationary noise: – Side talk – **Reverberation**	• Age • Gender • Culture • Accents • Dialect • Style • Emotion • Production • Speed	• Microphone – Single channel – Multiple channel • Loudspeaker • Mobile phone

1) *Environments.* Environmental interferences consist of stationary noises like babble and white noise, as well as non-stationary noises such as side talk (additive noise) and speech reverberation (convolutive noise) distort clean speech, resulting in many challenges for speech analysis. Typical scenarios are relevant to cocktail party, talking in the rooms, etc.

2) *Speakers.* It is widely accepted that the speech signals could reflect manifold clues related to the speakers (e.g., age, gender, cultural background, region, characteristics, emotion/affect), which in turn increase the acoustic variety.

3) *Devices.* Different sampling frequencies, coding protocols, transmission schemes (single channel or multi-channel), and the performance of microphones and loudspeakers will lead to different levels of speech analysis.

Reverberation

Particularly, *reverberation* is one of the greatest challenges. It happens when the speech signals from the user reach the microphone with different time delays and amplitude attenuations, caused by the various surfaces in the acoustic enclosure, such as a living room. The speech signal received by a microphone is a sum of three components: (a) the direct path signal, whose power is inversely proportional to the square of the distance from the speaker [79]; (b) the early reflections from the walls, floor, ceiling, etc., which depend on the position of the speaker; and (c) the late reverberation, which depends mainly on the size of the room and reflective properties of the room surface. This reverberation is considered to be less dependent on the position of the speaker [12, 79].

In the past decades, extensive research has been carried out to handle the harmful effects of non-stationary convolutive noise of reverberation. These works can generally be categorised in two ways depending on the amount of distant-talk training data available. When limited or no training data are available, the close-talk acoustic models are used, but the incoming reverberated speech signal is compensated either in the signal or the feature domain. When a large amount of distant-talk data are available, the acoustic models can be retrained or adapted using Maximum Likelihood Linear Regression (MLLR) or Constrained MLLR (CMLLR). Frequently, a combination of the above two methods is used.

The approaches based on the signal or the feature domain locate in the *front-end* of ISA systems. Therein, the goal of the *signal-based* approaches is to enhance the reverberant signal from temporal or spectral information. A typical method involves *inverse filtering* using a known or unknown Room Impulse Response (RIR) [80]. If the RIRs are known *a priori*, the inverse filter precisely recovers the quality of the source signal. However, such a scenario is not generalisable since, in most cases, the RIRs are not known in advance. Altering the recording rooms, or changing the relative position between the speakers and microphones, would result in the variation of RIRs. Thus, it needs to estimate the RIRs, or the equivalent inverse filter online during decoding. Such blind deconvolution methods use techniques like long-term linear prediction [81], maximum-likelihood objective [82], NMF [19, 83], and the like. Another common method is *spectral subtraction* that treats the late reverberation as additive noise in the spectral domain [84]. In addition to these microphone-array-independent methods, microphone-array-dependent techniques like *beamforming* are also used [12]. The basic idea is to delay and sum the signals from each microphone, whereby, it is executed under the assumption that the target and the undesired signals are uncorrelated, which is not true for reverberation. Recently, some new approaches, known as likelihood maximising beamforming [85] and two-stage beamforming [86], have been demonstrated to provide potential advantages over the standard ones. These microphone-array-based techniques can be followed with most of the above-outlined inverse filtering techniques, and jointly achieve noise and reverberation reduction [12].

The *feature-based* approaches attempt to remove the effect of reverberation directly from the corrupted feature vectors (feature enhancement) or extract specific noise-robust feature sets. *Cepstral Mean Normalisation* (CMN) [87] is an effective feature enhancement approach for mitigating early reverberation, as well as its advanced version of long-term spectral subtraction, where longer analysis windows are required to calculate the mean values since the typical room reverberation time constants are much longer (e.g., centiseconds) than the traditional analysis window (e.g., 20 ms) [88]. Additionally, taking into account the noise and reverberation information, some new feature sets like *RelAtive Spectral Transform - Perceptual Linear Prediction* (RASTA-PLP) [89] and *Delta-Spectral Cepstral Coefficient* (DSCC) [90] have been designed and often hand-crafted. In comparison with Perceptual Linear

Prediction (PLP), RASTA-PLP assumes that human speech perception is less sensitive to the steady-state spectral factor, and thus applies a bank-pass filter to the energy in each frequency subband in order to smooth over short-term noise variations and to remove any constant offset resulting from static spectral colouration in the speech channel [89]. Similarly, DSCC features are motivated by the vast difference between the rates of change of power for speech and noise [90] compared to MFCCs.

In contrast to the signal- or feature-based approaches, the *model-based* approaches are applied in the *back-end* of ISA systems. It adjusts the parameters of the acoustic model to the statistical properties of reverberant feature vectors or tailors the decoder to the reverberant feature vectors. One or more adaptation techniques are applied, for example, *Maximum A Posteriori* (MAP) [91], *MLLR* [92], and *CMLLR* (or feature-space MLLR [fMLLR]) [93], to reduce the mismatch of HMMs trained on clean speech and reverberant speech [94]. MAP combines the prior knowledge of the model parameters and the information obtained from the adaptation data. Thus, it normally needs a large amount of adaptation data. MLLR shifts the means and variances of the Gaussian Mixture Models (GMMs) used across a number of distributions. By contrast, CMLLR is a feature adaptation technique that estimates a set of linear transformations for the features.

Despite the fact that only a few efforts have been made in computational paralinguistics, but many in ASR to address the issue of reverberation over the past decades [95, 96], it is still of great importance to bring some fresh insights into in the area of ASR via advanced techniques. In this thesis, a state-of-the-art neural network with Long Short-Term Memory is used for an evaluation on ASR.

2.4 Classification

Classification is the process of finding a model (or function) that describes and distinguishes data classes or concepts. The derived model is based on the analysis of a set of training data (i.e., data objects whose class labels are known). Using this model, one could predict the class of objects whose class label is unknown.

2.4.1 Support Vector Machines

Support Vector Machines (SVMs) are effectively and frequently-used for statistic machine learning, as well as for ISA [2, 63, 4]. SVMs are supervised learning models based on the concept of decision hyperplanes that define decision boundaries, i.e., planes that separate sets of objects having different class memberships. SVMs perform classification tasks by constructing a set of hyperplanes in a multidimensional space that separates cases of different class labels. The goal of SVMs is to maximise the separation between classes, which consists of finding the hyperplane that has the largest distance to the nearest training data point of any class (aka functional margin), since the larger the margin, the lower the generalisation error of the classification task. In practice, training instances belonging to two or more categories are used to determine the hyperplane that best discriminates amongst different classes (that with the widest possible gap). The testing instances are then mapped to this multi-dimensional space, and the predicted categories are defined based on which side of the gap they fall onto.

Formally, given a set of examples $(\mathbf{x}_i, y_i), i = 1, 2, \dots, n$, where $\mathbf{x}_i \in \mathbb{R}^d$ is a d-dimensional feature vector, and $y_i : \mathcal{Y} \in \{-1, +1\}$ is a corresponding prediction of each example. To separate the two classes from each other, a hyperplane is defined as the set of points $\mathbf{x}$ satisfying

$$\mathbf{w}^T \cdot \mathbf{x} + b = 0, \tag{2.13}$$

where $\mathbf{w}$ is a normal vector and b is a bias.

If the training data are linear separable, one can select two hyperplanes written as

$$\begin{aligned} \mathbf{w}^T \cdot \mathbf{x} + b &= 1, \\ \mathbf{w}^T \cdot \mathbf{x} + b &= -1, \end{aligned} \tag{2.14}$$

between which there are no points. Given a test point $\mathbf{x}_i$, if it is subjective to $\mathbf{w}^T \cdot \mathbf{x}_i + b \geq 1$, it will have prediction $y_i = 1$; similarly, if it is subjective to $\mathbf{w}^T \cdot \mathbf{x}_i + b \leq -1$, it will have prediction $y_i = -1$.

The goal is to maximise their distance (aka functional margin) $\frac{2}{||\mathbf{w}||}$, that is, to minimise $||\mathbf{w}||$ or $\frac{1}{2}||\mathbf{w}||^2$ in $(\mathbf{x}, b)$ s.t. $\forall i$, $y_i(\mathbf{w}^T \cdot \mathbf{x}_i + b) \geq 1$. By applying Lagrange multipliers $\boldsymbol{\alpha}, (\alpha_i \geq 0, i = 1, \dots, n)$, this constrained problem can be converted to finding the saddle point of such a Lagrange function:

$$L(\mathbf{w}, b, \boldsymbol{\alpha}) = \arg \min_{\mathbf{w}, b} \max_{\boldsymbol{\alpha} \geq 0} \{\frac{1}{2}||\mathbf{w}||^2 - \sum_{i=1}^{n} \alpha_i [y_i(\mathbf{w}^T \cdot \mathbf{x}_i + b) - 1]\}. \tag{2.15}$$

So, the saddle point must satisfy the following condition:

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial b} = \sum_{i=1}^{n} y_i \alpha_i = 0, \tag{2.16}$$

$$\frac{\partial L(\mathbf{w}, b, \boldsymbol{\alpha})}{\partial \mathbf{w}} = \mathbf{w} - \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i = 0. \tag{2.17}$$

Substituting Equations (2.16) and (2.17) into Equation (2.15), the optimisation problem amounts to maximise the following formula

$$\begin{aligned} W(\boldsymbol{\alpha}) &= \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i \alpha_j y_i y_j K({\mathbf{x}_i}^T, \mathbf{x}_j) \\ \text{s. t.: } & 0 \leq \alpha_i \leq C, i = 1, \dots, n, \\ & \sum_{i=1}^{n} \alpha_i y_i = 0 \end{aligned} \tag{2.18}$$

where $K({\mathbf{x}_i}^T, \mathbf{x}_j) = \mathbf{x}_i^T \cdot \mathbf{x}_j$, and C is a freely defined constant. To calculate $K({\mathbf{x}_i}^T, \mathbf{x}_j)$, the Sequential Minimal Optimisation (SMO) algorithm can be used [97]. Finally, the normal vector $\mathbf{w}$ is determined by

$$\mathbf{w} = \sum_{i=1}^{n} \alpha_i y_i \mathbf{x}_i. \tag{2.19}$$

Up to now, linear separable problems have been analysed. However, it could also be extended to a nonlinear problem by applying a kernel trick [98]. All $\mathbf{x}_i$'s are replaced by $\phi(\mathbf{x}_i)$, where ϕ provides the higher-dimensional mapping. Thus, the kernel can be rewritten as

$$K(\mathbf{x}_i^T, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T \phi(\mathbf{x}_j). \tag{2.20}$$

The common kernel functions are the polynomial kernel with order d

$$K(\mathbf{x}_i^T, \mathbf{x}_j) = (\mathbf{x}_i \cdot \mathbf{x}_j)^d, \tag{2.21}$$

and the Gaussian Radial Basis Function (RBF)

$$K(\mathbf{x}_i^T, \mathbf{x}_j) = \exp(\frac{1}{2\alpha^2} ||\mathbf{x}_i - \mathbf{x}_j)||)^2, \tag{2.22}$$

with standard deviation α.

For the multiclass problem, *one-versus-all* or *one-versus-one* strategies can be used to reduce the multiclass problem into multiple binary classification problems [99].

2.4.2 Decision Trees

A decision tree is a simple representation for classifying examples. In the tree structures, each internal (non-leaf) node is labelled with an input feature; the arcs coming from an internal node are labelled with a possible value of feature; and each leaf of the tree is labelled with a class or a probability distribution over classes. Its main objective is to capture some meaningful relationship between class and the values of the attributes.

An earlier algorithm is *Iterative Dichotomiser 3* (ID3) developed by Ross Quinlan in 1986 [100]. It begins with the original data set $\mathcal{D}$ as the root node. In each iteration, it iterates through every unused attribute of the data set $\mathcal{D}$ and calculates the entropy $H(\mathcal{D})$ or information gain $IG(A)$ of that attribute A. It then selects the attribute that has the smallest entropy value. The data set $\mathcal{D}$ is then split by the selected attribute to produce subsets of data. The algorithm repeats on each subset until one of the following stopping criteria is met: 1) Every example in the subset belongs to the same class; 2) There are no more attributes to be selected; and 3) There are no examples in the subset. Then, the node is turned into a leaf with the most common class of the examples in the subset.

More specifically, for a given set $\mathcal{D}$ of examples $(\mathbf{x}, y)$, where $\mathbf{x} \in \mathbb{R}^d$ is a d-dimensional feature vector, and $y : \mathcal{Y} \in \{P, N\}$ (P: positive; N: negative) is a corresponding prediction of each example. Assume that $\mathcal{D}$ contain p examples of the class P and n examples of the class N, then the entropy can be calculated as

$$H(p,n) = -\frac{p}{p+n}\log_2\frac{p}{p+n} - \frac{n}{p+n}\log_2\frac{n}{p+n}. \tag{2.23}$$

If attribute A with values $A_1, A_2, \ldots, A_k$ is used for the root of the decision tree, it will partition data set $\mathcal{D}$ into k subsets $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_k$ where $\mathcal{D}_j$ $(j = 1, 2, \ldots, k)$ contains those examples that have value A_j $(j = 1, 2, \ldots, k)$ of A. Let $\mathcal{D}_j$ contain p_j examples of class P and n_j of class N. Then, the expected entropy remaining after trying attribute A (with branches $j = 1, 2, \ldots, k$) is obtained as the weighted average

$$H(A) = \sum_{j=1}^{k} \frac{p_j + n_j}{p+n} H(p_j, n_j). \tag{2.24}$$

Therefore, the information gained or reduction in entropy by branching on attribute A is

$$IG(A) = H(p,n) - H(A). \tag{2.25}$$

The attribute that can bring the most information gain (aka *gain criteria*) or reduction in entropy will be chosen to branch. Finally, use the above process recursively to form decision trees for the residual subsets $\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_k$.

A serious deficiency of this algorithm is that it has a strong bias in favour of the dominant class [101]. An extension algorithm of ID3 is *C4.5*. In contrast to the gain criterion employed for ID3, C4.5 uses the *Gain Ratio (GR) criterion* (or normalised information gain) for splitting data as follows

$$GR(A) = \frac{IG(A)}{H(A)}. \tag{2.26}$$

The attribute with the highest normalised information gain is chosen to make a decision. This kind of normalisation can rectify the bias-inherent problem in the gain criterion [101].

Furthermore, the above assumption for a two-class task can also be easily extended to any number of classes by altering the entropy function 2.23 to

$$H(\mathcal{Y}) = -\sum_{k=1}^{K} \frac{n_k}{\sum_{k=1}^{K} n_k} \log_2 \frac{n_k}{\sum_{k=1}^{K} n_k}, \tag{2.27}$$

where n_k denotes n examples belonging to the class k for a K-class task in a given data set.

In addition to the classification task, decision trees can also be used to deal with the regression task, known as the *Classification And Regression Tree* (CART) [102].

Random Forests

Decision tree is a simple and useful approach for pattern recognition, however, it will encounter a serious problem of over-fitting. To handle this issue, trunk-pruning is conducted for decision tree. Inspired by the ensemble algorithm of 'bagging' (cf. Section 3.1.3 for more details), the algorithm of random forests was first introduced by Leo Breiman [103]. The basic idea of random forests is to use several individual trees to make a final decision by taking the majority vote:

$$\hat{y} = \arg\max_{k_j \in K} \sum_{t=1}^{T} y_t(\mathbf{x}'_t), \tag{2.28}$$

where T is the predefined tree number, $\mathbf{x}'_t$ is the random subspace of the whole attributes for training the t-th tree. Therefore, each tree could be considered as an expert for a specific feature set. Normally, multiple trees are used depending on the size and nature of the training set. The majority vote by all trees could decrease the variance of models and ease the influence of over-fitting, thus making the classifier a strong predictor.

2.4.3 Long Short-Term Memory Neural Networks

Before giving an insight into Long Short-Term Memory (LSTM) neural networks, this section starts with the history of *Artificial Neural Networks* (ANNs). ANNs are mathematical models with the information processing capabilities of biological neural networks. It consists of a set of processing units or nodes that are jointed together by weighted connections. In terms of a biological model, the nodes represent the neurons, and the connection weights can be interpreted as the strength of the synapses between neurons. The network is activated by providing input to part/all of

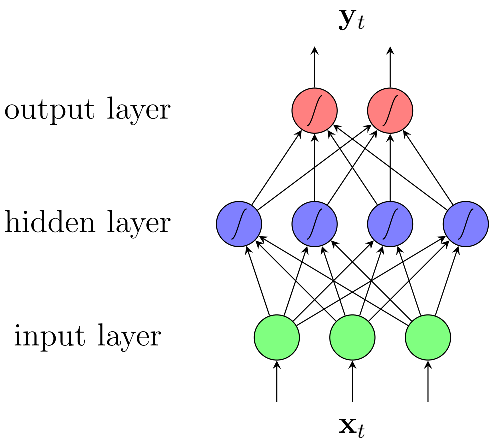


Figure 2.4: Multilayer Perceptron network.

the nodes. Then, the activations propagate through the network along the weighted connections. Based on whether the connections can form feedback loops (referred as feedback), ANNs can generally be categorised into *Feed-forward Neural Networks* (FNNs) and *Recurrent Neural Networks* (RNNs).

Multilayer Perceptrons

A well-known example of FNN is the *Multilayer Perceptrons* (MLP), whose nodes are arranged in layers, with connections feeding forward from one layer to the next one (see Figure 2.4). An MLP consists of an *input layer*, one or multiple *hidden layers*, and an *output layer*. Each layer is fully connected to all nodes of the subsequent layer, and no connections across multiple layers or within the same layer. The nodes or units in the hidden layers and output layers have (typically non-linear) *activation functions*.

In the context of pattern recognition, the activation of the input layer corresponds to the components of a feature vector ($\mathbf{x} = [x_1, x_2, \ldots, x_n]$). Each hidden or output node j receives its network input α_j from the I nodes in the preceding layer via the connections with the associated weights w_{ij} as follows:

$$\alpha_j = \sum_{i=1}^{I} w_{ij}\beta_i, \tag{2.29}$$

where w_{ij} denotes the weight from node i to node j, and β_i denotes the activation of the i-th node in the preceding layer. After applying an activation function f, the final activation of the j-th node can be written as

$$\beta_j = f(\alpha_j). \tag{2.30}$$

The most commonly used activation functions f are the non-linear hyperbolic tangent

$$\tanh(x) = \frac{e^{2x} - 1}{e^{2x} + 1}, \tag{2.31}$$

or the logistic sigmoid

$$\sigma(x) = \frac{1}{1 + e^{-x}}. \tag{2.32}$$

The activations of the output nodes are then obtained as $\mathbf{y}$, indicating the actual classification or regression results.

The network learning or training process is considered as the one of iteratively updating the weights of the networks by using *backpropagation* [104, 105], with the purpose of minimising the objective function (or cost function). Normally, the objective function is the *Sum of Squared Errors* (SSE) between the desired outputs (or targets) $\mathbf{z}$ and the actual network outputs $\mathbf{y}$ as follows:

$$\mathcal{J}(\theta) = E(\mathbf{z}, \mathbf{y}) = \frac{1}{2}\sum_{i=1}^{K}(z_i - y_i)^2, \tag{2.33}$$

where K is the number of output nodes. Then, the weight changes Δw_{ij} for each weight w_{ij} is obtained by

$$\Delta w_{ij} := \frac{\partial E}{\partial w_{ij}} = \delta_j \beta_i, \tag{2.34}$$

where δ_j is the error of the j-th node computed by the derivatives of the objective function (Equation (2.33)) with respect to this node. Assuming that the output activation function is a sigmoid function, then, if the j-th node is in the output layer, its error can be calculated as

$$\delta_j = \beta_j(1 - \beta_j)(\beta_j - z_j), \tag{2.35}$$

where β_j equals to y_j. In the hidden layer, the error δ_j is alternated into

$$\delta_j = \beta_j(1 - \beta_j)\sum_{k=1}^{K}\delta_k w_{jk}, \tag{2.36}$$

where k denotes the k-th node in the output layer.

During the network's training, some essential problems cannot be ignored, such as converging rate and over-fitting. To overcome the converging problem, a multitude of techniques have been proposed. *Gradient descent learning* is used to adjust the weights towards the negative error gradient in small steps [104]:

$$w_{ij,t+1} = w_{ij,t} - \lambda \Delta w_{ij,t}, \tag{2.37}$$

where $\lambda \in [0, 1]$ is the *learning rate*, which can be adaptively changed by the learning process [106].

As the gradient descent tends to get stuck in local minima, a so-called *momentum* term is normally added to avoid such a local minima, aiming to speed up the learning process [104]:

$$w_{ij,t+1} = w_{ij,t} - \lambda \Delta w_{ij,t} + \eta \Delta w_{ij,t-1}, \tag{2.38}$$

with the momentum $\eta \in [0, 1]$.

As the training iteration increases, the training error (see Equation (2.33)) will decrease more and more, and the network will become better and better to produce the desired outputs of the training set. At the same time, however, it will be more and more divergent in processing the test set. To cope with such an *over-fitting* phenomena, *generalisation* approaches are required. One approach is adding a penalty term in the objective function to avoid the weights changing too large [107, 108]:

$$\mathcal{J}(\theta) = E(\mathbf{z}, \mathbf{y}) + \frac{\gamma}{2} \sum_{i=1}^{I} \sum_{j=1}^{J} ||w_{ij}||^2, \tag{2.39}$$

where $\gamma \in [0, 1]$ is named *weight decay.* In this case, a new weight will be calculated by the iteration of w_{ij}

$$w_{ij,t+1} = w_{ij,t} - \lambda \Delta w_{ij,t} + \eta \Delta w_{ij,t-1} - \gamma w_{ij,t}. \tag{2.40}$$

In addition, there are some other tricks to avoid over-fitting like applying early stopping [109], or training with noise [109].

Recurrent Neural Networks

In contrast to FNNs whose connections did not form cycles, *Recurrent Neural Networks* (RNNs) allow cyclical connection, which consequently gives the RNNs the capability of accessing previously processed inputs. This section focuses on a simple RNN with self-connections in the hidden layer (referred to as Elman neural networks [110]), as shown in Figure 2.5. The self-connected hidden nodes can collect (weighted) activations not only from the input nodes but also from the hidden nodes in a previous time step. This implicitly allows a 'memory' of previous inputs that can be modelled in the internal state of the network.

Instead of merely considering *isolated* pattern vectors $\mathbf{x}$ as input for MLP, we now consider a *sequence* of vectors $\mathbf{x}_{1:T}$ with length T to a RNN with I input nodes, H hidden nodes, and K output nodes. Let $x_{i,t}$ be the value of the input i at time t, and let $\alpha_{j,t}$ and $\beta_{j,t}$ be the respective network input and the activation of the node j at time t. Thus, for the hidden nodes, we have

$$\alpha_{h,t} = \sum_{i=1}^{I} w_{ih} x_{i,t} + \sum_{h'=1}^{H} w_{h'h} \beta_{h',t-1}. \tag{2.41}$$

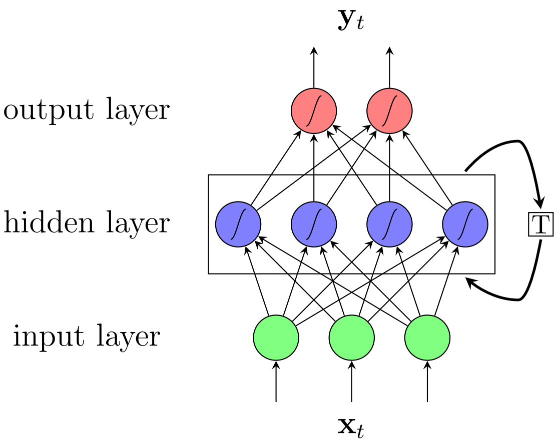


Figure 2.5: Recurrent Neural Network.

As to the backward pass, the partial derivatives of the objective function with respect to the weights are required. These derivatives can be determined via Back-Propagation Through Time (BPTT) [111, 105]. Similar to the backward pass for MLP, it is also a repeated application of the chain rule.

Bidirectional Recurrent Neural Networks

In sequential pattern recognition (e.g., handwriting, speech recognition), not only the past context but also the future context can improve the prediction capabilities [112, 113]. Unlike conventional RNNs that can only access the past inputs, bidirectional RNNs (BRNNs) [112] are proposed to access future inputs. The two separate recurrent hidden layers scan the input sequences in opposite directions [112]. As illustrated in Figure 2.6, the network calculates its forward hidden layer activations $\mathbf{h}_t^f$ from the beginning to the end of the sequence, and its backward hidden layer activations $\mathbf{h}_t^b$ from the end to the beginning of the sequence, then updates the output layer by

$$\mathbf{y}_t = \mathbf{W}_{fy}\mathbf{h}_t^f + \mathbf{W}_{by}\mathbf{h}_t^b + \mathbf{b}_y, \tag{2.42}$$

where $\mathbf{W}_{fy}$ and $\mathbf{W}_{by}$ are the forward and backward weight matrices, respectively, and $\mathbf{b}_y$ is the hidden bias vector. The forward and backward directed layers are connected to the same output layer, which can consequently access the whole context information.

Long Short-Term Memory

The conventional MLP, as outlined above, propagates the input signals unidirectionally layer-by-layer with sigmoid activations without any recurrent connec-

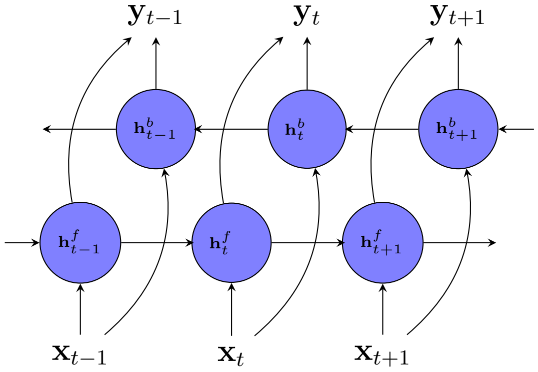


Figure 2.6: Structure of a bidirectional recurrent neural network.

tion. To exploit context information, one needs to stack several successive feature vectors as input [114]. Nevertheless, the capability of capturing context information is still limited [115]. Another method to address this issue is to employ RNNs, where the output of a previous time step is looped back and used as additional input. However, research shows that the standard RNNs cannot access long-range context since the backpropagated error either blows up or decays over time (the vanishing gradient problem) [116].

To overcome this limitation, the work in [17] introduced LSTM networks which are able to store information in memory cells over a long period. LSTM networks can be interpreted as RNNs in which the traditional neurons are replaced by so-called *memory blocks* (shown in Figure 2.7). Analogous to the cyclic connections in RNNs, these memory blocks are recurrently connected. Every memory block consists of self-connected linear memory cells and three multiplicative gate units: *input*, *output*, and *forget* gates. The input and output gates scale the input and output of the cell, respectively, while the forget gate scales the internal state. In other words, the input, output, and forget gates are responsible for writing, reading, and resetting the memory cell values, respectively. For example, if the forget gate is open and the input gate is closed (i.e., the input gate activation is close to zero), the activation of the cell will not be overwritten by new inputs. Therefore, the information from previous time t can be accessed by opening the output gate at the following arbitrary time steps.

In particular, for a memory block, the activation of the input gate i_t is composed of four components:

$$i_t = f_g\{\mathbf{W}_{xi}\mathbf{x}_t + \mathbf{W}_{hi}\mathbf{h}_{t-1} + \mathbf{W}_{ci}\mathbf{c}_{t-1} + b_i\}, \tag{2.43}$$

where $f_g\{\cdot\}$ denotes the logistic sigmoid function of the input unit; $\mathbf{W}_{xi}$, $\mathbf{W}_{hi}$, and $\mathbf{W}_{ci}$ are weight matrices of the connections from input gates, output gates, an forget

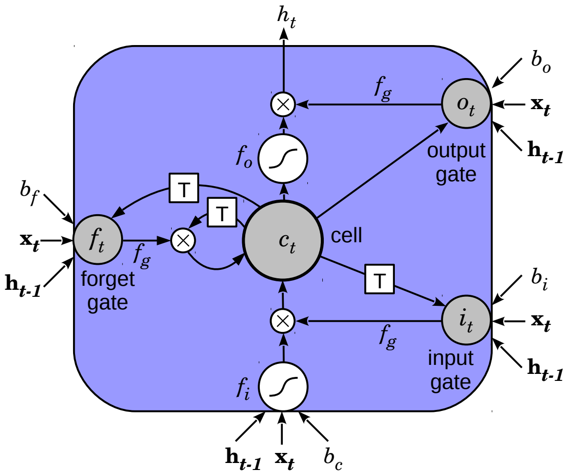


Figure 2.7: LSTM memory block. The symbols f_g, f_i, and f_o denote logistic sigmoid, tanh, and tanh activation functions, respectively; i_t, o_t, f_t are the activations of the input, output, and forget gates at time t, respectively; x_t, h_t, c_t represent input, output, and cell values of the memory block at time t, respectively; b is a bias.

gates in the same hidden layer to the input unit, respectively; $\mathbf{x}_t$ is the input vector; $\mathbf{h}_t$ is the hidden vector; and b_i is the unit bias. The activation of the forget gate f_t follows the same principle, and can be written as

$$f_t = f_g\{\mathbf{W}_{xf}\mathbf{x}_t + \mathbf{W}_{hf}\mathbf{h}_{t-1} + \mathbf{W}_{cf}\mathbf{c}_{t-1} + b_f\}. \tag{2.44}$$

The memory cell value c_t is the sum of the inputs at time step t and its previous time step activations that are multiplied by forget gate activation, and updated by:

$$c_t = i_t \cdot f_i\{\mathbf{W}_{xc}\mathbf{x}_t + \mathbf{W}_{hc}\mathbf{h}_{t-1} + b_c\} + \mathbf{f}_t \cdot \mathbf{c}_{t-1}, \tag{2.45}$$

where f_i is the tanh activation function. Finally, the output of the memory cell is controlled by the output gate activations of

$$o_t = f_g\{\mathbf{W}_{xo}\mathbf{x}_t + \mathbf{W}_{ho}\mathbf{h}_{t-1} + \mathbf{W}_{co}\mathbf{c}_t + b_o\}, \tag{2.46}$$

and delivered by

$$h_t = o_t \cdot f_o\{c_t\}, \tag{2.47}$$

where f_o is also a tanh activation function.

Note that each memory block can be regarded as a separate, independent unit. Therefore, if each memory block includes one memory cell, the activation vectors $\mathbf{i}_t$, $\mathbf{o}_t$, $\mathbf{f}_t$, and $\mathbf{c}_t$ are all of the same dimensional size as $\mathbf{h}_t$, i.e., the number of memory blocks in the hidden layer. From the formulas given above, it can be seen that the

Table 2.4: Confusion matrix.

Truth	Prediction	
	Positive	Negative
Positive	True Positive (t_p)	False Negative (f_n)
Negative	False Positive (f_p)	True Negative (t_n)

values from all the memory cells and block outputs at the previous time step $t-1$ in the same hidden layer will affect the activations of all input gates, output gates, and forget gates, as well as the input units at the current time step t. One exception is the case between the memory cell and the output gate that is the current state of memory cell $\mathbf{c}_t$ rather than the state from the previous time step, which contributes to forget gate activation.

Analogous to the standard ANNs, LSTM networks can be interpreted as differentiable 'function approximators' and can be trained by using BPTT in combination with gradient descent [117]. For more details, please refer to [118].

Overall, the memory cell of LSTM can store and access information over a long temporal range, thus avoid the vanishing gradient problem [17]. Therefore, one could also regard LSTM as a natural extension of *Deep Neural Networks* (DNNs) for temporal sequence data, where the depth comes from the layers through time.

To exploit both past and future context, LSTM neural networks can also be extended as *Bidirectional LSTM* (BLSTM) neural networks, as described for BRNNs. Each of the two separate recurrent hidden layers presents a training sequence forward and backward, and both of them are connected to the same output layer.

2.5 Evaluation Metrics

A number of measures have been defined to evaluate the performance of computational paralinguistics and speech recognition. Each of them only accesses one or several aspects of pattern recognition tasks, but none of them can be used to evaluate overall performance. The frequently-used measures in the present literature are Unweighted Average Recall (UAR), Weighted Average Recall (WAR), and Word Error Rate (WER) for classification tasks, and Coefficient Correlation (CC) for regression tasks.

Recall

Let us first define a two-class classification experiment from p positive instances and n negative instances. Then, the outcome can be calculated in a 2×2 confusion matrix as shown in Table 2.4.

The terms *positive* and *negative* in the first column refer to the ground truth or gold standard of data, and the ones in the second row denote the classifier's prediction (aka *expectation*). Furthermore, the terms of *true* and *false* indicate whether the prediction correctly corresponds to the external *observation*.

Recall is the proportion of the instances that are classified as class X among all the instances that are labelled as class X. Taking the positive class for example, it is defined as follows:

$$\text{Recall} = \frac{t_p}{t_p + f_n}. \tag{2.48}$$

This measure is helpful for evaluating the performance for each class separately, and for judging which classes can be distinguished better.

UAR and WAR

In order to evaluate the general performance of a classification over all classes, the most frequently-used and officially-recommended measure [63] is the Unweighted Average Recall (UAR) that emphasises the overall accuracy across all classes. It is defined as

$$\text{UAR} = \frac{\sum_{i=1}^{K} Recall_i}{\text{K}}, \tag{2.49}$$

where K is the number of classes.

Another common measure for classification is called Weighted Average Recall (WAR) that considers the unbalanced class distribution by the usage of the weight of each class.

$$\text{WAR} = \sum_{i=1}^{K} \lambda_i \cdot Recall_i, \tag{2.50}$$

where λ_i is the weight of the i-th class, and mathematically equals the proportion of the instance number of the i-th class over that of the whole data set.

In the case of a binary class task, the definitions of UAR and WAR are then simplified as

$$\text{UAR} = \frac{Recall_p + Recall_n}{2} \tag{2.51}$$

and

$$\text{WAR} = \lambda_p \cdot Recall_p + \lambda_n \cdot Recall_n. \tag{2.52}$$

The values of all these measures discussed above (i.e., Equation (2.49) – (2.52)) are in the interval [0,1]. The maximum value 1 indicates that all instances are perfectly predicted. In this thesis, the UAR and WAR are both adopted as the primary and secondary metrics, respectively, to evaluate the recognition performance of paralinguistic tasks in Sections 4.1 and 4.2.

Word Error Rate

For ASR, the general difficulty of measuring performance lies in the fact that the recognised word sequence might have a different length from the reference word sequence (the supposed correct one). One of the *de facto* standard measures applies to the Word Error Rate (WER) which is derived from the Levenshtein distance and works at the word level instead of the phoneme level. It is computed as

$$\mathrm{WER} = \frac{S + D + I}{N}, \tag{2.53}$$

where S, D, and I are the number of substitutions, deletions, and insertions, respectively, and N is the number of words in the reference. Note that the WER can be larger than 1.0 when the number of insertions I is large enough. In this thesis, it is used for speech recognition tasks in Section 4.3.

Correlation Coefficient

All the measures defined above are related to the classification tasks. For the regression tasks, Correlation Coefficient (CC, sometimes also called cross-correlation coefficient) is broadly used to evaluate the degree of correlation between two variables. There are several definitions for CC, among them is Pearson product-moment CC (PCC) which is one of the most frequently-used ones in statistics, which measures the strength and direction of the linear relationship between two variables. It is defined as the (sample) covariance of the variables divided by the product of their (sample) standard deviations:

$$\mathrm{CC} = \frac{cov(X, Y)}{\sigma_X \cdot \sigma_Y} = \frac{E[(X - \mu_X)(Y - \mu_Y)]}{\sigma_X \cdot \sigma_Y}, \tag{2.54}$$

where cov is the covariance, σ_X is the standard deviation of X, μ_X is the mean of X, and E is the expectation.

The value of CC varies from -1 to +1. The value of -1 indicates perfect negative correlation, 0 means no correlation, and +1 denotes perfect positive correlation. Note that the CC used in this thesis always refers to PCC.

3 Methodology for Data Enrichment and Optimisation

This chapter contributes to describing the approaches of data collection, selection, compression, and enhancement for the ISA systems. As most research topics (e.g., emotional speech recognition) in ISA are young and promising, such data processing work is still missing and is supposed to be the foremost work for building the ISA systems under real-life conditions. Here, the approaches of data processing, which are either transferred from the pre-existing classic algorithms in the community of other machine learning areas or proposed innovatively in this thesis, will be elaborately analysed.

From a technical point of view, data processing can be simply sorted into the concepts of *data enrichment* and *data optimisation*. *Data enrichment* is a general term which refers to the process of integrating the data that are available in the real world. There is an ever-lasting belief in the context of pattern recognition that 'there is no data like more data'. Although there are plenty of corpora comprising hundreds of hours of transcribed speech for ASR, the annotated data for other ISA tasks are still rare – in particular the publicly available ones as discussed in Section 2.2.2. The main purpose of data enrichment is to increase the *size* and *diversity* of speech data by, not only making good use of manifold labelled corpora, but also by exploiting the value of large-scale unlabelled data that are coloured with target information.

Typical approaches to data enrichment are related to data pooling, Semi-Supervised Learning (SSL), and Active Learning (AL). *Data pooling* is applied to merge pre-existing data from multiple, normally labelled but inconsistent, databases into one consistent database, whereas *SSL* or *AL* are used to semi or automatically develop the data that are missing labels.

Data optimisation is a general term regarding the process of improving data quality. The data coming from various sources, especially from real-life settings, tends to be dirty, incomplete, inconsistent, and heterogeneously distributed (cf.

Section 2.3.4). In the light of the principle that high-quality data will contribute to better performance of pattern recognisers, improving the data quality is also an essential step in the knowledge discovery process for ISA. The data quality in this thesis can be interpreted as 1) the most representative examples with the smallest data set; 2) the appropriate feature size and format that are suitable for the system structure; and 3) the least acoustic-characteristics mismatch between the training set and test set.

More specifically, data optimisation involves data balancing and data selecting. Generally, *data balancing* is used to adjust the data weights belonging to the minority class. *Data selecting* tries to choose the smallest, most representative data set, i.e., it seeks to eliminate redundant data that could result from environmental noise, seriously noise-contaminated data, high-uncertainty-labelled data, or even mislabelled data. When considering data distributions, data optimisation also refers to feature compression and enhancement. *Feature compression* can be applied to obtain the attributes with a reduced dimensionality, whereas *feature enhancement* attempts to boost the robustness of data by wiping off the additive or convoluted noise based on feature processing.

It is worth noting that the concepts of data enrichment and data optimisation are not mutually exclusive. For example, the process of data aggregation usually accompanies data selection; that is, only the data satisfied with predefined requirements will be chosen. Additionally, the goal of active learning methods is to integrate with the most informative data, which can also be viewed as a sort of data optimisation.

From an application point of view, this thesis simply categorises these techniques upon the processing objects – *labelled data*, *unlabelled data*, and *features*. These techniques will be presented in detail in Sections 3.1 to 3.3, respectively. Note that the concept of data is referred to as *instance*, *turn*, *record*, or *examples* in most sections, except Section 3.3, in which data means *attribute* or *feature*.

3.1 Exploiting Labelled Data

In the recent development of computational paralinguistics, especially speech emotion recognition, several databases (cf. Appendix A) have been released as benchmark sets that can be shared among researchers [119, 3, 2]. Hence, in general, the most straightforward way to increase the data is to reuse the existing labelled data. However, to build application systems in realistic conditions, almost all existing databases have limitations in the aspects of, for example, the naturalness of uttered emotional speech, emotional categories, class distributions, devices, languages, as well as acoustic backgrounds [119, 3, 2] (cf. Section 2.2.2).

To cope with these limitations and make the best advantage of each labelled database, many methods have been proposed in the context of machine learning, for example, data pooling [120], ensemble learning [121], data balancing [122], and data selection [123, 124]. However, most of these methods are neither investigated in computational paralinguistics, nor dealing with specific issues, such as label uncertainty, in the subjective tasks. In this thesis, these approaches are re-evaluated for speech emotion recognition as a representation task for general paralinguistics.

To do this, we firstly assume a database $\mathcal{D}$ that contains n examples, with the form $\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$, where $\mathbf{x} \in \mathcal{X}$, $y \in \mathcal{Y}$. The symbol of $\mathcal{X}$ represents a d-dimensional feature space $\mathbb{R}^d$, so $\mathbf{x} = [x_1, x_2, \ldots, x_d]$. Additionally, the symbol of $\mathcal{Y}$ denotes a prediction space, either from a discrete set of classes $\{1, 2, \ldots, k\}$ in the case of classification, or from continuous values in the case of regression. The goal of a learning algorithm $\mathcal{A}$ is to find a *decision function* $f : \mathcal{X} \mapsto \mathcal{Y}$ that correctly predicts the output of y from a future input drawn from the same distribution of $\mathcal{X}$. Finally, a classifier h characterised with the decision-function parameters is delivered. When there are q various labelled databases, these databases can be marked as $\{\mathcal{D}_1, \mathcal{D}_2, \ldots, \mathcal{D}_q\}$.

Before going into the details of each approach, we will first face the issue that the categories, $\mathcal{Y}$, differ among databases. To address this issue, the dimensional model of emotions offers an elegant solution, as emotion categories can be mapped onto coordinates to generate a unified set of labels. For example, the emotions can be mapped onto the arousal-valence dimensional coordinates [125, 126] from the cognitive psychology aspect. This mapping strategy is applied in the rest of this thesis. The specific way is displayed in Table A.3 and A.4.

3.1.1 Label Uncertainty

As to the tasks with subjective speech phenomena, such as the emotion and intoxication states, labels are determined by several labellers' personal judgement [52, 127] (cf. Section 2.2.1.2). Most previous studies on the speech pattern analysis, however, simply treated the labels as certain 'ground truth' [63, 65] without considering the label uncertainty among annotators. This, in fact, does not exactly express the whole pattern information embedded in speech [127]. In this thesis, the label uncertainty information is explored via data selection in Section 3.1.5.2 to well mine the most representative labelled examples.

A variety of methods can be employed to measure the label uncertainty (aka *human agreement level*) among human inter-raters. For example, the Spearman's rank correlation coefficient *rho* and the Pearson's intraclass correlation coefficient r are particularly suited for ranked intervals, albeit only for two raters [128, 129]. The same limitations also apply to the Scott's *pi* and Cohen's *kappa* [130, 131]. In this thesis, I employ the frequently-used Fleiss' Kappa coefficient [132], which is claimed to be a multi-rater generalisation of Scott's *pi*. It requires all raters to rate all data and is suited for larger data sets. It is defined as:

$$\kappa := \frac{p_0 - p_c}{1 - p_c}, \tag{3.1}$$

where p_0 is the observed agreement of annotators, and p_c is the chance-level agreement. For a single instance, the calculation of p_0 can be simplified by estimating the proportion of cases in which labellers agree on a common category:

$$p_0 = \frac{1}{m}\sum_{i=1}^{m} A_i, \tag{3.2}$$

where $A_i \in (0, 1)$ stands for a binary annotation of a specific category, and m is the number of annotators. Thus, the difference $p_0 - p_c$ indicates the proportion of the cases where 'beyond-chance agreement' occurs. It is normalised by the probability of disagreement $1 - p_c$ that is expected by chance.

3.1.2 Data Pooling

The idea of data pooling is integrating multiple databases with different distributions into a large size of pool $\mathcal{P} = \mathcal{D}_1 \cup \mathcal{D}_2 \cup \ldots \cup \mathcal{D}_q$. In this case, the mismatches over corpora exist due to the varieties of recording settings and languages. Thus, normalisation techniques are required to ease the mismatches among languages, speakers, and acoustic environments, so that the feature values can be unified within a small specified range, such as -1 to 1. In the ongoing, I introduce three kinds of data normalisation methods: *centring*, *normalisation*, and *standardisation*. Note that, apart from the data pooling, the process of data normalisation is also of importance for other methods dealing with labelled and unlabelled data, as well as features. For example, the feature sets are always normalised before being fed into neural networks, which helps in speeding up the learning phase [133, 134].

Centring is equal to the simple subtraction of the feature-wise mean, which corresponds to the following formula:

$$\mathbf{x}' = \mathbf{x} - \bar{\mathbf{x}}, \tag{3.3}$$

where $\bar{\mathbf{x}}$ is the mean feature vector over a specific corpus $\mathcal{D}_i$.

Min-max (range) normalisation forces the range of each feature to a predefined interval [a, b] by linear scaling. Suppose that $\mathbf{x}_{\min}$ and $\mathbf{x}_{max}$ are the minimum and maximum feature vectors over a specific corpus $\mathcal{D}_i$. When such a normalisation is applied, the representations per instance can be calculated by

$$\mathbf{x}' = \frac{\mathbf{x} - \bar{\mathbf{x}}}{\mathbf{x}_{max} - \mathbf{x}_{\min}} \cdot (b - a) + a. \tag{3.4}$$

Specifically, if the interval of [a, b] corresponds to [0,1], equation 3.4 evolves into

$$\mathbf{x}' = \frac{\mathbf{x} - \bar{\mathbf{x}}}{\mathbf{x}_{max} - \mathbf{x}_{\min}}, \tag{3.5}$$

and if it corresponds to [-1,1], it is altered by

$$\mathbf{x}' = 2 \cdot \frac{\mathbf{x} - \bar{\mathbf{x}}}{\mathbf{x}_{max} - \mathbf{x}_{\min}} - 1. \tag{3.6}$$

It is worth noting that an 'out-of-bounds' error could take place if an unknown future input case falls outside of the original data range. In this case, a simple way is assigning the extreme values (i.e., -1 or 1) to the 'out-of-bounds' data.

Standardisation (sometimes termed z-score normalisation) refers to the linear scaling to zero mean and unit variance, which is expressed as:

$$\mathbf{x}' = \frac{(\mathbf{x} - \bar{\mathbf{x}})}{\boldsymbol{\sigma}}, \tag{3.7}$$

where $\bar{\mathbf{x}}$ and $\boldsymbol{\sigma}$ are the mean and standard deviation vectors over the whole corpus. Compared to min-max normalisation, standardisation is more robust to outliers.

These three methods can not only be applied to each corpus separately (i.e., before data agglomeration), but also be used after building a joint training set from multiple databases, where the parameters $\{\bar{\mathbf{x}}, \mathbf{x}_{\min}, \mathbf{x}_{max}, \boldsymbol{\sigma}\}$ are extracted.

3.1.3 Ensemble Learning

In contrast to most machine learning approaches that consist in training *one* classifier, ensemble methods try to construct *a set of* classifiers (aka *weak* or *base learners*) and then classify new data by taking a weighted or unweighted vote through individual predictions. The robustness and effectiveness of ensemble methods are empirically and extensively verified in [135] and [136], where it found out that the predictions made by the fusion of a set of learners often perform better than the best single classifier. Prominent approaches include Bagging [137], Boosting [135, 136], and Stacking [138].

Bagging (aka bootstrap aggregating) [137] is an approach to training a plurality of weak learners, each of which is trained with different sets of bootstrap examples by a base learning algorithm. The bootstrap examples are obtained by randomly subsampling the training data set with replacement, where the number of the examples is the same as that of the training data set. Hence, some training examples may not appear, and some may appear even more than once. After obtaining a set of base learners h_i, $i = 1, \ldots, r$, Bagging combines their predictions by means of majority voting, and the most-voted class is predicted. That is, the final classifier is delivered by

$$\mathcal{H}(\mathbf{x}) = \arg\max_{y \in \mathcal{Y}} \sum_{i=1}^{r} 1(y = h_i(\mathbf{x})), \tag{3.8}$$

where the value of $1(a)$ is 1 if a is true and 0 otherwise. The pseudo-code description of Bagging is presented in Algorithm 1.

Algorithm 1: Bagging.

```
Input:
$\mathcal{D}$: training set $\mathcal{D} = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\}$;
$\mathcal{A}$: certain base learning algorithm;
$r$: number of learning rounds.
Output:
$\mathcal{H}$: final classifier, $\mathcal{H}(\mathbf{x}) = \arg\max_{y \in \mathcal{Y}} \sum_{i=1}^{r} 1(y = h_i(\mathbf{x}))$.
Process:
for $i = 1, \ldots, r$: do
    Generate bootstrap examples from training set $\mathcal{D}$, $\mathcal{S}_i = Bootstrap(\mathcal{D})$.
    Train the weak learner $h_i$ on the bootstrap examples, $h_i = \mathcal{A}(\mathcal{S}_i)$.
end
Ensemble the $r$ weak learners by (un-)weighted majority voting, and finally
deliver the output.
```

Upon Hansen and Salamon's suggestion in 1990 – a necessary and sufficient condition for an ensemble of weak learners to be more accurate than any of its individual members is if the base learners are accurate and diverse [121]. Thus, a variant of Bagging, also known as *cross-validation committees*, has been proposed in [139]. This method constructs base learners by leaving out disjoint subsets in the training data set.

Boosting consists in calculating the output using several different models and then averaging the results via a weighted average approach. Among a number of variants, which was developed by Freund and Schapire [140], **AdaBoost** is considered as the most popular boosting algorithm. Generating r classifiers for the ensemble requires r rounds through the algorithm. First, it assigns each training example an equal weight of $1/n$. Let's denote the distribution of these weights at the i-th learning round as $\mathcal{W}_i$. Upon the training data set $\mathcal{D}$ and $\mathcal{W}_i$, the algorithm generates a base learner h_i by calling the base learning algorithm. After that, it uses h_i to classify training examples, and the weights of the misclassified examples will increase. Then, an update weight distribution $\mathcal{W}_{i+1}$ is obtained. Again, upon the training set $\mathcal{D}$ and updated $\mathcal{W}_{i+1}$, AdaBoost generates another base learner. Such a process is repeated r times, leading to r base learners. The final strong learner is derived by weighted majority voting, where the weights of base learners are determined by a corresponding measured error rate on the training data set.

Stacking, in contrast to Bagging and Boosting that use a *one*-base learning algorithm to train all base learners, employs different learning algorithms $\mathcal{A}$'s (e.g., SVM and Decision Tree) to train individual learners [138]. Those individual learners are then combined with a higher-level learner that determines the final predictions.

Generally speaking, extensively empirical studies have shown that no ensemble method outperforms other ensemble methods in all scenarios [141]. More analysis about the influence of base learner's number, issues of overfitting and generalisation are provided in [142, 143].

In this thesis, Bagging is selected as a representative ensemble approach to be evaluated, because it is easier to adapt with multiple datasets, where each individual database is considered as a dependent base learner training set. In this case, the training set $\mathcal{D}$ is replaced by the data pool $\mathcal{P}$, the bootstrap data set $\mathcal{S}_i$ by the single database $\mathcal{D}_i$, and the number of learning rounds r by the number of databases n.

3.1.4 Data Balancing

A lion's share of the ubiquitous speech data is labelled as neutral, yet only a small share of them are characterised with interesting information like emotion and intoxication. Such an issue of highly imbalanced class distribution often results in a recognition engine with a poor prediction for the target classes. Under the assumption of a two-class task ($k = 2$), for simplicity, let us define the subsets $\mathcal{D}_{\text{min}} \subset \mathcal{D}$ and $\mathcal{D}_{\text{maj}} \subset \mathcal{D}$, where $\mathcal{D}_{\text{min}}$ and $\mathcal{D}_{\text{maj}}$ are the subsets belonging to the minority and majority class in $\mathcal{D}$, respectively. Thus, $\mathcal{D}_{\text{min}} \cap \mathcal{D}_{\text{maj}} = \phi$ and $\mathcal{D}_{\text{min}} \cup \mathcal{D}_{\text{maj}} = \mathcal{D}$.

Over the past decade, a number of studies in the field of machine learning have already laid heavy emphases on tackling this issue [144, 48, 145]. From the technical point of view, these studies can generally be categorised into three groups [145]: 1) sampling methods; 2) cost-sensitive methods; 3) kernel-based methods.

Sampling is the processing of repeating pre-existing data, or regenerating new data to modify the imbalanced data distribution, so as to provide a more balanced class distribution. This process counts on the findings that a balanced data set can usually provide more performance gain compared to an imbalanced data set [146, 147]. One specific way in them is *random sampling*, either by *random oversampling* (aka *upsampling*) – randomly selecting a set of examples $\mathcal{D}'_{\text{min}}$ in the minority subset $\mathcal{D}_{\text{min}}$ and then adding them to the original training set $\mathcal{D}$, $\mathcal{D} = \mathcal{D} \cup \mathcal{D}'_{\text{min}}$, or by *random undersampling* (aka *downsampling*) – randomly selecting a set of examples $\mathcal{D}'_{\text{maj}}$ in the majority subset $\mathcal{D}_{\text{maj}}$ and then removing them from the original training set $\mathcal{D}$, $\mathcal{D} = \mathcal{D} \setminus \mathcal{D}'_{\text{maj}}$. Yet, it is worth mentioning that, in the case of undersampling, the process of removing examples from the majority class may cause the loss of important information pertaining to the majority class. Another frequently-used and effective way of data sampling is the *Synthetic Minority Oversampling TEchnique (SMOTE)* [148]. The idea of this method is to create a new set of artificial examples belonging to the minority class. First, randomly select one example belonging to the minority class, $\mathbf{x} \in \mathcal{D}_{\text{min}}$. Then, find out its k nearest neighbours in the minority class set upon a Euclidean distance with $\mathbf{x}$, and randomly choose one $\mathbf{x}_{\text{neighbour}}$ among those neighbours. After that, create a new example by

$$\mathbf{x}_{\text{new}} = \mathbf{x} + \delta \cdot (\mathbf{x}_{\text{neighbour}} - \mathbf{x}), \tag{3.9}$$

where $\delta \in [0, 1]$ is a random number. This process is repeated until a predefined number or percentage of the minority examples are created. The pseudo-code description of the SMOTE algorithm can be found in [148].

In comparison with these data sampling methods that are with the aim of creating an evenly-distributed new data set, **cost-sensitive** methods attempt to use different cost metrics that describe the costs for misclassifying any particular data examples [149]. Similarly, the **kernel-based** methods try to adjust the decision boundary [150] in order to yield better performance for the less representative class. A typical kernel-based learning diagram is SVM [151].

In this thesis, the upsampling method is always applied to address the data imbalance issue due to its less complexity of computation and wide popularity of implementation.

3.1.5 Data Selection

With the implementation of the data enrichment methods presented from Section 3.1.2 to 3.1.4, the amount of labelled training data will increase, and the diversity of labelled training data will be enhanced. At the same time, however, they may also make for some problems [152]: 1) Data may be too overwhelming to be handled. In this case, the classifier training process could take a long time (Even though for most commercial applications the classifiers are usually trained in a once-off operation, a short training time is always desirable for researchers.) 2) There are redundant, mislabelled, and noise-distorted data fused into the prototypical data set, which damages the model performance.

These issues give rise to the necessity of data selection since the accurate decisions of a pattern recognition engine must be based on good quality data. The goal of data selection is to select a data source that is representative of the entire data universe of interest, and/or remove the superfluous and garbage data. Numerous methods have been proposed and investigated in the literature, and most of them can be assigned to one of the two groups from a technical point of view [152, 153].

The first data selection group is based on *wrappers*, where the selection criteria depend on the accuracy obtained by a classifier [153]. Those instances that do not improve predictive performance of the classification will be discarded from the training set. Most of the wrapper-based selection methods are relative to the k-nearest neighbour classifiers [154] like Condensed Nearest Neighbour (CNN) [155], Selective Nearest Neighbour rule (SNN) [156], or Decremental Reduction Optimisation Procedure (DROP) [157]. Taking CNN for example, the instances misclassified by the classifier will be selected and added into the initial training set.

Unlike the wrapper-based selection methods, the second data selection group is based on *filters* that attempt to select the instances by means of sampling or clustering, without depending on the prediction of classifier [153]. A prominent algorithm in them is RANdom SAmple Consensus (RANSAC) proposed by Fishler and Bolles [158]. It uses a set of data as small as possible to determine model parameters. Then, other data are tested via the fitted model, and those data that fit the estimated model within a predefined tolerance ϵ will be considered as part of the consensus set. When the fraction of the number of consensus data over the total number of data exceeds a predefined threshold, it re-estimates the model parameters using all consensus data and initial data. This procedure is repeated a fixed number of times. Another example comes to the Pattern by Ordered Projections (POP) [159] that discards the interior instances and selects some border instances. The border instance is defined if its nearest neighbour belongs to other classes, and the interior instance is defined in the other way around.

For ISA, especially for acoustic emotion recognition, however, only a handful of studies have placed their emphases on data selection so far [160]. To the best of my knowledge, almost all these studies just depended on the absolute labels (i.e., ground truth, or gold truth considered as ground truth), and none of these studies have directly exploited the agreement-level information of labelling (gold truth) for subjective tasks so far. In fact, due to the agreement levels among annotators, the instances with the same class may be with different label uncertainty, which is discussed in Section 2.2.1.2 and 3.1.1.

In the following, I introduce two methods to address the problem of data selection in the principle of 1) equal or improved performance: That means the performance of the model trained on a selected subset should be equal to, or better than, the performance of the model trained on all instances; and 2) reduction of training time. The two methods are Euclidean Distance-based Instance Selection, and Agreement- and Sparseness-based Instance Selection.

3.1.5.1 Euclidean Distance-based

The idea of *Euclidean Distance-based Data Selection* (**EDDS**) is to select the instances that are far away from the centre of the other classes (see Algorithm 2). Assume a two-class task with positive-negative dimensionality, where $\mathbf{x}_+$ and $\bar{\mathbf{x}}_+$ denote the feature vector of a particular positive instance and the averaged feature vector (centre) through all positive instances, respectively. The same definitions, $\mathbf{x}_-$ and $\bar{\mathbf{x}}_-$, apply to the negative class. Finally, only the instances that have long distances with the opposite class centre ($d(\mathbf{x}_+, \bar{\mathbf{x}}_-)$ and $d(\mathbf{x}_-, \bar{\mathbf{x}}_+)$) will be selected. In fact, this algorithm can also be considered for database selection.

3.1.5.2 Agreement- and Sparseness-based

Agreement- and Sparseness-based Instance Selection (**ASIS**) includes two steps: Agreement-based Instance Selection (AIS) and Sparseness-based Instance Selection (SIS) (see Algorithm 3). As discussed in Section 3.1.1, those instances labelled as the same class might have different label uncertainty among annotators. It is interest-

Algorithm 2: Euclidean Distance-based Data Selection (EDDS).

```
Input:
$\mathcal{D}$: Database of $n$ instances annotated in classes ;
$P$: Percentage of selected subset;
Output:
$\mathcal{S}$: Subset of database $\mathcal{D}$;
Process:
Obtain the proportional distribution of each class $R_+$, $R_-$ in the training set of $\mathcal{D}$.
Calculate the centres for positive data set $\bar{\mathbf{x}}_+ = E\{\mathbf{x}_+\}$ and negative data set $\bar{\mathbf{x}}_- = E\{\mathbf{x}_-\}$.
Calculate the Euclidean distance for each instance $\mathbf{x}_+$ or $\mathbf{x}_-$ to corresponding opposite class centre $\bar{\mathbf{x}}_-$ or $\bar{\mathbf{x}}_+$. That is, $d(\mathbf{x}_+, \bar{\mathbf{x}}_-)$ and $d(\mathbf{x}_-, \bar{\mathbf{x}}_+)$.
Sort the instances based on the distance values from high to low for both positive class ('positive' queue: $Q_+$) and negative class ('negative' queue: $Q_-$), respectively.
Select the most beginning $N_+$ ($N_+ = n \times P \times R_+$) instances in the 'positive' queue $Q_+$ and $N_-$ ($N_- = n \times P \times R_-$) instances in the 'negative' queue $Q_-$.
Fuse $N_+$ and $N_-$ into output subset $\mathcal{S}$.
```

ing to see what is the influence of these instances labelled by high-level uncertainty when building pattern recognition model. To this end, the AIS step aims to discard the instances with high-level label uncertainty. In this process, the instances are discarded proportionally over classes, whereby the proportion value depends on the original class distribution. On the one hand, it prevents the case of potential maldistribution of instances that might result in discarding such instances which mainly belong to certain classes, especially to the sparse ones; on the other hand, it improves the separability of classes by potentially removing the instances close to the class boundary in the feature space. Therefore, the larger the size of the discarded subset (P_D[%]) chosen, the fewer instances will be located near the class boundaries, and the less complex the model will become.

Moreover, the SIS step randomly selects an equivalent number of instances from each class set, so as to cope with the class imbalance problem that was introduced in Subsection 3.1.4. Note that, in the case of a class balanced task the size of the selected subset (P_S[%]) will satisfy $P_D + P_S \leq 1$, while in the case of a class imbalanced task, both P_D and P_S are constrained by the instance number of the most sparse class.

Algorithm 3: Agreement- and Sparseness-based Instance Selection (ASIS).

```
Input:
$\mathcal{D}$: Database of $n$ instances annotated in classes $C_i$ $(i = 1, \cdots, k)$ and corresponding human agreement levels $l$;
$P_D$: Percentage of discarded subset with low human agreement levels;
$P_S$: Percentage of selected subset;
$k$: number of classes;
Output:
$\mathcal{S}$: Subset of database $\mathcal{D}$;
Process:
Obtain the proportional distribution of each class $R_i$ $(i = 1, \cdots, k)$ in the training set of $\mathcal{D}$.
(Step: Agreement-based Instance Selection [AIS])
for $i = 1, \cdots, k$ do
    Sort the instances that are annotated as class $C_i$ by human agreement levels $l$ from low to high, producing queue $Q_i$.
    Delete $n_{Di} = n \times P_D \times R_i$ instances that are at the beginning of $Q_i$.
end
(Step: Sparsness-based Instance Selection [SIS])
for $i = 1, \cdots, k$ do
    Randomly select $n_{Si} = n \times P_S / k$ instances belonging to class $C_i$.
end
Fuse $n_{Si}$ $(i = 1, \cdots, k)$ into one output subset $\mathcal{S}$.
```

3.2 Exploiting Unlabelled Data

In contrast to the labelled data that are scanty, expensive, and time-consuming, unlabelled data are ubiquitously available in the real world and easily collected at little cost. This advantage motivates the exploitation of unlabelled data by advanced techniques that seek to directly extract the information embedded in the unlabelled data, so that the greatest performance gain can be delivered just by using a small human labelled data set.

In the past few decades, a multitude of methods in machine learning have been proposed and investigated, as will be introduced later. However, this thesis employs a *prediction uncertainty*-based learning strategy in terms of the *Confidence Value* (CV). This strategy involves the prediction work from the machine oracle (e.g., SSL), the human oracle (e.g., AL), and the combinations thereof (e.g., cooperative learning). The basic idea is to deliver the instances predicted with low uncertainty (or high CV) for the machine oracle, and the ones predicted with high/medium uncertainty (or low/medium CV) for the human oracle.

Input:
$\mathcal{L}$: small amount of labelled data (with gold standard);
$\mathcal{U}$: large amount of unlabelled data pool;
$\mathcal{S}$: selected data set;
k: number of classes;
n': number of selected instances for each repetition.
Output:
$\mathcal{H}$: enhanced emotion classifier.

Figure 3.1: Definitions of notations in the algorithms.

Before going into the algorithms from Section 3.2.2 to 3.2.4 in depth, Figure 3.1 provides the definitions of common notations. The initial labelled training set consists of inputs and outputs drawn from a joint distribution $\mathcal{L} = \{(\mathbf{x}_i, y_i)|(\mathbf{x}_i, y_i) \sim p(\mathbf{x}, y)\}_{i=1}^{l}$, and the unlabelled data pool from a marginal distribution $\mathcal{U} = \{\mathbf{x}_i|\mathbf{x}_i \sim p(\mathbf{x})\}_{i=l+1}^{l+u}$, typically $l \gg u$. Another point that needs to be mentioned is the stopping criterion of the learning process. There are several ways to stop the closed loop. The best time to stop training is when the anticipated performance is achieved. To do this, a separate validation set is involved. If there is no significant performance improvement in the validation set, the learning loop could be stopped [161]. Another way is that the loop could be automatically stopped once all the data in the unlabelled pool have been labelled by humans or machines. In this thesis, and for the sake of comparison, I adopt the strategy that the learning process stops when a predefined number of learning iterations are finished.

3.2.1 Prediction Uncertainty

Compared to the label uncertainty (cf. Section 3.1.1) which denotes the degree of labelling variability among annotators, the prediction uncertainty indicates the degree of predictions reliability of a classifier. In this thesis, the *prediction uncertainty* is derived from the poster probability of classification and measured by the level of *CV*.

For SVMs, as analysed in Section 2.4.1, the following function is implemented to classify a given test example:

$$f(\mathbf{x}) = \sum_{i}^{m} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) + b. \tag{3.10}$$

The sign of this function determines the category of the test example. Essentially, the output value of standard SVMs is the distance of a specific point from the separating hyperplane. Therefore, such standard SVMs are also known as the maximum margin algorithm.

To enable post-processing (e.g., for dealing with unbalanced data [162]), the output of a classifier should be a calibrated posterior probability $p(\text{class}|\text{input})$ [163] within the range of $[0, 1]$. To this end, there are various approaches including nonparametric and parametric methods [164]. As to the nonparametric methods, Zadrozny and Elkan proposed isotonic regression to transform the output of SVMs to probabilities, and demonstrated its effectiveness in [165]. Later on, they also reported a simple histogram method that outperforms isotonic regression for the properly chosen bin size [165]. As to the parametric methods, a typical one is to train a kernel directly with a regularised likelihood error function (e.g., logit link function). However, such an algorithm will produce non-sparse kernel machines. To overcome this limitation, Platt applied a sigmoid function to map the SVM outputs into posterior probabilities and demonstrated that the SVM+sigmoid combinations compete with the kernel methods. More specifically, it assumes that the posterior probability consists in finding the parameters A and B for a form of sigmoid function:

$$p(y|f(\mathbf{x})) = \frac{1}{1 + \exp(Af(\mathbf{x}) + B)}, \tag{3.11}$$

which maps the value $f(\mathbf{x})$ into the probability estimates $p(y|f(\mathbf{x}))$. For each instance, the sum of the posterior probability for all classes is equal to 1. In the special case of binary recognition tasks, the decision threshold is 0.5. Thus, the final prediction class is determined by comparing the posterior probability with 0.5.

However, sometimes we are not particularly concerned with the final prediction but more its uncertainty. For example, the prediction uncertainty can be used for SSL or AL. To measure such an uncertainty, a confidence value (CV) is proposed and can be calculated by the equation:

$$C(\mathbf{x}) = |p_0(\mathbf{x}) - p_1(\mathbf{x})|, \tag{3.12}$$

where $p_0(\mathbf{x}), p_1(\mathbf{x})$ are the posterior probabilities for classes '0' and '1', respectively. It can be seen that the CV is an indicator of the prediction uncertainty. The more certainty the predictor gives, the higher the CV. Its value ranges from 0 to 1.

3.2.2 Machine Oracle

Semi-Supervised Learning (SSL) techniques aim to use unlabelled data in an efficient way without any intervention from human annotators. A wide variety of methods exist [166], which can generally be distinguished as generative and discriminative models.

Generative model estimates the joint probability distribution $p(\mathbf{x}, y)$ of all variables, both the classes and the features. Common methods include the generative model with Expectation-Maximisation (EM). It assumes that the data (labelled or unlabelled) satisfy an identifiable mixture distribution like GMM, and use an EM algorithm to optimise the joint likelihood taking both labelled and unlabelled data into account [167]. It includes two steps: an expectation (E) step and a maximisation (M) step. In the E step, each unlabelled example is assigned a label distribution according to its expected value under the current model. In the M step, the multinomial parameters are re-estimated [168]. However, this method does not work when the independent input features are viable or when the best generative structure does not correspond to the decision boundary [169].

Discriminative model learns the probability distribution $p(y|\mathbf{x})$, that is, the probability of y given $\mathbf{x}$. It comprises Transductive SVM (TSVM) and graph-based methods [169]. TSVM is a variant of standard SVM with unlabelled data. The objective is to find labelling strategies for unlabelled data, so as to have a maximum margin on both original data and unlabelled data (labelled after the labelling process) [170]. One drawback of this method is that the margin may not exist if the classes strongly overlap. Graph-based methods define a graph where the nodes are labelled or unlabelled examples in the data set, and edges denote the similarity between examples [171]. The graph is usually constructed from the distances in the feature space. Indeed, this method is also sensitive to the overlapping classes.

In this section, I choose the discriminative models based on the prediction uncertainty of SVM classification as the evaluation methods, because SVM is frequently used for computational paralinguistics and does serve somewhat as a standard classifier. Two specific paradigms – self-training and co-training – are analysed.

3.2.2.1 Self-Training

Figure 3.2 and Algorithm 4 describe the flowchart and pseudocode of self-training. A classifier is firstly trained with a small amount of labelled data. After that, the classifier is used to recognise the unlabelled data. Typically, those unlabelled data that are classified with high confidence, together with their predicted labels, are added to the training set. The classifier is retrained, and the process is repeated. Formally, the selected example $\mathbf{x}' \in \mathcal{S}$ can be expressed as

$$\mathbf{x}' = \arg\max_{\mathbf{x}\in\mathcal{X}} C(\mathbf{x}). \tag{3.13}$$

Note that the classifier uses its own prediction to teach itself. Thus, this procedure is also named self-teaching.

Self-training is a widely used technique in SSL. It is simple and could be easily applied to an existing classifier. However, it also suffers from several drawbacks. Early mistakes could reinforce themselves, and be accumulated, possibly leading to a vicious circle of learning. Another drawback of self-training is the imbalanced class distribution. The prediction results are always biased to the most dominant class, and the opposite phenomenon occurs with the less representative one.

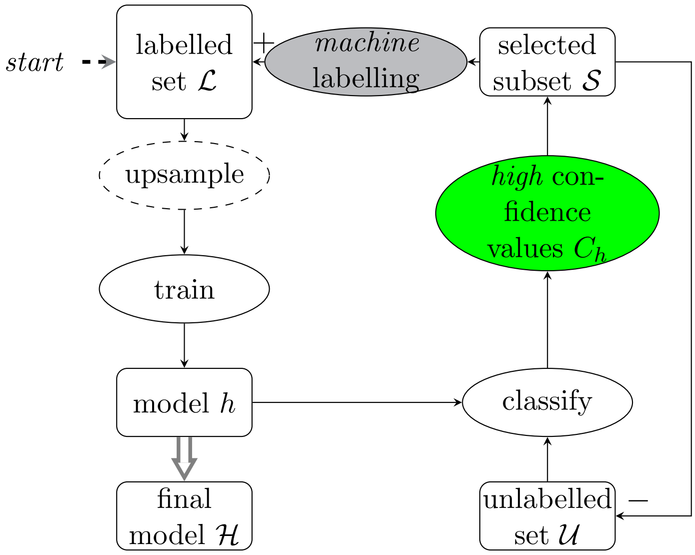

start
labelled set $\mathcal{L}$
+
machine labelling
selected subset $\mathcal{S}$
upsample
high confidence values C_h
train
model h
classify
final model $\mathcal{H}$
unlabelled set $\mathcal{U}$
−

Figure 3.2: Self-training.

Algorithm 4: Self-training.

```
repeat
    (Optional) Upsample training set $\mathcal{L}$ to even class distribution $\mathcal{L}_D$.
    Use $\mathcal{L}/\mathcal{L}_D$ to train a classifier $\mathcal{H}$, then classify $\mathcal{U}$.
    Select a subset $\mathcal{S}_{st}$ that contains those instances predicted with the
    highest confidence values.
    Remove $\mathcal{S}_{st}$ from the unlabelled set $\mathcal{U}$, $\mathcal{U} = \mathcal{U} \setminus \mathcal{S}_{st}$.
    Add $\mathcal{S}_{st}$ to the labelled set $\mathcal{L}$, $\mathcal{L} = \mathcal{L} \cup \mathcal{S}_{st}$.
until a predefined number of iterations is met.
```

3.2.2.2 Co-Training

Another SSL method is Multi-View Learning (MVL) [172, 166, 173], which focuses on improving the learning performance by training different models concurrently and optimising them by exploiting redundant feature sets (or 'views') of the same input data [169]. *Co-training* is one of the earliest schemes for MVL proposed in the literature [174]. Figure 3.3 and Algorithm 5 describe the flowchart and pseudocode of co-training.

We have a feature space $\mathcal{X} = \mathcal{X}^1 \times \mathcal{X}^2$, where $\mathcal{X}^1$ and $\mathcal{X}^2$ correspond to two different 'views' of an example, and $\mathcal{X}^1 \cap \mathcal{X}^2 = \varnothing$. That is, each example $\mathbf{x}$ is given as a pair of $(\mathbf{x}_1, \mathbf{x}_2)$. The two feature sets satisfy the following two assumptions [174]:

- *sufficiency* – Each view is sufficient for classification on its own. That is,

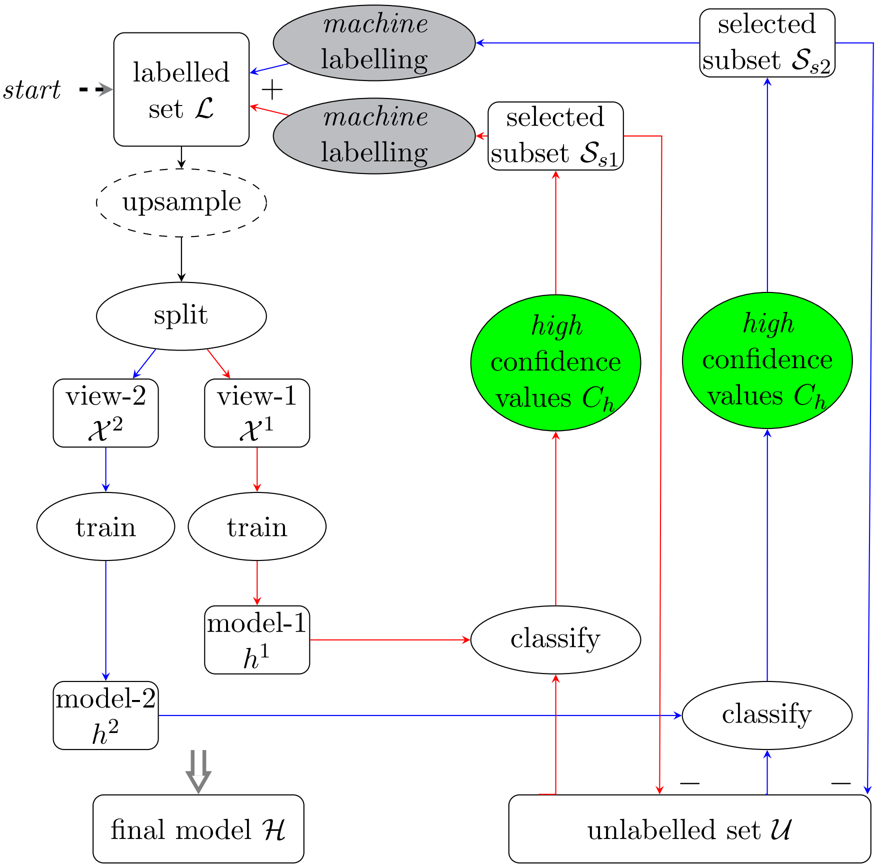

start
labelled set $\mathcal{L}$
machine labelling
machine labelling
+
selected subset $\mathcal{S}_{s2}$
selected subset $\mathcal{S}_{s1}$
upsample
split
view-2 $\mathcal{X}^2$
view-1 $\mathcal{X}^1$
train
train
high confidence values C_h
high confidence values C_h
model-1 h^1
classify
model-2 h^2
classify
final model $\mathcal{H}$
−
−
unlabelled set $\mathcal{U}$

Figure 3.3: Co-training.

the two hypotheses $f^1 : \mathcal{X}^1 \mapsto \mathcal{Y}$ and $f^2 : \mathcal{X}^2 \mapsto \mathcal{Y}$ are good enough for recognition.

- *conditional independence* – The views are conditionally independent given the class label [174], that is, $p(y_i|\mathbf{x}) \leftarrow p(y_i|\mathbf{x}_1)p(y_i|\mathbf{x}_2)$.

Initially, two models h^1 and h^2 are built on separate 'views', $\mathcal{X}^1$ and $\mathcal{X}^2$, respectively. Like self-training, each model then chooses the example $\mathbf{x}'^i \in \mathcal{S}$ from the unlabelled data pool $\mathcal{U}$ with the most confident predictions

$$\mathbf{x}'^i = \arg\max_{\mathbf{x}^i \in \mathcal{X}^i} C(\mathbf{x}^i), \tag{3.14}$$

where $i = 1, 2$ denotes the 'view' index. Both selected subsets S^1 and S^2 are finally added into the training set. Such a process is repeated several times until a predefined number of iterations is met. Essentially, each classifier is trained with its own data plus the additional training examples provided by the other classifier. Therefore, in one iteration, an instance is either discarded (low certainty predictions),

Algorithm 5: Co-training.

```
Input: (additional)
A learning domain with features $\mathcal{X}$.
repeat
    Divide the domain features $\mathcal{X}$ into two views: $\mathcal{X}^1, \mathcal{X}^2$, and $\mathcal{X}^1 \cap \mathcal{X}^2 = \varnothing$.
    for $i = 1, 2$ do
        (Optional) Upsample each view of data to even class distribution $\mathcal{X}_{Di}$.
        Use $\mathcal{X}_i/\mathcal{X}_{Di}$ to train classifier $\mathcal{H}_i$, and then classify $\mathcal{U}$.
        Select a subset $\mathcal{S}_{si}$ that contains those instances predicted with the
        highest confidence values.
    end
    Remove $\mathcal{S}_{ct} = \mathcal{S}_{s1} \cup \mathcal{S}_{s2}$ from the unlabelled set $\mathcal{U}$, $\mathcal{U} = \mathcal{U} \setminus \mathcal{S}_{ct}$.
    Add $\mathcal{S}_{ct} = \mathcal{S}_{s1} \cup \mathcal{S}_{s2}$ to the labelled set $\mathcal{L}$, $\mathcal{L} = \mathcal{L} \cup \mathcal{S}_{ct}$.
until a predefined number of iterations is met.
```

added once (high certainty predictions by one of the two classifiers), added twice with the same label (high certainty and similar predictions by the two classifiers), or added twice with different labels (high certainty but different predictions by the two classifiers).

Co-training makes strong assumptions on the splitting of features. However, in most application cases, such assumptions are hard to satisfy. Even so, co-training is still working in most applications and demonstrated empirically and theoretically [175, 176].

3.2.3 Human Oracle

As discussed in Section 3.2.2, SSL techniques can exploit the annotation work from machines without any human intervention, yet they often may not improve the performance as expected because of the issues of error accumulation and prediction inclination to the dominant class [169]. Alternatively, *Active Learning* (AL) [169] has the potential to achieve higher accuracy with fewer training labels by (actively) choosing the data from which it learns. Those instances that are 'most informative' for the task being modelled are selected from large pools of unlabelled data, and subsequently query an oracle or teacher for annotation. There are various strategies by which the informativeness of unlabelled examples can be processed (usually referred to as *query strategies*). One of the simplest strategies is to allow the model (or active learner) to determine the uncertainty of the predictions on unlabelled data based on a previously trained model (uncertainty sampling AL), and query an oracle (i.e., a human annotator) for the annotation of those with the least certain predictions [177]. Another common strategy is the so-called query-by-committee,

Algorithm 6: Passive Learning (PL).

```
repeat
    Randomly select subset $\mathcal{S}_p$ from unlabelled set $\mathcal{U}$.
    Ask human experts to label the selected subset $\mathcal{S}_p$.
    Remove $\mathcal{S}_p$ from the unlabelled set $\mathcal{U}$, $\mathcal{U} = \mathcal{U} \setminus \mathcal{S}_p$.
    Add $\mathcal{S}_p$ to the labelled set $\mathcal{L}$, $\mathcal{L} = \mathcal{L} \cup \mathcal{S}_p$.
until a predefined number of iterations is met.
```

whereby the predictions for unlabelled data are obtained from multiple models (previously) trained on the same data (typical models represent competing hypotheses to solve the same task). In this type of strategy, the data with the lowest agreement across classifiers are considered to be the most informative ones [178]. Other AL query strategies include the expected-error-reduction method that aims to measure how much its generalisation error is likely to be reduced [179], the expected-model-change-based method that chooses those instances that have a greater impact on the current model [180], and the diversity-density-related method that attempts to maximise the learning benefits of relevance feedback on retrieving documents [181].

In this thesis, I choose the query function based on the prediction uncertainty by SVM classification as mentioned in Section 3.2.1. In addition, as a baseline, the *Passive Learning* (PL) algorithm is briefly introduced in Algorithm 6, whereby unlabelled data are randomly selected from a pool, and subjected to human annotation, before being added to the training set.

3.2.3.1 Active Learning

Figure 3.4 gives the flowchart of AL based on the prediction uncertainty (confidence value). In particular, Algorithm 7 describes the traditional AL algorithm based on the least-certainty query strategy, and Algorithm 8 provides another AL algorithm with a novel query strategy based on the selection of those instances predicted with medium certainty levels for further annotation. The rationale behind the adoption of a medium-certainty query strategy is the fact that it has the potential advantage of avoiding the selection of noisy data, which can be caused by unreliable annotations [31] or distortions of the (acoustic) pattern [182] as demonstrated in [183]. This is particularly important for acoustic emotion recognition owing to the comparably high degree of ambiguity.

Formally, the query function is defined as:

$$Query(\mathbf{x}) = \begin{cases} 1, & \text{if } \mathbf{x} = \arg\min_{\mathbf{x} \in \mathcal{X}} |C(\mathbf{x}) - C_m|, \\ 0, & \text{otherwise}, \end{cases} \tag{3.15}$$

where $C(\mathbf{x})$ represents the prediction CV for a given instance $\mathbf{x}$, and C_m is the CV

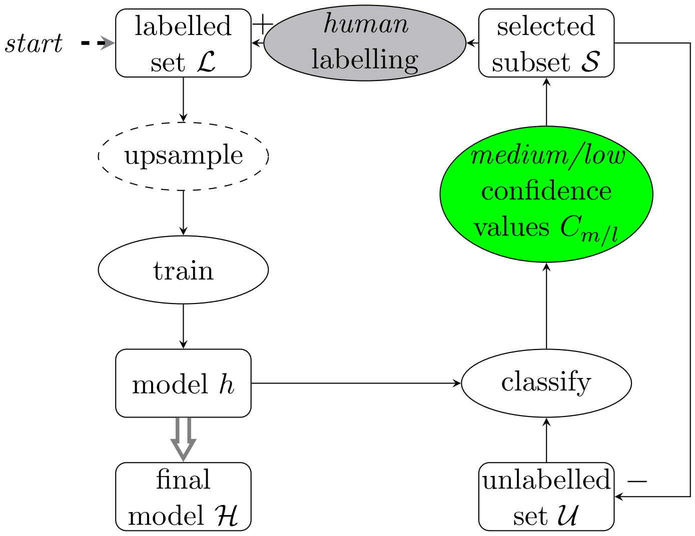

start
labelled set $\mathcal{L}$
+
human labelling
selected subset $\mathcal{S}$
upsample
medium/low confidence values $C_{m/l}$
train
model h
classify
final model $\mathcal{H}$
unlabelled set $\mathcal{U}$
−

Figure 3.4: Active Learning (AL) based on confidence values.

Algorithm 7: Active Learning (AL) with *least certainty query strategy.*

```
repeat
    (Optional) Upsample the training set $\mathcal{L}$ to obtain even class distribution $\mathcal{L}_D$.
    Use $\mathcal{L}/\mathcal{L}_D$ to train a classifier $\mathcal{H}$, and then classify the unlabelled set $\mathcal{U}$.
    Rank the data based on the prediction confidence values $C$ and store them in queue $Q$.
    Select a subset $\mathcal{S}_a$ whose elements are in the bottom of the ranking queue $Q$ (least certainty).
    Submit the selected subset $\mathcal{S}_a$ to human annotation.
    Remove $\mathcal{S}_a$ from the unlabelled set $\mathcal{U}$, $\mathcal{U} = \mathcal{U} \setminus \mathcal{S}_a$.
    Add $\mathcal{S}_a$ to the labelled set $\mathcal{L}$, $\mathcal{L} = \mathcal{L} \cup \mathcal{S}_a$.
until a predefined number of iterations is met.
```

of the instance located in the middle of the ranking queue. Ideally, for uniformly distributed predictions, C_m would be 0.5. Nonetheless, in practice this value is not fixed. Instead, it varies due to the changes on the unlabelled data pool as learning progresses (instances moved to the training set).

3.2.3.2 Co-Active Learning

Inspired by the concept of MVL and co-training, this thesis extends the AL to a novel method of 'Co-Active Learning' (hereafter coAL) and uses a prediction

Algorithm 8: Active Learning (AL) with *medium certainty query strategy.*

```
repeat
    (Optional) Upsample the training set $\mathcal{L}$ to obtain even class distribution $\mathcal{L}_D$.
    Use $\mathcal{L}/\mathcal{L}_D$ to train a classifier $\mathcal{H}$, and then classify the unlabelled set $\mathcal{U}$.
    Rank the data based on the prediction confidence values $C$ and store them in queue $Q$.
    Select subset $\mathcal{S}_a$ whose elements are in the middle of the ranking queue $Q$ (medium certainty).
    Submit the selected subset $\mathcal{S}_a$ to human annotation.
    Remove $\mathcal{S}_a$ from the unlabelled set $\mathcal{U}$, $\mathcal{U} = \mathcal{U} \setminus \mathcal{S}_a$.
    Add $\mathcal{S}_a$ to the labelled set $\mathcal{L}$, $\mathcal{L} = \mathcal{L} \cup \mathcal{S}_a$.
until a predefined number of iterations is met.
```

Algorithm 9: Co-Active Learning (coAL).

```
Input: (additional)
A learning domain with features $\mathcal{X}$.
repeat
    Split the domain features $\mathcal{X}$ into two views: $\mathcal{X}^1$, $\mathcal{X}^2$, and $\mathcal{X}^1 \cap \mathcal{X}^2 = \varnothing$.
    for $i = 1, 2$ do
        (Optional) Upsample each ‘view’ to even class distribution $\mathcal{X}_{Di}$.
        Use $\mathcal{X}_i/\mathcal{X}_{Di}$ to train classifier $\mathcal{H}_i$, and classify $\mathcal{U}$, respectively.
        Rank the data based on the prediction confidence values $C$ and store them in queue $Q$.
        Select a subset $\mathcal{S}_a$ whose elements are in the middle of the ranking queue $Q$ (medium certainty).
    end
    Submit the selected subsets $\mathcal{S}_{ca} = \mathcal{S}_{a1} \cup \mathcal{S}_{a2}$ to human annotation.
    Remove $\mathcal{S}_{ca}$ from the unlabelled set $\mathcal{U}$, $\mathcal{U} = \mathcal{U} \setminus \mathcal{S}_{ca}$.
    Add $\mathcal{S}_{ca}$ to the labelled set $\mathcal{L}$, $\mathcal{L} = \mathcal{L} \cup \mathcal{S}_{ca}$.
until a predefined number of iterations is met.
```

certainty query strategy. It consists of implementing two different views into AL. This strategy diverges from Co-Testing [172] by allowing both views to select the data to be annotated independently, rather than finding the ‘contention points’. Figure 3.5 and Algorithm 9 give the flowchart and pseudocode of coAL.

The feature domain $\mathcal{X}$ of a given data set needs to be separated into two independent and sufficient parts $\mathcal{X}^1, \mathcal{X}^2$, each of which is regarded as a ‘view’. Then, each view is used to create a model h that selects the instances by the same query

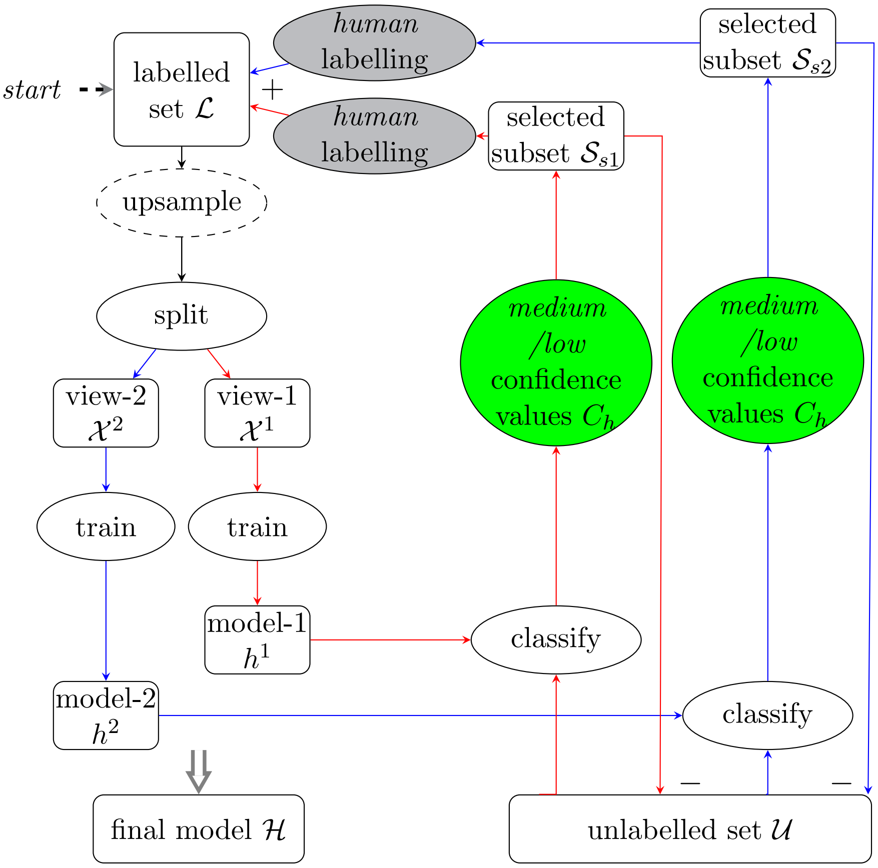

start
labelled set $\mathcal{L}$
human labelling
+
human labelling
selected subset $\mathcal{S}_{s2}$
selected subset $\mathcal{S}_{s1}$
upsample
split
view-2 $\mathcal{X}^2$
view-1 $\mathcal{X}^1$
medium /low confidence values C_h
medium /low confidence values C_h
train
train
model-1 h^1
classify
model-2 h^2
classify
final model $\mathcal{H}$
−
−
unlabelled set $\mathcal{U}$

Figure 3.5: Co-Active Learning.

function with the medium certainty-based AL as:

$$Query(\mathbf{x}^i) = \begin{cases} 1, & \text{if } \mathbf{x}^i = \underset{\mathbf{x}^i \in \mathcal{X}^i}{\arg\min} |C(\mathbf{x}^i) - C_m^i|, \\ 0, & \text{otherwise,} \end{cases} \tag{3.16}$$

where $i = 1, 2$ denotes the ‘view’ index. That is, the unlabelled instances in the data pool $\mathcal{U}$ predicted by each model with medium CVs ($\mathcal{S}^1$ and $\mathcal{S}^2$) are then delivered to a human oracle to be annotated. After the annotation, the subsets of $\mathcal{S}^1$ and $\mathcal{S}^2$ are added (together with the new label) to the training set and removed from the unlabelled data pool. There are three possibilities regarding the selection of a particular instance by the two views: 1) If an instance is not selected by any of the two views, it will be discarded in this iteration; 2) If an instance is selected by any of the two views, that instance plus the given label will be added to the training set once; 3) If an instance is selected by both views, it will be added twice to the training set together with the common class label (because it was annotated by a human oracle). The whole process is repeated until a predetermined number of iterations of the learning process is reached.

3.2.4 Cooperative Oracle

SSL techniques exploit the data labelling work without any human interaction, yet usually cannot acquire the gain as high as AL techniques when the same number of instances are labelled [172]. This phenomenon accounts for the essential drawbacks of SSL. One drawback relates to the absence of sufficient instances for a particular category in the initial training set, which leads to poor performance for that category. This drawback is because the instances with higher confidence estimates selected by the SSL algorithm are generally inclined to those categories with more examples and correct classification. This problem often engenders a vicious cycle in which the dominant categories are increasingly better recognised, with the opposite occurring with the less represented categories. Another common drawback of using SSL techniques is that noise can be added to the training set. Even though only the instances with the highest CVs are chosen, some of these instances are still misclassified. In this case, the noise is accumulated and increasingly affects the performance of the classifier. Both drawbacks are absent from AL. However, AL still requires a considerable amount of costly human intervention.

In order to take advantages of both approaches, the idea of jointly conducting AL and SSL was first introduced in [184], where the authors integrated Query-by-Committee-based AL and EM-based SSL. Later on, Tur et al. [185] used a combination of AL and Boosting-based SSL for spoken language understanding, showing that it significantly reduces human annotation effort; and Zhu et al. [186] took a combination of AL and SSL using Gaussian field and harmonic functions for graphic processing to recognise handwritten digit and text classification. Furthermore, in [172], Muslea et al. extended the idea of multi-view learning – considering the diversity of different views that are obtained from different feature subsets – into such a combination, namely *Co-EMT* (integrating co-testing with co-EM). After that, the work done by Wang and Zhou theoretically shows its efficiency [187].

In this thesis, the *Cooperative Learning* (CL) algorithm is further investigated, which integrates AL with SSL based on the prediction uncertainty with SVMs. It allows the sharing of the labelling effort between human and machine oracles, whilst being able to mitigate the limitations of both algorithms in the application of paralinguistic tasks (particularly for speech emotion recognition). In particular, CL always proceeds AL before SSL in each iteration. Such a process will repeat several times until a predefined number of iterations is satisfied. When taking the idea of multi-view learning into account, I consider the following three schemes: 1) *single-view CL* (svCL) – implementing AL followed by self-training (cf. Algorithm 10); 2) *mixed-view CL* (xvCL) – combining AL and co-training (cf. Algorithm 11); and 3) *multi-view CL* (mvCL) – fusing coAL and co-training (cf. Algorithm 12).

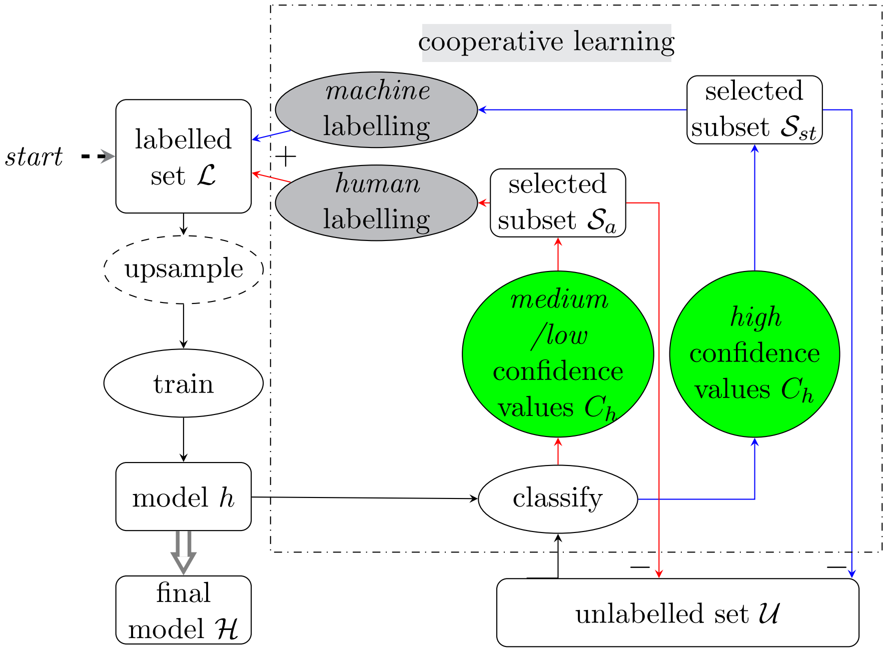

cooperative learning
start
labelled set $\mathcal{L}$
machine labelling
human labelling
+
selected subset $\mathcal{S}_{st}$
selected subset $\mathcal{S}_a$
upsample
train
medium /low confidence values C_h
high confidence values C_h
model h
classify
final model $\mathcal{H}$
−
−
unlabelled set $\mathcal{U}$

Figure 3.6: Single-view cooperative learning.

Algorithm 10: Single-view Cooperative Learning (svCL).

```
repeat
    Execute AL based on an initial training set $\mathcal{L}$, and obtain a subset $\mathcal{S}_a$ for human labelling (cf. Algorithm 7 or 8).
    Remove $\mathcal{S}_a$ from the unlabelled set $\mathcal{U}$ ($\mathcal{U}' = \mathcal{U} \setminus \mathcal{S}_a$), and add $\mathcal{S}_a$ to the labelled data set $\mathcal{L}$ ($\mathcal{L}' = \mathcal{L} \cup \mathcal{S}_a$).
    Execute self-training based on a training set $\mathcal{L}'$, and obtain a subset $\mathcal{S}_{st}$ for machine labelling (cf. Algorithm 4).
    Remove $\mathcal{S}_{st}$ from the unlabelled set $\mathcal{U}'$ ($\mathcal{U} = \mathcal{U}' \setminus \mathcal{S}_{st}$), and add $\mathcal{S}_{st}$ to the labelled set $\mathcal{L}'$ ($\mathcal{L} = \mathcal{L}' \cup \mathcal{S}_{st}$).
until a predefined number of iterations is met.
```

Particularly, taking the algorithm of svCL for example (the procedure is given by Figure 3.6), the algorithm distributes the data predicted with low certainty for human labelling at the first step (denoted as red lines), and the ones predicted with the highest certainty for machine labelling at the second step (denoted as blue lines) in each learning iteration.

In addition, to deal with the potential problem of imbalanced class distribution, data upsampling is employed in all algorithms by repeating a random subsample of the data set belonging to sparse categories.

Algorithm 11: Mixed-view Cooperative Learning (xvCL).

```
Input: (additional)
A learning domain with features $\mathcal{X}$.
repeat
    Execute AL based on initial training set $\mathcal{L}$, and obtain a subset $\mathcal{S}_a$ for human labelling (cf. Algorithm 7 or 8).
    Remove $\mathcal{S}_a$ from the unlabelled set $\mathcal{U}$ ($\mathcal{U}' = \mathcal{U} \setminus \mathcal{S}_a$), and add $\mathcal{S}_a$ to the labelled set $\mathcal{L}$ ($\mathcal{L}' = \mathcal{L} \cup \mathcal{S}_a$).
    Execute co-training based on training set $\mathcal{L}'$, and obtain a subset $\mathcal{S}_{ct}$ for machine labelling (cf. Algorithm 5).
    Remove $\mathcal{S}_{ct}$ from the unlabelled set $\mathcal{U}'$ ($\mathcal{U} = \mathcal{U}' \setminus \mathcal{S}_{ct}$), and add $\mathcal{S}_{ct}$ to the labelled data set $\mathcal{L}'$ ($\mathcal{L} = \mathcal{L}' \cup \mathcal{S}_{ct}$).
until a predefined number of iterations is met.
```

Algorithm 12: Multi-view Cooperative Learning (mvCL).

```
Input: (additional)
A learning domain with features $\mathcal{X}$.
repeat
    Execute coAL based on an initial training set $\mathcal{L}$, and obtain a subset $\mathcal{S}_{ca}$ (cf. Algorithm 9).
    Remove $\mathcal{S}_{ca}$ from the unlabelled set $\mathcal{U}$ ($\mathcal{U}' = \mathcal{U} \setminus \mathcal{S}_{ca}$), and add $\mathcal{S}_{ca}$ to the labelled set $\mathcal{L}$ ($\mathcal{L}' = \mathcal{L} \cup \mathcal{S}_{ca}$).
    Execute co-training based on training set $\mathcal{L}'$, and obtain a subset $\mathcal{S}_{ct}$ for machine labelling (cf. Algorithm 5).
    Remove $\mathcal{S}_{ct}$ from the unlabelled set $\mathcal{U}'$ ($\mathcal{U} = \mathcal{U}' \setminus \mathcal{S}_{ct}$), and add $\mathcal{S}_{ct}$ to the labelled set $\mathcal{L}'$ ($\mathcal{L} = \mathcal{L}' \cup \mathcal{S}_{ct}$).
until a predefined number of iterations is met.
```

Overall, all the above described learning algorithms will be elaborately analysed to leverage the unlabelled data by way of performance comparison in Chapter 4. However, it is necessary to point out that each algorithm has its own benefits and drawbacks. In realistic applications, the algorithm selection highly depends on the potential usage of human work and requirements of system performance.

3.3 Feature Optimisation

Following the stage of data preparation, features that are regarded as the representations for distinguishing patterns in the context of machine learning are successively exacted. Feature optimisation indeed is a general concept of reprocessing the original features with the goal of improving the recognition performance under certain circumstances. Within this thesis, these circumstances are defined as the distributed computing and the reverberant environments. That is because the distributed computing structure for pattern recognition task is emerging with the advance of networks and computer science recently, and the reverberant environments (e.g., room) are quite normal in our daily life. Each of the circumstances holds particular characteristics and requirements. In these cases, we need to find out a way to reprocess the raw features, contributing to higher recognition accuracy. Both cases will be elaborately discussed in Section 3.3.1 for computational paralinguistics and in Section 3.3.2 for ASR, respectively.

3.3.1 Feature Compression

Embedded vs. Client-Server-Based Recognition Systems

In the area of computational paralinguistics, most state-of-the-art academic research focuses on statically embedded recognition systems [14], [4], [64]. Such systems have a good degree of flexibility since they can be used without Internet access and therefore be applied in a wide range of practical scenarios. Nonetheless, at present, this advantage is becoming less significant. On one hand, because current data-driven pattern recognition systems largely benefit from processing large amounts of data for training and continuous development, which requires data transmission for the integration of data from multiple users, as well as vast storage and computational resources for training. Furthermore, sophisticated computational paralinguistic systems may require advanced computational models that are not possible to implement in users devices [188]. On the other hand, because Internet access is now ubiquitous on account of the advent of far-ranging coverage and high transmission speed wireless networks such as 3G, 4G and wireless LAN, and the breakout of mobile electronic devices like smartphones, laptops, and tablets.

One possible solution to these problems is to recur to client-server computing [189]. On the client side, the normal consumer devices with restricted computing ability can perform basic computational tasks; on the server side, super computers or computing centres can deal with the most expensive computational tasks. In the context of computational paralinguistics, the client is responsible for collecting realistic data (i.e., voice recordings in natural occurring scenarios) that is then sent to the server. On the server, the computational resources can be employed to integrate the data from various clients, build (and continuously improve) the target paralinguistic system(s), classify the data received from the clients for the task at hand, and feed back the final results to them.

Such a solution has several advantages for the future development and application of paralinguistic recognition systems in the real world. First, it can overcome one of the most important limitations for the development of robust paralinguistic recognition tasks – *data scarcity* (cf. Section 2.2.2). The client-server-based systems have the potential to allow the collection of large amounts of labelled and unlabelled realistic data from thousands of users in real-life scenarios, which can be exploited for training models and enhancing their performance using a multiplicity of machine learning techniques, as shown in Section 3.1 and 3.2. Second, it can accelerate the improvement of paralinguistic recognition systems, since having the data processed on the server side, computer scientists can continuously develop and apply more effective techniques (e.g., BLSTM [33] or cumulative evidence [34]) and combine various data sources to boost the systems' performance and robustness. Moreover, user profiles can be stored in the server to support long-term analysis and improve user-specific models. Third, on the client side, the requirements of computing power, the conditions of operating systems and hardware configurations are greatly relaxed, therefore making it possible to spread the use of paralinguistic analysis to a wide range of personal mobile and fixed devices.

Network vs. Distributed Recognition Systems

Concerning the location where the feature extraction takes place, client-server architectures for computational paralinguistics can be categorised into two classes: network recognition systems and distributed recognition systems [189]. The former uses conventional speech coders for transmission of speech from a client device to a server where feature extraction and recognition decoding are undertaken. The latter implies that the feature extraction stage is processed on the client side, but the recognition is made on the server [189].

One of the major advantages of adopting a network recognition approach is that it is not necessary to develop a completely new system for paralinguistic recognition tasks. Indeed, numerous commercial applications already implement speech coding, and so, without the need to change the applications on existing devices and networks, we can simply use preexisting recognition models on the server side to process the encoded speech signals. Moreover, it shares all the advantages of server-based systems in terms of system maintenance, update and device requirements [189]. Nonetheless, network recognition systems pose various challenges related to privacy and transmission bandwidth limitations as discussed later.

In distributed pattern recognition systems, instead, the feature extraction process occurs on the client side, where a representation of the speech signal with a lower dimensionality and redundancy can be obtained and optimised for transmission. Such systems have been adopted in various applications, being some of the most impressive and successful ones developed in the context of speech recognition, where both theoretical (e.g., packet loss via transmission [190], feature compression techniques [191], and noise robustness [192]) and experimental (e.g., Google search engines and Apple's Siri) research have been conducted. In other fields, distributed pattern recognition has also been applied, for instance, to the recognition of human faces [193], actions [194], and nature elements (such as trees or weeds).

In relation to computational paralinguistics, if distributed computing can be demonstrated to be feasible and reliable, some of the current limitations preventing recognition systems to be applied to a variety of realistic applications can be greatly mitigated. More importantly, this would be beneficial to a variety of areas, such as, remote medicine treatment, remote conferences or negotiations, remote education, and even advanced driver assistance systems, where paralinguistic recognition systems have manifold applications.

3.3.1.1 Distributed Speech Analysis System

Upon the discussion outlined above, a framework of distributed structure is proposed and illustrated in Figure 3.7. It is indeed inspired by the standardisation work of distributed speech recognition performed by the Aurora group from the ETSI [18]. Compared to the unified ISA framework shown in Section 2.1, this distributed system maintains all original components, whereas the front-end and back-end modules turn into the *client* and *server* modules. Moreover, two new components related to signal transmission and feature (de-)compression are added. Similar to any other network-based systems, the process of data framing, bit-stream formatting, error protection, and secure coding are demanded before entering data into the physical transmission channel. This process targets at meeting the physical transmission requirements (e.g., IP routing, clock recovery), preventing the channel distortion (e.g., channel noise, packet loss), and guaranteeing the information security. The component of feature (de-)compression plays a vital role in dealing with the main concerns as follows.

Major Concerns of Distributed Recognition Systems

One major concern of network-based systems is *transmission bandwidth*. As discussed in Section 2.3.3, the statistic feature set delivers the smallest feature size compared to the raw coded speech and LLD feature sets. Nevertheless, it is still a challenge for both network transmission and memory storage in consideration of a target scenario involving a large number of users/devices.

Another major concern in paralinguistic recognition is *security*, as the privacy of the speakers has to be guaranteed. This protection is particularly important in real-life contexts in which personal information and sensitive data may be collected. Paralinguistic information is indeed of a highly private nature (e.g., emotional statements, information about alcohol intoxication, tiredness) [195]. The transmission of raw coded speech is a common approach in client-server architectures. The speech data are normally coded by protocols such as G.711, G.726, and AMR-WB [196]. However, the goal of these methods is to have the speech signals well recovered to ensure better communication quality. As mentioned earlier, a possible alternative is

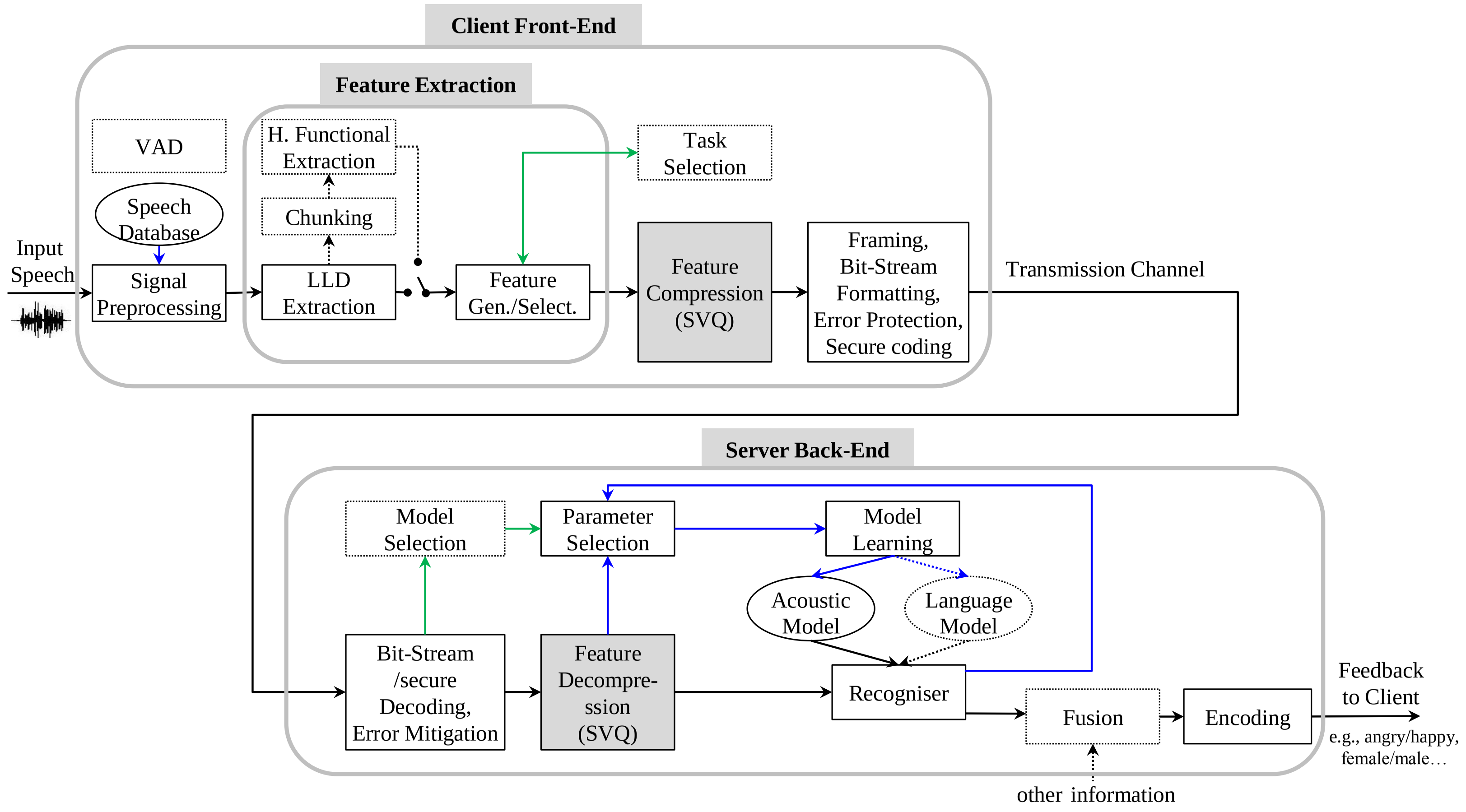


Figure 3.7: Framework of distributed paralinguistic recognition system. Dotted boxes indicate optional components. Blue arrows show steps carried out only during system training or adaptation phases. Green arrows indicate steps carried out when multiple recognition tasks need to be processed at the same time or separately.

to perform feature extraction directly in the client and transmit LLDs [18], therefore preventing direct access to users' speech. Previous work has shown that it is feasible to reconstruct audio from static feature vectors, such as MFCCs and pitch (e.g., [197], [198]). Unfortunately, this feasibility once more generates important privacy-related issues. For the system, I propose to generate and transmit statistical feature vectors obtained by applying functionals over LLDs for each utterance. The procedure for generating such feature vectors is irreversible, and, therefore, it avoids the reconstruction of speech signals. Because of this irreversibility, the speakers' speech content is fully protected, which is significantly important because the speech content is widely admitted as the most important personal information. Moreover, even though we can acquire some age and gender information from the statistical features, it is difficult to confirm a speaker's identify among billions of people of different genders and ages. Furthermore, in the context of state-of-the-art computational paralinguistic research, statistical features are currently well-accepted for extracting relevant information from speech (e.g., [14, 4, 53]).

3.3.1.2 Split Vector Quantisation

Given the concerns outlined above, Split Vector Quantisation (SVQ) [78] is considered for feature compression in the proposed distributed systems. The rationales behind this choice are: i) The assignment of prototype numbers from a codebook finally eliminates any direct feature information from the user thus ensuring high privacy [18]; ii) SVQ is the officially recommended method by the ETSI standards [18] for distributed speech recognition; and iii) It is a well established and efficient feature compression technique [77, 74, 199].

SVQ algorithms split the high dimensional feature vectors into several subvectors that automatically group the original feature set through some sort of clustering algorithm (e.g., k-means). Each subvector is then represented by the centroid of each group. Figure 3.8 shows a diagram depicting the SVQ algorithm. The encoding scheme firstly partitions the whole d-dimensional feature vector $\mathbf{x} = [x_1, x_2, \ldots, x_d]^T$ into p subvectors, each of which $\mathbf{s}_i = [x_i, \ldots, x_{i+k_i}]^T$ is with k_i dimensions, where $i \in [1, \ldots, p]$. Thus, $\mathbf{x} = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_p]^T$, and d is equal to the sum of dimensions of each subvector, $d = k_1 + k_2 + \ldots + k_p$. In the particular case of having the same number of dimensions in each subvector, then $d = k \times p$. Following, each subvector is quantised using a separated VQ codebook, $\boldsymbol{Q} = vq(\mathbf{x}) = [vq(\mathbf{s}_1), vq(\mathbf{s}_2), \ldots, vq(\mathbf{s}_p)]^T = [\boldsymbol{v}_1, \boldsymbol{v}_2, \ldots, \boldsymbol{v}_p]^T$, where $\boldsymbol{v}_i \in \boldsymbol{C}_i$. Note that, the codebook ($\boldsymbol{C}_i$) pertaining to a particular subvector can be different from that of other codebooks, not only in the clustering space but also in size.

In this implementation, a k-means algorithm is used for clustering. That is, each observation belongs to the closest quantisation centroid that is found by using a weighted Euclidean distance to determine the index:

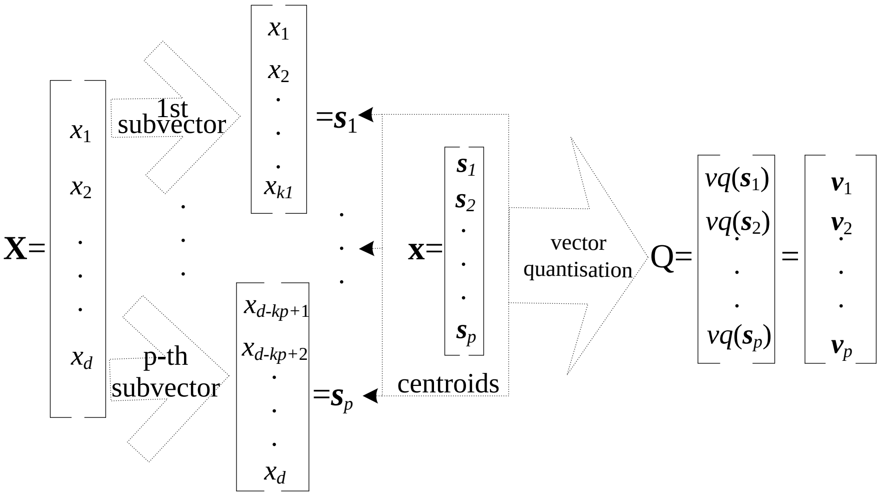


Figure 3.8: Diagram of the Split Vector Quantisation (SVQ) algorithm. [8]

$$\boldsymbol{d}_i^j = \mathbf{s}_i - \boldsymbol{v}_i^j, \quad i = 1, \ldots, p; j = 1, \ldots, N_i, \tag{3.17}$$

$$idx_i = \underset{1 \leq j \leq (N_i)}{\arg\min} \, (\boldsymbol{d}_i^j)^T \boldsymbol{W}_i (\boldsymbol{d}_i^j), \tag{3.18}$$

where $\boldsymbol{v}_i^j$ means the j-th codevector in the codebook $\boldsymbol{C}_i$, $\boldsymbol{d}_i^j$ denotes the Euclidean distance between subvector $\mathbf{s}_i$ and codevector $\boldsymbol{v}_i^j$, N_i is the size of the codebook, $\boldsymbol{W}_i$ is the weight matrix, e.g., identity matrix, to be applied to the codebook $\boldsymbol{C}_i$, and the idx_i denotes the codevector index chosen to represent the vector $\mathbf{s}_i$.

The final set of quantised vectors, $[idx_1, idx_2, \ldots, idx_P]^T$, is used to represent the corresponding speech chunk, and transmitted to the server back-end. On the server back-end, the SVQ process is reversed by using the same codebook used in the front-end for each subvector:

$$\hat{\mathbf{s}}_i = \boldsymbol{v}_i^{idx_i}, \tag{3.19}$$

where $\hat{\mathbf{s}}_i$ denotes the estimate of $\mathbf{s}_i$. Then, we unify all estimated subsets of features into one set, $\hat{\mathbf{x}} = [\hat{\mathbf{s}}_1, \hat{\mathbf{s}}_2, \ldots, \hat{\mathbf{s}}_p]^T$.

Finally, it is important to mention that there are various aspects that need to be taken into consideration when using SVQ as they can impact the performance of the various recognition tasks. Those will be exemplified in Section 4.2.2.

3.3.2 Feature Dereverberation

To tackle the reverberation problem of speech recognition, many techniques using neural networks have been proposed. A prominent technique is to train DNNs [16] using a wide variety of reverberated data sources. The key objective is to derive the original speech features to a high-level representation. Its potential capability for noise robust ASR has been demonstrated in [200, 201]. Another approach that has lately received increasing attention is to use neural networks for feature enhancement, which aims to remove the reverberation characteristic information from the distant-talk speech by means of learning a mapping (or transforming) rule from the distant-talk feature space to its close-talk counterpart. The main advantage of this approach is that it leaves the feature extraction and the back-end untouched, as the mapping is performed after feature extraction and prior to decoding. Therefore, the technique can be easily integrated with any existing ASR systems. This work was firstly realised in [202], in which an MLP was employed via mapping multiple channel array speech to clean speech. Then, it was extended using RNNs [203] for the 2nd CHiME challenge [204], where reduction in word error rates was observed.

Recently, long short-term memory RNNs (LSTM-RNNs) [17], a more sophisticated form of RNNs, have been successfully applied to a variety of pattern recognition tasks, especially to sequential pattern tasks, e.g., handwriting recognition [205], continuous speech recognition [206], and driver distraction detection [207]. Compared to 'classic' RNNs, LSTM-RNNs adopt memory blocks to replace the individual artificial neurons. Therefore, these networks can learn an optimised range of contextual information, aiming to overcome the vanishing gradient problem of conventional RNNs [116, 17]. The superiority of LSTM neural networks (especially the bidirectional type of BLSTM) when compared to DNNs and conventional RNNs has been empirically confirmed in several recent comparative studies [206, 208, 209]. The effectiveness of LSTM networks in handling non-stationary noisy speech was first demonstrated in [210], and this was later extended to enhance reverberated noisy speech in [211]. Applying LSTM networks to enhance non-stationary noisy speech was firstly introduced in [210]. It was further shown to enhance noise-reverberated speech in [211].

3.3.2.1 Feature-Enhanced Speech Recognition System

The framework of BLSTM models for dereverberation in distant-talk ASR is illustrated in Figure 3.9. The clean talk signal $s(t)$ is corrupted by convolutional noise $r(t)$ and additive noise $n(t)$ when transmitting through space channel. So, the observed distant-talk signal $\hat{s}(t)$ at the microphone can be written as:

$$\hat{s}(t) = s(t) * r(t) + n(t). \tag{3.20}$$

For the sake of simplification, I ignore additive noise in this thesis. Thus, Equation (3.20) becomes

$$\hat{s}(t) = s(t) * r(t). \tag{3.21}$$

The total length of RIR can be denoted as T_{60} that represents the time taken for the energy in the impulse response to decay by 60dB compared to the direct

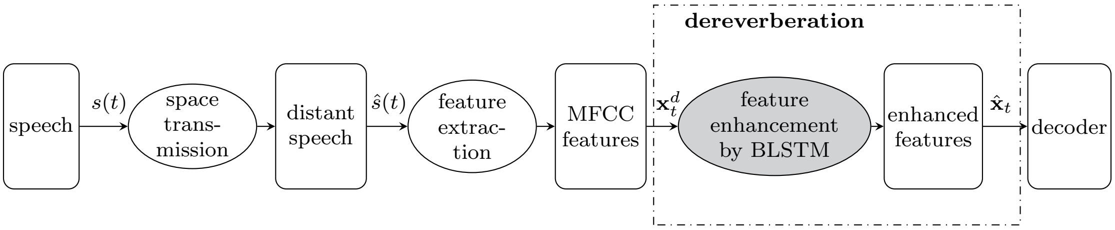


Figure 3.9: Framework of BLSTM models for dereverberation in distant-talk ASR. [212]

sound. The RIR $r(t)$ can be divided into two portions: the early reflection $r_e(t)$ that includes several strong reflections, and the late reverberation $r_l(t)$ that consists of a series of numerous indistinguishable reverberation. That is,

$$r(t) = r_e(t) + r_l(t), \tag{3.22}$$

where

$$r_e(t) = \begin{cases} r(t) & 0 \leq t < T \\ 0 & \text{otherwise,} \end{cases} \quad r_l(t) = \begin{cases} r(t+T) & 0 \leq T \\ 0 & \text{otherwise,} \end{cases} \tag{3.23}$$

and T is the length of the spectral analysis window ($20 \sim 30\,\text{ms}$)[1]. Thus, Equation (3.21) can be changed into

$$\hat{s}(t) = s(t) * r_e(t) + s(t-T) * r_l(t). \tag{3.24}$$

When the length of RIR T_{60} is much shorter than the analysis window size T, $r(t)$ is equal to $r_e(t)$, which only effects the speech signals within a frame (analysis window). This linear distortion in the spectral domain can be effectively mitigated by conventional techniques like CMN [87]. For most applications (e.g., occurring in typical office and home environment), however, the reverberation time T_{60} ranges from 200 to 1 000 ms [213] that is much longer than the analysis window size, resulting in an undesirable influence on the following speech frames. For example, if the duration of a RIR is 1 s (T_{60}) and a feature frame is extracted every 10 ms, one RIR would smear across the following 100 frames. Therefore, this distorted speech, after applying Short-Time Discrete Fourier Transform (STDFT), can be formulated by:

$$\hat{S}(t,f) = S(t,f)R_e(t,f) + \sum_{\tau=1}^{N-1} S(t-\tau,f)R_l(t-\tau,f), \tag{3.25}$$

where $R(\tau, f)$ denotes the part of $R(f)$ (i.e., STDFT of RIR $r(t)$) corresponding to frame delay τ, N is the number of influenced frames following one RIR. In this

[1] In some studies, T is denoted as the boundary between the early reflection and late reverberation. Its typical value is 50 ms [213, 79].

case, the channel distortion is no more of multiplicative nature in a linear spectral domain – rather it is convolutional.

Assuming the phases of different frames are non-correlated for simplification, the power spectrum of Equation (3.25) can be approximated as

$$\begin{aligned}|\hat{S}(t,f)|^2 \approx & |S(t,f)|^2|R_e(t,f)|^2 \\ & + \sum_{\tau=1}^{N-1} |S(t-\tau,f)|^2|R_l(t-\tau,f)|^2.\end{aligned} \tag{3.26}$$

To extract the standardised feature vectors in cepstral domain for ASR [18], logarithms and DCT are executed over the above spectral signals. So,

$$\begin{aligned}\mathcal{D}(\ln|\hat{S}(t,f)|^2) \approx & \mathcal{D}(\ln|S(t,f)|^2) + \mathcal{D}(\ln|R_e(t,f)|^2) \\ & + \mathcal{D}(\ln|M(t,f)|^2),\end{aligned} \tag{3.27}$$

where $\mathcal{D}$ denotes the discrete cosine transformation matrix, and

$$\begin{aligned}|M(t,f)|^2 &= 1 + \frac{\sum_{\tau=1}^{N-1} |S(t-\tau,f)|^2|R_l(t-\tau,f)|^2}{|S(t,f)|^2|R_e(t,f)|^2} \\ &= \frac{|\hat{S}(t,f)|^2}{|S(t,f)|^2|R_e(t,f)|^2}.\end{aligned} \tag{3.28}$$

If the speech signal transmission channel is invariable within the sentence period, the second term of $\mathcal{D}(\ln|R_e(t,f)|^2)$ in Equation (3.27) can be treated as a constant, and can be theoretically removed just by subtracting the cepstral mean over each utterance [214]. Therefore, the objective of the strategy is to get rid of the third term, $\mathcal{D}(\ln|M(t,f)|^2)$, which is the proportion of the power spectrum of the whole observed distorted speech and the distorted speech only convoluted by early reverberation (cf. Equation (3.28)). The specific way to realise such a strategy in this thesis is to apply neural networks to map the feature vectors $\mathbf{x}_t^r$ that are extracted from the distant-talk speech signals $\hat{s}(t)$ to the target ones frame by frame. Finally, the enhanced feature vectors $\hat{\mathbf{x}}_t$ will be fed into the ASR decoder.

3.3.2.2 Feature Enhancing by Neural Networks

From Equation (3.27) and Equation (3.28), one can observe that the term of $M(t,f)$ is not only relative to the early reflection, but also convoluted to the late reverberation of previous speech signals. Such a highly nonlinear and nonstationary characteristic makes dereverberation an extremely challenging task [12, 213, 19]. To this end, using a *nonlinear* system to predict this term might be a potentially promising approach. On the other hand, the close relationship of $M(t,f)$ with the numerous previous speech frames also implies the possibility of compensating for the late reverberation by leveraging the *long-term acoustic context*. That is, exploiting the sequence of reverberant feature vectors preceding the current ones might also be beneficial for mitigating the late reverberation. The traditional way to capture such contextual information is to use triphone HMMs, which is empirically proved not sufficient for this task [204].

Motivated by these analyses, I explore an approach based on the usage of a nonlinear and more efficient context-learning-ability neural network [17] – BLSTM-RNN – to remove such convoluted late reverberation in the cepstral domain. More specifically, two ways could be applied according to Equation (3.28). They are via transforming the distorted feature vectors $\mathbf{x}_t^d$ from the distant speech signal $\hat{s}(t)$ into:

1) the corresponding *absolute* (clean) ones $\mathbf{x}_t^c$ from close-talk speech signals $s(t)$ by minimising the following objective function of the Mean Squared Error (MSE):

$$\mathcal{J}(\theta) = \sum_{i=1}^{r} (x_t^c - \hat{x}_t^c)^2, \tag{3.29}$$

where $\hat{x}_t^c$ is the predicted close-talk feature, and r is the dimensionality of feature vector. This *direct* channel mapping strategy has already been investigated in [210] and [211].

2) the corresponding *differential* (delta) ones $\mathbf{x}_t^\Delta$ that are obtained from later reverberation of $M(t, f)$ (cf. Equation (3.28)). Before training the neural networks, the differential vectors are calculated by subtracting the feature vectors of distant talk $\mathbf{x}_t^d$ from those of the corresponding close talk $\mathbf{x}_t^c$. When training the neural networks, the parameters are optimised by minimising:

$$\mathcal{J}(\theta) = \sum_{i=1}^{r} (x_t^\Delta - \hat{x}_t^\Delta)^2, \tag{3.30}$$

where $\hat{x}_t^\Delta$ is the predicted differential feature. After that, these mapped differential vectors are added to the original distant-talk feature vectors $\mathbf{x}_t^d$ frame by frame, so as to compensate the distortion by reverberation. This *indirect* channel mapping strategy will be well investigated in this thesis.

4 Applications in Intelligent Speech Analysis

Speech is broadly considered to be the most natural communication form for humans. When we are interacting with humans or mediating between humans, both spoken content and non-verbal information, such as emotion and gender, play critical roles.

In this vein, this chapter focuses on designing and executing experiments regarding speech to verify the effectiveness of the presented approaches in Chapter 3. It starts with the one of the most representative paralinguistic tasks – emotion. The challenge of data scarcity is extensively investigated by not only well-developed machine learning algorithms, such as confidence-value-based SSL and AL, but also by some novel methods like labelling agreement-based data selection. Among these learning algorithms, co-training is further extended to be accessed for general paralinguistic tasks, for example, sleepiness, intoxication, gender, and age. Such a process of data collection and model updating can benefit from a distributed recognition structure, which is then preliminarily investigated via compressing feature size. Subsequently, it moves to the linguistic task of ASR, in which the context-sensitive LSTM neural networks are illustrated to enhance reverberant features.

4.1 Speech Emotion Recognition

ASR has already served as an integral part in many intelligent systems, such as personal assistants (e.g., Siri, Google Now, and Cortana) in smartphones. However, the analysis of other patterns (e.g., speakers' characteristics, personalities, and states) are still needed to enable machines to permanently observe and react to its conversational partner in a socially competent way.

Emotion, which is widely admitted as a fundamental component of being human, has received much attention over the past years [215, 216, 63, 2]. Taking the automatic call centre for example, the systems can be changed into a manual service mode after detecting customers' specific states, such as anger. Indeed, any interface that ignores users emotional states or fails to manifest the appropriate emotions can dramatically impede performance and risks being perceived as cold and socially inept.

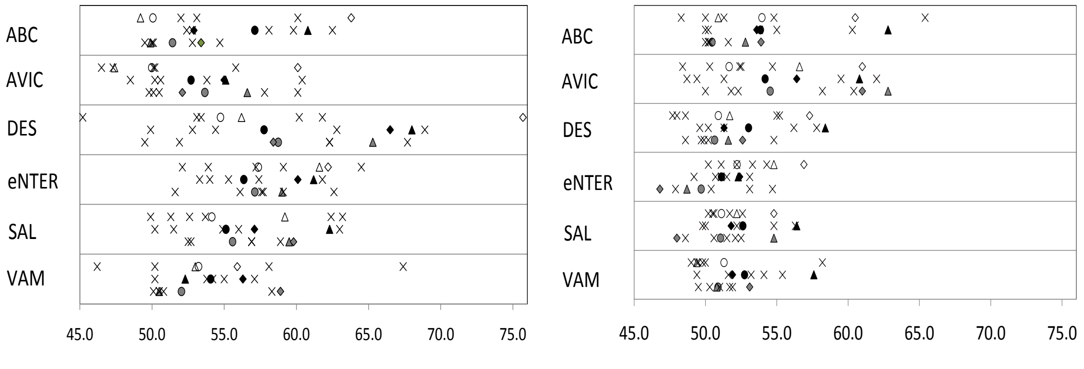
In this light, *speech emotion recognition* (SER) is selected as the representative task for data enrichment and optimisation evaluation in this section. More specifically, Sections 4.1.1 - 4.1.4 examine the data pooling, data bagging, data sampling, and data selection techniques, respectively, with the objective of profiting from existing databases for the same or similar targets. Yet, it is still difficult for these databases to meet the application requirements. Thus, Sections 4.1.5 and 4.1.6 use machines to complete the annotation work using the approaches of SSL and co-training, respectively, so as to exploit the value of unlabelled data and further boost the 'limitedly' trained models. Alternatively, Section 4.1.7 asks humans for annotation, with as little work as possible via the use of certain queries. Finally, Section 4.1.8 distributes the annotation work among machines and humans.

4.1.1 Human Speech Data Fusion

This experiment is designed to aggregate several pre-existing separated databases or classifiers into a large database (*pooling*) or a strong classifier (*voting* or *bagging*). Here, six frequently-used databases were selected – DES, eNTERFACE (eNTER), ABC, AVIC, SAL, and VAM. They range from acted over induced to spontaneous affect portrayal. For better comparability of obtained performances among corpora, the diverse emotion groups were additionally mapped onto the two most popular axes in the dimensional emotion model as in [23, 25]: arousal (i.e., passive ['-'] vs. active ['+']) and valence (i.e., negative ['-'] vs. positive ['+']). These mappings were not straight forward, which favours better balance among target classes. The introduction of each database as well as its category mapping strategy is presented in Appendices A.1.2 and A.1.4.

For the acoustic feature set, brute-forced feature vectors of 6 552 dimensions were employed (see [217] for more details) by applying 39 functionals over 56 acoustic LLDs including first and second order delta regression coefficients. For the classifiers, SVM, RF, and Naïve Bayes (NB) [218] were all considered due to their very good generalisation properties in SER (cf. Section 2.4). Further, the kernel of SVM is linear with a complexity constant of 0.05, and RF is with 10 trees and 200 attributes per tree.

Compared to the bagging algorithm shown in Section 3.1.3, each randomly selected subset is replaced by each database in this experiment. Thus, note that the *data bagging* is here referred to as *data voting.*



(a) arousal (b) valence

Figure 4.1: Distributions of UARs for cross-corpus binary arousal/valence classification of six test databases: Single-database classifiers (crosses), average of single-database classifiers (circles), and classifier fusion by voting (triangles) and pooling (diamonds). The top row per test database depicts results obtained by SVM, the middle one by RF, and the bottom one by NB. [217]

Data Fusion vs. Single Classifier

The results of fusion by pooling and voting against the results obtained by pairwise cross-corpus SER are compared, as has been investigated, e.g., in [23]: There, classifiers are simply trained on a single database and tested on another. In particular, the average UAR is calculated for each test database obtained in pairwise classification. These evaluation procedures represent fully realistic conditions where the classifier cannot simply adapt to the peculiarities of a single database as in 'traditional' intra-corpus evaluation.

Results for each test database and different classifiers (top: SVM, middle: RF, bottom: NB) are depicted in Figure 4.1 as one-dimensional scatter plots. Circles depict the average performance of pairwise cross-corpus, and triangles indicate the UAR obtained by voting and diamonds the one obtained by pooling. Exact values are given in Table 4.1. On average, it can be seen that both, voting and pooling are superior to the average UAR for pairwise classification. Thus, on average one expects a gain by fusing databases instead of selecting a single one that performs best.

Comparing the results by data fusion to the individual single-corpora results, data fusion seems most promising for valence recognition, where it often outperforms the best single-corpus classifier. On the other hand, this trend is not as strongly visible in arousal recognition. Still, from the application point of view, this is very interesting: When designing an SER system, we normally do not know which training database performs best. In that case, using multiple training databases and fusing decisions can dispose of the need for extensive validation experiments with

Table 4.1: UARs for cross-corpus binary arousal/valence classification: Average UAR of single database classifiers (Avg), and UAR of cross-corpus fusion by classifier voting and data pooling, for SVM, RF, and NB. Mean UAR across classifier type, and UAR of two-stage multi-classifier vote (2-Vote). [217]

UAR [%]	**SVM**			**RF**			**NB**			**Mean**			**2-Vote**
Test on	**Avg**	**Vote**	**Pool**	**Avg**	**Vote**	**Pool**	**Avg**	**Vote**	**Pool**	**Avg**	**Vote**	**Pool**	
							Arousal						
ABC	50.1	49.2	63.8	57.1	60.8	52.9	51.4	49.9	53.4	52.9	53.3	56.7	55.5
AVIC	50.0	47.4	60.1	52.7	55.1	55.0	53.7	56.6	52.1	52.1	53.0	55.7	53.9
DES	54.7	56.2	75.7	57.8	68.0	66.5	58.7	65.3	58.4	57.1	63.2	66.9	68.7
eNTER	57.4	61.6	62.2	56.4	61.2	60.1	57.1	59.0	59.1	56.9	60.6	60.5	61.3
SAL	54.1	59.2	62.4	55.1	56.3	57.1	55.6	59.5	59.8	55.0	60.3	59.8	63.3
VAM	53.2	53.0	55.9	54.1	52.3	56.3	52.0	50.5	58.9	53.1	51.9	57.0	51.0
Mean	53.3	54.4	**63.4**	55.5	**60.0**	58.0	54.8	56.8	**57.0**	54.5	57.1	**59.4**	59.0
							Valence						
ABC	54.0	50.9	60.5	53.9	62.8	53.6	50.5	52.8	53.9	52.8	55.5	56.0	61.0
AVIC	51.7	56.6	61.0	54.2	60.8	56.4	54.5	62.8	61.0	53.5	60.1	59.5	65.7
DES	50.9	51.7	57.3	53.0	58.4	51.3	50.6	51.6	52.6	51.5	53.9	53.7	58.2
eNTER	52.2	54.8	56.9	51.1	52.3	52.4	49.7	48.7	46.8	51.0	51.9	52.0	52.2
SAL	51.1	52.2	54.8	52.6	56.4	51.8	51.1	54.8	48.0	51.6	54.5	51.5	56.7
VAM	51.3	49.4	49.4	52.7	57.6	51.9	50.9	50.8	53.1	51.6	52.6	51.5	54.8
Mean	51.9	52.6	**56.7**	52.9	**58.1**	52.9	51.2	**53.6**	52.6	52.0	**54.7**	54.0	58.1

different training sets.

Pooling vs. Voting

As to comparison of fusion strategies with one another, it seems that pooling (63.4 % UAR on average over all test databases) generally outperforms voting (54.4 %) for the SVM classifier, even drastically for the arousal recognition on the DES database (75.7 % vs. 56.2 % UAR). However, this can be observed neither for the RF nor for the NB classifier. A possible explanation might be that the SVM training algorithm automatically weights training instances by selecting them as support vectors; thus, it seems more suited to training on large, heterogeneous data sets. On the other hand, voting with random forests outperforms the voting from other classifiers significantly, both, for valence and arousal; this probably indicates that using confidence scores indeed increases robustness of the voting strategy. Finally, on average over classifiers, pooling is superior to both, single-database classification of arousal and voting, delivering a relative improvement of UAR by 9.0 % over the former. For valence recognition, pooling and voting perform almost equally, and voting is observed slightly better.

Two-Stage Voting

As the above evaluation revealed very notable differences in the performance of different classifiers, I investigated the performance of a two-stage voting process, where a secondary majority vote among the three classifiers is performed. Again, this majority vote is well-defined in any case. From Table 4.1, it can be seen that for recognition of arousal, the two-stage vote is on average slightly inferior to the voting by random forests (59.0 % vs 60.0 % UAR); for valence, the accuracy of two-stage voting (58.1 %) is even equal to the best possible configuration of classifier and fusion strategy (voting by random forests).

This result, in fact, suggests that when designing an emotion recogniser from multiple databases using fusion by voting – in that case, it is not clear a priori that classifier performs best – a majority vote among classifiers delivers almost equal accuracy to the best classifier. Thus, it will be an interesting issue for future research to evaluate the two-stage scheme for pooled training as well.

4.1.2 Synthesised Speech Data Fusion

Contrary to the method of fusing *natural* speech data, an alternative approach is fusing artificially generated speech – *synthesised* speech. If such data are suitable for training or adapting models for the recognition of human emotional speech, countless options open up: Not only could training data be generated in virtually infinite quantities, but emotional speech could be produced for different target groups (e.g., by varying parameters of the synthesiser corresponding to different ages or genders), for various and also under-resourced languages, and for fitting to the spoken content at hand. The latter could help for the design of dialogue systems with specific vocabularies, and could also be promising to address the challenge of text-independent emotion recognition: Assuming reliable ASR, one could first recognise the phonetic content, and then reproduce this content in various emotional facets for adaptation of acoustic emotion models. The general feasibility of this idea has been repeatedly demonstrated: For example, Microsoft's Kinect sensor uses synthesised user models to provide for different body shapes, postures, etc. Concerning the field of audio processing, improved recognition of chords in music was enabled by the synthesis of training material from symbolic music using various sound fonts (sets of instrument samples) in [219]. Finally, the work in [220] achieved tentative results showing that using synthesised speech for training benefits emotion recognition from human speech in a pair-wise cross-database evaluation using the eNTERFACE and EMO-DB corpora, i.e., training on one database and testing on the other. Therein, using synthesised speech for training could often outperform training with human speech.

This section aims to consolidate these promising results by providing extensive empirical evaluations on eight *human* emotional speech databases – ABC, AVIC, DES, EMO-DB (EMOD), eNTER, SAL, SUSAS, and VAM, as well as two *synthesised* emotional speech databases – TXT2PHO and OpenMary. Such two synthesised databases were generated by two different phonemisation components, in combination with Emofilt and Mbrola. In regard to the introduction of the eight human and the two synthesised speech databases, as well as the category mapping strategy, please refer to Appendices A.1.2, A.1.3, and A.1.4. Moreover, the same acoustic features (6 552 attributes) and classifier (SVM with a linear kernel and a complexity constant of 0.05) as those in Section 4.1.1 were chosen.

In the following, the first baseline experiment employs pair-wise cross-corpus training and testing on the eight databases of human emotional speech (HS), i.e., for each test database, each of the remaining seven databases is used once as a training set. This protocol results in $56 = 7 \times 8$ cross-corpus classifications for each dimension (arousal and valence). Then, to assess the suitability of synthesised training data for analysing human speech, the experiment is repeated by training with each of the two sets of synthesised emotional speech (SS), and testing on each of the human speech databases ($16 = 2 \times 8$ evaluations per dimension). Finally, to investigate the benefit of joint training with human and synthesised speech (HS+SS), all $16 = 2 \times 8$ combinations of human and synthesised data sets are considered for training, and evaluate on each of the seven human databases not found in the training data. The last experiment results in $112 = 16 \times 7$ combinations of training and test data per dimension. To generally enhance performance in cross-corpus SER, the thesis employed feature standardisation per corpus (cf. Section 3.1.3). This process helps to reduce trivial cases of feature mismatch due to different microphone-to-mouth distances, etc.

Experimental Results

Table 4.2 shows the UARs obtained for the two-class arousal and valence classification tasks when following the three above-mentioned evaluation protocols (HS, SS, and HS+SS). In summary, the results of the baseline experiment (cross-corpus training and testing on human speech, HS) corroborate the results of other cross-corpus SER studies [23], indicating that although arousal classification is somewhat stable, cross-corpus valence classification cannot be performed robustly using the acoustic features used in this study. In fact, results are often found below chance level UAR (50 %) for valence. Furthermore, in arousal classification, we find that testing on highly prototypical emotions (e.g., EMO-DB [EMOD] or DES) generally leads to higher performance than testing on spontaneous emotions (e.g., in the SAL or SUSAS databases), which is expected. A notable exception from this general pattern is the comparably high UAR (73.2 %) when testing on the VAM database; this can be attributed to the fact that although the emotions in this database are naturalistic, the talk-show recording scenario is much more likely to elicit strong emotions than, for example, the human–computer interaction scenario in the SAL database.

Comparing synthesised and human speech for training purposes, it is highly interesting that in the SS scenario (training on synthesised speech only) the average UAR of binary arousal classification across all test databases (64.8 %) is significantly higher than in the HS scenario (training on human speech only, 62.6 %). In contrast, for valence, the performance of synthesised training data (50.4 % average UAR) is observed significantly below the one of human data (53.3 % UAR), and is near chance level UAR (50 %). This phenomenon indicates a large mismatch between the features of the synthesised speech that is supposed to express negative valence, and the human utterances actually corresponding to negative valence (or being perceived as such by the human labellers). Generally, this result corroborates the well-known fact that variation of valence can only partly be modelled, and hence be generated, by variation of acoustic features.

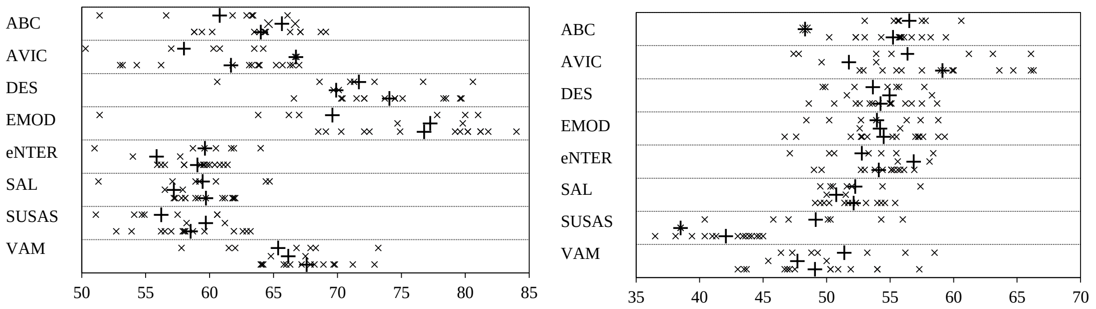
Third, when considering the performance of merging ‘HS’ and ‘SS’ data in training (65.2 % average UAR), we find a slight enhancement over training with only synthesised speech (64.8 % UAR), and a significant gain of 3 % absolute across all databases with respect to training with only human speech (62.6 % UAR). This performance enhancement by agglomeration of HS and SS training data is to be expected, since the performances of HS and SS on the individual databases suggest that they may have complementary strengths when used for model training (see Figure 4.2 and the discussion below).

Besides these promising improvements in a large scale perspective, without a doubt there are several noteworthy singular results that should not be overlooked. For example, we see that the synthesised speech prevails over human training data when testing on the EMO-DB (77.3 % on average); this is probably a consequence

Table 4.2: Mean and maximum UARs for varying training data in cross-corpus binary arousal/valence classification on eight test databases of human speech. Training with HS: eight databases of human speech, SS: two databases of synthesised speech, HS+SS: all possible permutations of human speech and synthesised speech databases. EMOD: EMO-DB, eNTER: eNTERFACE. [28]

Train on	**Test on**								
UAR [%]	ABC	AVIC	DES	EMOD	eNTER	SAL	SUSAS	VAM	**mean**
				Arousal					
Mean									
HS	60.8	58.0	71.7	69.6	59.6	59.4	56.2	65.4	**62.6**
SS	65.7	66.8	69.9	77.3	55.9	57.2	59.7	66.2	**64.8**
HS+SS	64.0	61.7	74.1	76.8	59.0	59.7	58.5	67.6	**65.2**
Max									
HS	66.1	64.2	80.3	71.0	64.0	64.7	60.6	73.2	**69.3**
SS	66.7	66.8	70.1	79.8	57.7	57.9	61.2	67.5	**66.0**
HS+SS	69.1	67.0	79.7	84.0	61.4	62.0	63.2	72.9	**69.9**
				Valence					
Mean									
HS	56.5	56.4	53.6	54.0	52.8	52.2	49.1	51.4	**53.3**
SS	48.3	51.8	55.0	54.2	56.9	50.8	38.5	47.7	**50.4**
HS+SS	55.2	59.1	54.2	54.5	54.1	52.1	42.1	49.1	**52.6**
Max									
HS	60.6	66.1	57.7	58.8	58.4	57.4	56.0	58.5	**59.2**
SS	48.4	53.9	58.3	55.8	58.1	51.5	38.5	50.0	**51.8**
HS+SS	59.4	66.3	58.7	59.3	56.9	55.4	45.0	57.3	**57.3**

of text dependency, because the sentences from the EMO-DB were used to synthesise the emotional training speech. In the same vein, the overall best result on the EMO-DB (84.0 %) is achieved by joining the DES database of acted emotions with synthesised speech from Mary. This, however, should not suggest that synthesised speech is only useful when the textual content matches: On the spontaneous, free-text AVIC database, both variants of synthesised speech deliver 8.8 % absolute higher UAR (66.8 %) than human speech on average (58.0 %), and are observed above the best single human speech database, which is, interestingly, the acted DES database (64.2 % UAR). Looking at the maximum UAR values in Table 4.2, we find other surprising cases of databases that seem to 'match' particularly well. For example, the best result in cross-corpus arousal classification on the DES database is achieved by using the spontaneous VAM database for training (80.3 % UAR);



(a) arousal (b) valence

Figure 4.2: Distributions of UARs for cross-corpus binary arousal/valence classification of eight test databases: Single-database classifiers are depicted by crosses and the average of single-database classifiers by a plus sign. For each test database, the top row corresponds to training on human speech, the middle row to training with synthesised speech, and the bottom row to merging of one human database with one synthesised speech database. [28]

even more notably, the same holds *vice versa* (training on DES and testing on VAM delivers the best single result of 73.2 % UAR on VAM). This apparent similarity of DES and VAM is also reflected in the fact that both are 'equally hard' to classify by synthesised speech as opposed to human speech (max. UAR of 70.1/80.3 % for DES, max. UAR of 67.5/73.2 % for VAM). The latter also indicates that the evident mismatch between the synthesised speech and DES is not simply caused by different languages (Danish/German): VAM is in German as is the synthesised speech.

To give an overview of the performance and its variability of the various training and testing permutations, I visualise the distributions of the UAR for the three kinds of training scenarios in Figure 4.2 as one-dimensional scatter plots. For each human testing database, the top row shows results obtained by human speech training sets, the middle row corresponds to synthesised speech training sets, and the bottom row refers to training sets obtained by merging one human database with one synthesised speech database. The 'plus' symbols indicate the average performance per row. From Figure 4.2, it is obvious that training with single databases of human speech results in greatly varying performance. This effect is most visible for the acted test databases (DES and EMO-DB), where the UAR of binary arousal classification with training on human speech ranges from 60.6 % to 80.6 % (DES), and from 51.4 % to 81.0 % (EMO-DB). The latter is somewhat expected since for databases with limited variation of content, there is a larger chance that some training databases will strongly 'mismatch' the testing data. One can see that especially for arousal classification (Figure 4.2 (a)), this variability can be partly compensated by adding synthesised data to the training set; this effect is clearly visible for all but the AVIC and SUSAS databases. For valence classification (Figure 4.2 (b)), we cannot

Table 4.3: Min(imum), max(imum) and mean UARs when training on one of 8 databases and testing on the 7 remaining databases. [221]

Train on **UAR** [%]	**Test on 7 remaining databases**					
	Arousal			Valence		
	min	max	mean	min	max	mean
ABC	52.5	73.6	59.8	47.8	58.5	53.3
AVIC	55.0	66.6	59.5	43.7	56.6	51.7
DES	58.8	80.4	66.6	**49.2**	**64.1**	**54.8**
EMOD	54.9	72.9	62.5	45.6	60.5	51.3
eNTER	51.1	68.4	60.0	48.9	57.9	54.3
SAL	54.1	76.7	63.8	47.0	57.8	51.4
SUSAS	52.2	69.5	57.1	47.1	56.3	51.7
VAM	**60.6**	**80.6**	**67.7**	48.8	51.3	50.2

observe such decreases in variability, which can be attributed to the generally lower classification performance that is often near chance level.

4.1.3 Distance-Based Data Selection

By implementing the methods evaluated in Section 4.1.1 and 4.1.2, a growing amount of training data can be accumulated. At the same time, however, it has some drawbacks, for example, the model complexity and training time are increased, as stated in Section 3.1.5. To overcome these issues, this section tries to examine the *Euclidean distance-based data selection* (EDDS) (cf. Section 3.1.5.1) in a cross-corpus scenario.

Methods for pruning atypical instances from training have been thoroughly explored in pattern recognition [222] and particularly SER [223]; still, such experiments are limited to training and testing on the same data set. On the other hand, first studies on feature selection in cross-corpus SER suggest that training optimisations do not always generalise across different data sets [224]. Here, the goal is to find objective measures for databases and instances that are correlated with the expected accuracy in cross-corpus SER. Particularly, it deals with the question whether selecting the most ‘prototypical’ instances and databases for model building enables generalisation across corpora.

For the experiments, the same eight human emotional speech databases (i.e., ABC, AVIC, DES, EMOD, eNTER, SAL, SUSAS, and VAM), as well as the same feature set (6 552 attributes) and the classifier (i.e., SVM, c=0.05) as those in Section 4.1.2 were selected. Note that, the signs of ‘+’ and ‘-’ in the following indicate the positive and negative classes, respectively.

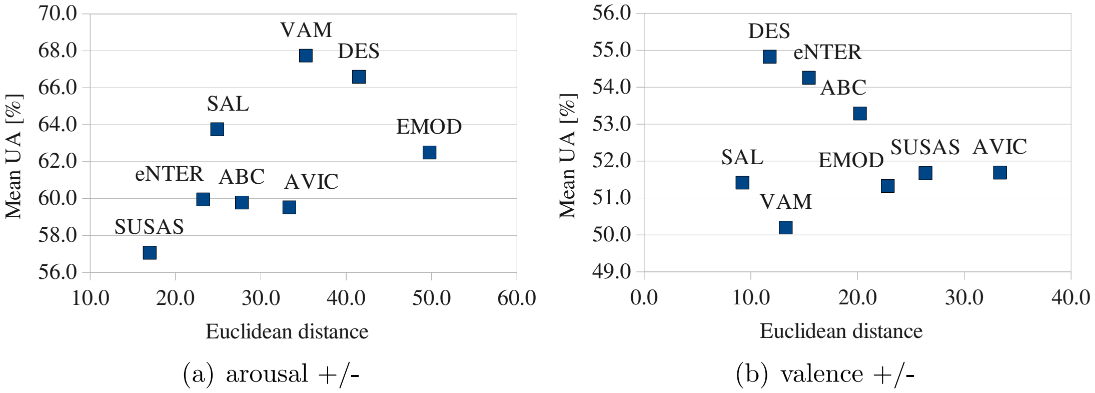
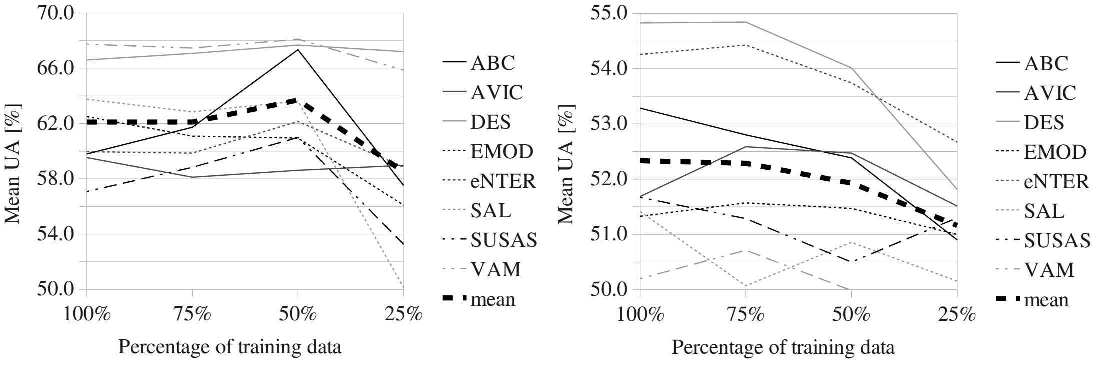


Figure 4.3: Training database selection: Mean UAR in cross-corpus testing – i.e., training on one database and testing on the remaining seven – and its relation to the Euclidean distance of class centres of '+' and '-' instances (after z-normalisation), for arousal (a) and valence (b). [221]

Database Selection

For each of the eight databases, the performance in a cross-corpus evaluation on the seven other databases was firstly evaluated. Results are shown in Table 4.3. For arousal, interestingly, training with the VAM database of spontaneous, natural speech yields highest average UAR (67.7 %), minimum UAR (60.6 % on SUSAS), and maximum UAR (80.6 % on DES). The second best training corpus is the DES database of acted emotions (66.6 % mean UAR). In contrast to arousal, recognition of valence seems to be very challenging, resulting in no more than 54.8 % average UAR, which is achieved by training with DES. In fact, it is well known that valence recognition from purely acoustic features is challenging, and even more so in cross-corpus testing.

In the following, I investigated the relation between the 'prototypicality' of a database and the expected UAR in cross-corpus SER when using that database for training. As a measure of prototypicality, I calculated the Euclidean distance d of the class centre of 'positive' instances, $\overline{\mathbf{x}}_+ = \mathrm{E}\{\mathbf{x}_+\}$, and the one of 'negative' instances, $\overline{\mathbf{x}}_- = \mathrm{E}\{\mathbf{x}_-\}$. Figure 4.3 (a) shows the results for recognition of positive and negative arousal. Generally, training with databases that exhibit large distances between positive and negative classes delivers higher UA; notably, this prototypicality in the feature space does not exactly correspond to the notion of acted vs. spontaneous emotion: Consider the similar prototypicality measure of the VAM and DES databases. Furthermore, training on the highly prototypical EMOD only delivers mediocre results (62.5 %), seemingly due to insufficient generalisation. Overall, the Spearman (rank) correlation between $d(\overline{\mathbf{x}}_+, \overline{\mathbf{x}}_-)$ and the mean UAR is $\rho = .571$, which is however not of statistical significance due to the small sample size (8).



(a) arousal +/-

(b) valence +/-

Figure 4.4: Training instance selection: Relation between mean UAR in cross-corpus testing – i.e., training on one database and testing on the remaining seven – and the amount of training data chosen in order of prototypicality, measured as the Euclidean distance from the class centre of the opposite class. [221]

Analogously, results for valence recognition are shown in Figure 4.3 (b). As opposed to the arousal case, for valence there is no clear trend as to whether one can expect a gain by using more prototypical databases as training data (Spearman's $\rho = -.060$). Still, the lower recognition rates compared to arousal are reflected in generally smaller distance between the class centres.

Instance Selection

The second experiment evaluated the effect of restricting the training to prototypical instances, for each database. To this end, I computed for each positive instance $\mathbf{x}_+$ the distance to the class centre of the negative instances, $d(\mathbf{x}_+, \overline{\mathbf{x}}_-)$. Then, I computed the quartiles of the distribution of $d(\mathbf{x}_+, \overline{\mathbf{x}}_-)$ and selected only the instances corresponding to the fourth quartile (in other words, the 25 % most prototypical positive instances). An analogous procedure was followed for selection of negative instances, which were selected according to the distance $d(\mathbf{x}_-, \overline{\mathbf{x}}_+)$ from the positive class centre. For each database, the selected positive and negative instances were joined, and the resulting model was evaluated on the seven other databases. We repeated the experiment using the 50 % (quartiles 3 and 4) and 75 % (quartiles 2–4) most prototypical instances, respectively.

Results are shown in Figure 4.4. For arousal recognition (Figure 4.4(a)), one gains almost 2 % absolute UAR on average across the eight databases when using only the 50 % most prototypical instances for training—this result suggests that the 'manual' process of instance selection is complementary with the instance weighting performed in SVM optimisation. The improvement is most visible for training with the ABC database, where an absolute gain of 7.5 % UAR is achieved. However, a drop in performance occurs when further restricting the amount of training data.

Furthermore, cross-corpus *valence* recognition (Figure 4.4(a)) cannot generally (i.e., on average) be improved by selecting training instances using the proposed method, despite slight UAR gains for the eNTERFACE, EMO-DB, ABC and VAM databases.

4.1.4 Agreement- and Sparseness-Based Data Selection

This section examines another data selection method, named Agreement- and Sparseness-Based Instance Selection (ASIS) (cf. Section 3.1.5.2), by which 1) the instances with lowest labelling agreement are pruned; and 2) an equal number of instances from each class are selected.

One of the motivations of this method is the subjectivity of paralinguistic phenomena. Unlike traditional pattern recognition tasks where a true 'ground truth' is available, those tasks only have 'gold standard' labels, which are often assigned by (sometimes weighted) majority voting over multiple human ratings (cf. Section 2.2.1.2). In fact, instance labelling for such tasks highly depends on the labellers' personal judgements. Instances with high labelling uncertainty could potentially cause the model to over-fit these 'noisy' instances, which results in increased complexity [225]. This case thus would deteriorate the generalisation performance.

Another motivation for using this method is relevant to the unbalanced distribution of classes, which is most pronounced in natural and spontaneous speech databases, where 'neutral' speech is much more frequent than clear-cut cases of emotional or other target speech. This phenomenon results in some models favouring the majority classes and showing bad performance on the sparse (minority) classes. However, these sparse classes are indeed usually of most interest in practical applications.

To proceed with the experiments, I selected the well-standardised machine learning task and corresponding database from the INTERSPEECH 2009 Emotion Challenge (EC) [63]. The introduction of the database – FAU Aibo Emotion Corpus (AEC) – can be found in Appendix A.1.1. Particularly, based on the definition of *human agreement level* in Section 3.1.1, Figure 4.5 displays the instance distribution of training set with human agreement levels. In addition, the same classifier of SVM and the acoustic features (384 attributes) were kept in line with the EC [63].

Experimental Results

The following experiments were executed for SER using different variations of the instance selection algorithm: 1) only agreement-based instance selection ('AIS') based on discarding low-agreement instances (cf. Step 'AIS' in Algorithm 1); 2) only sparseness-based instance selection ('SIS') by selecting sparse instances (cf. Step 'SIS' in Algorithm 1); 3) combining both steps (ASIS) (random selection with

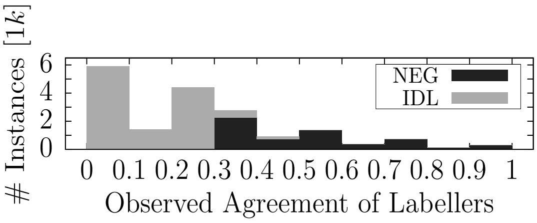


Figure 4.5: Number of instances with according observed agreement of labellers in the AEC. [183]

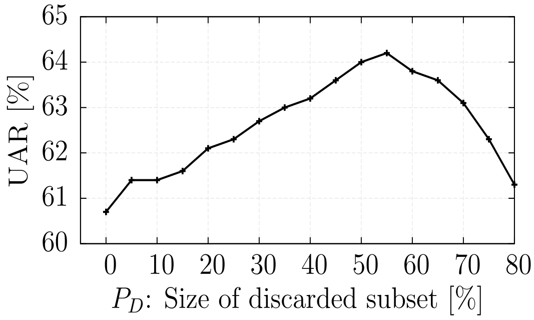


Figure 4.6: Agreement-based Instance Selection (AIS): UAR on the AEC test set after discarding low agreement training instances (no balancing). [183]

balancing of instances across classes). For comparison, I denote the control methods of Random Instance Selection (RIS) as randomly selecting a predefined number of instances from the whole set.

Figure 4.6 provides an overview of performances after discarding a certain ratio of instances with low human agreement (AIS). Note that the human agreement levels by discarding {5, 10, 20, 30, 40, 50, 60}% of the instances for the class IDL are {0.4, 0.52, 0.60, 0.60, 0.60, 0.72, 0.90}, respectively; and for the class NEG, they are {-0.28, -0.2, -0.2, -0.2, -0.08, 0.06, 0.2}, respectively. No instance balancing is performed here. The performance of the classifier improves continuously and significantly (one-sided z-test) until 55 % of the training set instances with human agreement are discarded (from 60.7 % to 64.2 % UAR). Figure 4.7 compares the performance of two instance sub-sampling strategies (with (SIS) and without balancing), both without any prior discarding of low agreement instances. As expected, UAR is increased by about 8 % absolute when balancing is performed, showing the importance of a balanced distribution for SVM (and further) classifiers. Figure 4.8 shows results obtained when randomly sub-sampling the training set and balancing after discarding low agreement instances (ASIS). At a certain ratio of discarded instances, increasing the number of selected instances enhances the system robustness.

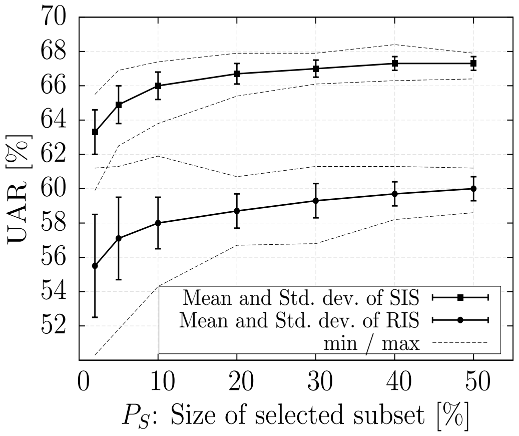


Figure 4.7: Sparseness-based Instance Selection (SIS): UAR mean, standard deviation (std. dev.), minimum (min), and maximum (max) on the AEC test set over 40 independent runs. Comparison of balanced SIS and random instance selection (RIS) from the training set. No discarding of instances with low agreement ($P_D = 0$). [183]

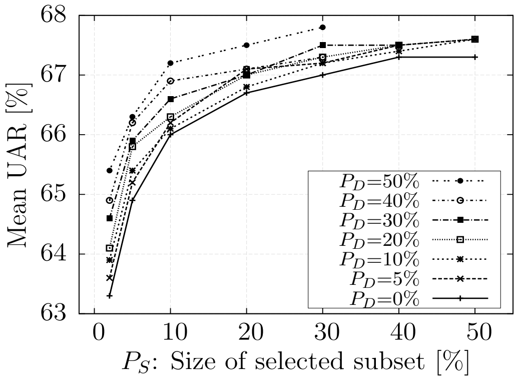


Figure 4.8: Agreement- and Sparseness-based Instance Selection (ASIS): Mean UAR on the AEC test set in 40 independent runs of *balanced sub-sampling* after discarding P_D instances with lowest labelling agreement from the training set of the AEC. [183]

As more instances are added, however, the increase of UAR converges. At a certain amount of sub-sampling, discarding up to 50 % of low agreement instances improves UAR. Note that this improvement is more obvious for a small subset size, as in this case when the disturbing influence of the low agreement instances has a larger relative impact on the model. The best result of 67.8 % of UAR is achieved by discarding 50 % of lowest agreement instances and selecting only 30 % of instances (relative to the whole set) for model building. This result is equivalent to the baseline (67.7 %

of UAR) in [63], in which the whole training set with SMOTE is considered (for balancing). Note that in the experiments, the amount of sub-sampling is limited by the size of the minority class 'NEG'.

4.1.5 Semi-Supervised Learning

To evaluate the effectiveness of Semi-Supervised Learning (SSL) for SER, I selected six databases, i.e., ABC, AVIC, DES, eNTER, SAL, and VAM. These corpora cover a broad variety of data from acted (DES) over simulated (ABC, eNTERFACE) to spontaneous emotional speech (AVIC, VAM), and from strictly limited textual context (DES) over more variation (eNTERFACE) to full variance (AVC, AVIC, SAL, VAM). Three languages (English, German, and Danish) belonging to the same family of Germanic languages are contained. Moreover, the speaker characteristics, the recording conditions, as well as the annotators vary greatly among these databases. An overview of these six corpora and their category mapping strategy are shown in Appendices A.1.2 and A.1.4, respectively.

Moreover, the acoustic feature set (6 552 attributes) and classifier (i.e., SVM, c=0.05) were kept in line with the experiments in Section 4.1.2 [226]. In the experiments, a Leave-One-Corpus-Out (LOCO) strategy was used, i.e., one corpus was used as test set and the remaining five were used for (supervised or unsupervised) training. Training data were always agglomerated on the instance level by simply joining databases for training ('pooling'), as this strategy has shown superior classification performance in comparison with late decision fusion for cross-corpus LOCO evaluation with SVM in Section 4.1.1.

Normalisation

In automatic speech and speaker recognition, methods such as cepstral mean subtraction or joint factor analysis are widely used to mitigate the diversities among speakers and acoustic environments. In the case of cross-corpus SER, differences do not only exist within corpora among different speakers (intra-corpus) but also in between corpora (inter-corpus) due to various recording settings and languages (cf. Table A.2); consequently, the impact of normalisation techniques on cross-corpus recognition rates has been demonstrated [23]. In this section, I investigate three kinds of normalisation methods: centring, normalisation and standardisation (cf. Section 3.1.3). These three methods can be applied to each corpus separately (i.e., before data agglomeration) or after building a joint training set from multiple databases. In Table 4.4, I compare the mean UAR across databases in LOCO evaluation with the three above named normalisation methods. Since the databases vary greatly in size (cf. Table A.2), I present both, the unweighted mean and the mean UAR weighted by number of instances in the database.

Table 4.4: Normalisation in leave-one-corpus-out cross-corpus binary arousal/valence classification: Test on 6 databases and training on 5 remaining databases. UAR for centring (C), min-max normalisation (M) and z-normalisation (Z) on corpus before and after data agglomeration (agg.), and both. W-Mean: mean weighted by number of instances as opposed to Mean: mean over the results of the corpora without weighting by the number of instances within the corpora. [226]

UAR [%]	**Before agg.**			**After agg.**			**Both**	
Test on	**C**	**M**	**Z**	**C**	**M**	**Z**	**C**	**Z**
				Arousal				
ABC	63.1	64.5	66.6	64.3	60.2	61.0	63.4	65.5
AVIC	55.9	55.1	62.0	55.9	59.0	62.7	55.8	61.4
DES	76.1	79.1	78.3	74.9	66.3	74.4	77.9	80.1
eNTER	62.7	60.0	61.6	61.7	57.8	61.6	63.3	60.8
SAL	60.0	55.4	61.6	64.4	51.2	64.7	62.9	63.3
VAM	64.6	58.2	69.2	65.8	58.3	67.4	67.4	69.7
W-Mean	60.5	58.2	63.9	62.9	57.5	64.1	61.6	64.0
Mean	63.7	62.1	66.6	65.2	58.8	65.3	65.1	66.8
				Valence				
ABC	63.6	62.2	62.3	63.3	58.0	59.7	63.6	62.3
AVIC	61.8	51.7	57.8	61.8	50.1	60.0	61.8	57.9
DES	57.0	56.3	59.7	57.0	61.1	57.9	56.8	59.7
eNTER	57.4	56.0	58.2	56.5	55.2	57.4	57.4	58.2
SAL	54.3	50.0	53.4	54.3	51.1	55.5	54.3	53.4
VAM	54.4	51.5	52.0	56.4	53.0	54.0	54.5	52.0
W-Mean	58.4	52.8	56.6	58.5	52.5	57.7	58.4	56.6
Mean	58.1	54.6	57.2	58.2	54.8	57.4	58.1	57.3

When applying normalisation per corpus before data agglomeration, it can be seen that z-normalisation delivers a vast improvement for arousal recognition both over min-max normalisation and centring: The (unweighted) mean UAR is 66.6 % for z-normalisation compared to 62.1 % (min-max normalisation) and 63.7 % (centring). For valence, interestingly, simple centring delivers best results (58.1 %), and min-max normalisation severely deteriorates the results (54.6 %). The largest improvement in accuracy of arousal classification by standardisation instead of centring or min-max normalisation is found for the AVIC and VAM databases of spontaneous speech. These results are mirrored to a great extent in the results for normalisation *after* data agglomeration, and in general the per-corpus normalisation cannot be outperformed. I also investigated a combination of normalisation both before and

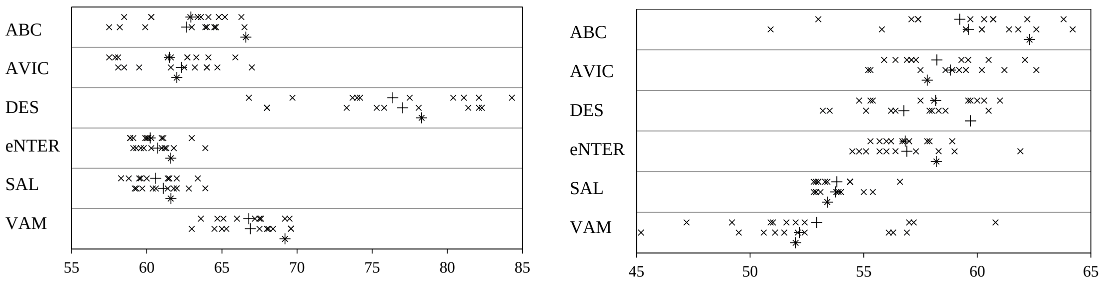
Table 4.5: Mean and the maximum UAR of supervised and unsupervised training for cross-corpus binary arousal/valence classification. Pool 3: agglomeration of three corpora; Pool 5: agglomeration of five corpora; Pool 3 + 2: agglomeration of three labelled corpora and two unlabelled corpora for unsupervised learning; W-Mean: mean weighted by number of instances. [226]

UAR [%]	**Pool 3**		**Pool 3 + 2**		**Pool 5**
Test on	**Mean**	**Max.**	**Mean**	**Max.**	**Value**
			Arousal		
ABC	62.9	66.3	62.7	66.5	66.6
AVIC	61.5	65.9	62.3	67.0	62.0
DES	76.4	84.3	77.0	86.1	78.3
eNTER	60.2	63.0	60.7	63.9	61.6
SAL	60.6	63.4	61.1	63.9	61.6
VAM	66.8	69.5	66.9	69.6	69.2
W-Mean	62.6	66.3	63.2	67.1	63.9
Mean	64.7	68.7	65.1	69.5	66.6
			Valence		
ABC	62.2	65.0	62.3	64.7	63.6
AVIC	56.6	60.9	60.5	64.6	61.8
DES	52.6	57.0	54.4	56.6	57.0
eNTER	55.8	58.3	55.7	58.4	57.4
SAL	53.5	56.0	53.1	55.2	54.3
VAM	54.2	56.3	54.5	58.6	54.4
W-Mean	55.6	58.9	57.1	60.4	58.4
Mean	55.8	58.9	56.6	59.7	58.1

after agglomeration; the mean UAR in arousal recognition in that case is 65.1 % for centring and 66.8 %. In all further experiments, I used z-normalisation on the corpus level for arousal and centring for valence recognition. Note that the per corpus normalisation strategy is very convenient in practice as it does not require retraining when adding further databases to the training set.

Unsupervised vs. Supervised Learning

To determine the potential of unsupervised learning for SER, I considered three different experimental settings: First, I agglomerated ('pooled') together three corpora for training and tested on one database (corresponding to 'Pool 3' in Table 4.5). This results in ten possible training set permutations for each of the six test sets. Second, I agglomerated three corpora for training and two corpora for unsupervised adaptation, and tested on the remaining corpus (i.e., I used three corpora



(a) Arousal (b) Valence

Figure 4.9: Distributions of UARs for cross-corpus binary arousal/valence classification of six test databases. Crosses refer to the individual training set combinations, and the plus sign refers to the average performance. The top row per test database depicts results obtained by pooling three training corpora, the row in the middle refers to pooling three corpora and fusing two corpora for unsupervised adaptation, and the bottom row represents pooling five training corpora. Note that in the last case no permutations are possible, as six corpora are used and five are pooled for training, whereas in the other cases several permutations exist. [226]

to build models that are used to generate predictions for the two further corpora that in turn are used for unsupervised learning). This series of experiments is denoted by 'Pool 3 + 2' in Table 4.5. Note that due to the varying size of the corpora, this covers both settings where little labelled data are available as a 'seed', and an 'unsupervised adaptation' scenario where the amount of unlabelled data are rather small compared to the available labelled data. Finally, as a reference for supervised learning, I considered agglomerating together five databases for training, again testing on the remaining corpus. Table 4.5 shows the UAR obtained for the two-class arousal and valence classification task when evaluating the three training scenarios. Using a set of three databases for training leads to an average UAR of 64.7 % and 55.8 % for arousal and valence, respectively. Unsupervised adaptation with two additional corpora increases average recognition performance to 65.1 % and 56.6 %, respectively. The most impressive gain is seen for the AVIC database of spontaneous speech: Here, unsupervised training even slightly outperforms supervised training for arousal recognition, and gives a boost in accuracy of almost 4 % absolute for valence (compared to 5 % for supervised training). Still, as expected, the best average result is obtained when using the labels of all five corpora for training (UAR of 66.6 % and 58.1 %, respectively).

Figure 4.9 depicts the distributions of UAR for the six test databases. The plus sign indicates the UAR averaged over all test sets. The top row per test database depicts results obtained by agglomerating three training corpora, the middle row refers to agglomerating 3 corpora and fusing two further corpora for unsupervised learning, and the bottom row shows the results for agglomerating all five training corpora with known ground truth. From Figure 4.9, it can be seen that unsupervised learning outperforms the baseline setting (i.e., using only three corpora without further data agglomeration) in 5 of 6 cases for arousal, but only 3 of 6 cases for valence, which can probably be attributed to generally insufficient robustness of cross-corpus valence recognition from acoustic features. Overall, in terms of (weighted) mean UAR in arousal and valence recognition, addition of unlabelled training data delivers roughly half of the gain that can be expected from adding labelled training data, as in previous studies in speech recognition [227].

4.1.6 Co-Training

The results obtained in Section 4.1.5 demonstrate the efficiency of SSL. Another SSL method with the potential to exploit unlabelled data is co-training [174]. As described in Section 3.2.2.2, co-training focuses on building two learners by maximising the mutual agreement on two distinct 'views' of the unlabelled data set. In previous work, co-training was applied in several areas, such as documents classification [174] and handwriting [228] classification. In this section, I attempt to investigate the performance of co-training for SER. To do this, the conventional SSL is referred to as self-training (cf. Section 3.2.2.2), which is considered to be the baseline.

Experiments

For the experiments, two databases were selected – the AEC and SUSAS. Both databases consist of natural speech samples and are widely used in the field of SER [63, 229, 23]. The description of the two databases can be found in Sections A.1.1 and A.1.2. Similarly to the AEC database, four stress classes of SUSAS are divided into two stress-intensity cover classes – **HIGH** (i.e., *high stress* and *screaming*) and **LOW** (i.e., *neutral* and *medium stress*). In order to perform a speaker independent evaluation, the validation set contains 1 064 instances recorded from one male speaker and one female speaker, and the unlabelled pool set includes the remaining instances (2 529). Table 4.6 gives the distribution of speakers and instances per partition for both databases.

In addition, to verify the robustness of co-training, two standard sets of acoustic features were selected – the INTERSPEECH 2009 (IS09) EC [63] and the INTERSPEECH 2010 (IS10) Affect Sub-Challenge (ASC) [64]. The former contains 384 features that result from a systematic combination of 16 LLDs and corresponding first order delta coefficients with 12 functionals. The latter is an extension of the former, designed to cover a wider range of features relevant for paralinguistic information retrieval [64]. All features were extracted using the openSMILE framework

Table 4.6: Distributions of speakers and instances per partition of the FAU Aibo Emotion Corpus (AEC) [53] and the Speech Under Simulated and Actual Stress (SUSAS) [230]. NEG: negative emotions; IDL: neutral and positive emotions; HIGH: high stress; LOW: low stress.

	# speakers		**# instances**		
AEC	male	female	NEG	IDL	Σ
Pool	13	13	3 358	6 601	9 959
Validation	8	17	2 465	5 792	8 257
Σ	21	30	5 823	12 393	18 216
SUSAS	M	F	HIGH	LOW	Σ
Pool	3	2	1 116	1 413	2 529
Validation	1	1	500	564	1 064
Σ	4	3	1 616	1 977	3 593

[62]. Table 4.7 shows the LLDs and functionals for both feature sets. For more details, please see [63, 64]

As described in Section 2.4.1, SVMs were used as the modelling paradigm. In accordance with the IS09 EC baseline specifications, the SVMs were initially trained with SMO algorithm with a linear kernel and a complexity constant of 0.05. Logistic regression modelling was enabled to allow converting the SVMs' output distances to confidence values (cf. Section 3.2.1). In addition, an upsampling strategy was adopted for evening class distribution (cf. Section 3.1.4). The training process was repeated 20 times with different initialisations of the random generator for each experimental condition.

Four different experiments were conducted to evaluate the performance and robustness of co-training. The first two experiments were designed to evaluate the performance of the various learning methods with different numbers of initial training instances using the AEC corpus and the IS09 EC feature set. In this section, 200 and 500 instances of the AEC database were used for initial training, which corresponds to approximately 2 % and 5 %, respectively, of the whole pool. In the third experiment, the various learning strategies were evaluated with the AEC corpus and a new feature set (IS10 ASC) so as to establish the robustness of co-training for different feature sets (using 200 initial training instances). In the final experiment, an additional corpus (SUSAS) was used with the IS10 ASC feature set to evaluate the robustness of co-training across tasks (with 100 initial training instances, approximately 5 % of the whole pool). For the four experiments, the UARs obtained after the *initial* supervised training were: 1) 60.9 % (std = 1.8); 2) 62.6 % (std = 1.1); 3) 64.4 % (std = 1.3); and 4) 58.6 % (std = 2.5). The performance when training the SVMs with the *full* set of training data was: 1) 67.7 %; 2) 67.7 %; 3) 67.2 %; and 4)

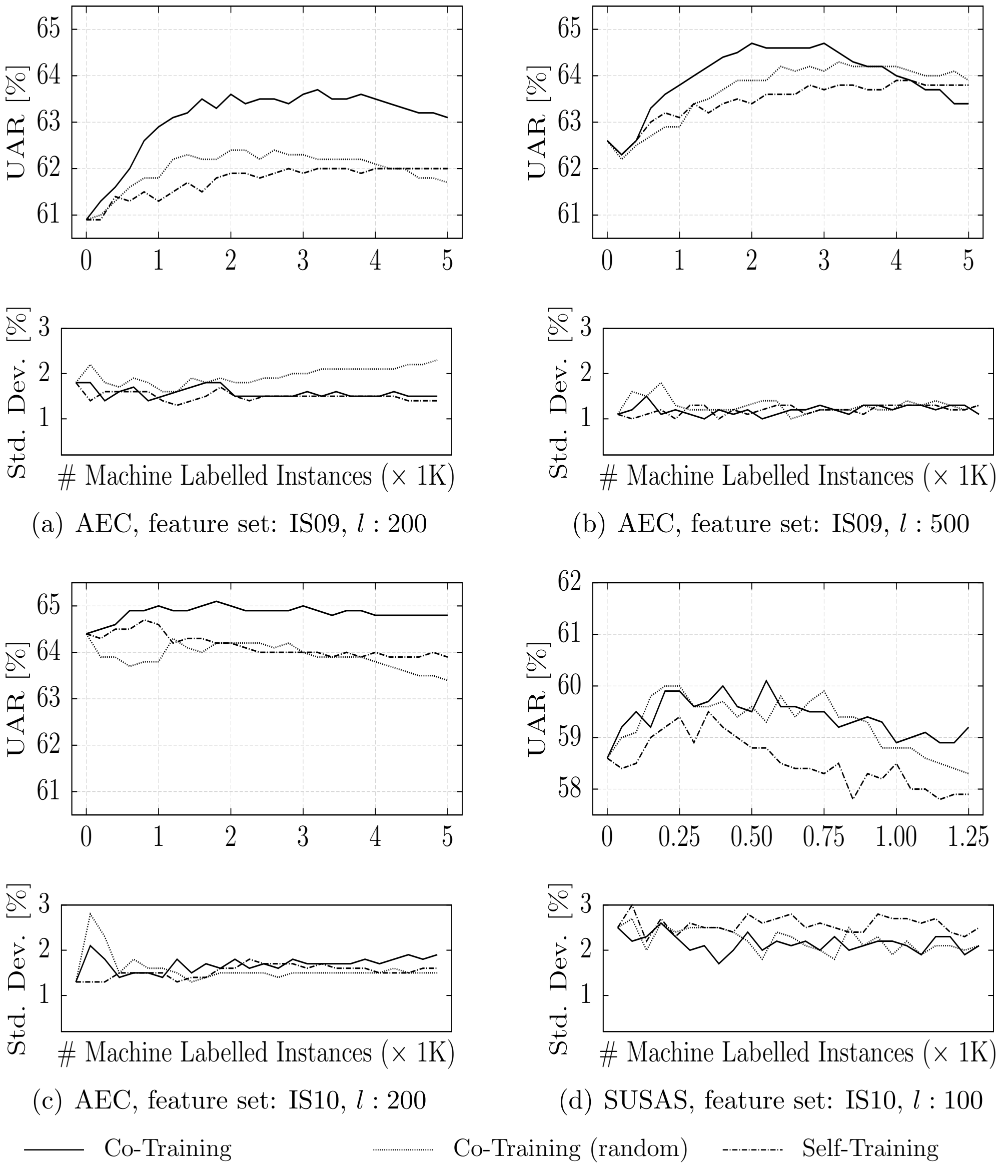
Table 4.7: The IS09 EC and the IS10 ASC Acoustic feature sets used in these experiments: LLDs and respective functionals. The $*$ symbol indicates the features belonging to view-1 for the co-training. [231]

LLD (Δ)	**Functionals**
IS09 EC feature set (384)	
ZCR	mean
RMS Energy	standard deviation energy
F0	kurtosis, skewness
HNR	extremes: value, rel. position, range
MFCC 1-12*	linear regression: offset, slope, MSE
IS10 ASC feature set (1 582)	
PCM loudness	position maximum/minimum
MFCC 0-14*	algorithmic mean, standard deviation
log Mel freq. band 0-7	skewness, kurtosis
line spectral pairs freq. 0-7	linear regression coefficients 1/2
F0	linear regression error quadratic/absolute
F0 envelope	quartile 1/2/3
voicing probability	quartile range 2-1/3-2/3-1
jitter local	percentile 1/99
jitter consec. frame pairs	percentile range 99-1
shimmer local	up-level 75/90

64.6 % UARs.

In all experiments, the instances that are not used for the initial training were used for the unlabelled data pool. For AEC, 500 instances are selected per iteration for self-training and co-training. Thus, each ‘view’ of co-training selects 250 instances due to the equal number of selected instances per ‘view’. For SUSAS, given the smaller size of the database (approximately 25 % of the AEC), only 125 instances are selected in each learning iteration for self-training and co-training.

For the creation of each ‘view’ used for co-training, the full feature set is split into two partitions - one comprising MFCCs (view-1) and the other the remaining LLDs (view-2). This partitioning is well motivated by size of the feature sets and different characteristics. In fact, more ‘traditional’ paralinguistic feature sets use prosodic and further non-cepstral information, whereas MFCCs on their own are a common set in speaker identification and speech recognition that increasingly found its way into general paralinguistics. Nevertheless, although such feature separation is only related to LLDs and not to higher level features of functionals or linguistics, the feature sets in the two views may not be conditionally independent, as, for example, a change in the signal which affects F0 or energy, etc., will also affect the



(a) AEC, feature set: IS09, l : 200

(b) AEC, feature set: IS09, l : 500

(c) AEC, feature set: IS10, l : 200

(d) SUSAS, feature set: IS10, l : 100

Figure 4.10: Comparison between co-training using the feature separation method based on cepstral LLDs, co-training using a random feature separation method, and self-training. The charts show the average UARs across 20 independent runs (and respective standard deviations) vs. number of *machine* labelled instances for the four experiments described in this section: a) AEC database with the IS09 EC feature set and 200 initial supervised training instances; b) AEC database with the IS09 EC feature set and 500 initial supervised training instances; c) AEC database with the IS10 ASC feature and 200 initial supervised training instances; and d) the SUSAS database with the IS10 ASC feature set and 100 initial training instances. [231]

MFCCs. However, the effect will be different, thus likely adding complementary information. Furthermore, the experimental results in [232] demonstrate that such feature separation criterion applied to multi-view learning is valid and effective. The ratio of attributes (view-1/view-2) is 288/96 for the IS09 EC feature set and is 630/952 for the IS10 ASC feature set.

Results

Figure 4.10 shows the average and standard deviation of the UAR measure for the self-training and co-training approaches under study. The error measures shown correspond to the average of the individual performances across 20 independent runs of the learning process for all four experiments described in this thesis.

The first observation is that co-training using the feature separation based on cepstral LLDs improves the initial classification performance in all above mentioned four experimental scenarios. Co-training using random feature separation does not lead to improvements using the IS10 feature set and AEC database (see subfigure 4.10 (c)). Self-training leads to improvements in the experiments using the IS09 feature set, but not in those with the IS10 one (see subfigures 4.10 (c) and (d)). Overall, co-training with cepstral LLDs feature separation seems to be more robust than the other two approaches when using different number of initial supervised training instances, different databases and different feature sets. Furthermore, it outperforms the other approaches after only a few iterations, which suggests that this algorithm leads to faster learning process and better generalisation performance. Finally, it is also noticeable that the performance of co-training degrades after a certain number of learning iterations. Previous work (e.g., [174, 233]) has demonstrated that this phenomenon can be attributable to the exchange of mislabelled instances between the different ‘views’.

4.1.7 Active Learning and Co-Active Learning

SSL techniques could deliver good results without any intervention of human annotators for SER (cf. Section 4.1.5 and 4.1.6). Yet, they have several disadvantages, for example, prediction bias and noise accumulation (cf. Section 3.2.4).

This section attempts to evaluate the human oracles by means of Active Learning (AL) and co-Active Learning (coAL), both of which are based on CV. In the ‘hay stack’ of speech data, AL is a way to automatically identify the ‘needles’, i.e., the most informative instances (cf. Section 3.2.3.1). Motivated by the ‘multi-view’ idea of co-training, a novel AL – coAL (cf. Section 3.2.3.2) – is also accessed in this section.

Experiments and Results

For the experiments, all setups were kept in line with the ones in Section 4.1.6 for the sake of comparison except the instance number selected per learning iteration. Here, for AL and coAL, only 200 instances were selected for labelling. As a result, each view of coAL selected 100 instances per learning iteration.

The performance of the PL, AL with least (lc) and medium (mc) certainty query strategies, and coAL algorithms (cf. Section 3.2.3.1) are shown in Figure 4.11, which depicts the performance figures averaged across 20 independent runs of the whole training process (and respective standard deviations) for the four experimental scenarios (the results of CL, also shown, will be described later).

As can be seen, the sequential addition of human-labelled instances to the training set (200 for AEC and 50 for SUSAS per iteration) lead to improvements of classifier performance for all four supervised learning approaches. Nonetheless, contrary to our expectations, the coAL approach does not show an improvement over the AL algorithms. The best global performance is delivered by the AL with medium certainty query strategy, especially in relation to the AEC database. The exception to this rule, as can be seen on Figure 4.11 (d), is the performance of the SUSAS database, which is particularly worse than the other algorithms for fewer human-labelled instances. In this task, the AL with the least certainty query strategy performs better. Regarding the amount of labelled data used, the AL approaches with either least or medium certainty strategies achieve a similar performance to that of the baselines when the models are trained with the full set of training data. Nevertheless, it uses 55 %, 50 %, 70 %, and 65 % fewer human-labelled instances in each of the four experimental scenarios. Therefore, AL methods efficiently reduce the amount of required human-labelled data.

4.1.8 Cooperative Learning

Both SSL and AL have their own advantages and disadvantages: AL algorithms generally improve a model's performance, but they still require a considerable amount of costly human intervention. SSL techniques, instead, exploit machine labelling of data, yet usually cannot improve the performance of an existing classifier as much as AL techniques when the same number of instances are labelled [172]. In order to take advantage of the best of both approaches, a cooperative learning (CL) method is proposed. It combines AL and SSL, which allows sharing the labelling effort between human and machine oracles while being able to ease the drawbacks of each method.

In this section, three types of schemes – single-view CL (svCL), mixed-view CL (xvCL), and multi-view CL (mvCL) – are examined for SER.

Table 4.8: Mean and standard deviations of UAR performance measure obtained by averaging the results between iterations 4 and 12 (800 ∼ 2 400 instances for AEC, and 200 ∼ 600 instances for SUSAS). Values are shown for Passive Learning (PL), Active Learning (AL), co-Active Learning (coAL), and single-/mixed-/multi-view Cooperative Learning (svCL/xvCL/mvCL) for the four experimental scenarios. [231]

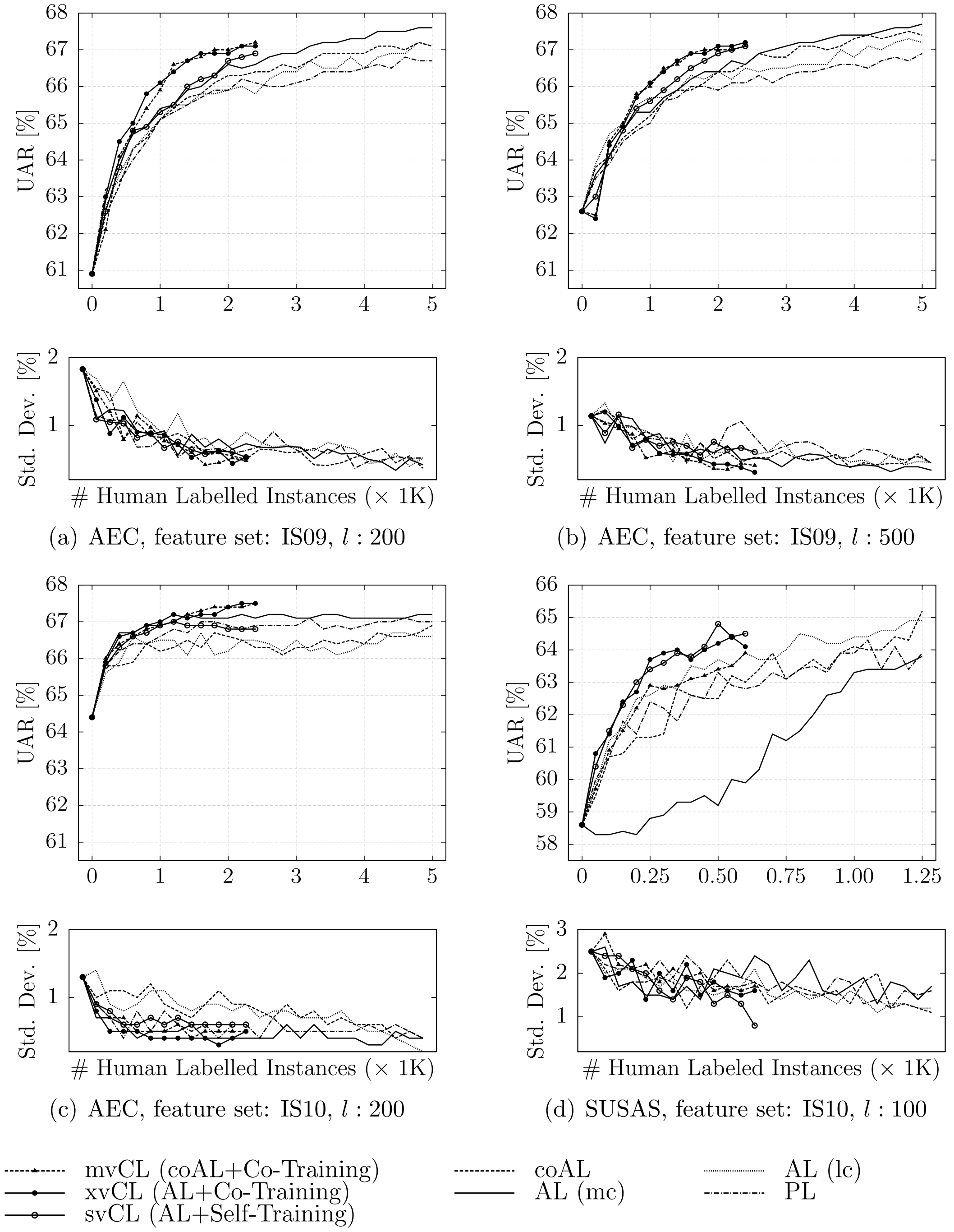
Avg. **UAR** [%]	(a) AEC IS09, *l*:200	(b) AEC IS09, *l*:500	(c) AEC IS10, *l*:200	(d) SUSAS IS10, *l*:100
PL	$65.7_{\pm 0.8}$	$66.8_{\pm 0.6}$	$65.6_{\pm 0.7}$	$62.4_{\pm 2.0}$
AL	$66.1_{\pm 0.6}$	$67.0_{\pm 0.5}$	$66.0_{\pm 0.8}$	$63.2_{\pm 1.7}$
coAL	$65.9_{\pm 0.6}$	$66.4_{\pm 0.9}$	$65.7_{\pm 0.7}$	$62.3_{\pm 1.8}$
svCL	$66.4_{\pm 0.7}$	$66.9_{\pm 0.6}$	$66.1_{\pm 0.8}$	$\mathbf{63.9}_{\pm 1.5}$
xvCL	$\mathbf{66.7}_{\pm 0.5}$	$\mathbf{67.2}_{\pm 0.4}$	$\mathbf{66.7}_{\pm 0.8}$	$\mathbf{63.9}_{\pm 1.7}$
mvCL	$\mathbf{66.7}_{\pm 0.5}$	$\mathbf{67.2}_{\pm 0.5}$	$66.6_{\pm 0.8}$	$63.1_{\pm 1.9}$

Experiments and Results

All the experimental setups in Section 4.1.6 and 4.1.7 were maintained in this section for the sake of comparison. In particular, given that more unlabelled data are necessary for machine-supervised learning than for human-supervised learning, at each learning iteration, 200 instances were selected for labelling for AL and coAL algorithms, and 500 instances for self-training and co-training on the database of AEC. For SUSAS, because of the smaller size of this database, fewer instances are selected in each learning iteration: 50 (AL and coAL) and 125 (self-training and co-training).

In these approaches, only a maximum of 2 400 and 600 human-labelled instances could be considered for the AEC and the SUSAS databases, respectively. This decision is due to the fact that both AL and SSL algorithms independently select instances from the unlabelled data pool for human and machine (respectively) labelling at each learning iteration. Therefore, the comparisons with the previous models in Section 4.1.6 and 4.1.7 are only made for a maximum of 12 iterations of the learning algorithm (when the maximum number of human-labelled instances is achieved). Given the inconclusive results obtained from Section 4.1.7 regarding to the query strategy, the AL algorithms used in the CL approaches make use of the medium certainty query strategy for the experiments with the AEC database and least certainty query strategy for those with the SUSAS database. The results are shown in Figure 4.11.

As depicted in Figure 4.11, the three CL methods perform globally better than all other algorithms for different numbers of initial training instances, databases



(a) AEC, feature set: IS09, $l: 200$

(b) AEC, feature set: IS09, $l: 500$

(c) AEC, feature set: IS10, $l: 200$

(d) SUSAS, feature set: IS10, $l: 100$

Figure 4.11: Comparison between the supervised (PL, least certainty AL, medium certainty AL, and coAL) and cooperative (AL+self-training, AL+co-training, and coAL+co-training) learning. The performance measures shown are mean and standard deviations of UAR averaged across 20 independent runs of each algorithm vs. the number of *manually* labelled instances for the AEC with IS09 EC feature set by 200 (a) or 500 (b) initial supervised training instances, as well as with the IS10 ASC feature set by 200 (c) initial supervised training instances, and the SUSAS with the IS10 ASC feature set by 100 (d) initial training instances. [231]

Table 4.9: Significance levels obtained from the statistical comparison (Student's t-test) of the UAR performance measures between iterations 4 and 12 (800 $\sim$ 2 400 instances for AEC, and 200 $\sim$ 600 instances for SUSAS). Values are shown for Passive Learning (PL), Active Learning (AL), co-Active Learning (coAL), and single-/mixed-/multi-view Cooperative Learning (svCL/xvCL/mvCL) for the four experimental conditions. [231]

Sign. levels	PL	AL	coAL	svCL	xvCL	mvCL	PL	AL	coAL	svCL	xvCL	mvCL
	(a) AEC, IS09, l : 200						(b) AEC, IS09, l : 500					
PL												
AL	█						█					
coAL	█	╲					░	╲				
svCL	▒	╲	█				▒	░	▒			
xvCL	█	█	█	█			█	░	▒	░		
mvCL	█	▒	█	▒	╲		█	▒	▒	░	╲	
	(c) AEC, IS10, l : 200						(d) SUSAS, IS10, l : 100					
PL												
AL	█						█					
coAL	╲	╲					╲	╲				
svCL	╲	╲	█				█	█	█			
xvCL	█	▒	█	█			█	█	█	╲		
mvCL	█	░	█	▒	╲		█	╲	█	╲	╲	

╲ $p > .05$ ░ $p < .05$ ▒ $p < .01$ █ $p < .001$

and feature sets. The improvement is evident in all experiments just after a few iterations of the learning algorithms, the only exception being the experiment with the AEC and the IS10 feature set where the improvement is clearer at the end of the learning process. Moreover, the standard deviation of UAR exhibits a descending trend, which indicates that increasingly adding more human-labelled instances to the training set makes the system more stable. In relation to the global performance improvement and human effort minimisation, the best UARs obtained with CL algorithms in the four experimental scenarios (67.2 %, 67.2 %, 67.6 %, 64.9 %) are very close to the baseline performance of the models trained on the whole pool of labelled data (67.7 %, 67.7 %, 67.2 %, 64.6 %). Nevertheless, CL uses about 75 % fewer labelled instances in all scenarios and is, therefore, less expensive.

In order to analyse in more detail the performance of the various algorithms, Table 4.8 presents the average UAR across iterations 4 and 12, and Table 4.9 computes Student's t-tests statistically to compare the performances of the various algorithms. The analysis of both tables confirms our previous observations and clearly indicates that all three CL approaches (single-, mixed-, and multi-view) generally lead to significantly better performance than all other approaches. This observation is particularly evident for xvCL (AL and co-training), the algorithm that led to the best performance in all four experiments by consistently and robustly outperforming the other methods. This conclusion is consistent with the best performance of co-training over self-training as demonstrated in Section 4.1.6.

4.1.9 Summary

SER, one of the most important computational paralinguistic tasks, was chosen to verify the effectiveness of a variety of data enrichment and optimisation algorithms. The main objective of these algorithms is to exploit the value of available data (either labelled or unlabelled), so as to maximise the performance of pre-existing modellings with less human effort.

Sections 4.1.1 and 4.1.2 introduced the data learning by aggregating human speech data or synthesised speech data in a cross-corpus evaluation scenario. The results in Section 4.1.1 show that the systems based on pooled data considerably surpass the performance of the ones based on single training corpora. Concerning the majority voting of individually trained learners, as opposed to the data pooling in a single classifier, the results largely depend on the classifier architecture (e.g., SVMs or RFs). The results in Section 4.1.2 illustrate that combining human and synthesised speech increases the expected performance while decreasing the performance variability caused by training with 'matching' or 'mismatching' human speech databases. Furthermore, in many cases, training on synthesised speech alone has been shown to be at least competitive with training on disjoint human speech databases. However, these trends are not observed for cross-corpus valence classification. This fact shows the difficulty not only of building generalising models for acoustic valence classification, but also the difficulty of synthesising speech that matches the human perception of positive/negative valence.

In contrast to Sections 4.1.1 and 4.1.2, in which the algorithms agglomerated data as much as possible, Sections 4.1.3 and 4.1.4 attempted to pick the most useful data via certain algorithms. Section 4.1.3 evaluated a distance-based data selection over eight emotional databases. Its effectiveness is shown by computing the prototypical data in cross-corpus arousal recognition, even though such a conclusion is not observed in cross-corpus valence recognition. Section 4.1.4 verified another data selection method, namely agreement- and sparseness-based instance selection. An obvious improvement of performance is observed by balancing the instance distribution among classes through random sub-sampling. Furthermore, in conjunction with data balancing, discarding the instances with low agreement levels bring further improvement.

The data collected by the above methods are labelled by experts, which are time- and money-consuming. Alternative data that could be collected for SER occurs to the ubiquitously available label-missed data, which can be easily acquired. Three possible ways can be used for annotation: by machine oracle, by human oracle, or by a combination thereof.

Section 4.1.5 investigated the SSL (self-training) in extensive studies of six different emotional databases in a various permutation of databases. It is interesting to see that, adding unlabelled data significantly exceeds not adding at the common .05 level (one-sided z-test) for valence on average. An enhanced SSL technique, namely co-training, divides the whole feature sets into two conditionally independent parts ('view'), which then learn each other. Extensive experiments on two emotional databases (i.e., AEC and SUSAS) in four experimental scenarios suggest that the proposed co-training with an LLD feature separation is more robust than the other two approaches: self-training and co-training with random feature separation. However, it is also observed that the performance of co-training degrades after a certain number of learning iterations. This phenomenon is possibly due to the exchange of mislabelled instances between different 'views'.

Because of the drawbacks of SSL such as noise accumulation and prediction tendency, a human oracle could provide relatively reliable and trustworthy annotations. In comparison with the passive learning, the confidence-value-based AL with the medium or least certainty query strategies generally perform significantly better (Section 4.1.7). When achieving a similar performance to that of the baselines (trained with the full set of training data), the models only use approximately half of the data, which resulted in a high human-work reduction. However, it still needs medium-level workload from a human.

For leveraging the merits and restraining the drawbacks of both SSL and AL, CL was estimated in Section 4.1.8. The underlying idea is to efficiently share the labelling work between humans and machines efficiently in such a way that instances predicted with insufficient CVs are subject to human labelling, and those with high CVs are subject to machine labelling. In particular, three approaches were considered: 1) single-view CL – combination of AL and self-training; 2) mixed-view CL – combination of AL and co-training; and 3) multi-view CL – combination of coAL and co-training. Furthermore, I evaluated the use of a medium certainty query strategy for instances selection in AL. Various test runs were conducted on two well-defined SER tasks (i.e., AEC and SUSAS) with a variable number of initial supervised training instances. The results show that all three suggested CL algorithms are superior to all other approaches when using the same number of human-labelled instances for retraining. The results also show that not only is the accuracy of the classifier improved, but its stability is enhanced. Furthermore, by varying the amount of instances used in the initial supervised training phase, using different feature sets, and testing different classification tasks, it demonstrates that the CL is a robust method. In particular, the best performance and robustness were obtained with the mixed-view CL algorithm. In relation to the type of query strategy used for instance selection in AL, the results indicate that medium certainty may be a feasible way to improve the classification performance of pre-trained models. Its robustness with different initial training set sizes and features sets using the AEC is observed. Nevertheless, the lowest certainty query strategy leads to better results with the SUSAS database and, thus, the results are not conclusive in this respect.

As data scarcity is considered to be the frontier challenge for computational paralinguistics, all these conclusions displayed above are significantly important for computational paralinguistics, or rather SER, in real-life applications. As a matter of fact, all these approaches can not only work independently but also jointly. However, there are still numerous points that could be improved or extended. A lesson learnt from Section 4.1.2 tells us that using multi- or cross-lingual speech synthesis methods could benefit cross-lingual emotion recognition, and developing synthesis methods to simulate different target groups of computer users, from children to elderly, or even pathologic voices. If a meaningful synthesis of such voices can be established, it would be a major step forward in the generalisation of SER to target groups that are nowadays overlooked by the lion's share of research in the (certainly justified) quest for stable results in 'controlled' evaluation scenarios involving healthy adult speech.

In addition, with the conclusions from Section 4.1.5 to 4.1.8, one can further improve the modelling robustness by considerably large databases that are not coming from predefined speech databases per se, but stemming from the richly available resources, such as Youtube online audio, and recordings of everyday-life conversations, among others.

Further promising directions can also be found in novel learning algorithms. On the one hand, for the subjective tasks, ground truth annotation differs severely among human raters. The most labelling uncertainty instances indeed deteriorate the modelling performance. On the other hand, the sparsest class often does not have enough information to yield robust model parameters. The conclusions in Section 4.1.4 imply that this information regarding annotation agreement, as well as class-sparseness, can in turn be used as queries for advanced learning. Tentative experiments in [234, 235] have demonstrated their value.

4.2 General Paralinguistic Tasks

Rather than emotion, this section proceeds extensive experiments on more general paralinguistic tasks, such as sleepiness and gender. Here, co-training is re-analysed because it demonstrated a remarkable performance without any manual annotations in Section 4.1.6. To support the data collection and continuous model updating, Section 3.3.2 proposed a distributed structure that increase the potential of computational paralinguistics to escape from the current 'lab-based' state to 'realistic' applications. One of the key issues – feature compression – is analysed elaborately here.

4.2.1 Co-Training

For the experiments, three common representative tasks to span the time continuum were selected, which were officially studied in INTERSPEECH Challenges from 2009-2011: short-term-related *emotion* [63], mid-term-related *sleepiness* [65], and long-term-related *gender* [64] of speakers. The corresponding official databases are the AEC, the Sleepy Language Corpus (SLC), and the Agender database. The main tasks of the three corpora cover different time-relations of paralinguistic groups from the short-term state of emotion, over the medium-term phenomena of sleepiness, to the long-term trait of gender. Speaker-independent partitioning of instances is shown in Table A.6(a). For more details of these databases, please refer to Appendices A.1.1 and A.2.

In order to keep in line with the INTERSPEECH Challenge 2009–2011 conditions, I employ the same feature sets per task in these experiments as those in the respective original Challenge. Thus, for SER, 384 features by brute-forcing based on 31 LLDs and 42 functionals are implemented; for sleepiness detection, 4 368 features comprising 59 LLDs and 39 functionals are used, and for gender classification, 450 features composed by 38 LLDs and 21 functionals. For more details of the LLDs and functionals, please refer to [63, 65, 64].

Concerning the two assumptions of co-training (i.e., compatibility and independence) as stated in Section 3.2.2.2, I firstly split the whole LLDs into three partitions: energy-related, spectral, and cepstral. Taking the (largest) feature set for sleepiness recognition as an example, Table 4.10 depicts this feature splitting. In comparison to the LLD groups shown in [65], the feature separation described in Table 4.10 differs – the one chosen here proved more suitable for the assumption of independence. For the other two tasks of emotion and gender recognition, the feature separation rule is the same. After divided into three partitions, the features are rather unbalanced across partitions. To solve this problem and satisfy the first assumption of sufficiency for each view, I rearrange the three groups into two views. That is, the two groups including fewer features are agglomerated as one view. In Table 4.10, the symbols of $\dagger, *, \sharp$ mark the feature group of which view-1 comprises for emotion, sleepiness, and gender recognition, respectively (remember that, the feature sets differ from task to task). Thus, the remaining two feature groups form view-2. Eventually, attribution ratios of view-1/view-2 are obtained as 288/96, 2 294/2 074, and 240/210 for the three tasks, respectively.

In addition, the classifier of SVMs with linear kernel and its parameters (complexity constant optimised on development data of 0.05, 0.02, 0.1 for emotion, sleepiness, and gender recognition) were also kept in line with the INTERSPEECH Challenge 2009–2011.

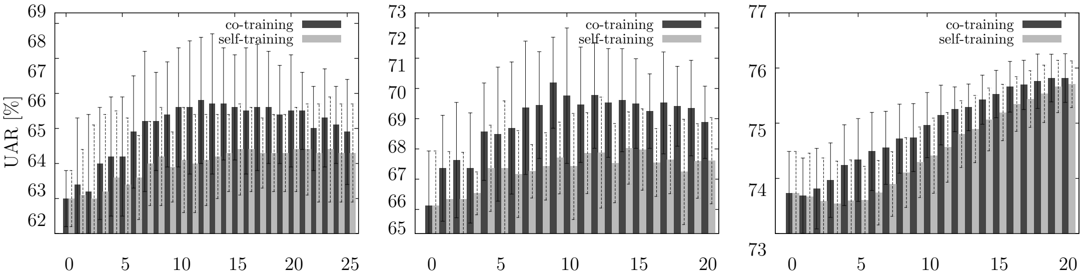
Table 4.10: Feature separation based on LLDs. The symbols $\dagger, *, \sharp$ indicate the feature group on which view-1 of co-training bases for emotion, sleepiness, and gender recognition, respectively. [232]

Group	Features in Group
Energy-related*	Sum of energy in auditory bands (loudness) Sum of RASTA-style filtered auditory spectral band energies RMS Energy RASTA-style filtered auditory spectral bands 1–26 (0–8 kHz) Spectral energy 25–650 Hz, 1 k–4 kHz
Spectral	Zero-Crossing Rate Spectral Roll Off Point 0.25, 0.50, 0.75, 0.90 Spectral Flux, Entropy, Variance, Skewness, Kurtosis, Slope F0, Probability of voicing Jitter (local, delta), Shimmer (local)
Cepstral$^{\dagger,\sharp}$	MFCC 1–12

Table 4.11: Experimental set-ups for AEC, SLC, and Agender. R: round number of whole processing; n_0: number of initial human-labelled training instances; N_1+N_2: number of instances selected by view-1 and view-2 per iteration; I: iteration times per round.

#	R	n_0	N_1+N_2	I
AEC	5	500	100+100	25
SLC	5	500	100+100	20
Agender	5	4 000	100+100	20

I randomly select 500 instances as initial human-labelled training set from AEC and SLC, and 4 000 instances from Agender due to its larger size, which resembles approximately 3 %, 6 %, 8 % of each database. At each new iteration, 100 instances are selected by each view of co-training. Thus, for the baseline experiment of single-view self-training, 200 instances are chosen per iteration to provide a fair comparison. Finally, 25, 20, and 20 times of SSL iterations for AEC, SLC, and Agender are executed per learning round. Furthermore, to reduce the influence of 'lucky' or 'unlucky' selection for the initial training set, I repeat five times with different random generator initialisations ('seeds'), leading to five learning rounds executed. In addition, to deal with class imbalance, instance upsampling is used per iteration for emotion and sleepiness recognition. Details of the three experimental set-ups are given in Table 4.11.



(a) Emotion: AEC (b) Sleepiness: SLC (c) Gender: Agender

Figure 4.12: UAR vs. number of iterations. Comparison between single-view semi-supervised learning and co-training in five independent rounds for three paralinguistic corpora – AEC, SLC, and Agender.

Experimental Results

The chance level of UAR is 50.0 % for the binary emotion and sleepiness classification, and 33.3 % for the three-class gender classification.

Figure 4.12 displays a comparison of average performance and standard deviations between co-training (dark grey histograms with solid error lines) and single-view self-training (light grey histograms with dotted error lines) in five independent rounds for the three experiments based on the AEC, SLC, and Agender databases.

For the SER based on AEC, as seen in Figure 4.12 (a), the best mean UAR obtained by co-training with two-view learning based on feature partition in five independent rounds is 64.8 % UAR at the 12th iteration (24K instances combined by co-training). This value boosts the initial mean UAR of 62.0 % without any SSL iteration at the .001 significance level in a one-side z-test, and even greatly higher than the best mean UAR of 63.4 % achieved by single-view self-training at the 15th iteration at the .05 significance level (see Table 4.12). This improvement means that, the two-view SSL of co-training incorporates more additional information than single-view self-training. Further, one can also notice that the performance degrades quicker than in single-view self-training after the highest UAR gain. This phenomenon can probably be attributed to falsely labelled data by both views when the instances with increasingly lower confidence score are selected as iteration goes on, leading to doubling or accelerating error numbers added per iteration.

Figure 4.12 (b) depicts the UAR for recognition of sleepiness based on SLC. The gain obtained by co-training is also notable with a boost in mean UAR of almost 4.1 %, and 2.2 % absolute in comparison to the initial results (UAR of 65.1 %) and the best mean UAR achieved by single-view self-training (UAR of 67.0 %), respectively. Both improvements are significant at the .001 and .05 level (one-side z-test, see Table 4.12).

Table 4.12: Classification evaluation comparison of co-training and single-view self-training in five independent rounds for three corpora of AEC, SLC, Agender. Initial: initial supervised learning result; delta: absolute improvement of co-training over single-view self-training. [232]

Mean of UAR[%]	**initial**	**self-training**	**co-training**	**delta**
AEC	62.0	63.4 ●○	**64.8** ●●	1.4 ●○
SLC	65.1	67.0 ○○	**69.2** ●●	2.2 ●○
Agender	73.7	75.7 ●●	**75.8** ●●	0.1 ○○

Significance levels [236]: ○○ not significant ●○ 0.05 ●● 0.001

Table 4.13: Features used for five paralinguistic tasks.

# Features	**Emotion**	**Intoxication**	**Pathology**	**Age/Gender**
LLDs	16	59	64	29
Functionals	12	39	61	8
Total	384	4 368	6 125	450

Finally, Figure 4.12 (c) shows the performance for recognition of gender based on the Agender database. It can be seen that, both co-training and single-view self-training significantly increase the initial UAR from 73.7 % to 75.8 %, and 75.7 %, at the significance level of .001 in a one-side z-test (see Table 4.12).

Overall, in terms of UAR for emotion, sleepiness, and gender recognition, the gain achieved by co-training based on feature multi-view is highly significant in comparison with the initial results of supervised learning for all three tasks. This conclusion holds even when compared to the baseline SSL approach. Details of performance improvement are given in Table 4.12.

4.2.2 Feature Compression

As discussed in Section 3.3.1, SVQ is selected for feature compression for distributed computational paralinguistics systems. To access the feasibility of this method, I proceeded the experiments recurring to five well-defined paralinguistic tasks (i.e., emotion, intoxication, pathology, age and gender) in correspondence with four databases (i.e., AEC, ALC, NCSC, and Agender) that are well-defined in the INTERSPEECH Challenges 2009–2012. For the details of these databases, please refer to Appendix A.2.

For the experiments, both acoustic feature sets and classifiers (i.e., SVMs) are kept in line with the INTERSPEECH 2009–2012 EC [63], Paralinguistic Challenge (PC) [64], Speaker State Challenge (SSC) [65], and Speaker Trait Challenge (STC) [66]. I also followed the classifier setups of the five Sub-Challenges: the complexity constants optimised on the development or cross-validation of the training data with values 0.05, 0.01, 0.001, 0.05, and 0.05 for emotion, intoxication, pathology, age, and gender, respectively. Furthermore, as in the challenge, to alleviate the influence of instance imbalance, I implemented instance upsampling before any learning process that produces a random subsample of the data set belonging to sparse category with-/out replacement.

In the experiments on ALC and NCSC tasks, the training and development sets are combined for training, and the test sets are used for testing. In relation to the Agender task, the development set is used for testing given that there is no test set. The UAR baseline for the binary classification on the emotion, intoxication, and pathology classification tasks are 67.6 %, 66.0 %, 69.0 %, respectively. The baseline for the three-class gender classification is 76.0 %, and the baseline for the four-class age classification is 45.7 %. It should be pointed out that the baselines obtained in this section are different from those reported in the 2009–2012 INTERSPEECH due to the use of a different Weka version.

For the sake of simplicity, in each paralinguistic task I split the whole feature vector (d dimensions) into multiple subvectors with the same dimensionality k (note that the last subvector dimensionality may be smaller than k and equal to d mod k). I also adopted the same codebook size N ($N = 2^L$, where L is codevector length) for all subvectors. In this case, the transmission bandwidth B_w for such compressed features is

$$B_w = (\lceil \frac{d}{k} \rceil \cdot L)/T, \tag{4.1}$$

where T is the average length of chunks. Hence, its corresponding feature compression rate R for a transmission bandwidth requirement $B_{w/o}$ (no feature compression) is calculated by the equation

$$R = \frac{B_{w/o}}{B_w} = \frac{(32 \cdot d)/T}{(\lceil \frac{d}{k} \rceil \cdot L)/T} \cong 32 \cdot \frac{k}{L}, \tag{4.2}$$

assuming a single-precision floating point for the transmission of uncompressed data (32 bits). Obviously, the feature compression rate R is in direct proportion to the subvector dimension k and in inverse proportion to the codevector length L.

Influence of Attributes Independence

As discussed in Section 3.3.1, the central issue of SVQ is splitting the whole feature set into multiple subvectors in an effective way. The most important factor is arguably the cross correlation of attributes in the feature domain. A simple method to deal with this issue is to adopt a splitting strategy based on the types of LLDs,

Table 4.14: Performance comparison for five paralinguistic tasks using two types of vector splitting strategies: LLD-based (D) and random (R). BL: baseline; k: dimension of subvector; N: codebook size for each subvector.

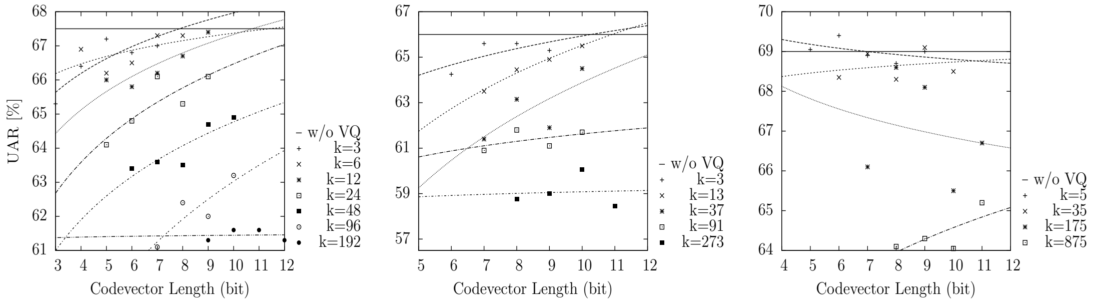
UAR [%]	**BL**	$k =$	$N = 128$ D	$N = 128$ R	$N = 256$ D	$N = 256$ R	$N = 512$ D	$N = 512$ R
Emotion	**67.6**	12	66.1	65.3	66.7	65.8	67.4	66.8
Intoxication	**66.0**	37	61.4	59.7	63.2	60.4	61.9	60.7
Pathology	**69.0**	35	69.0	66.6	68.3	66.8	69.1	66.2
Age	**45.7**	8	44.5	43.6	44.6	43.8	44.9	44.0
Gender	**76.0**	8	75.0	73.9	75.3	74.1	75.2	74.2

that is, the statistical features belonging to the same LLD are grouped into one subvector. In order to test this method, I compared the performance of this strategy with the performance achieved using a random clustering of the features on the five paralinguistic tasks. The dimensionality of all subvectors was set to the same value in each task – $k = 12, 37, 35, 8, 8$ for emotion, intoxication, pathology, age, and gender recognition, respectively. Given that the number of functionals over each LLD within each task may be different, I defined the dimensionality of the subvectors for each task as the maximum number of functionals over all LLDs. Table 4.14 shows the results obtained for the various tasks.

As it can be seen in Table 4.14, the performance achieved through LLD-based vector splitting strategy is always better than the strategy that used a random splitting strategy. This improvement is evident for all codebook sizes and across all tasks, and lies in the range of $1 \sim 3\,\%$ (absolute UAR). Results also show that the improvement delivered by the LLD-based splitting strategy over the random one is more noticeable for the tasks with larger features spaces, i.e., Intoxication (absolute average improvement of 1.9 %) and Pathology (absolute average improvement of 2.3 %). The absolute average improvement on the Emotion, Age and Gender tasks is less pronounced: 0.8 %, 0.8 % and 1.1 %, respectively.

Distributed Paralinguistic Tasks Classification

In the context of distributed speech recognition, the feature set typically comprises 14 features, and adjacent features are grouped into pairs [18]. This feature grouping method is quite different from the one for distributed paralinguistic tasks recognition, since the feature spaces are much larger (cf. Table 4.14). Therefore, grouping features into pairs would lead to a very large number of subvectors and low compression rates, which is not ideal given the bandwidth limitations. In order to investigate the influence of the dimensionality of the subvectors as well as the codebook sizes



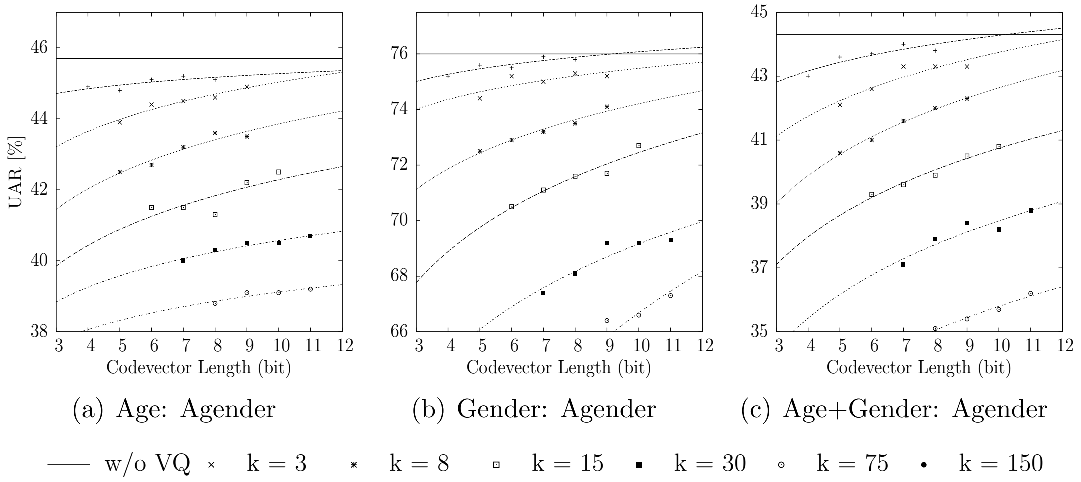
(a) Emotion: AEC (b) Intoxication: ALC (c) Pathology: NCSC

Figure 4.13: Performance for distributed *short-term* (*emotion*, AEC) and *medium-term* (*intoxication*, ALC; *pathology*, NCSC) paralinguistic tasks. k: subvector size. [8]

and their impact on the recognition performance, I considered several permutations of these two parameters for each task. Given the results presented in the previous section, I adopted an LLD-based splitting strategy, and so, each subvector is quantised using the same codevector length and their own codebook.

Figure 4.13 to 4.14 depict the classifier performance for the short-term, medium-term and long-term recognition tasks for various codevector lengths (the length of each codevector is $L = log_2 N$, where N is the codebook size) and subvector sizes (k; increasing values of k indicate higher compressions rates). The horizontal lines in each figure indicate the baseline performance for each task. As expected, for increasing codevector lengths (i.e., smaller quantisation error) and lower subvector dimensionalities (i.e., lower compression rates) the recognition performance is improved for all tasks, except some cases of the 'Pathology' task ($k = 5$ and $k = 175$; discussed below). Taking the 'Emotion' task as a representative example (see Figure 4.13(a)), we can observe that for $k = 24$ the UAR varies between (approximately) 62.6 % ($L = 3$) to 67.0 % ($L = 12$), a value very close to the baseline (67.6 %). If we increase the subvector dimensionality (e.g., $k = 48$), the performance varies between 61.0 % ($L = 3$) to (approximately) 65.3 % ($L = 12$), which is further away from the baseline. Naturally, with a higher value of k a smaller bandwidth is required. In the example given, for a codevector of length 12, the bandwidth would be $(384/24) * 12/1.7 \approx 113b/s$ ($k = 24$) and $(384/48) * 12/1.7 \approx 57b/s$ ($k = 48$). Compared to the no compression case the bandwidth reduction would be of 98.4 % and 99.2 %, respectively.

As mentioned above, the Pathology task does not follow the same pattern and shows a more complex relationship between the code vector length and subvector dimensionality. As it can be seen in Figure 4.13(c) for different values ok k the performance either decreases ($k = 5$ and $k = 175$) or increases ($k = 35$ and $k = 875$) for increasing code vector lengths. In my view, this phenomenon might be caused by



(a) Age: Agender (b) Gender: Agender (c) Age+Gender: Agender

Figure 4.14: Performance for distributed *long-term* paralinguistic tasks: *age*, *gender*, and *age* + *gender* classification on Agender. k: dimension of subvector. [8]

data scarcity. As it can be observed in Table A.6(a), there are only 2 386 instances in total for this task, which is potentially an insufficient number of instances to train a robust SVQ model and/or recogniser. This conclusion seems to be corroborated by the results of the 'Agender' task, where we have 53 074 instances, and the stability and reliability of the system is much higher (and also the fact that age and gender recognition tasks have a more solid ground truth). Despite this unexpected result, and as it will be shown in the next section, the relationship between recognition performance, feature compression rate, and bandwidth follows a pattern that is similar to that of other tasks (see Figures 4.15 and 4.16).

It is also noticeable that in this task compressing the feature set to a certain degree increased the performance of the model over the baseline – in the case of $k = 5/L = 6$ the UAR is 69.4 %, and in the case of $k = 35/L = 9$ the UAR is 69.1 %. This phenomenon may indicate that, to a certain degree (both values are actually not significantly higher than the baseline) the compression process attributed more weight to relevant features and reduced the impact of less relevant information.

Performance, Feature Compression Rate, and Bandwidth

Figure 4.15 provides a combined overview of the relationship between performance and feature compression rate for the five distributed paralinguistic classification presented in this section. The distributed gender recognition seems to be the most sensitive to the feature compression rate as it can be inferred from the slope of the trend line (dash-dot). It can also be observed that, for most tasks (i.e., four of five tasks, except the case of "age"), the performance degradation is not significant (one-side z-test, $p > .05$) when the feature compression rate is less than 40, but it

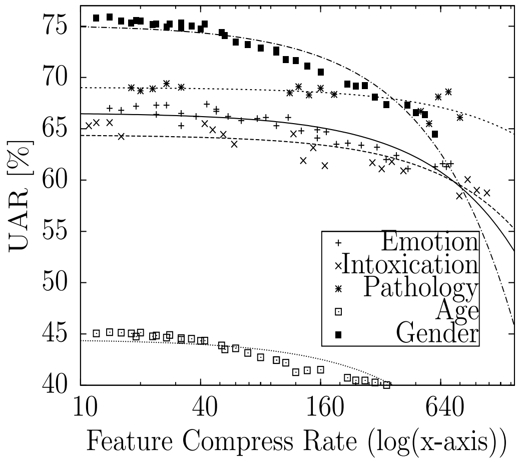


Figure 4.15: Relationship between recognition performance (UAR) and *feature compression rate* for the various tasks with manifold permutations of codevector length and subvector dimensionality. [8]

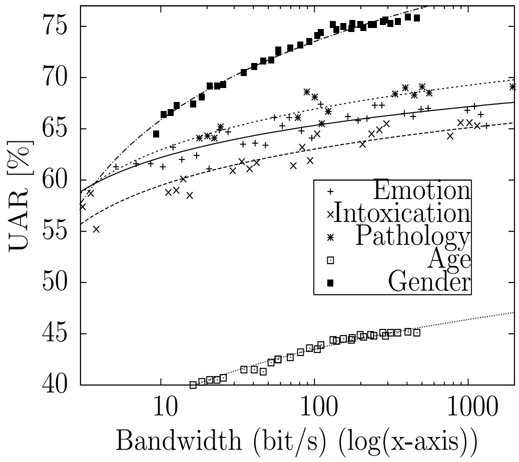


Figure 4.16: Relationship between recognition performance (UAR) and *bandwidth requirements* for the various tasks with manifold permutations of codevector length and subvector dimensionality. [8]

is increasingly pronounced for values over 40, and especially over 160. This is an interesting result given that in a multi-task scenario where the best permutation of the subvector dimension and codebook sizes for a given task may be unknown and varied, guaranteeing a compression rate below 40 warrants a good performance for all tasks.

Given that a crucial aspect of a distributed recognition system is a trade-off between recognition accuracy and bandwidth limitations, Figure 4.16 shows the relationship between recognition performance and required transmission bandwidth for all five tasks (the transmission bandwidth is calculated by the Equation (4.1)). This figure can be used to obtain an estimation of the recognition task accuracy for a particular transmission bandwidth, and *vice versa*. As expected, the performance decreases for lower transmission bandwidth rates, and is particularly degraded for rates below 100 bit/s. For instance, considering the 'Gender' classification task, if a transmission bandwidth of 10 bit/s is available the recognition accuracy would be of about 65.0 %. If a better performance is required, for instance 75.0 %, then a transmission bandwidth of more than 100 bit/s would be necessary.

4.2.3 Summary

The suitability of co-training was evaluated by investigating three representative cases of personal affect, speaker state, and speaker trait recognition. The results indicate that adding unlabelled data with co-training can significantly enhance the performance of initial supervised learning – here by 2.8 %, 4.1 %, and 2.1 % UAR absolutely for emotion, sleepiness, and gender classification, respectively –, and even impressively improve upon the performance of commonly used single-view self-training for the former two cases with a mean UAR of 1.4 % and 2.2 % absolutely (one-side z-test, $p < .05$), respectively. This renders co-training beneficial in real-world applications of computational paralinguistics, in which labelled data are scanty, but unlabelled data is sufficient.

Moreover, to support the process of data collection and continuous modelling updating, Section 4.2.2 further investigated a general distributed architecture, which holds manifold advantages compared to embedded or network-based structures. Because of the security, transmission bandwidth, and storage capacity requirements, the statistic feature set was chosen for transmission. However, these features could be further optimised by means of feature compression when considering a large number of users/devices. To this end, this section also focused on the evaluation of SVQ due to its efficiency, security, and the fact that it is the official compression method recommended by ETSI for distributed speech recognition. Various experiments were conducted to investigate the feasibility of the system on large-scale paralinguistic tasks, including short-term states, medium-term phenomena, and long-term traits.

The results report that there is a strong influence of feature attributes on the performance of the compression algorithm. Compared to a random clustering strategy, grouping the feature attributes under same LLDs reduces the information loss when implementing compression of the feature set using SVQ. It has also shown that subvector and codebook size have a critical impact on the system's performance – the classification performance degrades for almost all tasks when either large subvector or small codebook size (or both) is used. Overall, the results demonstrated that when the feature compression rate is less than 40, the classification performance is similar to that with no compression.

These results are very informative and encouraging for future exploitation of the system proposed in this thesis. Nonetheless, this work is only a first step towards the creation of large-scale distributed paralinguistic information analysis systems for applications in real-life contexts, and several issues still need to be addressed. A central issue is the optimisation of various modules. In this section, I focused on demonstrating the feasibility of the whole system, but there is plenty of room for improvements in the various modules. For instance, I used a common feature compression technique (SVQ), but given the demonstrated importance of the compression stage it would be very beneficial to explore other state-of-the-art feature compression techniques such as PCA [72], LDA [73], HQ [191], and sparse presentation [237]. Additionally, even though I used preselected features sets for each task, it is worth exploring the use of feature selection as it could improve compression rates and reduce the required bandwidth while maintaining or improving the recognition tasks performance. Furthermore, given that the dimensionality of statistical features vectors used in this thesis is always the same per turn, the transmission bandwidth will vary as a function of turn duration, which may lead to bandwidth bursts for consecutive short turns. A possible way of overcoming such a problem is to consider different methods for dealing with long and short turns so as to avoid its negative impact on the client-server communication. Another possible solution would be to evaluate the contribution of the features used for each classification task, and to vary the dimensionality and codebook size for the attributes with different levels of importance (in this thesis, all features were equally important to the classification tasks).

In addition to the optimisation issues, there are various important challenges particular to paralinguistic recognition systems that should be addressed in the future. A central one is to deal with multiple paralinguistic tasks simultaneously, which could be handled by the ways of task selection or multi-task learning. In relation to the first way, if the tasks are not selected manually on the client side, methods such as CASA could be used to analyse the characteristics of the acoustic environment and to infer the paralinguistic task(s) of interest. Concerning the second way, and given that it has been continuously demonstrated that paralinguistic tasks benefit from contextual knowledge (for example, gender, social background, and other information can improve SER tasks [238, 239]), it would be relevant to exploit the use of mutual information in multi-task learning scenarios to improve the performance for a particular task. Furthermore, given the overlaps between the feature sets used for different tasks, it is plausible to pursue a common set of features that can be applied to all tasks. In this case the distributed framework can be simplified since modules related to task selection are not necessary anymore. To this end, DNNs or sparse coding could be used to extract high-level feature representations that may be shared by the various paralinguistic tasks.

Another aspect that should be at the very centre of future developments is the enhancement of the system robustness, being relevant to, in particular, device disparity, environmental noise and reverberation, voice variation across users, security protection, and the impact of packet loss during data transmission. This point is crucial for the implementation and use of the system proposed in this thesis.

Despite the many issues still to be addressed in this area, this section has shown very promising results and demonstrated that we are not far from the creation of robust distributed multi-task paralinguistic recognition systems that can be applied to a myriad of everyday life scenarios, such as remote medical treatments, remote conferences or negotiation, remote education, or even advanced driver assistance systems. Also, and very importantly, as previously mentioned, this type of systems may also be crucial to the future of computational paralinguistics by providing the essential speech signals for the development of robust ISA systems.

4.3 Speech Recognition

4.3.1 BLSTM Model for Dereverberation

The ultimate user experience for human–machine interaction is the ability to communicate hands-free from a distance, typically a few meters. In this case, however, the distant-controlled speech can undergo significant distortion due to room reverberation, echo from a loudspeaker, and additive noise sources, which consequently leads to high WER of speech recognition. To eliminate the reverberation, this section aims to explore the advantage of Bidirectional Long Short-Term Memory Recurrent Neural Networks (BLSTM-RNNs) to learn the nonlinear feature mapping rule, and to contribute to (1) evaluating the BLSTM dereverberation approach by executing extensive experiments on realistic and synthesised reverberated speech, by comparing the approach with other traditional network structures, such as MLP and (B)RNN in order to exploit the potential value of memory networks; (2) evaluating the proposed *differential* feature vectors between the distant-talk (reverberant/distorted) speech and close-talk (clean) speech as training targets, which differs with the ones used in [211] where only the *absolute* feature vectors of close-talk speech were adopted as training targets; (3) comparing and integrating the feature enhancement methods with the widely used adaptation algorithms like MLLR [92] and CMLLR [93]; and (4) assessing the robustness of the techniques in the scenarios of mismatched recording environments between training and evaluation sets.

Experimental Setup

To demonstrate the effectiveness of the proposed methods, two databases – a *French* and an *English* corpus were recorded beforehand in a realistic acoustic space environment. Both databases are collected for speech controlled TV application. (For more details of the two corpora, please refer to Appendix A.3.) In the ongoing, the effectiveness of the mapping techniques is mainly evaluated on the French corpus. The English database is used to study the impact of mismatch in acoustic (room) environments between training and testing conditions.

The stereo training (close-talk and distant-talk) feature vectors are time aligned such that the Pearson product-moment correlation coefficient (PCC) is maximised between the MFCC-0 time series. The training utterances with maximum PCC coefficient lower than 0.9 were dropped to avoid utterances with severe channel distortions.

The mapping techniques were evaluated on the standard MFCCs. The 12-dimensional static MFCCs were appended to their first, second, and third order regression coefficients, resulting in a feature vector of size 48. The feature vectors of $\mathbf{x}_t^c$ and $\mathbf{x}_t^d$ were extracted from the close- and distant-talk signals, respectively, every 10 ms using a window size of 25 ms. Then, the differential feature vectors of $\mathbf{x}_t^\Delta$ were acquired by $\mathbf{x}_t^c - \mathbf{x}_t^d$. Furthermore, before training the neural networks I calculated the global means and variances of the close-talk, distant-talk, and their differential feature vectors over the whole neural network's training sets. Then, mean and variance normalisation were performed over the network inputs and targets (i.e., the absolute close-talk feature vectors or the differential feature vectors) using the means and variances from the corresponding sets, respectively.

For the neural networks, both input and output node numbers are equal to the dimension of the feature vector (48 in our case) except that stacked frames are used as input. Moreover, one hidden layer with 200 neurons was chosen. Particularly, for the LSTM memory block, input and output gates adopt hyperbolic tangent (tanh) activation functions, and the forget gates take logistic activation functions.

During network training, gradient descent is implemented with a learning rate of 10^{-5} and a momentum of 0.9. Zero mean Gaussian noise with standard deviation 0.1 is added to the input activations in the training phase in order to improve generalisation. All weights are randomly initialised in the range from -0.1 to 0.1. Finally, the early stopping strategy is used as no improvement of the MSE on the evaluation set has been observed during 20 epochs.

The acoustic models were trained on mobile data collected on hand-held devices. The performance of the ASR is measured and compared in terms of WER and its relative reduction (WERR) metrics (cf. Section 2.5), and the baselines for the close talk and distant talk of the French corpus are 11.8 % and 19.41 % WERs, respectively (see Table 4.15).

Neural Network Architectures

To verify the effectiveness of BLSTM networks for dereverberation, I compare it with MLP, and recurrent networks without memory (RNNs) or with memory (i.e., LSTM). The comparison results are displayed in Table 4.15. Note that, according to

Table 4.15: Performance of the baseline recogniser and dereverberant systems by adopting various neural network architectures like MLP, RNN, BRNN, LSTM, and BLSTM, with different number of hidden neurons and stacked feature frames. [212]

networks	**# neurons**	**# frames**	**# weights**	WER [%]	WERR [%]
w/o mapping (close talk)				11.81	
w/o mapping (distant talk)				19.41	
MLP	200	1	19K	24.15	-24.4
	200	5	58K	18.74	3.5
	200	7	77K	18.72	3.6
	200	9	96K	18.74	3.5
	600	7	230K	18.06	7.0
RNN	200	1	59K	19.07	1.8
BRNN	200	1	118K	17.42	10.3
LSTM	200	1	180K	18.47	4.8
BLSTM	200	1	360K	**16.38**	15.6
BLSTM	200	7	590K	16.43	15.4
BLSTM	144-200-144	1	1M	**16.32**	15.9

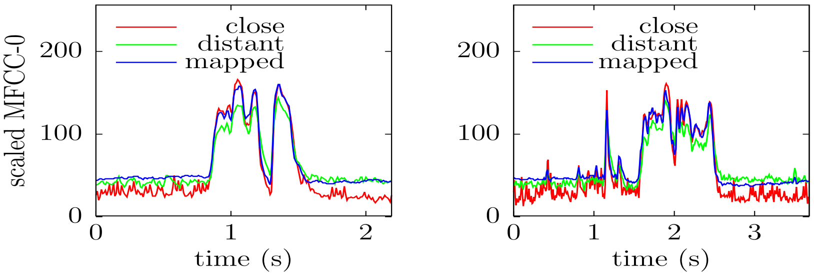
the empirical experience, the best performance for training MLP and (B)RNN was achieved by a learning rate of 10^{-6}, as opposed to 10^{-5} for the (B)LSTM networks.

It can be seen that, when no context is used at the input of the MLP, there is an increase in WER compared to the baseline. Whereas, the recurrent neural networks (standard RNN and more sophisticated LSTM) show lower WERs. This is because of their ability to capture the contextual information implicitly. When the temporal context is increased at the input of the MLP, there is a steady decrease in WERs and for 600 hidden nodes and a context of seven frames, we see a WERR of 7 % over the baseline system.

RNN and LSTM models capture only the past information. However, for dereverberation, it is important to learn the temporal smearing in the future frames because the distant-talk signals are delayed (future) and attenuated version of the close-talk signals (cf. Section 3.3.2). The bidirectional RNN and LSTM yield a significant (one-side z-test, $p < .001$) reduction in WERs compared to the corresponding uni-directional models capturing past information.

It can also be seen that both uni- and bi- directional LSTM models give lower WERs compared to the simple RNN models. This observation can be attributed to the sophisticated architecture of the individual neurons compared to the simple neuron. Previous acoustic information can be stored in the memory cell until the input gates and the forget gates allow (partly) changing it (cf. Section 2.4.3).

Moreover, as seven successive frames are simultaneously fed into BLSTM networks, no improvement is observed from this side (see Table 4.15). Hence, the



(a) Example 1

(b) Example 2

Figure 4.17: The scaled MFCC-0 (1∼256) of a close-talk utterance (red), a distant-talk one (green), and a mapped close-talk one (blue) for two examples. [212]

BLSTM seems to learn the context better if feature frames are presented one by one and the increased size of the input layer rather harms recognition performance. In addition, when increasing one hidden layer with 200 neurons to three hidden layers with 144-200-144 neurons, the performance improvement is not obvious. In the following experiments, I keep using one hidden layer with 200 neurons as the BLSTM network's architecture on the French corpus.

To visualise the mapping learned by the BLSTM model, I plot the trajectories of MFCC-0 for two randomly selected utterances in Figure 4.17. The figure shows three trajectories – close talk (red), distant talk (green), and mapped (or estimated) close talk (blue). It can be seen that the MFCC-0 curves of mapped close-talk speech (by BLSTM networks) are closer to the original one than the distant-talk speech during the speaking period, and are smoother during the silence period. This observation indicates that the reverberant signals and channel noise are successfully suppressed. Such a feature enhancement phenomenon can be further confirmed over the entire training set and the whole feature vectors. Figure 4.18 presents the PCCs of the 48 MFCCs between distant-talk utterances (hollow circle and dotted line)/mapped utterances (solid circle and line) and close-talk utterances over the whole training set. Obviously, the PCCs is boosted after reverberated features are enhanced, which could demonstrate the performance improvement of ASR by using a BLSTM dereverberation model.

Training on Differential Targets

As discussed in Section 3.3.2, there are two ways to obtain the enhanced features from distant talk, either by direct way (training networks with absolute targets) or by indirect way (training networks with differential targets). Table 4.16 compares the performance of the two mapping ways in ASR systems.

By checking three types of BLSTM network structure, the BLSTM dereverber-

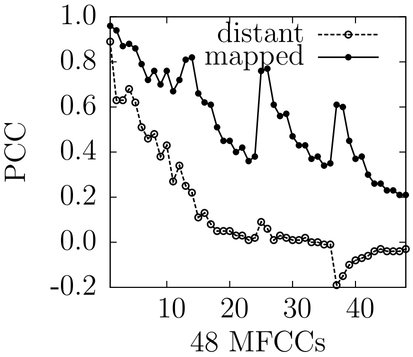


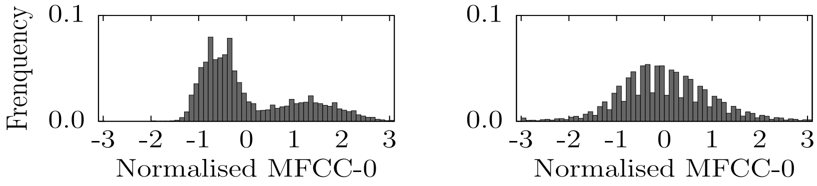
Figure 4.18: Pearson product-moment Correlation Coefficient (PCC) of 48 MFCCs between distant-talk utterances (hollow circle and dotted line)/mapped close-talk utterances (solid circle and line) and close-talk utterances over the whole training set. [212]

Table 4.16: Performance comparison by using *absolute* and *differential* targets.

methods	**# neurons**	WER [%]	WERR [%]
absolute	144	17.04	12.2
differential	144	16.52	14.9
absolute	200	16.38	15.6
differential	200	16.43	15.4
absolute	144-200-144	16.32	15.9
differential	144-200-144	16.29	16.1

ation models trained on differential targets perform better than the models trained on the absolute targets when the network structure is simpler. It can be seen that a gain of about 3 % relative WERR (at the .05 significance level in a one-side z-test) is achieved when only 144 neurons are used in only one hidden layer, compared to using absolute targets.

To find out the rationale behind this phenomenon, I plot the distribution of globally normalised log energies (MFCC-0) on the absolute targets (a) and the differential targets (b) over the whole French corpus in Figure 4.19. Obviously, the differential targets have a symmetrical unimodal distribution that is centred around zero. In contrast, the absolute target has a bimodal distribution that could be harder to learn. Therefore, the simpler the neural networks are, the higher a gain would be obtained via training on the differential targets. Such superiority of differential targets-based learning can further be verified in the following.



(a) targets: absolute

(b) targets: differential

Figure 4.19: Distribution of normalised log energy (MFCC-0) of absolute targets (a) and differential targets (b). [212]

Incorporating CMLLR and MLLR

As the distant-talk speech is passing through the BLSTM dereverberation models, its feature vectors are transformed (almost) to the clean target, on which most pre-existing acoustic models are trained. Thus, this technique could also be considered as a sort of feature adaptation. It is interesting to see whether incorporating back-end adaptation techniques like CMLLR and MLLR can further enhance the ASR performance. As expected, without the mapping technique the WERs for distant talk decrease from 19.41 %, over 19.01 %, to 17.19 % with no adaptation, CMLLR, and CMLL + MLLR, respectively (as shown in Table 4.17). The WERs drop further to 16.43 %, 16.34 %, and 15.68 % when integrating with the suggested mapping technique, which results in 15.4 %, 13.8 % and 7.8 % relative WERR, respectively (All improvements are at the .001 significance level in a one-side z-test). Overall, the best result is achieved by combining both mapping and adaptation (CMLLR + MLLR) techniques, with 8.8 % and 19.2 % performance improvement in WERR in comparison with adaptation techniques only and the baseline (w/o adaptation and mapping), respectively. Additionally, Table 4.17 also shows that if the close talk was falsely detected as distant talk and fed into the mapping and adaptation systems, the WER would increase about 10 % relatively.

Cross-Room Evaluation

In the above experiments, the data set used for training the dereverberation model is recorded in the same room with the evaluation set. In a real-life application, however, the evaluation scenarios are always unpredictable. That is, the acoustic environments (i.e., room size, type) for creating the training data normally mismatch with the evaluation scenarios. To cope with this issue, I firstly artificially generated several reverberant corpora on the close-talk set of French by convolving various RIRs and adding a little noise. The rooms to create the RIRs are different with the ones for creating the French corpus. When generating the simulated corpora, three elements were taken into account: positions variation of the speakers w.r.t

Table 4.17: ASR evaluation on distant-talk and close-talk sets by combining BLSTM dereverberation and adaptation (CMLLR and MLLR) techniques.

[%]	**targets**	**distant talk**		**close talk**	
adaptation		WER	WERR	WER	WERR
w/o adaptation					
w/o mapping		19.41		11.81	
w/ mapping	absolute	16.38	15.6	14.47	-22.5
w/ mapping	differential	16.43	15.4	14.02	-18.7
w/ CMLLR					
w/o mapping		19.01	2.0	11.78	-0.3
w/ mapping	absolute	16.14	16.8	13.70	-16.0
w/ mapping	differential	16.34	15.8	13.46	-13.9
w/ CMLLR+MLLR					
w/o mapping		17.19	11.4	11.63	-1.5
w/ mapping	absolute	15.70	19.1	13.33	-12.9
w/ mapping	differential	**15.68**	19.2	13.04	-10.4

Table 4.18: ASR evaluation on the *artificial* distant-talk set using the BLSTM dereverberation models trained on the *natural* distant-talk set. pos: position of speakers w.r.t microphones; R/N: reverberant/noisy signal weights (dB); w/o: without mapping.

[%]	**w/o**	***absolute***		***differential***	
	WER	WER	WERR	WER	WERR
pos-1,R:-100,N:-30	20.86	20.06	3.8	19.24	7.8
pos-1,R:-30,N:-100	21.28	20.59	3.2	19.97	6.2
pos-2,R:-100,N:-30	20.24	19.25	4.9	18.78	7.2
pos-2,R:-30,N:-100	19.89	19.70	1.0	18.71	5.9
Average	20.57	19.90	3.3	**19.18**	6.8

the microphones, the weights of the reverberation signals and the weights of the noise signals. The first column of Table 4.18 shows the four scenarios of simulated speech. The second to sixth columns represent the WER and WERR for each simulated corpus without mapping, mapping to the absolute targets, and mapping to the differential targets, respectively. As observed from the table, the BLSTM dereverberant ASR systems prevail over the systems without dereverberation, which overall leads to a reduction of WER with 3.3 % relatively by the usage of absolute targets and 6.6 % relatively by the usage of differential targets.

In addition, I repeated the experiments on a realistic English corpus, of which the training and test sets are recorded in totally different rooms (cf. Section A.3).

Table 4.19: ASR evaluation on the training and test sets of the *English* corpus by using the BLSTM (one hidden layer with 128 neurons) feature dereverberation model trained on the training set. CMN (utt.): utterance level cepstral mean normalisation.

[%]	**targets**	**training set**		**test set**	
		WER	WERR	WER	WERR
w/o mapping (close talk)		9.27		9.48	
w/o mapping (distant talk)		18.30		18.77	
BLSTM	absolute	15.80	13.7	20.67	-10.0
BLSTM	differential	15.26	16.6	**17.73**	5.5
BLSTM+CMN (utt.)	absolute	14.61	20.2	19.38	-3.0
BLSTM+CMN (utt.)	differential	14.96	18.3	**17.32**	7.7

As shown in Table 4.19, the baselines of the distant talk of the English corpus are WERs of 18.30 % and 18.77 % for the training and test sets, both of which almost double the baseline of close talk (WERs of 9.27 % and 9.48 % for the training and test sets). As expected, a high gain is obtained for the training set when applying channel mapping. Nevertheless, such a high gain is not observed for the test set. Only when using the differential targets to train neural networks, a gain can be obtained by 5.5 % of WERR on the mismatched test set, and can be enlarged to 7.7 % WERR when I further implement utterance level CMN that mainly aims to remove static noise as well as the early reverberation [87]. In this experiment, it can also be noticed that the indirect mapping way (using differential targets for networks training) significantly overcomes the direct mapping way (using absolute targets for networks training).

From the above two experiments, the results imply that the inter-room scenario is more challenging when compared to the intra-room scenario shown above. This observation leads to the following conclusions: On the one hand, the performance improvement on both training and test sets indicates that different rooms share some common reverberation information. These shared information can be learned by the BLSTM networks. On the other hand, the different gains obtained by the training and test sets suggest that the networks probably learn too much information from a specific acoustic environment. Therefore, to generalise the neural networks and exploit more common information, I further did some preliminary experiments on the English corpus for generalisation, on the means of adding more Gaussian noise to the input when training networks. The results in Table 4.20 indicate that by improving the generalisation capability, the networks can learn more common information, as the WER reduces from 17.32 % to 16.89 % with about 3 % relative performance improvement.

Table 4.20: Performance of the test set of the *English* corpus by adding more training noise to regularise the BLSTM-RNN (one hidden layer with 128 neurons) feature dereverberation model.

noise variation	**targets**	WER [%]	WERR [%]
0.1	differential	17.32	7.7
0.2	differential	17.15	8.6
0.3	differential	**16.89**	10.0
0.4	differential	16.96	9.6

4.3.2 Summary

In this section, a feature-based dereverberation method was investigated for a realistic distant-talk ASR system. The basic idea is to use BLSTM-RNNs for channel mapping – from distant-talk cepstral feature space to its close-talk counterpart.

In the scenario of distant talk, the speech signals at each frame time will impact the subsequent frames in long term. This issue consequentially requires a learning algorithm that has the potential to not only access long-term context information but also make use of the future information. The bidirectional structure (past and future) of LSTM neural networks is capable of dealing with these issues. The experimental results on a French corpus show a WERR of more than 16 % for ASR, which significantly outperform the 'conventional' networks MLPs (one-side z-test, $p < .001$) and BRNNs (one-side z-test, $p < .05$). Such effectiveness of the feature mapping method is further confirmed by integrating widely-used adaptation techniques of MLLR or/and fMLLR, which yields the best performance of about 20 % of WERR. It is also confirmed in the scenario of cross- or inter-room evaluation, as the evaluation sets in a mismatched acoustic environment also obtain a gain via channel mapping when using BLSTM.

This study also presents another indirect way for channel mapping – the differential feature vectors (between the distant-talk speech and the close-talk speech) as network targets, then adding the estimated differential feature vectors to the counterpart of original distant-talk ones. The results based on a rich number of experiments show that this indirect mapping strategy can compete with the previously used direct mapping strategy, particularly in some cases like using a simple network structure and evaluating mismatched data sets. All these cases are quite welcome for real-life applications.

Due to a gain gap between matched and mismatched evaluation cases, future work will focus on the further exploitation of joint acoustic information across different rooms with the goal of 'blind' dereverberation application. One way to achieve this is to train the networks by a vast amount of reverberant speech collected in a variety of rooms. Furthermore, one can apply to the objective functions some generalisation terms such as weight decay. In addition, it seems also beneficial to develop a way of selecting predefined mapping models for different room categories, in order to ultimately explore the advantages of the room-specific models.

5

General Summary and Outlook

Intelligent speech analysis (ISA) systems can be applied to, for example, intellectual interaction between humans and machines where linguistic subsystems (e.g., automatic speech recognition [ASR] systems) are able to interpret semantic content, while computational paralinguistics subsystems are capable of analysing information about the speaker, such as characteristics and states (e.g., emotion). When applying such systems in real-world scenarios, however, developers will encounter numerous challenges with respect to realistic conditions, such as the diversity of speakers, background noise, and reverberation. Computational paralinguistics, which is still a new research topic, suffers from a shortage of training data. This problem is widely considered the most frontier challenge in the field, which, in turn, prevents the creation of a robust acoustic model. In addition, large feature set size impedes real-world application when implementing a distributed structure. For speech recognition, especially in a distant-talk scenario, performance is significantly undermined by noise and reverberation.

To deal with these challenges, the key strategy of this thesis is to exploit data quantity and quality in an efficient way. Thus, the focus of this thesis is semi-autonomous data enrichment and data optimisation either by transferring the effective approaches developed by the machine learning community or proposing novel approaches. More precisely, this work presents and evaluates a variety of approaches to achieve the four major objectives: (1) reuse of pre-existing heterogeneous data, (2) efficiently exploiting the vast amount of unlabelled data, (3) reducing feature size to satisfy the requirements of physical transmission, storage, and security, and (4) enhancing the features corrupted as a result of reverberation.

Here, the conclusions of this thesis are summarised and suggestions made for further research.

5.1 Summary

After an introduction of the general framework of ISA systems in Chapter 2, each of the four aforementioned objectives of this thesis is addressed by proposing and investigating appropriate approaches in both theoretical aspects (Chapter 3) and empirical aspects (Chapter 4).

(1) Unified perspectives on leveraging available labelled data were illustrated to overcome the varieties among different databases. In particular, the approaches of data pooling, bagging, sampling, and selection were presented in Section 3.1, and further examined using the speech emotion recognition (SER) task from Sections 4.1.1 to 4.1.4. The corresponding experimental results in Section 4.1.1 show that, in the case of cross-corpus evaluation [217], pooling or bagging multiple similar databases boosts the performance of the model trained on a single database. The idea of data pooling was then extended to aggregate synthesised emotional speech, and the results obtained in Section 4.1.2 further prove its efficiency [28]. However, such a direct data fusing approach overlooks the issues of noisy data (i.e., severely mislabelled or highly distorted data) and imbalanced class distribution. Accordingly, distance-based data selection [221], and agreement- and sparseness-based data selection [183] were proposed. The results in Section 4.1.3 and 4.1.4 show that the selected smaller size of subsets achieved almost the same performance as the baseline model.

(2) Manifold semi-automatic self-optimisation ways were proposed and presented in Section 3.2 and investigated from Sections 4.1.5 to 4.1.8, so as to enrich the data quantity by exploiting massive unlabelled data for SER using less human effort. One way of achieving this was to leverage the ability of machines (pre-trained model) to predict unlabelled data, and, in turn, to integrate these machine labelled data to retrain a new model. The success of this method was empirically confirmed [226]. Another way was to seek help from humans by iteratively making queries to select the most informative data. A traditional confidence-value-based active learning was examined and proved to be highly efficient. Both ways have their own advantages and disadvantages: The semi-supervised learning often accumulates the mislabelled data and tends to predict the dominant class, whereas the active learning needs more laborious human work. In an attempt to compensate respective drawbacks, a novel solution – cooperative learning (CL) – was proposed to iteratively proceed using both methods. Extensive experiments were devoted to comparing the performance of CL for SER to the other methods. The observations show that CL significantly outperforms the learning methods that merely use the effort from machines or humans. Furthermore, the idea of multi-view learning was merged into the CL method, which gave rise to better performance [231].

(3) The recommended compression method of split vector quantisation for distributed speech recognition was transferred to distributed computational paralinguistics (cf. Section 3.3.1). By conducting extensive experiments on five time-scaled

paralinguistic tasks (i.e., emotion, intoxication, pathology, age, and gender), the proposed compression method on the statistic features was demonstrated to be superior (cf. Section 4.2.2): Even if the compression ratios are up to 40, the recognition accuracies are still very close to the baseline [8].

(4) A novel feature enhancement method was proposed in Section 3.3.2 with the aim of alleviating the influence of noise, especially reverberation, on speech signals for distant-talk ASR. In particular, this method is by means of mapping the feature space from distorted speech to the one from clean speech via Bidirectional Long Short-Term Memory modelling for regression [212]. It was evaluated on a realistic digital TV application in Section 4.3. The results indicate that such a context-sensitive neural networks-based method outperforms alternative neural networks (e.g., Multilayer Perceptrons and conventional recurrent neural networks), delivering a relative word error rate reduction of approximately 16 % in an intra room scenario and 10 % in an inter room scenario.

5.2 Outlook

Despite the techniques that were proposed and investigated to address the data scarcity and distortion issues for ISA in this thesis, there are many thinkable and promising possibilities to enrich and optimise the data in future. The first possibility goes to utilise truly unlabelled data collected from the real world, such as YouTube, movies, call centres, and video games. Contrary to the use of pre-prepared data, constantly mining useful information from such applications is more challenging. Thus, it will be interesting to examine the feasibility and efficiency of the proposed methods in these cases. Moreover, a major part of paralinguistic tasks (e.g., emotion) are subjective, which means that multiple annotators are demanded for labelling, so as to obtain a gold standard. In this case, the annotation work for a subjective task requires several times the work for an objective task. To fulfil this work, one way called crowd sourcing might be feasible, which takes the advantage of the Internet, i.e., anyone in the world can contribute to this annotation work. In addition, more sophisticated algorithms on the basis of agreement levels could also further ease the burdensome work. In comparison with the investigated CL methods, the information of labelling agreement levels could also be helpful to improve the system performance. Some preliminary results have already been shown in [235].

So far, although some of the most frontier issues regarding the ISA were tackled, there are several other issues that are still laying ahead when applying such systems in the real world. State-of-the-art machine learning algorithms could be further employed to increase the recognition accuracy. One potential direction is to apply multi-task learning that aims to learn the knowledge of a target task jointly with other related tasks, for example, using shared representations [240]. For computational paralinguitics, each paralinguistic task (e.g., gender) also plays an important role in the analysis of almost all other tasks (e.g., emotion) [4]. Thus, this mutually joint information may benefit the individual task recognition. In the context of computational paralinguistics, however, multi-task learning is still a fresh topic, as there are only a few preliminary studies on this topic [241, 238]. Another possible research direction involves distributed computational paralinguistics. Several related issues are still waiting to be solved, for example, the impact of package loss and security. Moreover, the paralinguistic subsystems and linguistic subsystems were considered independently, heretofore. Thus, it seems promising to investigate a unified scheme that could merge the two subsystems into one.

Overall, this thesis has investigated all four goals in detail and hopefully the field has gained from the research presented here.

A Databases

A large number of databases are used for the performance evaluation throughout the thesis. This appendix summarises the technical details of each database. Particularly, Appendix A.1 introduces multiple speech emotion databases (i.e., FAU Aibo Emotion Corpus, other eight related corpora, as well as synthesised OpenMary and TXT2PHO databases) that are employed in Section 4.1. Then, Appendix A.2 describes four general paralinguistic databases (i.e., Alcohol Language Corpus, Sleepy Language Corpus, NKI CCRT Speech Corpus, and Agender Database) for the experiments in Section 4.2. Finally, Appendix A.3 gives a description of the reverberant French and English databases for speech recognition in Section 4.3.

A.1 Emotion Databases

In this part, the FAU Aibo Emotion Corpus is presented in detail. It is adopted in a plethora of experiments in Section 4.1, as well as widely used in others' research [63, 53, 242, 243]. The other eight emotional corpora are also frequently-used databases in the field of emotion analysis [23, 25]. In this thesis, the eight databases are mainly employed to exploit the value of labelled data. Among these eight databases, some of them are mixed audiovisual collections, whereas only the audio information is used in this thesis. In comparison to human speech databases, OpenMary and TXT2PHO are two artificially synthesised speech databases.

A.1.1 FAU Aibo Emotion Corpus

FAU Aibo Emotion Corpus (AEC) [53] is the official corpus of the INTERSPEECH 2009 Emotion Challenge (EC) [63]. This corpus contains audio recordings of German-speaking children interacting with Sony's pet robot Aibo [53]. For the construction of this database, children were led to believe that the Aibo was responding to their commands by producing a series of fixed and predetermined behaviours.

Table A.1: Distribution of speakers and instances per partition of the FAU Aibo Emotion Corpus (AEC) [53]. NEG: negative emotions; IDL: neutral and positive emotions.

	# speakers		# instances		
	male	female	**NEG**	**IDL**	Σ
Pool	13	13	3 358	6 601	9 959
Validation	8	17	2 465	5 792	8 257
Σ	21	30	5 823	12 393	18 216

Nevertheless, the Aibo robot did sometimes disobey to the children's commands, which provoked various types of emotional reactions. The recordings include speech samples from 51 children (30 females) with ages ranging from 10 to 13 years that were taken at two different German schools to which I will refer to in this thesis as MONT and OHM. The whole corpus comprises a total of 9.2 hours of speech without pauses and was recorded through a DAT-recorder (16 bit, 48 kHz down-sampled to 16 kHz) placed on a wireless head set. The recordings were segmented into turns using a pause threshold of 1 s. Five students of advanced linguistics were then asked to listen to the various samples and to annotate each one of them by selecting one specific label (from a set of 11 predefined labels) to describe the emotional character of the sample. The labels used were: *neutral*, *angry*, *touchy*, *reprimanding*, *emphatic*, *surprise*, *joyful*, *helpless*, *motherese*, *bored*, and *rest*. If more than three labellers assigned a specific label to a speech sample (majority voting), that label was chosen to describe the emotional character of the segment.

In this thesis, I employed the same natural speech corpus used in the INTERSPEECH 2009 EC [63] that consists of 18 216 instances taken from the full database. Each instance consists of a manually defined chunk of speech longer than a word and shorter than a turn defined based on syntactic-prosodic criteria. The original 11 classes were mapped onto two cover classes: one consisting of **NEG**ative emotion labels (*angry*, *touchy*, *reprimanding*, *emphatic*), and the other consisting of non-negative (**IDL**e) states (for more information about the database development and data processing please refer to [63]). In order to guarantee speaker independence, I used the data recorded at the OHM school as the unlabelled data pool (9 959), and the data recorded at the MONT school as the validation set (8 257). Table A.1 shows the details of the data sets used.

A.1.2 Other Eight Emotion Databases

The details of the eight emotional human speech databases are summarised in Table A.2 (also includes two synthesised speech databases.). In the following, I briefly introduce the eight emotion databases.

ABC

The Airplane Behaviour Corpus (ABC) [244] is an audiovisual emotion database. It is crafted for the special target application of public transport surveillance. In order to induce a certain mood, a script was used, which lead the subjects through a guided storyline: Prerecorded announcements by five different speakers were automatically played back controlled by a hidden test-conductor. As a general framework, a vacation flight was chosen, consisting of several scenes such as start, serving of wrong food, turbulences, falling asleep, conversation with a neighbour, or touch-down. The general setup consisted of an airplane seat for the subject, positioned in front of a blue screen. Eight subjects in gender balance from 25 to 48 years (mean 32 years) took part in the recording. The language throughout recording is German. A total of 11.5 h video was recorded and annotated independently after pre-segmentation by three experienced male labellers within a closed set. The average length of the 431 clips is 8.4 s.

AVIC

The AudioVisual Interest Corpus (AVIC) [35] is another audiovisual emotion corpus. In its scenario setup, a product presenter leads one of 21 subjects (10 female) through an English commercial presentation. The level of interest is annotated for every turn reaching from boredom (subject is bored with listening and talking about the topic, very passive, does not follow the discourse; this state is also referred to as level of interest [loi] 1, i.e., loi1), over neutral (subject follows and participates in the discourse, it cannot be recognised, if she/he is interested or indifferent in the topic; loi2) to joyful interaction (strong wish of the subject to talk and learn more about the topic; loi3). Additionally, the spoken content and non-linguistic vocalisations are labelled in the AVIC set. For the evaluation I use all 3 002 phrases, in contrast to only 996 phrases with high inter-labeller agreement as e.g., employed in [35].

EMO-DB

The Berlin Emotional Speech Database (EMO-DB) [245] was created by 10 professional actors (5 males and 5 females), who were asked to express 10 predefined German sentences like ‘Der Lappen liegt auf dem Eisschrank’ (The cloth is lying on the fridge) in seven emotional states that include anger, boredom, disgust, fear, joy, neutral, and sadness. Finally, around 900 utterances were obtained.

DES

The Danish Emotional Speech (DES) [246] database has been chosen as one of the 'traditional representatives', because it is easily accessible and well annotated. The data used in the experiments are nine Danish sentences, two words and chunks that are located between two silent segments of two passages of fluent text, for example: 'Nej' (No), 'Ja' (Yes), 'Hvor skal du hen?' (Where are you going?). The set used contains 419 speech utterances (i.e., speech segments between two silence pauses) that are expressed by four professional actors, two males and two females. Speech is expressed in five emotional states: anger, happiness, neutral, sadness, and surprise. Twenty judges (native speakers from 18 to 58 years old) verified the emotions with a score rate of 67 %.

eNTERFACE

The eNTERFACE [247] corpus is a further public audiovisual emotion database. It consists of induced anger, disgust, fear, joy, sadness, and surprise speaker emotions. 42 subjects (eight female) from 14 nations are included. It consists of office environment recordings of predefined spoken content in English. Each subject was instructed to listen to six successive short stories, each of them eliciting a particular emotion. They then had to react to each of the situations by uttering previously read phrases that fit the short story. Five phrases are available per emotion as 'I have nothing to give you! Please dont hurt me!' in the case of fear. Two experts judged whether the reaction expressed the emotion in an unambiguous way. Only if this was the case, the sample was added to database. Overall, the database consists of 1 277 samples.

SAL

The Belfast Sensitive Artificial Listener (SAL) data is part of the final HUMAINE database [248]. This subset contains 25 recordings in total from 4 speakers (2 male, 2 female) with an average length of 20 minutes per speaker. The data contains audiovisual recordings from natural human–computer conversations that were recorded through a SAL interface designed to let users work through a range of emotional states. The data has been labelled continuously in real time by four annotators with respect to valence and activation using the feel-trace system: The annotators used a sliding controller to annotate both emotional dimensions separately whereas the adjusted values for valence and activation were sampled every 10 ms to obtain a temporal quasi-continuum. To compensate linear offsets that are present among the annotators, the annotations were normalised to zero mean globally. Further, to ensure common scaling among all annotators, each annotator's labels were scaled so that 98 % of all values are in the range from -1 to +1. The 25 recordings have been split into turns using an energy based voice activity detection. A total of 1 692 turns is accordingly contained in the database. Labels for each turn are computed by averaging the frame level valence and activation labels over the complete turn. Apart from the necessity to deal with continuous values for time and emotion, the great challenge of the SAL database is the fact that one must deal with all data.

SUSAS

The Speech under Simulated and Actual Stress (SUSAS) database [230] contains audio recordings of speakers in various (actual and simulated) stress conditions and organised in different domains. In this thesis, I focus on the 'Actual Speech Under Stress' domain, which includes audio recordings of speech produced in the 'Scream Machine' scenario, one of the subject motion-fear tasks. In this scenario, 7 speakers (3 female) were taken in a roller-coaster (the 'Scream Machine') ride for about 90 s and asked to repeat words from a 35-word vocabulary card held in their hands at different moments. Each speaker performed the task twice.

In the task scenario, different levels of stress are spontaneously evoked by the dynamics of the roller-coaster ride, resulting in the various levels of stress being expressed in the voice. A total of 1 642 utterances were collected during the rides (sampled at 8 kHz, 16 bit). Subsequently these utterances were segmented into words, resulting in 3 593 instances that were then annotated for stress levels (i.e., neutral, medium, high stress, and screaming) based on the time and position during the ride.

VAM

The Vera-Am-Mittag (VAM) corpus [249] consists of audiovisual recordings taken from a German TV talk show. The set contains 946 spontaneous and emotionally coloured utterances from 47 guests of the talk show that were recorded from unscripted, authentic discussions. The topics were mainly personal issues such as friendship crises, fatherhood questions, or romantic affairs. To obtain non-acted data, a talk show in which the guests were not being paid to perform as actors was chosen. The speech extracted from the dialogues contains a large amount of colloquial expressions as well as nonlinguistic vocalisations and partly covers different German dialects. For annotation of the speech data, the audio recordings were manually segmented to the utterance level, whereas each utterance contained at least one phrase. A large number of human labellers was used for annotation (17 labellers for one half of the data, six for the other). The labelling bases on a discrete five point scale for three dimensions mapped onto the interval of [-1,1]: The average results for the standard deviation are 0.29, 0.34, and 0.31 for valence, activation, and dominance. The averages for the correlation between the evaluators are 0.49, 0.72, and 0.61, respectively. The correlation coefficients for activation and dominance show suitable values, whereas the moderate value for valence indicates that this emotion primitive was more difficult to evaluate, but may partly also be a

Table A.2: Overview of the eight emotional human speech corpora and two emotional synthesised speech corpora (Lan: languages; Sp: speech, Em: emotion, Lab: labellers, Rec: recording environment, f/m: (fe-)male subjects), synth: synthesised.

Corpus	**Lan**	**Sp**	**Em**	**# Arousal**		**# Valence**		**# All**	**h:mm**	**# m**	**# f**	**# Lab**	**Rec**	**Hz**
				-	+	-	+							
						Human Speech (HS)								
ABC	German	fixed	induced	104	326	213	217	430	1:15	4	4	3	studio	16k
AVIC	English	free	natural	553	2449	553	2449	3002	1:47	11	10	4	studio	44k
DES	Danish	fixed	acted	169	250	169	250	419	0:28	2	2	–	studio	20k
EMO-DB	German	fixed	acted	248	246	352	142	494	0:22	5	5	–	studio	16k
eNTER	English	fixed	induced	425	852	855	422	1277	1:00	34	8	2	studio	16k
SAL	English	free	natural	884	808	917	779	1692	1:41	2	2	4	studio	16k
SUSAS	English	fixed	natural	701	2892	1616	1977	3593	1:01	4	3	–	noisy	8k
VAM	German	free	natural	501	445	875	71	946	0:47	15	32	6/17	noisy	16k
						Synthesised Speech (SS)								
OpenMary	German	fixed	synth	280	350	420	210	630	0:33	4	3	–	–	22k
TXT2PHO	German	fixed	synth	280	350	420	210	630	0:33	4	3	–	–	16k

result of the smaller variance of valence.

A.1.3 Two Synthesised Emotion Databases

The two synthesised speech databases – **TXT2PHO** and **OpenMary** – are generated by the emotional speech synthesis system – Emofilt [250]. TXT2PHO and OpenMary are also the names of their two phonemisation components. Ten sentences of the EMO-DB (cf. Section A.1.2) are served as the content of synthesised speech. Both synthesised databases cover eight target emotions (happiness, joy, boredom, yawning, fear, despair, hot anger, sadness) plus a neutral state, using all seven German voices (four female and three male), thus obtaining 1 260 samples (10 × 2 × 9 × 7, cf. Table A.2). For more details of the generation process, please refer to [28, 250].

A.1.4 Mapping and Clustering

Upon the above databases description, it can been seen that the eight human speech databases, as well as the two synthesised speech databases, are annotated in various emotion categories or continuous valued dimensions. In this case, I mapped the diverse emotion groups into the quadrants of two-dimensional arousal-valence space as in [25]: arousal (i.e., high vs. low) and valence (i.e., positive vs. negative) in order to generate a unified set of labels that can be used for cross-corpus experiments. It is because on the one hand, most categorical emotion labels (such as the 'Big Six' emotions joy, sadness, anger, fear, surprise and disgust) can be expressed as points in the arousal-valence space [251]; on the other hand, the majority of the considered databases is annotated by emotion categories instead of a more fine-grained, continuous annotation – this is mostly due to the type of emotion elicitation used for creating these databases. In addition, these mappings are not straight forward – it also favours better balance among target classes. The specific mapping strategies are given in Tables A.3 and A.4 for arousal and valence, respectively.

A.2 General Paralinguistic Databases

This part of the Appendix focuses on introducing four databases in accordance with different paralinguistic tasks, covering speakers' medium-term states, such as intoxication, sleepiness, pathology, and speakers' long-term characteristics like age and gender. All these databases are prepared for Section 4.2.

Table A.3: Mapping the classes of various databases to a binary arousal (High or Low).

Corpus	**High**	**Low**
	Eight Human Speech Databases	
ABC	aggressive, cheerful, intoxicated, nervous	neutral, tired
AVIC	loi2, loi3	loi1
DES	angry, happy, surprise	neutral, sad
EMO-DB	anger, fear, joy	boredom, disgust, neutral, sadness
eNTERFACE	anger, fear, happiness, surprise	disgust, sadness
SAL	q1, q4	q2, q3
SUSAS	high stress, medium stress, screaming, fear	neutral
VAM	q1, q4	q2, q3
	Two Synthesised Speech Databases	
OpenMary / TXT2PHO	despair, fear, happiness, hot anger, joy	boredom, neutral, sadness, yawning

Table A.4: Mapping the classes of various databases to a binary valence (Positive or Negative).

Corpus	**Positive**	**Negative**
	Eight Human Speech Databases	
ABC	cheerful, neutral, intoxicated	aggressive, nervous, tired
AVIC	loi2, loi3	loi1
DES	happy, surprise, neutral	angry, sad
EMO-DB	neutral, joy	anger, boredom, disgust, sadness, fear
eNTERFACE	happiness, surprise	anger, fear, disgust, sadness
SAL	q1, q2	q3, q4
SUSAS	medium stress, neutral	high stress, screaming, fear
VAM	q1, q2	q3, q4
	Two Synthesised Speech Databases	
OpenMary / TXT2PHO	happiness, joy, neutral	boredom, despair, fear, hot anger, sadness, yawning

Alcohol Language Corpus

The Alcohol Language Corpus (ALC) [252] contains 38 hours of genuine alcohol intoxicated and sober speech. For the experiments, as for the 2011 Speaker State Challenge (SSC) [65], I use a gender balanced subset of the ALC with 154 speakers (77 male, 77 female). Speakers are within the age range of 21 to 75 years and were selected to ensure a balance of German dialects. The corpus is subdivided into training, testing and development partitions guaranteeing speaker independence.

To create the corpus, speakers were recorded at self-chosen blood alcohol concentrations (BACs) ranging from 0.28 to 1.75 per mill. The intoxicated speech material in the ALC was obtained by a speech test that the speakers were asked to perform immediately after taking a blood sample. Since the speech test did not last longer than 15 minutes, it is ensured that the BAC throughout the speech test remains roughly equal to the measured BAC before the test. At least two weeks after the intoxicated speech test, each speaker returned to undergo a second recording in sober condition. The sober recordings were chosen to be roughly twice as long as the intoxicated recordings.

Three different speech styles are part of each ALC recording: read speech, spontaneous speech, and command & control. For the experiments in this thesis, the recordings from speakers with BAC ≤ 0.5 per mill were labelled as non-alcoholised (**NAL**). All other were labelled as alcoholised (**AL**).

Sleepy Language Corpus

The Sleepy Language Corpus (SLC) [38] is the official corpus of the Sleepiness Sub-Challenge of the INTERSPEECH 2011 Speaker State Challenge (SSC) [65]. To build this corpus, 99 participants with an age range of 20–52 years took part in six partial sleep deprivation studies. The recording took place in a realistic car environment or in lecture-rooms, including read and spontaneous German speech as detailed in [65]. To annotate the value of sleepiness, the Karolinska Sleepiness Scale (KSS) was used by the subjects and two raters. Scores ranging from 1–10 are given from *extremely alert* (1) to *cannot stay awake* (10). For training and classification purpose, the recordings (mean = 5.9, standard deviation = 2.2) were binarised into two classes: not sleepy ('**NSL**') and sleepy ('**SL**') with the threshold of 7.5 on the KSS.

NKI CCRT Speech Corpus

The 'NKI CCRT Speech Corpus' (NCSC) [44] is the official corpus of the Pathology Sub-Challenge of the INTERSPEECH 2012 Speaker Trait Challenge (STC)

[1] Test labels of Agender are not freely available. Thus, only its partitions of train and develop are used in Section 4.2.2.

Table A.5: Overview of selected corpora for emotion (AEC), intoxication (ALC), sleepiness (SLC), pathology (NCSC), age and gender (Agender) recognition tasks. Languages (LA): German (DE) and Dutch (NL); speech types (TY): spontaneous (S) and promoted (P); number of subjects (S) and instances (INST); total speech time (TT) and average speech time per chunk (TC); recording rate (Hz).

Corpus	**LA**	**TY**	**S #**	**TT[H]**	**TC[s]**	**INST #**	**Hz**
AEC	DE	S	51	8.9	1.8	18 216	16k
ALC	DE	P	162	43.8	12.8	12 360	16k
SLC	DE	S	99	–	–	9 089	16k
NCSC	NL	P	55	2.0	3.0	2 386	16k
Agender[1]	DE	P	770	35.9	2.4	53 074	8k

Table A.6: Instances distribution per partition (Train, Develop or Test) for four paralinguistic corpora—ALC, SLC, NCSC, and Agender. (N)AL: (non-)intoxicated; (N)SL: (non-)sleepiness; (N)I: (non-)intelligible; C/Y/A/S: children/young/adult/senior; X/M/F: children/male/female.

(a) ALC, SLC, NCSC

	ALC		**SLC**		**NCSC**	
	NAL	**AL**	**NSL**	**SL**	**I**	**NI**
Train	3 750	1 650	2 215	1 241	384	517
Develop	2 790	1 170	1 836	1 079	341	405
Test	1 620	1 380	1 957	851	475	264
Σ	8 160	4 200	5 918	3 171	1 200	1 186

(b) Agender

	Agender: Age				**Agender: Gender**		
	C	**Y**	**A**	**S**	**X**	**M**	**F**
Train	4 406	8 657	8 990	10 473	4 406	13 985	14 135
Develop	2 396	4 892	5 873	7 387	2 396	8 508	9 644
Σ	6 802	13 549	14 863	17 860	6 802	22 493	23 779

[66]. The database was created at the Department of Head and Neck Oncology and Surgery of the Netherlands Cancer Institute and consists of speech recordings from 55 speakers (10 female; mean age is 57 y.o.) before and after Chemo-Radiation Treatments (CCRT). All speakers read a text in the Dutch language with an emotionally neutral content. Thirteen speech pathologists evaluated the speech recordings in an on-line experiment on an intelligibility scale ranging from 1 to 7. Finally, evalua-

Table A.7: Distribution of speakers, sentences, words, and recording time of close talk per partition of French and English corpora.

	French		**English**	
	train	test	train	test
# speakers (f/m)	5/6	6/5	5/4	5/6
# sentences	2 231	4 619	1 430	1 801
# words	15 148	30 094	7 886	9 907
time (hours)	4.1	4.2	2.9	3.4

tor weighted estimator (EWE) was used to compute and discretise the ratings into binary classes—intelligible (**I**) and non-intelligible (**NI**)—wising the median of the ratings distribution.

Agender Database

The Agender database [47] is the official corpus of the INTERSPEECH 2010 Paralinguistic Challenge (PC) Age and Gender Sub-Challenges [64]. This database was collected by a commercial company with the aiming of identifying people of specific targeted ages and genders. The participants were asked six times to call an automated Interactive Voice Response system and to repeat various German utterances or to produce free speech content. The calls were made in through a mobile phone in various recording environments, and in different days and times so as to ensure more variation in the voices of each speaker. In the Challenge task, four classification classes were used for age—**C**hildren, **Y**oung, **A**dult, and **S**enior—and three for gender—Children (**X**), **M**ale, and **F**emale. Additionally, I also consider seven new classes that are generated by combining the various age and gender classes. Hereinafter, this classification task is referred to as ‘Age+Gender’ (cf. [64]).

A.3 Speech Recognition Databases

The French and English Reverberation Databases

The *French* and *English* corpora were recorded in a realistic acoustic space environment. Both databases were collected for a speech-controlled TV application. This application was designed to enable the user to change the TV controls (volume, brightness, etc.) or to browse the programs using her voice. Table A.7 shows the statistics of the two databases. The French corpus was recorded in a living room containing furniture, where one microphone near the mouth recorded the close talk, and another microphone array, consisting of 16 channels, recorded the distant talk. Twenty-two native French speakers (11 females) were asked to speak naturally so as to control the TV as their wish, i.e., ‘je veux un film avec Cameron Diaz (I want a movie with Cameron Diaz)’. Finally, 8.3 h recordings were obtained, including about 5K sentences and 30K words in total. The distant talk data obtained from a 16-channel microphone array was grouped into four disjoint sets (1-4, 5-8, 9-12, and 13-16). The four channel speech in each of the sets was beamformed and noise reduced to obtain a single speech signal. As a result, the amount of distant talk training/test data was four times that of its close talk counterpart. The whole database was then divided into training and test sets, both speaker-independently and equally. Likewise, 6.3 h of recordings were captured for the English corpus that comprised 20 speakers (10 females), and approximately 3K sentences and 18K words in total. For French, the training and test data sets were recorded in the same room, but for English, these data sets were recorded in different rooms. The details of the French and English corpora are shown in Table A.7.

Acronyms

ABC	Airplane Behaviour Corpus
ACF	AutoCorrelation Function
AIS	Agreement-based Instance Selection
AL	Active Learning
ALC	Alcohol Language Corpus
ANN	Articial Neural Network
ASC	Affect Sub-Challenge
ASIS	Agreement- and Sparseness-based Instance Selection
ASR	Automatic Speech Recognition
AVIC	AudioVisual Interest Corpus
BAC	Blood Alcohol Concentration
BLSTM	Bidirectional Long Short-Term Memory
BPTT	BackPropagation Through Time
BRNN	Bidirectional Recurrent Neural Network
CART	Classication And Regression Tree
CC	Coefficient Correlation
CCRT	Chemo-Radiation Treatments
CL	Cooperative Learning
CMLLR	Constrained MLLR
CMN	Cepstral Mean Normalisation
CNN	Condensed Nearest Neighbour

coAL	co-Active Learning
CV	Confidence Value
DCT	Discrete Cosine Transform
DES	Danish Emotional Speech
DNN	Deep Neural Network
DROP	Decremental Reduction Optimisation Procedure
DSCC	Delta-Spectral Cepstral Coefficient
EC	Emotion Challenge
EDIS	Euclidean Distance-based Instance Selection
EM	Expectation-Maximisation
EMO-DB	Berlin Emotional Speech Database
ETSI	European Telecommunications Standards Institute
EWE	Evaluator Weighted Estimator
FNN	Feed-forward Neural Network
GMM	Gaussian Mixture Model
GR	Gain Ratio
HMM	Hidden Markov Model
HNR	Harmonics-to-Noise Ratio
HQ	Histogram-based Quantisation
HS	Human Speech
ID3	Iterative Dichotomiser 3
ISA	Intelligent Speech Analysis
KSS	Karolinska Sleepiness Scale
LDA	Linear Discriminant Analysis
LLD	Low-Level Descriptor
LOCO	Leave-One-Corpus-Out
LPCC	Linear Prediction Cepstral Coefficient
LPC	Linear Prediction Coefficient
LPC	Linear Prediction Coding
LP	Linear Prediction

LSTM	Long Short-Term Memory
MAP	Maximum A Posteriori
MFCC	Mel-Frequency Cepstral Coefficient
MLLR	Maximum Likelihood Linear Regression
MLP	Multilayer Perceptrons
MSE	Mean Squared Error
MVL	Multi-View Learning
NB	Naïve Bayes
NCSC	NKI CCRT Speech Corpus
NMF	Non-negative Matrix Factorisation
PCA	Principle Component Analysis
PCC	Pearson product-moment Correlation Coefficient
PC	Paralinguistic Challenge
PL	Passive Learning
PLP	Perceptual Linear Prediction
POP	Pattern by Ordered Projections
RANSAC	RANdom SAmple Consensus
RASTA-PLP	RelAtive Spectral Transform - Perceptual Linear Prediction
RBF	Radial Basis Function
RIR	Room Impulse Response
RIS	Random Instance Selection
RNN	Recurrent Neural Network
SAL	Belfast Sensitive Artificial Listener
SIS	Sparseness-based Instance Selection
SLC	Sleepy Language Corpus
SMO	Sequential Minimal Optimisation
SMOTE	Synthetic Minority Oversampling TEchnique
SNN	Selective Nearest Neighbour rule
SSC	Speaker State Challenge
SSE	Sum of Squared Errors

SSL	Semi-Supervised Learning
SS	Synthesised Speech
STC	Speaker Trait Challenge
STDFT	Short-Time Discrete Fourier Transform
STFT	Short-Time Fourier Transform
SUSAS	Speech under Simulated and Actual Stress
svCL	single-view Cooperative Learning
xvCL	mixed-view Cooperative Learning
mvCL	multi-view Cooperative Learning
SVM	Support Vector Machine
SVQ	Split Vector Quantisation
TSVM	Transductive SVM
TTS	Text-To-Speech
UAR	Unweighted Average Recall
VAD	Voice Activity Detection
VAM	Vera-Am-Mittag
VQ	Vector Quantisation
WAR	Weighted Average Recall
WER	Word Error Rate
WOZ	Wizard-of-Oz
ZCR	Zero-Crossing Rate

List of Symbols

Acoustic Feature Extraction

n	index of sampling
$S(n)$	discrete speech signal
k	time-shift
$R(k)$	autocorrelation function
F_0	fundamental frequency (pitch)
T_0	pitch period
E	Energy
Z	zero-crossing rate
$sgn[\cdot]$	sign function
f	frequency
$Mel(f)$	Mel-scale frequency
$m(l)$	the l-th log filter bank amplitudes on Mel-scale frequency
$c(i)$	the i-th Mel-frequency correlation coefficient
n'	index of frame
$J(n')$	jitter
$A(n')$	peak to peak amplitude difference
$sh(n')$	shimmer

Classification

n	number of examples
$\mathbf{x}$	feature vector
y	referenced class
$\mathcal{X}$	feature space
$\mathcal{Y}$	prediction space (class domain)
$\mathbb{R}^d$	d-dimensional feature space
$\mathbf{w}$	normal vector of SVM hyperplane
b	bias
α	Lagrange multiplier
$L(\cdot)$	Lagrange function
$K(x_i, x_j)$	kernel function
C	constant
$\phi(\mathbf{x})$	high-dimensional feature mapping
$\mathcal{D}$	database
p	number of positive examples
n	number of negative examples
k	number of subsets
A	attribute set
$H(\cdot)$	entropy
$IG(\cdot)$	information grain
$GR(\cdot)$	gain ratio
T	number of trees
$\mathbf{x}'_t$	random attribute subspace for training the t-th tree
β_i	activation of the i-th node
α_i	input of the i-th node
w_{ij}	the weight from node i to node j
f	activation function
$\mathbf{y}$	predicted output vector
$\mathbf{z}$	designed output vector

$\mathcal{J}(\theta)$	objective function
$E(\mathbf{z}, \mathbf{y})$	sum of squared errors between **z** and **y**
Δw_{ij}	weight change
δ_i	error of the i-th node
λ	learning rate
η	momentum
γ	weight decay
$\mathbf{x}_{1:T}$	time sequential (1:T) feature vectors
I	input nodes
H	hidden nodes
K	output nodes
$x_{i,t}$	input of the i-th node at time t
$\mathbf{b}$	bias vector
f_g	logistic sigmoid activation function of forget gates
f_i	tanh activation function of input gate
f_o	tanh activation function of output gate
i_t	activation of input gate at time t
o_t	activation of output gate at time t
f_t	activation of forget gate at time t
x_t	input of node at time t
h_t	output of hidden node at time t
h_t^f	output of forward hidden node at time t
h_t^b	output of backward hidden node at time t
c_t	cell value of memory block at time t

Evaluation Metrics

t_p	case is positive and predicted positive
t_n	case is negative and predicted negative
f_p	case is positive and predicted negative

f_n	case is negative and predicted positive
λ	weight
K	number of classes
S	number of substitutions
D	number of deletions
I	number of insertions
N	number of total words
X, Y	variables
σ_X	standard deviation of variables X
μ_X	mean of variables X
cov	covariance
$E[\cdot]$	expectation

Methodology for Data Enrichment

$\mathcal{D}$	database
q	number of databases
n	number of examples
k	number of classes
m	number of labellers
$\mathbf{x}$	feature vector
$\mathbb{R}^d$	d-dimensional space
y	preferred label
h	classifier
p_c	probability of chance-level
p_o	probability of observed labellers' agreement
κ	Kappa coefficient
A_i	binary annotation by the i-th labeller
$\mathcal{P}$	fused database
$\mathbf{x}'$	normalised feature vector

$\mathbf{x}_{min}$	minimum feature vector
$\mathbf{x}_{centre}$	central feature vector
$\mathbf{x}_{max}$	maximum feature vector
$\bar{\mathbf{x}}$	mean feature vector
$\bar{\sigma}$	standard deviation vector
$\mathcal{A}$	learning algorithm
r	number of learning rounds
$\mathcal{D}_{maj}$	data subset of majority class
$\mathcal{D}_{min}$	data subset of minority class
δ	random value
$E\{\mathbf{x}\}$	centre of a set of data
$d(\mathbf{x}, \mathbf{x}')$	distance of data sets $\mathbf{x}$ and $\mathbf{x}'$
P_D	percentage of discarded subset
P_S	percentage of selected subset
R_i	quantity ratio of examples belonging to class-i
l	human agreement level
$p(y \mid \mathbf{x})$	conditional probability of class y given input $\mathbf{x}$
$p(\mathbf{x}, y)$	joint probability distribution
$\mathcal{L}$	labelled data set
$\mathcal{L}_D$	labelled data set after class balancing
$\mathcal{U}$	unlabelled data set
$\mathcal{S}$	data subset selected per learning iteration
n'	number of selected instances per learning iteration
$\mathcal{H}$	acoustic model
$C_{h/m/l}$	high/medium/low confidence value
$\mathcal{X}$	feature space

Methodology for Data Optimisation

p	number of subvectors

$\mathbf{s}$	feature subvector
k	dimensionality of subvector
$\boldsymbol{v}$	codevector
$\boldsymbol{C}$	codebook
$\mathbf{d}_i^j$	Euclidean distance between $\mathbf{s}_i$ and $\boldsymbol{v}_i^j$
idx	index of codevector
$s(t)$	clean speech signal
$\hat{s}(t)$	distorted speech signal
$r(t)$	convolutional noise
$n(t)$	additive noise
$r_e(t)$	early reverberation
$r_l(t)$	late reverberation
$S(t,f)$	STDFT of speech signal
$R(t,f)$	STDFT of reverberation signal
τ	frame delay
$\mathcal{D}(\cdot)$	discrete cosine transform
$\mathbf{x}^d$	feature vector of distorted speech
$\mathbf{x}^c$	feature vector of clean speech
$\mathbf{x}^\Delta$	delta feature vector between clean and distorted speech
$\hat{\mathbf{x}}^c$	enhanced feature vector of distorted speech
$\hat{\mathbf{x}}^\Delta$	enhanced delta feature vector between clean and distorted speech

Bibliography

[1] D. Abercrombie, *Elements of General Phonetics.* Edinburgh, UK: Edinburgh University Press, 1967, vol. 203.

[2] M. E. Ayadi, M. S. Kamel, and F. Karray, "Survey on speech emotion recognition: Features, classification schemes, and databases," *Pattern Recognition*, vol. 44, no. 3, pp. 572–587, 2011.

[3] D. Ververidis and C. Kotropoulos, "A review of emotional speech databases," in *Proc. of Panhellenic Conference on Informatics (PCI)*, Thessaloniki, Greece, 2003, pp. 560–574.

[4] B. Schuller, S. Steidl, A. Batliner, F. Burkhardt, L. Devillers, C. Müller, and S. Narayanan, "Paralinguistics in speech and language – state-of-the-art and the challenge," *Computer Speech & Language*, vol. 27, no. 1, pp. 4–39, 2013.

[5] J. Liu, C. Chen, J. Bu, M. You, and J. Tao, "Speech emotion recognition using an enhanced co-training algorithm," in *Proc. of ICME*, Beijing, China, 2007, pp. 999–1002.

[6] A. Mahdhaoui and M. Chetouani, "A new approach for motherese detection using a semi-supervised algorithm," in *Proc. of IEEE International Workshop on MLSP.* Grenoble, France: IEEE, 2009, pp. 1–6.

[7] W. Han, Z. Zhang, J. Deng, M. Wöllmer, F. Weninger, and B. Schuller, "Towards distributed recognition of emotion in speech," in *Proc. of ISCCSP*, Rome, Italy, 2012, pp. 1–4.

[8] Z. Zhang, E. Coutinho, J. Deng, and B. Schuller, "Distributed recognition in computational paralinguistics," *IEEE Transactions on Affective Computing*, 2014, to appear.

[9] T. Vogt and E. André, "Comparing feature sets for acted and spontaneous speech in view of automatic emotion recognition," in *Proc. of ICME*, Amsterdam, Netherlands, 2005, pp. 474–477.

[10] J. Rong, G. Li, and Y.-P. P. Chen, "Acoustic feature selection for automatic emotion recognition from speech," *Information processing & management*, vol. 45, no. 3, pp. 315–328, 2009.

[11] D. Ververidis and C. Kotropoulos, "Fast and accurate sequential floating forward feature selection with the bayes classifier applied to speech emotion recognition," *Signal Processing*, vol. 88, no. 12, pp. 2956–2970, 2008.

[12] P. A. Naylor and N. D. Gaubitch, *Speech Dereverberation.* Berlin: Springer, 2010.

[13] D. Braha, *Data mining for design and manufacturing: methods and applications.* Kluwer academic publishers, 2001.

[14] B. Schuller, "The computational paralinguistics challenge," *IEEE Signal Processing Magazine*, vol. 29, no. 4, pp. 97–101, 2012.

[15] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[16] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A.-r. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[17] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[18] *Speech Processing, Transmission and Quality Aspects (STQ); Distributed Speech Recognition; Extended Advanced Front-End Feature Extraction Algorithm; Compression Algorithms; Back-End Speech Reconstruction Algorithm.* ETSI ES 202 212 V1.1.1, ETSI standard, 2003.

[19] J. Benesty, S. Makino, and J. Chen, *Speech Enhancement.* Berlin: Springer, 2005.

[20] A. J. Ferreira, "Combined spectral envelope normalization and subtraction of sinusoidal components in the ODFT and MDCT frequency domains," in *Proc. of WASPAA*, New Paltz, New York, 2001, pp. 51–54.

[21] A. Batliner, S. Steidl, D. Seppi, and B. Schuller, “Segmenting into adequate units for automatic recognition of emotion-related episodes: a speech-based approach,” *Advances in Human-Computer Interaction*, vol. 2010, pp. 3:1–3:15, 2010.

[22] F. Eyben, F. Weninger, S. Squartini, and B. Schuller, “Real-life voice activity detection with lstm recurrent neural networks and an application to hollywood movies,” in *Proc. of ICASSP*. Vancouver, Canada: IEEE, 2013, pp. 483–487.

[23] B. Schuller, B. Vlasenko, F. Eyben, M. Wöllmer, A. Stuhlsatz, A. Wendemuth, and G. Rigoll, “Cross-corpus acoustic emotion recognition: Variances and strategies,” *IEEE Transactions on Affective Computing*, vol. 1, no. 2, pp. 119–131, 2010.

[24] J. Wilting, E. Krahmer, and M. Swerts, “Real vs. acted emotional speech.” in *Proc. of INTERSPEECH*, Pittsburgh, PA, 2006, pp. 805–808.

[25] B. Schuller, B. Vlasenko, F. Eyben, G. Rigoll, and A. Wendemuth, “Acoustic emotion recognition: A benchmark comparison of performances,” in *Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding*, Merano, Italy, 2009, pp. 552–557.

[26] A. Batliner, K. Fischer, R. Huber, J. Spilker, and E. Nöth, “How to find trouble in communication,” *Speech communication*, vol. 40, no. 1, pp. 117–143, 2003.

[27] J. L. Flanagan, *Speech analysis: Synthesis and perception.* Berlin: Springer-Verlag, 1972.

[28] B. Schuller, Z. Zhang, F. Weninger, and F. Burkhardt, “Synthesized speech for model training in cross-corpus recognition of human emotion,” *International Journal of Speech Technology*, vol. 15, no. 3, pp. 313–323, 2012.

[29] M. Carbone, Y. Gal, S. Shieber, and B. Grosz, “Unifying annotated discourse hierarchies to create a gold standard,” in *Proc. of the 5th Sigdial Workshop on Discourse and Dialogue*, Boston, MA, 2004, 9 pages.

[30] B. J. Grosz and J. Hirschberg, “Some intonational characteristics of discourse structure.” in *Proc. of ICSLP*, Pittsburgh, PA, 1992.

[31] M. Grimm and K. Kroschel, “Evaluation of natural emotions using self assessment manikins,” in *Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding*, Cancun, Mexico, 2005, pp. 381–385.

[32] P. Ekman, “An argument for basic emotions,” *Cognition & Emotion*, vol. 6, no. 3-4, pp. 169–200, 1992.

[33] A. Metallinou, M. Wollmer, A. Katsamanis, F. Eyben, B. Schuller, and S. Narayanan, “Context-sensitive learning for enhanced audiovisual emotion classification,” *IEEE Transactions on Affective Computing*, vol. 3, no. 2, pp. 184–198, 2012.

[34] B. Schuller, S. Steidl, A. Batliner, F. Schiel, J. Krajewski, F. Weninger, and F. Eyben, “Medium-term speaker states – a review on intoxication, sleepiness and the first challenge,” *Computer Speech & Language*, vol. 28, no. 2, pp. 346–374, 2014.

[35] B. Schuller, R. Müller, F. Eyben, J. Gast, B. Hörnler, M. Wöllmer, G. Rigoll, A. Höthker, and H. Konosu, “Being bored? recognising natural interest by extensive audiovisual integration for real-life application,” *Image and Vision Computing Journal, Special Issue on Visual and Multimodal Analysis of Human Spontaneous Behavior*, vol. 27, no. 12, pp. 1760–1774, 2009.

[36] J. B. Hirschberg, S. Benus, J. M. Brenier, F. Enos, S. Friedman, S. Gilman, C. Girand, M. Graciarena, A. Kathol, L. Michaelis *et al.*, “Distinguishing deceptive from non-deceptive speech,” pp. 1833–1836, 2005.

[37] S. Kim, M. Filippone, F. Valente, and A. Vinciarelli, “Predicting the conflict level in television political debates: an approach based on crowdsourcing, nonverbal communication and gaussian processes,” in *Proc. of the 20th ACM international conference on Multimedia.* Nara, Japan: ACM, 2012, pp. 793–796.

[38] J. Krajewski, A. Batliner, and M. Golz, “Acoustic sleepiness detection – framework and validation of a speech adapted pattern recognition approach,” *Behavior Research Methods*, vol. 41, pp. 795–804, 2009.

[39] F. Schiel, C. Heinrich, and S. Barfüsser, “Alcohol language corpus: the first public corpus of alcoholized German speech,” *Language Resources and Evaluation*, vol. 46, no. 3, pp. 503–521, 2012.

[40] B. Schuller, F. Friedmann, and F. Eyben, “The munich biovoice corpus: Effects of physical exercising, heart rate, and skin conductance on human speech production,” p. to appear, 2014.

[41] T. F. Yap, “Speech production under cognitive load: Effects and classification,” Ph.D. dissertation, The University of New South Wales, Sydney, Australia, 2012.

[42] G. Mohammadi, A. Vinciarelli, and M. Mortillaro, “The voice of personality: Mapping nonverbal vocal behavior into trait attributions,” in *Proc. of the 2nd*

international workshop on Social signal processing. Florence, Italy: ACM, 2010, pp. 17–20.

[43] F. Burkhardt, B. Schuller, B. Weiss, and F. Weninger, "'would you buy a car from me?'-on the likability of telephone voices." in *INTERSPEECH*, Florence, Italy, 2011, pp. 1557–1560.

[44] L. Van Der Molen, M. van Rossum, A. Ackerstaff, L. Smeele, C. Rasch, and F. Hilgers, "Pretreatment organ function in patients with advanced head and neck cancer: clinical outcome measures and patients' views," *BMC Ear Nose and Throat Disorders*, vol. 9, no. 10, 2009.

[45] F. Ringeval, J. Demouy, G. Szaszák, M. Chetouani, L. Robel, J. Xavier, D. Cohen, and M. Plaza, "Automatic intonation recognition for the prosodic assessment of language-impaired children," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 5, pp. 1328–1342, 2011.

[46] A. Hanani, M. Russell, and M. J. Carey, "Speech-based identification of social groups in a single accent of British English by humans and computers," in *Proc. of ICASSP.* Prague, Czech Republic: IEEE, 2011, pp. 4876–4879.

[47] F. Burkhardt, M. Eckert, W. Johannsen, and J. Stegmann, "A database of age and gender annotated telephone speech," in *Proc. of the 10th International Conference of Language Resources and Evaluation (LREC)*, Valletta, Malta, 2010, pp. 1562–1565.

[48] N. V. Chawla, N. Japkowicz, and A. Kotcz, "Editorial: Special issue on learning from imbalanced data sets," *SIGKDD Explorations Newsletter*, vol. 6, no. 1, pp. 1–6, 2004.

[49] S. Li, Z. Wang, G. Zhou, and S. Y. M. Lee, "Semi-supervised learning for imbalanced sentiment classification," in *Proc. of IJCAI.* Barcelona, Spain: AAAI Press, 2011, pp. 1826–1831.

[50] G. E. Batista, R. C. Prati, and M. C. Monard, "A study of the behavior of several methods for balancing machine learning training data," *ACM Sigkdd Explorations Newsletter*, vol. 6, no. 1, pp. 20–29, 2004.

[51] K. R. Scherer, A. E. Schorr, and T. E. Johnstone, *Appraisal processes in emotion: Theory, methods, research.* Oxford University Press, 2001.

[52] I. Sneddon, M. McRorie, G. McKeown, and J. Hanratty, "The Belfast induced natural emotion database," *IEEE Transaction on Affective Computing*, vol. 3, no. 1, pp. 32–41, 2012.

[53] S. Steidl, *Automatic Classification of Emotion-Related User States in Spontaneous Children's Speech.* Berlin: Logos Verlag, 2009.

[54] P. Boersma, "Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound," in *Proc. of the institute of phonetic sciences*, vol. 17, no. 1193. Amsterdam, 1993, pp. 97–110.

[55] R. C. Snell and F. Milinazzo, "Formant location from LPC analysis data," *IEEE Transactions on Speech and Audio Processing*, vol. 1, no. 2, pp. 129–134, 1993.

[56] S. H. Nawab and T. F. Quatieri, "Short-time fourier transform," *Advanced topics in signal processing*, vol. 6, no. 2, pp. 289–337, 1988.

[57] C. K. Chui, Ed., *Wavelets: a tutorial in theory and applications*, San Diego, CA: Academic Press, 1992, vol. 1.

[58] J. Makhoul, "Linear prediction: A tutorial review," *Proceedings of the IEEE*, vol. 63, no. 4, pp. 561–580, 1975.

[59] H. Hermansky, "Perceptual linear predictive (PLP) analysis of speech," *The Journal of the Acoustical Society of America*, vol. 87, no. 4, pp. 1738–1752, 1990.

[60] T. L. Nwe, S. W. Foo, and L. C. De Silva, "Speech emotion recognition using hidden markov models," *Speech communication*, vol. 41, no. 4, pp. 603–623, 2003.

[61] B. Schuller, G. Rigoll, and M. Lang, "Hidden markov model-based speech emotion recognition," in *Proc. of ICASSP*, vol. 2. Hong Kong, China: IEEE, 2003, pp. 1–4.

[62] F. Eyben, M. Wöllmer, and B. Schuller, "openSMILE – the Munich versatile and fast open-source audio feature extractor," in *Proc. of ACM MM*, Florence, Italy, 2010, pp. 1459–1462.

[63] B. Schuller, S. Steidl, and A. Batliner, "The INTERSPEECH 2009 emotion challenge," in *Proc. of INTERSPEECH*, Brighton, UK, 2009, pp. 312–315.

[64] B. Schuller, S. Steidl, A. Batliner, F. Burkhardt, L. Devillers, C. Müller, and S. Narayanan, "The INTERSPEECH 2010 paralinguistic challenge," in *Proc. of INTERSPEECH*, Makuhari, Japan, 2010, pp. 2794–2797.

[65] B. Schuller, S. Steidl, A. Batliner, F. Schiel, and J. Krajewski, "The INTERSPEECH 2011 speaker state challenge," in *Proc. of INTERSPEECH*, Florence, Italy, 2011, pp. 3201–3204.

[66] B. Schuller, S. Steidl, A. Batliner, E. Nöth, A. Vinciarelli, F. Burkhardt, R. van Son, F. Weninger, F. Eyben, T. Bocklet, G. Mohammadi, and B. Weiss, "The INTERSPEECH 2012 speaker trait challenge," in *Proc. of INTERSPEECH*, Portland, OR, 2012, 4 pages.

[67] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *The Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[68] A. Jain and D. Zongker, "Feature selection: Evaluation, application, and small sample performance," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 2, pp. 153–158, 1997.

[69] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 8, pp. 1226–1238, 2005.

[70] S. Planet and I. Iriondo, "Comparative study on feature selection and fusion schemes for emotion recognition from speech," *International Journal of Interactive Multimedia and Artificial Intelligence, Special Issue on Intelligent Systems and Applications*, vol. 1, no. 6, pp. 44–51, 2012.

[71] A. Batliner, S. Steidl, B. Schuller, D. Seppi, T. Vogt, J. Wagner, L. Devillers, L. Vidrascu, V. Aharonson, L. Kessous *et al.*, "Whodunnit–searching for the most important feature types signalling emotion-related user states in speech," *Computer Speech & Language*, vol. 25, no. 1, pp. 4–28, 2011.

[72] I. T. Jolliffe, *Principal Component Analysis.* Berlin: Springer, 1986.

[73] Z. Fan, Y. Xu, and D. Zhang, "Local linear discriminant analysis framework using sample neighbors," *IEEE Transactions on Neural Networks*, vol. 22, no. 7, pp. 1119–1132, 2011.

[74] A. Gersho and R. M. Gray, *Vector Quantization and Signal Compression.* Boston, MA: Kluwer Academic Publishers, 1992, vol. 159.

[75] J. Arrowood and M. Clements, "Extended cluster information vector quantization (ECI-VQ) for robust classification," in *Proc. of ICASSP*, Montreal, Canada, 2004, pp. 889–892.

[76] C. Wan and L. Lee, "Joint Uncertainty Decoding (JUD) with Histogram-Based Quantization (HQ) for robust and/or distributed speech recognition," in *Proc. of ICASSP*, Toulouse, France, 2006, pp. 125–128.

[77] A. Vasuki and P. Vanathi, "A review of vector quantization techniques," *IEEE Potentials*, vol. 25, no. 4, pp. 39–47, 2006.

[78] V. Digalakis, L. G. Neumeyer, and M. Perakakis, “Quantization of cepstral parameters for speech recognition over the world wide web,” *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 1, pp. 82–90, 1999.

[79] T. Virtanen, R. Singh, and B. Raj, *Techniques for Noise Robustness in Automatic Speech Recognition*. New York, NY: John Wiley & Sons, 2012.

[80] M. Miyoshi and Y. Kaneda, “Inverse filtering of room acoustics,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 36, no. 2, pp. 145–152, 1988.

[81] T. Nakatani, T. Yoshioka, K. Kinoshita, M. Miyoshi, and B.-H. Juang, “Speech dereverberation based on variance-normalized delayed linear prediction,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1717–1731, 2010.

[82] K. Kumar and R. Stern, “Maximum-likelihood-based cepstral inverse filtering for blind speech dereverberation,” in *Proc. of ICASSP*, Dallas, TX, 2010, pp. 4282–4285.

[83] H. Kameoka, T. Nakatani, and T. Yoshioka, “Robust speech dereverberation based on non-negativity and sparse nature of speech spectrograms,” in *Proc. of ICASSP*, Taipei, Taiwan, 2009, pp. 45–48.

[84] K. Lebart, J.-M. Boucher, and P. Denbigh, “A new method based on spectral subtraction for speech dereverberation,” *Acta Acustica united with Acustica*, vol. 87, no. 3, pp. 359–366, 2001.

[85] M. Seltzer, B. Raj, and R. Stern, “Likelihood-maximizing beamforming for robust hands-free speech recognition,” *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 5, pp. 489–498, 2004.

[86] E. Habets and J. Benesty, “A two-stage beamforming approach for noise reduction and dereverberation,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 5, pp. 945–958, 2013.

[87] S. Furui, “Cepstral analysis technique for automatic speaker verification,” *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 29, no. 2, pp. 254–272, 1981.

[88] D. Gelbart and N. Morgan, “Evaluating long-term spectral subtraction for reverberant ASR,” in *Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding*, Madonna di Campiglio, Italy, 2001, pp. 103–106.

[89] H. Hermansky and N. Morgan, “RASTA processing of speech,” *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 4, pp. 578–589, 1994.

[90] K. Kumar, "A spectro-temporal framework for compensation of reverberation for speech recognition," Ph.D. dissertation, Carnegie Mellon University, 2011.

[91] J.-L. Gauvain and C.-H. Lee, "Maximum a posteriori estimation for multivariate gaussian mixture observations of markov chains," *IEEE Transactions on Speech and audio processing*, vol. 2, no. 2, pp. 291–298, 1994.

[92] C. J. Leggetter and P. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden Markov models," *Computer Speech & Language*, vol. 9, no. 2, pp. 171–185, 1995.

[93] M. J. Gales, "Maximum likelihood linear transformations for HMM-based speech recognition," *Computer Speech & Language*, vol. 12, no. 2, pp. 75–98, 1998.

[94] H.-G. Hirsch and H. Finster, "A new approach for the adaptation of HMMs to reverberation and background noise," *Speech Communication*, vol. 50, no. 3, pp. 244–263, 2008.

[95] F. Eyben, F. Weninger, and B. Schuller, "Affect recognition in real-life acoustic conditions – a new perspective on feature selection." in *Proc. of INTERSPEECH*, vol. 2013, Lyon, France, 2013.

[96] J. Li, L. Deng, Y. Gong, and R. Haeb-Umbach, "An overview of noise-robust automatic speech recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 4, pp. 745–777, 2014.

[97] J. C. Platt, "Fast training of support vector machines using sequential minimal optimization," in *Advances in Kernel Methods: Support Vector Learning.* Cambridge, MA: MIT press, 1999, pp. 185–208.

[98] B. E. Boser, I. M. Guyon, and V. N. Vapnik, "A training algorithm for optimal margin classifiers," in *Proc. of the fifth annual workshop on Computational Learning Theory.* Montreal, Canada: ACM, 1992, pp. 144–152.

[99] C.-W. Hsu and C.-J. Lin, "A comparison of methods for multiclass support vector machines," *IEEE Transactions on Neural Networks*, vol. 13, no. 2, pp. 415–425, 2002.

[100] J. R. Quinlan, "Induction of decision trees," *Machine learning*, vol. 1, no. 1, pp. 81–106, 1986.

[101] ——, *C4.5: programs for machine learning.* Morgan kaufmann, 1993, vol. 1.

[102] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and regression trees.* Monterey, CA: Wadsworth & Brooks, 1984.

[103] L. Breiman, “Random forests,” *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[104] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning internal representations by error propagation,” DTIC, Tech. Rep., 1985.

[105] P. J. Werbos, “Backpropagation through time: what it does and how to do it,” *Proceedings of the IEEE*, vol. 78, no. 10, pp. 1550–1560, 1990.

[106] R. A. Jacobs, “Increased rates of convergence through learning rate adaptation,” *Neural networks*, vol. 1, no. 4, pp. 295–307, 1988.

[107] J. Moody, S. Hanson, A. Krogh, and J. A. Hertz, “A simple weight decay can improve generalization,” *Advances in neural information processing systems*, vol. 4, pp. 950–957, 1995.

[108] R. Reed, “Pruning algorithms – a survey,” *IEEE Transactions on Neural Networks*, vol. 4, no. 5, pp. 740–747, 1993.

[109] C. M. Bishop *et al.*, “Neural networks for pattern recognition,” 1995.

[110] J. L. Elman, “Finding structure in time,” *Cognitive science*, vol. 14, no. 2, pp. 179–211, 1990.

[111] R. J. Williams and D. Zipser, *Gradient-based learning algorithms for recurrent networks and their computational complexity*, 1995.

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

[113] T. G. Dietterich, “Machine learning for sequential data: A review,” in *Structural, syntactic, and statistical pattern recognition.* Springer, 2002, pp. 15–30.

[114] N. Morgan and H. Bourlard, “Continuous speech recognition using multilayer perceptrons with hidden markov models,” in *Proc. of ICASSP.* Albuquerque, NM: IEEE, 1990, pp. 413–416.

[115] O. Vinyals, S. V. Ravuri, and D. Povey, “Revisiting recurrent neural networks for robust ASR,” in *Proc. of ICASSP*, Kyoto, Japan, 2012, pp. 4085–4088.

[116] S. Hochreiter, Y. Bengio, P. Frasconi, and J. Schmidhuber, “Gradient flow in recurrent nets: The difficulty of learning long-term dependencies,” in *A Field Guide to Dynamical Recurrent Neural Network*, S. C. Kremer and J. F. Kolen, Eds. New York, NY: IEEE Press, 2001, pp. 1–15.

[117] A. Graves and J. Schmidhuber, “Framewise phoneme classification with bidirectional LSTM and other neural network architectures,” *Neural Networks*, vol. 18, no. 5, pp. 602–610, 2005.

[118] A. Graves, *Supervised sequence labelling with recurrent neural networks.* Springer, 2012, vol. 385.

[119] E. Douglas-Cowie, N. Campbell, R. Cowie, and P. Roach, “Emotional speech: Towards a new generation of databases,” *Speech Communication*, vol. 40, no. 12, pp. 33–60, 2003.

[120] U. M. Fayyad, G. Piatetsky-Shapiro, P. Smyth, and R. Uthurusamy, *Advances in knowledge discovery and data mining.* Cambridge, MA: The MIT Press, 1996.

[121] L. Hansen and P. Salamon, “Neural network ensembles,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 10, pp. 993–1001, 1990.

[122] A. Folleco, T. Khoshgoftaar, and A. Napolitano, “Comparison of four performance metrics for evaluating sampling techniques for low quality class-imbalanced data,” in *Proc. of ICMLA*, 2008, pp. 153–158.

[123] D. LaLoudouana, M. B. Tarare, L. T. Center, and G. Selacie, “Data set selection,” *Journal of Machine Learning Gossip*, vol. 1, pp. 11–19, 2003.

[124] J. Han and M. Kamber, *Data Mining, Southeast Asia Edition: Concepts and Techniques.* Morgan kaufmann, 2006.

[125] L. A. Feldman, “Valence focus and arousal focus: Individual differences in the structure of affective experience.” *Journal of personality and social psychology*, vol. 69, no. 1, p. 153, 1995.

[126] L. F. Barrett, “Discrete emotions or dimensions? the role of valence focus and arousal focus,” *Cognition & Emotion*, vol. 12, no. 4, pp. 579–599, 1998.

[127] B. Schuller, “Multimodal affect databases – collection, challenges & chances,” in *Handbook of Affective Computing*, R. A. Calvo, S. DMello, J. Gratch, and A. Kappas, Eds. Oxford, UK: Oxford University Press, 2013.

[128] C. Spearman, “The proof and measurement of association between two things,” *The American journal of psychology*, vol. 15, no. 1, pp. 72–101, 1904.

[129] K. Pearson, “Note on regression and inheritance in the case of two parents,” *Proc. of the Royal Society of London*, vol. 58, no. 347-352, pp. 240–242, 1895.

[130] W. A. Scott, "Reliability of content analysis: The case of nominal scale coding." *Public opinion quarterly*, 1955.

[131] J. Cohen, "Weighted kappa: Nominal scale agreement provision for scaled disagreement or partial credit." *Psychological bulletin*, vol. 70, no. 4, p. 213, 1968.

[132] J. L. Fleiss, "Measuring nominal scale agreement among many raters." *Psychological bulletin*, vol. 76, no. 5, p. 378, 1971.

[133] J. Sola and J. Sevilla, "Importance of input data normalization for the application of neural networks to complex industrial problems," *IEEE Transactions on Nuclear Science*, vol. 44, no. 3, pp. 1464–1468, 1997.

[134] Y. A. LeCun, L. Bottou, G. B. Orr, and K.-R. Müller, "Efficient backprop," in *Neural networks: Tricks of the Trade.* Berlin: Springer, 2012, pp. 9–48.

[135] R. E. Schapire, "The strength of weak learnability," *Machine learning*, vol. 5, no. 2, pp. 197–227, 1990.

[136] Y. Freund and R. E. Schapire, "A decision-theoretic generalization of on-line learning and an application to boosting," *Journal of computer and system sciences*, vol. 55, no. 1, pp. 119–139, 1997.

[137] L. Breiman, "Bagging predictors," *Machine learning*, vol. 24, no. 2, pp. 123–140, 1996.

[138] D. H. Wolpert, "Stacked generalization," *Neural networks*, vol. 5, no. 2, pp. 241–259, 1992.

[139] B. Parmanto, P. W. Munro, and H. R. Doyle, "Improving committee diagnosis with resampling techniques," in *Advances in neural information processing systems*, D. S. Touretzky, M. C. Mozer, and M. E. Hesselmo, Eds. Cambridge, MA: MIT Press, 1995, pp. 882–888.

[140] Y. Freund, "Boosting a weak learning algorithm by majority," *Information and computation*, vol. 121, no. 2, pp. 256–285, 1995.

[141] E. Bauer and R. Kohavi, "An empirical comparison of voting classification algorithms: Bagging, boosting, and variants," *Machine learning*, vol. 36, no. 1-2, pp. 105–139, 1999.

[142] Z.-H. Zhou, J. Wu, and W. Tang, "Ensembling neural networks: many could be better than all," *Artificial intelligence*, vol. 137, no. 1, pp. 239–263, 2002.

[143] T. Ditterrich, “Machine learning research: four current directions,” *Artificial Intelligence Magazine*, vol. 4, pp. 97–136, 1997.

[144] N. Japkowicz *et al.*, “Learning from imbalanced data sets: a comparison of various strategies,” in *Proc. of AAAI workshop on learning from imbalanced data sets*, vol. 68. Austin, Texas: Menlo Park, CA, 2000.

[145] H. He and E. A. Garcia, “Learning from imbalanced data,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263–1284, 2009.

[146] G. M. Weiss and F. Provost, “The effect of class distribution on classifier learning: an empirical study,” Department of Computer Science, Rutgers University, New Brunswick, NJ, Tech. Rep. ML-TR-43, 2001.

[147] A. Estabrooks, T. Jo, and N. Japkowicz, “A multiple resampling method for learning from imbalanced data sets,” *Computational Intelligence*, vol. 20, no. 1, pp. 18–36, 2004.

[148] N. Chawla, K. Bowyer, L. Hall, and W. Kegelmeyer, “SMOTE: Synthetic minority over-sampling technique,” *Journal of Artificial Intelligence Research*, vol. 16, pp. 321–357, 2002.

[149] K. M. Ting, “An instance-weighting method to induce cost-sensitive trees,” *IEEE Transactions on Knowledge and Data Engineering*, vol. 14, no. 3, pp. 659–665, 2002.

[150] G. Wu and E. Y. Chang, “Class-boundary alignment for imbalanced dataset learning,” in *Proc. of ICML 2003 workshop on learning from imbalanced data sets II*, Washington, DC, 2003, pp. 49–56.

[151] R. Akbani, S. Kwek, and N. Japkowicz, “Applying support vector machines to imbalanced datasets,” in *Lecture Notes in Computer Science.* Berlin: Springer, 2004, vol. 3201, pp. 39–50.

[152] H. Liu and H. Motoda, “On issues of instance selection,” *Data Mining and Knowledge Discovery*, vol. 6, no. 2, pp. 115–130, 2002.

[153] J. A. Olvera-López, J. A. Carrasco-Ochoa, J. F. Martínez-Trinidad, and J. Kittler, “A review of instance selection methods,” *Artificial Intelligence Review*, vol. 34, no. 2, pp. 133–143, 2010.

[154] T. Cover and P. Hart, “Nearest neighbor pattern classification,” *IEEE Transactions on Information Theory*, vol. 13, no. 1, pp. 21–27, 1967.

[155] P. Hart, “The condensed nearest neighbor rule,” *IEEE Transactions on Information Theory*, vol. 14, no. 3, pp. 515–516, 1968.

[156] G. Ritter, H. Woodruff, S. Lowry, and T. Isenhour, “An algorithm for a selective nearest neighbor decision rule,” *IEEE Transactions on Information Theory*, vol. 21, no. 6, pp. 665–669, 1975.

[157] D. R. Wilson and T. R. Martinez, “Reduction techniques for instance-based learning algorithms,” *Machine learning*, vol. 38, no. 3, pp. 257–286, 2000.

[158] M. A. Fischler and R. C. Bolles, “Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography,” *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[159] J. C. Riquelme, J. S. Aguilar-Ruiz, and M. Toro, “Finding representative patterns with ordered projections,” *Pattern Recognition*, vol. 36, no. 4, pp. 1009–1018, 2003.

[160] C. E. Erdem, E. Bozkurt, E. Erzin, and A. T. Erdem, “Ransac-based training data selection for emotion recognition from spontaneous speech,” in *Proc. the 3rd international workshop on Affective interaction in natural environments*, New York, NY, 2010, pp. 9–14.

[161] M. Li and I. Sethi, “Confidence-based active learning,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 8, pp. 1251–1261, 2006.

[162] Q. Tao, G.-W. Wu, F.-Y. Wang, and J. Wang, “Posterior probability support vector machines for unbalanced data,” *IEEE Transactions on Neural Networks*, vol. 16, no. 6, pp. 1561–1573, 2005.

[163] J. Platt, “Probabilistic outputs for support vector machines and comparisons to regularized likelihood methods,” in *Advances in large margin classifiers*, A. Smola, P. Bartlett, B. Schölkopf, and D. Schuurmans, Eds. Cambridge, MA: MIT Press, 1999, pp. 61–74.

[164] R. Duda, P. Hart, and D. Stork, *Pattern Classification*, 2nd ed. New York, NY: John Wiley, 2001.

[165] B. Zadrozny and C. Elkan, “Transforming classifier scores into accurate multiclass probability estimates,” in *Proc. of the eighth ACM SIGKDD.* Edmonton, Canada: ACM, 2002, pp. 694–699.

[166] X. Zhu, “Semi-supervised learning literature survey,” Department of Computer Sciences, University of Wisconsin at Madison, Madison, WI, Tech. Rep. TR 1530, 2006.

[167] J.-T. Huang and M. Hasegawa-Johnson, "On semi-supervised learning of Gaussian mixture models for phonetic classification," in *Proc. of the NAACL HLT 2009 Workshop on Semi-Supervised Learning for Natural Language Processing.* Boulder, CO: Association for Computational Linguistics, 2009, pp. 75–83.

[168] G. Druck, C. Pal, A. McCallum, and X. Zhu, "Semi-supervised classification with hybrid generative/discriminative methods," in *Proc. of the 13th ACM SIGKDD.* San Jose, CA: ACM, 2007, pp. 280–289.

[169] B. Settles, "Active learning literature survey," Department of Computer Sciences, University of Wisconsin–Madison, Wisconsin, WI, Tech. Rep., 2009.

[170] V. Vapnik, *The nature of statistical learning theory*, 2nd ed. Berlin: springer, 1999.

[171] B. Kulis, S. Basu, I. Dhillon, and R. Mooney, "Semi-supervised graph clustering: a kernel approach," *Machine Learning*, vol. 74, no. 1, pp. 1–22, 2009.

[172] I. Muslea, S. Minton, and C. Knoblock, "Active + semi-supervised learning = robust multi-view learning," in *Proc. of ICML*, Sydney, Australia, 2002, pp. 435–442.

[173] X. Cui, J. Huang, and J.-T. Chien, "Multi-view and multi-objective semi-supervised learning for HMM-based automatic speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 7, pp. 1923 –1935, 2012.

[174] A. Blum and T. Mitchell, "Combining labeled and unlabeled data with co-training," in *Proc. of 11th annual conference on Computational Learning Theory*, Madison, WI, 1998, pp. 92–100.

[175] W. Wang and Z.-H. Zhou, "A new analysis of co-training," in *Proc. of ICML*, Haifa, Israel, 2010, pp. 1135–1142.

[176] J. Du, C. X. Ling, and Z. Zhou, "When does co-training work in real data?" *IEEE Transactions on Knowledge Discovery and Data Mining*, vol. 23, no. 5, pp. 788–799, 2011.

[177] J. Zhu, H. Wang, B. K. Tsou, and M. Ma, "Active learning with sampling by uncertainty and density for data annotations," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 6, pp. 1323 –1331, 2010.

[178] R. Liere and P. Tadepalli, "Active learning with committees for text categorization," in *Proc. of AAAI/IAAI*, Providence, RI, 1997, pp. 591–596.

[179] N. Roy and A. McCallum, “Toward optimal active learning through sampling estimation of error reduction,” in *Proc. of ICML*, Williamstown, MA, 2001, pp. 441–448.

[180] B. Settles and M. Craven, “An analysis of active learning strategies for sequence labeling tasks,” in *Proc. of Empirical Methods in Natural Language Processing (EMNLP)*, Honolulu, HY, 2008, pp. 1070–1079.

[181] Z. Xu, R. Akella, and Y. Zhang, “Incorporating diversity and density in active learning for relevance feedback,” in *Proc. of European Conference on Information Retrieval (ECIR)*, Rome, Italy, 2007, pp. 246–257.

[182] M. You, C. Chen, J. Bu, J. Liu, and J. Tao, “Emotion recognition from noisy speech,” in *Proc. of ICME.* Toronto, Canada: IEEE, 2006, pp. 1653–1656.

[183] Z. Zhang, F. Eyben, J. Deng, and B. Schuller, “An agreement and sparseness-based learning instance selection and its application to subjective speech phenomena,” in *Proc. of 5th International Workshop on Emotion Social Signals, Sentiment & Linked Open Data, satellite of LREC 2014*, Reykjavik, Iceland, 2014, pp. 21–26.

[184] A. McCallum and K. Nigam, “Employing EM in pool-based active learning for text classification,” in *Proc. of ICML*, Madison, WI, 1998, pp. 359–367.

[185] G. Tur, D. Hakkani-Tür, and R. E. Schapire, “Combining active and semi-supervised learning for spoken language understanding,” *Speech Communication*, vol. 45, no. 2, pp. 171–186, 2005.

[186] X. Zhu, J. Lafferty, and Z. Ghahramani, “Combining active learning and semi-supervised learning using Gaussian fields and harmonic functions,” in *Proc. of 20th International Conference on Machine Learning (ICML) Workshop on The Continuum from Labelled to Unlabelled Data*, Washington DC, 2003, pp. 58–65.

[187] W. Wang and Z. Zhou, “On multi-view active learning and the combination with semi-supervised learning,” in *Proc. of ICML*, Helsinki, Finland, 2008, pp. 1152–1159.

[188] A. Arimond, “A distributed system for pattern recognition and machine learning,” M. Eng. thesis, TU Kaiserslautern & DFKI, Kaiserslautern, Germany, 2010.

[189] Z. Tan and I. Varga, “Network, distributed and embedded speech recognition: An overview,” in *Automatic Speech Recognition on Mobile Devices and over Communication Networks*, Z.-H. Tan and B. Lindberg, Eds. London, UK: Springer, 2008, pp. 1–23.

[190] A. Gomez, A. Peinado, V. Sanchez, and A. Rubio, “Combining media-specific FEC and error concealment for robust distributedspeech recognition over loss-prone packet channels,” *IEEE Transactions on Multimedia*, vol. 8, no. 6, pp. 1228–1238, 2006.

[191] C. Wan and L. Lee, “Histogram-based quantization for robust and/or distributed speech recognition,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 4, pp. 859–873, 2008.

[192] R. Flynn and E. Jones, “Robust distributed speech recognition using speech enhancement,” *IEEE Transactions on Consumer Electronics*, vol. 54, no. 3, pp. 1267–1273, 2008.

[193] G. Nagy, “Interactive, mobile, distributed pattern recognition,” in *Proc. of ICIAP*, Cagliari, Italy, 2005, pp. 37–49.

[194] A. Yang, R. Jafari, S. Sastry, and R. Bajcsy, “Distributed recognition of human actions using wearable motion sensor networks,” *Journal of Ambient Intelligence and Smart Environments*, vol. 1, no. 2, pp. 103–115, 2009.

[195] A. Batliner and B. Schuller, “More than fifty years of speech processing – the rise of computational paralinguistics and ethical demands,” in *Proc. of ETHICOMP*, Paris, France, 2014, no pagination.

[196] T. S. Rappaport, *Wireless communications: principles and practice.* New Jersey: Prentice Hall PTR, 1996, vol. 2.

[197] D. Chazan, R. Hoory, G. Cohen, and M. Zibulski, “Speech reconstruction from mel frequency cepstral coefficients and pitch frequency,” in *Proc. of ICASSP*, vol. 3. IEEE, 2000, pp. 1299–1302.

[198] B. Milner and X. Shao, “Clean speech reconstruction from MFCC vectors and fundamental frequency using an integrated front-end,” *Speech Communication*, vol. 48, no. 6, pp. 697–715, 2006.

[199] J. Makhoul, S. Roucos, and H. Gish, “Vector quantization in speech coding,” *Proc. of the IEEE*, vol. 73, no. 11, pp. 1551–1588, 1985.

[200] T. Yoshioka, X. Chen, and M. J. Gales, “Impact of single-microphone dereverberation on dnn-based meeting transcription systems,” in *Proc. of ICASSP*, Florence, Italy, 2014, pp. 5527–5531.

[201] M. Seltzer, D. Yu, and Y. Wang, “An investigation of deep neural networks for noise robust speech recognition,” in *Proc. of ICASSP*, Vancouver, Canada, 2013, pp. 7398–7402.

[202] W. Li, J. Dines, and M. Magimai.-Doss, “Robust overlapping speech recognition based on neural networks,” Martigny, Switzerland, Tech. Rep. IDIAP-RR-55-2007, 2007.

[203] A. L. Maas, T. M. O’Neil, A. Y. Hannun, and A. Y. Ng, “Recurrent neural network feature enhancement: The 2nd CHiME challenge,” in *Proc. of CHiME Workshop*, Vancouver, Canada, 2013, pp. 79–80.

[204] E. Vincent, J. Barker, S. Watanabe, J. Le Roux, F. Nesta, and M. Matassoni, “The second ‘CHiME’ speech separation and recognition challenge: Datasets, tasks and baselines,” in *Proc. of ICASSP*, 2013, pp. 126–130.

[205] A. Graves, M. Liwicki, S. Fernandez, H. Bertolami, R. andBunke, and J. Schmidhuber, “A novel connectionist system for unconstrained handwriting recognition,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, pp. 855–868, 2009.

[206] C. Plahl, M. Kozielski, R. Schlüter, and H. Ney, “Feature combination and stacking of recurrent and non-recurrent neural networks for LVCSR,” in *Proc. of ICASSP*, Vancouver, Canada, 2013, pp. 6714–6718.

[207] M. Wöllmer, C. Blaschke, T. Schindl, B. Schuller, B. Farber, S. Mayer, and B. Trefflich, “Online driver distraction detection using long short-term memory,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 12, no. 2, pp. 574–582, 2011.

[208] M. Wöllmer, B. Schuller, and G. Rigoll, “Keyword spotting exploiting long short-term memory,” *Speech Communication*, vol. 55, no. 2, pp. 252–265, 2013.

[209] H. Sak, A. Senior, and F. Beaufays, “Long short-term memory based recurrent neural network architectures for large vocabulary speech recognition,” *arXiv preprint arXiv:1402.1128*, 2014.

[210] M. Wöllmer, Z. Zhang, F. Weninger, B. Schuller, , and G. Rigoll, “Feature enhancement by bidirectional LSTM networks for conversational speech recognition in highly non-stationary noise,” in *Proc. of ICASSP*, Vancouver, Canada, 2013, pp. 6822–6826.

[211] F. Weninger, J. Geiger, M. Wöllmer, B. Schuller, and G. Rigoll, “The Munich feature enhancement approach to the 2nd CHiME challenge using BLSTM recurrent neural networks,” in *Proc. of CHiME Workshop*, Vancouver, Canada, 2013, pp. 86–90.

[212] Z. Zhang, J. Pinto, C. Plahl, B. Schuller, and D. Willett, “Channel mapping using bidirectional long short-term memory for dereverberation in hand-free

voice controlled devices," *IEEE Transactions on Consumer Electronics*, 2014, to appear.

[213] T. Yoshioka, A. Sehr, M. Delcroix, K. Kinoshita, R. Maas, T. Nakatani, and W. Kellermann, "Making machines understand us in reverberant rooms: Robustness against reverberation for automatic speech recognition," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 114–126, 2012.

[214] Q. Jin, T. Schultz, and A. Waibel, "Far-field speaker recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 2023–2032, 2007.

[215] R. Cowie, E. Douglas-Cowie, N. Tsapatsoulis, G. Votsis, S. Kollias, W. Fellenz, and J. G. Taylor, "Emotion recognition in human-computer interaction," *IEEE Signal Processing Magazine*, vol. 18, no. 1, pp. 32–80, 2001.

[216] B. Schuller, A. Batliner, S. Steidl, and D. Seppi, "Recognising realistic emotions and affect in speech: State of the art and lessons learnt from the first challenge," *Speech Communication*, vol. 53, no. 9, pp. 1062–1087, 2011.

[217] B. Schuller, Z. Zhang, F. Weninger, and G. Rigoll, "Using multiple databases for training in emotion recognition: To unite or to vote?" in *Proc. of INTERSPEECH*, Florence, Italy, 2011, pp. 1553–1556.

[218] D. D. Lewis, "Naive (bayes) at forty: The independence assumption in information retrieval," in *Machine learning: ECML-98.* Berlin Heidelberg: Springer, 1998, pp. 4–15.

[219] K. Lee and M. Slaney, "Automatic chord recognition from audio using a supervised hmm trained with audio-from-symbolic data," in *Proc. of the 1st ACM workshop on Audio and music computing multimedia table of contents.* Santa Barbara, CA: ACM, 2006, pp. 11 – 20.

[220] B. Schuller and F. Burkhardt, "Learning with synthesized speech for automatic emotion recognition," in *Proc. of ICASSP*, Dallas, Texas, 2010, pp. 5150–5153.

[221] B. Schuller, Z. Zhang, F. Weninger, and G. Rigoll, "Selecting training data for cross-corpus speech emotion recognition: Prototypicality vs. generalization," in *Proc. of 2011 Speech Processing Conference*, Tel Aviv, Israel, 2011, 4 pages.

[222] A. Angelova, Y. Abu-Mostafa, and P. Perona, "Pruning training sets for learning of object categories," in *Proc. of CVPR*, San Diego, CA, 2005, pp. 494–501.

[223] C. E. Erdem, E. Bozkurt, E. Erzin, and A. T. Erdem, "RANSAC-based training data selection for emotion recognition from spontaneous speech," in *Proc.*

of the 3rd international workshop on Affective Interaction in Natural Environments (AFFINE), Firenze, Italy, 2010, pp. 9–14.

[224] F. Eyben, A. Batliner, B. Schuller, D. Seppi, and S. Steidl, “Cross-corpus classification of realistic emotions – some pilot experiments,” in *Proc. 3rd International Workshop on EMOTION (satellite of LREC): Corpora for Research on Emotion and Affect.* Valletta, Malta: ELRA, 2010, pp. 77–82.

[225] A. N. Angelova, “Data pruning,” Ph.D. dissertation, California Institute of Technology, Pasadena, CA, 2004.

[226] Z. Zhang, F. Weninger, M. Wöllmer, and B. Schuller, “Unsupervised learning in cross-corpus acoustic emotion recognition,” in *Proc. of IEEE workshop on Automatic Speech Recognition and Understanding*, Big Island, HY, 2011, pp. 523–528.

[227] G. Tur and A. Stolcke, “Unsupervised language model adaptation for meeting recognition,” in *Proc. of ICASSP*, Honolulu, HY, 2007, pp. 173–176.

[228] V. Frinken, A. Fischer, H. Bunke, and A. Foornes, “Co-training for handwritten word recognition,” in *Proc. of 2011 Document Analysis and Recognition (ICDAR)*, Beijing, China, 2011, pp. 314–318.

[229] L. Zao, D. Cavalcante, and R. Coelho, “Time-frequency feature and AMS-GMM mask for acoustic emotion classification,” *IEEE Signal Processing Letters*, vol. 21, no. 5, pp. 620–624, 2014.

[230] J. Hansen and S. Bou-Ghazale, “Getting started with SUSAS: A speech under simulated and actual stress database,” in *Proc. of EUROSPEECH*, Rhodes, Greece, 1997, pp. 1743–1746.

[231] Z. Zhang, E. Coutinho, J. Deng, and B. Schuller, “Cooperative learning and its application to emotion recognition from speech,” *IEEE Transactions on Audio, Speech, and Language Processing*, 2014, in peer review.

[232] Z. Zhang, J. Deng, and B. Schuller, “Co-training succeeds in computational paralinguistics,” in *Proc. of ICASSP*, Vancouver, Canada, 2013, pp. 8505–8509.

[233] X. Zhu and A. B. Goldberg, “Introduction to semi-supervised learning,” *Synthesis lectures on artificial intelligence and machine learning*, vol. 3, no. 1, pp. 1–130, 2009.

[234] Z. Zhang and B. Schuller, “Active learning by sparse instance tracking and classifier confidence in acoustic emotion recognition,” in *Proc. of INTERSPEECH*, Portland, OR, 2012, 4 pages.

[235] Z. Zhang, J. Deng, E. Marchi, and B. Schuller, "Active learning by label uncertainty for acoustic emotion recognition," in *Proc. of INTERSPEECH*, Lyon, France, 2013, pp. 2856–2860.

[236] S. Salzberg, "On comparing classifiers: Pitfalls to avoid and a recommended approach," *Data mining and knowledge discovery*, vol. 1, no. 3, pp. 317–328, 1997.

[237] K. Huang and S. Aviyente, "Sparse representation for signal classification," in *Proc. of NIPS*, Vancouver, Canada, 2006, pp. 609–616.

[238] B. Schuller, M. Wöllmer, F. Eyben, G. Rigoll, and D. Arsic, "Semantic speech tagging: Towards combined analysis of speaker traits," in *Proc. of AES 42nd International Conference*, Ilmenau, Germany, 2011, pp. 89–97.

[239] D. Ververidis and C. Kotropoulos, "Automatic speech classification to five emotional states based on gender information," in *Proc. of European Signal Processing Conference (EUSIPCO)*, Vienna, Austria, 2004, pp. 341–344.

[240] R. Caruana, "Multitask learning," *Machine learning*, vol. 28, no. 1, pp. 41–75, 1997.

[241] R. Reisenzein and H. Weber, "Personality and emotion," in *The Cambridge handbook of personality psychology.* Cambridge, UK: Cambridge University Press, 2009, pp. 54–71.

[242] A. Batliner, C. Hacker, S. Steidl, E. Nöth, S. D'Arcy, M. Russell, and M. Wong, ""you stupid tin box" – children interacting with the AIBO robot:a cross-linguistic emotional speech corpus," in *Proc. of the 4th International Conference of Language Resources and Evaluation (LREC)*, Lisbon, Portugal, 2004, pp. 171–174.

[243] T. Polzehl, S. Sundaram, H. Ketabdar, M. Wagner, and F. Metze, "Emotion classification in children's speech using fusion of acoustic and linguistic features," in *Proc. of INTERSPEECH*, Brighton, UK, 2009, pp. 340–343.

[244] B. Schuller, M. Wimmer, D. Arsic, G. Rigoll, and B. Radig, "Audiovisual behavior modeling by combined feature spaces," in *Proc. of ICASSP*, vol. II, Honolulu, HY, 2007, pp. 733–736.

[245] F. Burkhardt, A. Paeschke, M. Rolfes, W. Sendlmeier, and B. Weiss, "A database of German emotional speech," in *Proc. of INTERSPEECH*, Lisbon, Portugal, 2005, pp. 1517–1520.

[246] I. S. Engberg and A. V. Hansen, "Documentation of the Danish emotional speech database DES," Center for Person Kommunikation, Aalborg University, Denmark, Tech. Rep., 1996.

[247] O. Martin, I. Kostsia, B. Macq, and I. Pitas, "The eNTERFACE'05 audio-visual emotion database," in *Proc. of IEEE Workshop on Multimedia Database Management*, Atlanta, GA, 2006.

[248] E. Douglas-Cowie, R. Cowie, I. Sneddon, C. Cox, O. Lowry, M. McRorie, J.-C. Martin, L. Devillers, S. Abrilan, A. Batliner, N. Amir, and K. Karpousis, "The HUMAINE database: Addressing the collection and annotation of naturalistic and induced emotional data," in *Affective Computing and Intelligent Interaction*, A. Paiva, R. Prada, and R. W. Picard, Eds. Berlin-Heidelberg: Springer, 2007, pp. 488–500.

[249] M. Grimm, K. Kroschel, and S. Narayanan, "The Vera am Mittag German audio-visual emotional speech database," in *Proc. of ICME*, Hannover, Germany, 2008, pp. 865–868.

[250] F. Burkhardt, "Emofilt: The simulation of emotional speech by prosody transformation," in *Proc. Interspeech 2005*, Lisbon, Portugal, 2005.

[251] J. A. Russell, "A circumplex model of affect," *Journal of Personality and Social Psychology*, vol. 39, no. 6, pp. 1161–1178, 1980.

[252] F. Schiel and C. Heinrich, "Laying the foundation for in-car alcohol detection by speech," in *Proc. of INTERSPEECH*, Brighton, UK, 2009, pp. 983–986.