UNIVERSIDAD DE CHILE
FACULTAD DE CIENCIAS FÍSICAS Y MATEMÁTICAS
DEPARTAMENTO DE INGENIERÍA ELÉCTRICA

ROBUST SPEECH RECOGNITION IN NOISY AND REVERBERANT ENVIRONMENTS USING DEEP NEURAL NETWORK-BASED SYSTEMS

TESIS PARA OPTAR AL GRADO DE DOCTOR EN

INGENIERÍA ELÉCTRICA

JOSÉ EDUARDO NOVOA ILIC

PROFESOR GUÍA:
NÉSTOR BECERRA YOMA

MIEMBROS DE LA COMISIÓN:
MANUEL DUARTE MERMOUD
CESAR AZURDIA MEZA
JOHN ATKINSON ABUTRIDY
CARLOS BUSSO RECABARREN

SANTIAGO DE CHILE
2018

RESUMEN DE LA TESIS PARA OPTAR AL GRADO DE: Doctor en Ingeniería Eléctrica
POR: José Eduardo Novoa Ilic
FECHA: 06/08/2018
PROFESOR GUIA: Néstor Becerra Yoma

ROBUST SPEECH RECOGNITION IN NOISY AND REVERBERANT ENVIRONMENTS USING DEEP NEURAL NETWORK-BASED SYSTEMS

In this thesis an uncertainty weighting scheme for deep neural network-hidden Markov model (DNN-HMM) based automatic speech recognition (ASR) is proposed to increase discriminability in the decoding process. To this end, the DNN pseudo-log-likelihoods are weighted according to the uncertainty variance assigned to the acoustic observation. The results presented here suggest that substantial reduction in word error rate (WER) is achieved with clean training. Moreover, modelling the uncertainty propagation through the DNN is not required and no approximations for non-linear activation functions are made. The presented method can be applied to any network topology that delivers log-likelihood-like scores. It can be combined with any noise removal technique and adds a minimal computational cost. This technique was exhaustively evaluated and combined with uncertainty-propagation-based schemes for computing the pseudo-log-likelihoods and uncertainty variance at the DNN output. Two proposed methods optimized the parameters of the weighting function by leveraging the grid search either on a development database representing the given task or on each utterance based on discrimination metrics. Experiments with Aurora-4 task showed that, with clean training, the proposed weighting scheme can reduce WER by a maximum of 21% compared with a baseline system with spectral subtraction and uncertainty propagation using the unscented transform.

Additionally, it is proposed to replace the classical black box integration of automatic speech recognition technology in human-robot interaction (HRI) applications with the incorporation of the HRI environment representation and modeling, and the robot and user states and contexts. Accordingly, this thesis focuses on the environment representation and modeling by training a DNN-HMM based automatic speech recognition engine combining clean utterances with the acoustic-channel responses and noise that were obtained from an HRI testbed built with a PR2 mobile manipulation robot. This method avoids recording a training database in all the possible acoustic environments given an HRI scenario. In the generated testbed, the resulting ASR engine provided a WER that is at least 26% and 38% lower than publicly available speech recognition application programming interfaces (APIs) with the loudspeaker and human speakers testing databases, respectively, with a limited amount of training data.

This thesis demonstrates that even state-of-the-art DNN-HMM based speech recognizers can benefit by combining systems for which the acoustic models have been trained using different feature sets. In this context, the complementarity of DNN-HMM based ASR systems trained with the same data set but with different signal representations is discussed. DNN fusion methods based on flat-weight combination, the minimization of mutual information and the maximization of discrimination metrics were proposed and tested. Schemes that consider the combination of ASR systems with lattice combination and minimum Bayes risk decoding were also evaluated and combined with DNN fusion techniques. The experimental results were obtained using a publicly-available naturally-recorded highly reverberant speech data. Significant improvements in WER were observed by combining DNN-HMM based ASR systems with different feature sets, obtaining relative improvements of 10% with two classifiers and 18% with four classifiers, without any tuning or a priori information of the ASR accuracy.

Resumen

En esta tesis se propone un esquema de ponderación por incertidumbre para sistemas de reconocimiento automático de voz (ASR) basados en redes neuronales profundas y modelos ocultos de Markov (DNN-HMM) para incrementar la discriminabilidad en el proceso de decodificación. Para esto, los *pseudo-log-likelihoods* de la DNN son ponderados de acuerdo a la varianza de incertidumbre asignada al vector de observación. Los resultados presentados aquí sugieren que se obtiene una reducción sustancial en la tasa de error de palabra (WER) con entrenamiento *clean*. Además, no se requiere modelar la propagación de incertidumbre a través de la DNN y no se realizan aproximaciones para las funciones de activación no-lineal. El método presentado se puede aplicar a cualquier topología de red que entregue valores de tipo *log-likelihood*. Este puede ser combinado con cualquier técnica de supresión de ruido y adiciona un mínimo costo computacional. Esta técnica fue exhaustivamente evaluada y combinada con esquemas basados en propagación de incertidumbre para el cómputo de los *pseudo-log-like-lihoods* y la varianza de incertidumbre a la salida de la DNN. Dos métodos propuestos optimizaron los parámetros de la función de ponderación al aprovechar la búsqueda de grilla ya sea en una base de datos de desarrollo representativa de la tarea dada o en cada elocución en base a métricas de discriminación. Experimentos con la tarea Aurora-4 muestran que, con entrenamiento *clean,* el método de ponderación propuesto puede reducir el WER en un máximo de 21% comparado con el sistema baseline con sustracción espectral y propagación de incertidumbre usando la transformada *unscented*.

Adicionalmente, se propone reemplazar la clásica integración black box de tecnología de reconocimiento de voz en aplicaciones de interacción humano-robot (HRI) con la incorporación de la representación y modelado del entorno HRI y los estados y contextos del robot y usuario. En consecuencia, esta tesis se centra en la representación y modelado del entorno entrenando un motor de reconocimiento automático de voz basado en DNN-HMM combinando elocuciones *clean* con las respuestas del canal acústico y el ruido que se obtuvieron en un banco de pruebas de HRI construido con un robot de manipulación móvil PR2. Este método evita grabar una base de datos de entrenamiento en todos los entornos acústicos posibles dado un escenario de HRI. En el banco de pruebas generado, el motor de ASR resultante proporcionó un WER que es al menos 26% y 38% menor que las interfaces de programación de aplicaciones (APIs) de reconocimiento de voz disponibles públicamente para las bases de datos de *loudspeaker* y *speakers* humanos, respectivamente, con una cantidad limitada de datos de entrenamiento.

Esta tesis demuestra que incluso los reconocedores de voz basados en DNN-HMM de última generación se pueden beneficiar al combinar sistemas para los cuales los modelos acústicos han sido entrenados usando diferentes conjuntos de características. En este contexto se discute la complementariedad de los sistemas de ASR basados en DNN-HMM entrenados con los mismos conjuntos de datos pero con diferentes representaciones de la señal. Se propusieron y probaron los métodos de fusión de DNN basados en la combinación de peso plano, la minimización de la información mutua y la maximización de métricas de discriminación. Esquemas que consideran la combinación de sistemas de ASR con la combinación de *lattice* y la decodificación con *minimum Bayes risk* fueron evaluados y combinados con técnicas de fusión de DNN. Los resultados experimentales fueron obtenidos usando una base de datos de voz de dominio público grabada de forma natural. Se obtuvieron mejoras significativas en el WER combinando sistemas de ASR basados en DNN-HMM con diferentes conjuntos de características, obteniendo mejoras relativas de 10% con dos clasificadores y 18% con cuatro clasificadores, sin ningún ajuste o información a priori de la precisión del ASR.

To my parents Marta and José

Acknowledgements

Quiero dar las gracias a mi familia por apoyarme en todos los proyectos que he realizado, así como también a mi amada Carolina por acompañarme y ser parte de esta gran aventura.

Quiero agradecer a mis compañeros de Laboratorio Jorge Wuth, Josué Fredes, Rodrigo Mahu, Sebastián Guerrero, Víctor Poblete, Felipe Espic y Juan Pablo Escudero por la disposición y voluntad que tuvieron en todo momento para ayudarme a crecer.

Quiero dar las gracias a los Profesores miembros de la comisión Manuel Duarte, César Azurdia, John Atkinson y Carlos Busso por sus valiosos comentarios y sugerencias para mejorar la calidad de este manuscrito.

Quisiera agradecer especialmente al Profesor Néstor Becerra por mostrarme el maravilloso y apasionante mundo de la investigación y por darme la oportunidad de formar parte del gran equipo humano que es el Laboratorio de Procesamiento y Transmisión de Voz (LPTV) del Departamento de Ingeniería Eléctrica de la Universidad de Chile.

Finalmente quiero agradecer a CONICYT por otorgar el financiamiento para realizar mis estudios de postgrado a través de la beca CONICYT-PCHA/Doctorado Nacional/2014-21140711.

Contents

Index of tables

Index of figures

Chapter 1
Introduction

Speech recognition systems have become increasingly important in uses as varied as HRI scenarios, smart homes and applications where users are not directly tethered to a microphone. Major problems with ASR performance often occur with additive noise and reverberation in acoustic environments in which the ASR has to function. A wide range of algorithms have been developed to reduce issues with these variables. However, many problems still remain to be resolved to improve speech recognition accuracy.

In this thesis, three topics are proposed to improve speech recognition accuracy. First, uncertainty variance is used to manage additive noise problem in DNN-based speech recognition systems. Second, ASR training considers the time-varying acoustic channel in to increase robustness of performance in a specific acoustic environment. Finally, to reduce reverberation effects, new methods and schemes are proposed to combine different ASR systems.

1.1. Speech recognition and the additive noise problem

Additive noise is present in many of the speech recognition applications and can greatly affect the performance of recognition systems especially when this noise changes over time or is non-stationary. To address this problem, a variety of algorithms have been developed, ranging from techniques using spectral subtraction to adaptive filters. Furthermore, after the application of these noise removal techniques, the enhanced features can be considered random variables where the mean is given by the enhanced features and the variance by the uncertainty associated to these features.

Depending on the distortion in the acoustic observation, the scale of scores delivered by the acoustic model may vary frame by frame. In these cases, it is important to preserve the balance between the acoustic/phonetic and language model scores in the decoding stage. Additionally, the original motivation to use the uncertainty variance in noise cancelling was to weight the information of each frame according to its reliability and then use that variance to modify the probability of observation in Gaussian mixture model-hidden Markov model (GMM-HMM) systems.

Recently, several uncertainty-of-observation techniques have been developed by extending the idea of using the uncertainty to modify the acoustic model probability. Methods such as unscented transform [1] and piecewise exponential approximation [2] have been proposed in the literature to propagate the uncertainty variance through the DNN. Surprisingly, the uncertainty variance propagated through the DNN in DNN-HMM systems has not been employed yet.

In this thesis, the uncertainty variance in noise cancelling and the uncertainty propagated through the DNN are used to address the additive noise problem in the proposed uncertainty weighting scheme. This scheme uses the original idea of uncertainty variance in noise cancelling to underestimate frames with high uncertainty, thus increasing the discrimination of the decoding stage in state-of-the-art DNN-based ASR systems.

1.2. Time-varying acoustic channel in speech recognition

The presence of robots in our society is a reality, and the speech technologies could play an important role in improving robot social integration. Robust speech recognition is one of the key features that must be provided for successful HRI. To achieve robustness in speech recognition performance in HRI scenarios, several challenges must be addressed, including how to manage the time-varying acoustic channel due to the relative motion between the sound sources and directivity patterns of robot microphones.

Several authors in the HRI community use speech processing technologies, particularly speech recognition, in a plug-and-play fashion without considering issues that can arise in real scenarios such as noises produced by robot movements or variations that can occur with acoustic channels during interaction. Typically, these authors use toolkits or speech recognition engines without making any adaptations according to the scenario that will be used. Also, many authors consider to use a Wizard-of-Oz approach for applications that involve speech recognition instead of implementing a recognition engine (*e.g.* [3,4,5]).

Furthermore, as robots are being developed for a wide range of tasks, robots will need to operate in a wide variety of scenarios with acoustic channel variations, noises from the scenario where HRI occurs, etc. Typically the noises generated by motors, fans and robot mechanisms turn out to be non-stationary, so addressing the noise problem in HRI scenarios becomes even more complex.

Given the possible acoustic variations in HRI scenarios, a generic HRI scenario may be best used for modelling. Generic HRI scenarios can also take into consideration that when the robot or the user changes its position, there are changes in the acoustic channel that are reflected in variations of the reverberation time. In addition, generic scenarios can also manage the direction of the directivity patterns of robot microphones and sources.

This thesis addresses the problem of time-varying acoustic channels and robot noise through a DNN-based ASR training strategy, which considers the acoustic environment representation. Here, also a model is proposed that considers the acoustic environment, and the state and context of the user and robot as input of ASR to more accurately reflect real situations. Additionally, applications are proposed that involve automatic speech recognition at the different speaker-microphone distances that can occur in rooms with high reverberation.

1.3. Speech recognition in real reverberant environments

Many applications that consider the recognition of speech for their operation are developed to work in rooms with adverse acoustic conditions, such as reverberant environments in which the performance of recognition systems can be reduced due to successive sound reflections. This reverberation problem can be found in places such as: parking lots, halls, multipurpose rooms, gymnasiums and museums. While humans can recognize the speech signals in these highly-reverberant environments, achieving this task with machines is not trivial.

Additionally, distant speech recognition represents a major challenge because the effects of noise and reverberation on the speech signal increase with the speaker-to-microphone distance [6]. Distant speech recognition leads to degradation of speech recognition systems due to the room acoustics and must be considered in real applications. Several authors have performed or proposed distant speech database recording for developing and testing techniques with real data. In recent years, many researchers have tried to improve the performance of recognition systems by incorporating new training strategies and acoustic modelling techniques. However, despite the recent advances in speech recognition technology, successful distant speech recognition in real reverberant environments remains an important challenge [7].

One way to approach the reverberation problem is to design or use engineered features. Parameterization techniques for such features designed to offer robustness in certain conditions can be found in the literature. In addition, acoustic modeling based on deep learning can extract complex relationships from the input features that could help in part to deal with the reverberation problem. Another alternative to address the reverberation problem is the application of enhancement techniques. Several algorithms have been proposed to enhance the reverberated speech signals, including non-negative matrix factorization (NMF), suppression of slowly-varying components and the falling edge (SSF), and weighted prediction error (WPE).

In this thesis the reverberation problem is addressed combining acoustic models of state-of-the-art DNN-based ASR systems trained with the WPE enhancement method and different parametrization techniques. The proposed methods were exhaustively assessed in highly-reverberant real and controlled environments using the HRRE database.

1.4. Hypothesis

In this research four hypotheses are established in the ASR field:

- Underweighting frames with high uncertainty can lead to better performance in DNN-HMM based ASR systems.
- The robustness over time-varying spectral tilt of the Locally-Normalized Filter-Bank (LNFB) features can improve the performance of ASR systems in HRI scenarios.
- Considering the acoustic environment information could improve the performance of ASR systems in HRI scenarios.
- Combining ASR systems trained with different features can lead to higher accuracy in challenging environments such as those with high reverberation.

1.5. Objectives

To test these hypotheses, a general objective and three specific objectives have been established.

1.5.1. General objective

- To improve the effectiveness of the ASR systems in noisy and reverberant environments in terms of the WER.

1.5.2. Specific objectives

- To improve the performance of DNN-based ASR systems in environments with additive noise by adapting the relative weights of the language model and the acoustic-phonetic model.
- To improve the performance of DNN-based ASR systems in HRI scenarios with a time-varying acoustic channel using LNFB and environment-based training.
- Improve the robustness of DNN-based systems in reverberant environments by combining acoustic models trained with different parameterization techniques.

1.6. Thesis structure

In Chapter 2, a novel uncertainty weighting scheme is proposed to address the additive noise problem. This chapter discusses the uses and propagation through DNN of uncertainty variances. Here, two schemes to estimate the uncertainty variances are presented: at the observation vector and at the DNN output. Different configurations of the uncertainty weighting scheme are also proposed according to the estimation of uncertainty variance and the propagation of features/variance. The propagation of features and uncertainty was carried using the unscented transform. Additionally, two schemes are presented for the estimation of the weighting parameters: task-dependent and utterance-dependent. The reported results were achieved with the Aurora-4 database. The propagation of features and uncertainty variances with the unscented transform as well as the estimation of weighting parameters using the task-dependent and utterance-dependent schemes are also discussed. The results show that the improvements obtained with the proposed weighting scheme with clean training could reduce the gap between clean and multi-noise/multi-condition training, which could be very useful when it is not feasible to train an ASR system in the same testing conditions.

Chapter 3 addresses the robustness of ASR in HRI scenarios. It is proposed to replace the classic black box integration of ASR technologies with the incorporation of the HRI environment representation and modelling, and the robot and user states and contexts. Moreover, different test conditions were generated by recording two types of acoustic sources, i.e. loudspeaker and human speaker, with the PR2 robot while performing azimuthal head rotations and movements towards and away from the fixed sources. Additionally, this chapter proposes the use of locally normal features to address the problem of time-varying acoustic channel in human-robot interaction scenarios. Also, a DNN-HMM system training strategy that considers the acoustic channel responses and noise to characterize the acoustic environment in the ASR engine was proposed. Results show that the locally normalized features can lead to greater robustness regarding the acoustic channel variation. Moreover, the proposed environment-based training scheme is compared with publicly available APIs.

Chapter 4 addresses the problem of reverberation by means of the combination of complementary acoustic models trained with different parameterization techniques. DNN fusion methods based on flat-weight combination, the minimization of mutual information and the maximization of discrimination metrics were proposed and evaluated. Also, schemes that consider the combination of ASR systems with lattice combination and minimum Bayes risk decoding were tested and combined with DNN fusion techniques. Results with CHiME-2 and HRRE (highly-reverberant real environments) databases are presented and discussed. Finally, Chapter 5 summarizes the general conclusions and proposes topics for future research.

Chapter 2
Uncertainty weighting and propagation in DNN-HMM based speech recognition

2.1. Introduction

Uncertainty variance in noise removal was firstly proposed to weight the information provided by frames according to their reliability in dynamic time warping (DTW) and HMM algorithms [8,9,10]. To this end, the enhanced features, *e.g.* Mel frequency cepstral coefficients (MFCC) or filter-bank log-energies, should be considered random variables with the corresponding mean and variance. In [11] it was proposed "the replacement of the ordinary output probability with its expected value if the addition of noise is modelled as a stochastic process, which in turn is merged with the HMM in the Viterbi algorithm." As a result, the new output probability for the generic case of a mixture of Gaussians can be regarded as the definition of a stochastic version of the weighted Viterbi algorithm. This is because the final variances of the Gaussians correspond to the sum of the HMM and uncertainty variances. If the uncertainty variances increase, the discriminability of the GMM observation probability decreases and the decoding process relies more on the language model [12]. The Viterbi decoding algorithm, which incorporates the uncertainty in noise cancelling is called stochastic weighted Viterbi (SWV) algorithm because the increase of the GMM variances leads to a discriminability reduction of those frames with high uncertainty. Results with GMM-HMM-based speaker verification [11] and speech recognition [12,13] suggested that SWV can lead to significant WER reductions when speech signals are corrupted with additive, convolutional and coding-decoding distortion.

In [14], a similar result was later obtained by marginalizing the joint conditional pdf of the original and corrupted cepstral features over all possible unseen clean speech cepstra. Instead of using a model for additive noise, as in [11], the pdf of the noisy features, given the clean coefficients, was assumed to be as a Gaussian distribution. However, this result employed the same idea of uncertainty proposed in [8,9,10]. Additionally, in [14], the weighting nature of the use of uncertainty was not analyzed. In [15], a new classification rule was presented by proposing an integration over the feature space instead of over the model-parameter space. It was tested with connected speech recognition. The enhancement uncertainty variances were estimated by using a probabilistic and parametric model of speech distortion. In [16], two adaptation schemes were proposed to preserve the observation uncertainty. The results were obtained with connected digits. It is worth noting that in [12] and [13] a generalization of the model presented in [11] was successfully applied to a continuous speech recognition task.

The uncertainty estimation of speech features was also later addressed in [17,18,19,20]. Particularly in [20], it was shown that short-term Fourier transform (STFT) uncertainty propagation can be combined with the Wiener filter to compute minimum mean square error (MMSE) estimations in the feature domain for various parameter extraction methods. In contrast, despite the noise cancelling uncertainty being presented only for band-pass filters and MFCC coefficients, the proposed modelling employed in [8,9,10,11] does not require consideration of a Gaussian distribution for the additive noise in the STFT domain. Moreover, the non-linear log function is included by definition in the uncertainty estimation with spectral subtraction.

As mentioned above, in the context of band-pass filter bank analysis based features, the uncertainty in noise cancelling was proposed initially in [8,9], and further developed in [11]. According to [11] the uncertainty variance in noise cancelling in a band-pass filter is expressed as:

$$\mathrm{Var}\left[\log\left(\overline{s_m^2}\middle|\overline{y_m^2}\right)\right] = \begin{cases} \dfrac{2 \cdot c_m \cdot \mathrm{E}\left[\overline{n_m^2}\right]}{\overline{y_m^2} - \mathrm{E}\left[\overline{n_m^2}\right]} & \text{, if } \overline{y_m^2} - \mathrm{E}\left[\overline{n_m^2}\right] \geq 10 \cdot c_m \cdot \mathrm{E}\left[\overline{n_m^2}\right] \\ -\dfrac{\overline{y_m^2} - \mathrm{E}\left[\overline{n_m^2}\right]}{50 \cdot c_m \cdot \mathrm{E}\left[\overline{n_m^2}\right]} + 0.4 \text{ , else} & \end{cases} , \tag{2.1}$$

where $\overline{s_m^2}$, $\overline{y_m^2}$ and $\mathrm{E}[\overline{n_m^2}]$ are the estimated original clean energy, observed noisy energy and estimated noise energy at filter m, respectively. In addition, c_m is a correction coefficient that considers the short-term correlation between the clean and noise signals. According to [8], $\mathrm{E}\left[\log\left(\overline{s_m^2}\middle|\overline{y_m^2}\right)\right] = \log\left(\overline{y_m^2} - \mathrm{E}[\overline{n_m^2}]\right)$, where $\overline{y_m^2} - \mathrm{E}[\overline{n_m^2}]$ can be seen as the spectral subtraction estimate of the clean signal. As shown in [8] and [11], the uncertainty variance of the Mel filter bank (MelFB) and MFCC can be obtained with (2.1). The uncertainty variance of delta and delta-delta features can also be estimated as in [11]. This uncertainty variance is a key component of the SWV algorithm, which can lead to significant improvements in HMM-based speaker verification and speech recognition tasks.

Uncertainty propagation has attracted the attention of several authors in the last few years. Various uncertainty-of-observation (UoO) techniques have been developed by extending the idea of using the uncertainty in noise cancelling to modify the acoustic model probability [14,17,21,22,23,24]. The main motivation in this regard is the same as the one in SWV. It considers the enhanced features as random variables, rather than estimated coefficients. Thus, the uncertainty introduced by the enhancement process is considered the variance of the obtained feature. Then, these random variables are analytically propagated and modify the acoustic model variance. However, when applying this strategy to a DNN-based system, the problem of uncertainty propagation cannot be analytically handled without important approximations. Because a DNN is not a probabilistic model it is not clear how to modify the acoustic pseudo-likelihood given the feature uncertainty. Some methods for uncertainty propagation, such as the unscented transform (UT) [1] and piecewise exponential approximation (PIE) [2], have been proposed. Considering the results presented in [24], this chapter focuses on the UT scheme for uncertainty variance propagation through a DNN. UT is a method for propagating the statistics of a random variable through a nonlinear transformation. A set of *sigma points* is deterministically chosen to represent the distribution of the random variable. Then, these points are propagated using a given nonlinear function, the DNN in this case, and the mean and variance of the transformed set are computed. This method differs from the Monte Carlo technique in that no random samples are required, and only a low number of points is needed [25].

In the DNN-based UoO techniques published elsewhere, it is surprising that only the mean or expectation of the DNN-output is considered, and the variance is not used to modify the acoustic probability. As mentioned above, the DNN is not a parametric framework, and the uncertainty variance estimated by propagating random variables through the hidden layers requires many approximations and assumptions that can hardly be considered realistic. Moreover, the resulting uncertainty variance at the DNN output has not yet been employed in the decoding process. It is also surprising that the original motivation to define uncertainty in noise cancelling was to weight the information provided by frames according to the reliability of the information they provide. This philosophy has not been pursued by uncertainty propagation methods in DNN-based systems.

In this Chapter, an uncertainty weighting scheme for DNN-HMM-based speech recognition is presented. The motivation is to increase the discriminability in the decoding process by weighting the DNN pseudo-log-likelihoods according to the uncertainty variance assigned to the acoustic observation. The parameters of the weighting function can be optimized by a grid search on a development database or on an utterance-by-utterance basis. Optimizing the combination of the acoustic/phonetic and language models is not a problem that has been exhaustively addressed. In [26], it was suggested that the combination of the language and acoustic/phonetic models needs to be explored. The dynamic adaptation of the language model weight was addressed in [27]. The language model weight was modified according to the state of the dialogue or to the current utterance; however, a limited improvement in accuracy was observed. The advantages of the proposed method are outlined as follows: a) substantial reductions in WER are achieved with clean training; b) no degradation is introduced with multi-noise and multi-condition training; c) modelling the uncertainty propagation through the DNN is not required despite the fact that it leads to further improvements; d) no approximations for non-linear activation functions are made; e) there is no need to consider the hidden layer pre-activations as uncorrelated; f) it can be applied to any network topology that delivers log-likelihood-like scores; g) the proposed weighting scheme can be combined with any noise removal and, h) it incurs a minimal additional computational cost in its most basic configuration.

The proposed technique is exhaustively evaluated and combined with uncertainty-propagation-based schemes for computing the pseudo-log-likelihoods and the uncertainty variance at the DNN output. It is worth emphasizing that special attention is given to optimize the DNN-HMM baseline system, which in turn provides a baseline WER that is competitive with those published elsewhere. The results presented here suggest that the proposed uncertainty weighting scheme outperforms the existing propagation strategy. The latter scheme can hardly lead to improvements in recognition accuracy with an optimized DNN-HMM system. Moreover, the uncertainty weighting method reduces the gap between clean and multi-noise/multi-condition training, which can be convenient when it is not easy to train a DNN-HMM system in conditions that are similar to the testing ones.

2.2. Uncertainty weighting

In a DNN-HMM system, the DNN provides a pseudo-log-likelihood defined as [28]:

$$\log[p(x_t|q_t = s)] = \log[p(q_t = s|x_t)] - \log[p(s)]\,, \tag{2.2}$$

where x_t is the acoustic observation at time t, which is defined as a window of input feature frames. In addition, q_t denotes one of the states or senones, $s \in [1, S]$, S is the number of states or senones, and $p(s)$ is the prior probability of state s. The final decoded word string, $\widehat{W}$, is determined by [28]:

$$\widehat{W} = \arg\max_{W} \{\log[p(X|W)] + \lambda \cdot \log[p(W)]\}\,, \tag{2.3}$$

where X denotes the sequence of acoustic observations x_t, and $p(X|W)$ is the acoustic model probability that depends on the pseudo log-likelihood delivered by the DNN, $\log[p(x_t|q)]$. Furthermore, $p(W)$ is the language model probability of word string W and λ is a real constant that is employed to balance the acoustic model and language model scores. The scheme adopted in this chapter corresponds to modification of the DNN-HMM decoding process by incorporating an uncertainty weight, UW, in (2.3):

$$\widehat{W} = \arg\max_{W} \{UW \cdot \log[p(X|W)] + \lambda \cdot \log[p(W)]\}, \tag{2.4}$$

where UW is defined for each x_t, i.e. $UW[x_t]$. Accordingly, $UW[x_t] \rightarrow 0$ if the uncertainty of frames in x_t, as employed in DNN-HMM systems, is high. In addition, $UW[x_t] \rightarrow 1$ if the uncertainty of x_t is low. For a given x_t, DNN estimates S pseudo-log-likelihoods, $\log[p(x_t|q = s)]$. At each x_t, the dispersion of $\log[p(x_t|q = s)]$ is defined as its variance estimated over all the possible states or senones s. The weighted pseudo-log-likelihoods correspond to $\mathcal{L}_w = UW[x_t] \cdot \log[p(x_t|q)]$. Consequently, $\mathcal{L}_w$ has a lower variance or dispersion than $\log[p(x_t|q = s)]$ when $UW[x_t] < 1$. Observe that the closer $UW[x_t]$ is to zero, the less dispersed is the distribution of $\mathcal{L}_w$. As a consequence, the information provided by the acoustic model loses discriminability and the decoding process tends to rely more on the language model than on the acoustic model. In contrast, if the uncertainty associated to x_t is low, i.e. $UW[x_t]$ tends to be one, (2.4) is reduced to (2.3).

Hence, the motivation is to estimate and employ the uncertainty variance associated with the acoustic observation by providing an alternative technique to the uncertainty propagation methodology, which in turn requires many assumptions and approximations in the DNN framework. It should be noted that the use of the uncertainty variance at the DNN output remains unsolved to date.

In this chapter, the following weighting function which is a generalization of the function presented in [8,9,10] is proposed:

$$UW[x_t] = \begin{cases} 1 & , \text{if } UV[x_t] \leq Th \\ \dfrac{Th}{K(UV[x_t] - Th) + Th} & , \text{if } UV[x_t] > Th \end{cases}, \tag{2.5}$$

where $UV[x_t]$ is the uncertainty variance assigned to the acoustic observation, x_t; Th is a threshold; and K is a constant that can be tuned on a task-by-task basis or estimated by optimizing the discriminability sentence-by-sentence, as explained in Section 2.3. Figure 2.1 shows the weighting function defined in (2.5) with $Th = 1$ and several values of K.

In this way, two schemes to compute the uncertainty variance required by the weighting function in (2.5) have been proposed.

2.2.1. Uncertainty variances estimated at the observation vector

If $UV_{l,n}$ corresponds to the uncertainty variance of feature n at frame l, N is the number of features, and $2L + 1$ is the size of the context window of input x_t in the DNN, $UV[x_t]$ in (2.5) can be made equal to the averaged uncertainty variance within $x_t = [O_{t-L} \cdots O_t \cdots O_{t+L}]$:

$$UV[x_t] = \frac{1}{N \cdot (2L+1)} \cdot \sum_{\substack{1 \leq n \leq N \\ t-L \leq l \leq t+L}} UV_{l,n}, \tag{2.6}$$

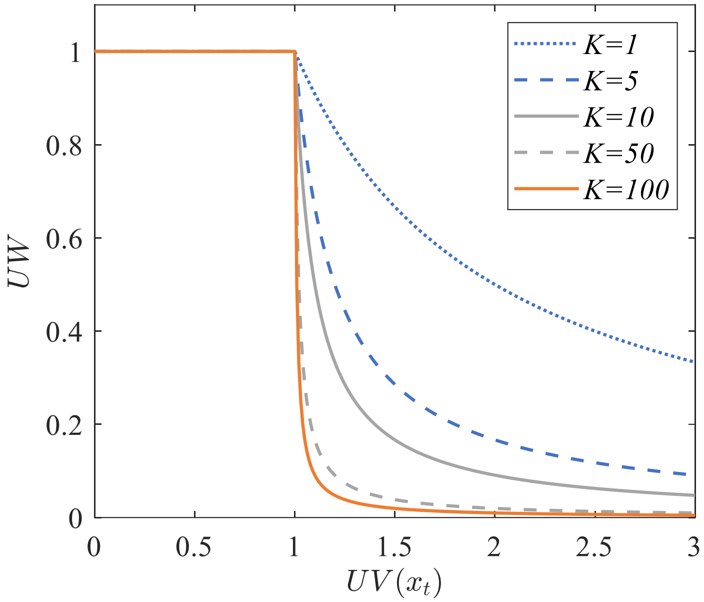


Figure 2.1 Proposed uncertainty weighting function. Th was made equal to 1 and K was made equal to 1, 5, 10, 50, and 100.

where O_t represents the observation vector at frame t composed of N features, $O_t = \left[O_{t,1} \cdots O_{t,n} \cdots O_{t,N}\right]^T$.

2.2.2. Uncertainty variances estimated at the DNN output

The averaged uncertainty variance assigned to x_t is also defined when the uncertainty variance is propagated through the DNN as a random variable. Therefore, in this case $UV[x_t]$ represents the averaged uncertainty over all states at the DNN output:

$$UV[x_t] = \frac{1}{S} \sum_{s \in [1,S]} \mathrm{Var}[\log[p(x_t|q_t = s)]]\,. \tag{2.7}$$

Observe that according to (2.7), the proposed uncertainty weighting technique provides a model in which the DNN-output variance can be employed, which remains not possible so far. The propagation of the uncertainty variances of the context window should lead to a more effective representation of $UV[x_t]$ in (2.5) because it incorporates information of the whole DNN.

2.2.3. Uncertainty variance estimation vs. feature propagation

Similarly to the uncertainty variance estimation, there are two methods to propagate the features through a DNN: first, by considering the features as constants; second, by considering the features as random variables by leveraging uncertainty propagation schemes, *e.g.* UT. This scenario generates the four combinations or possible configurations shown in Fig. 2.2 and summarized in Table 2.1. Each configuration is defined as outlined below:

Configuration I: The features of acoustic observation x_t are propagated through the DNN as constants and $UV[x_t]$ in (2.5) is directly computed at the observation vector with (2.6), as illustrated in Fig. 2.2.a.

Configuration II: The features of acoustic observation x_t are propagated through the DNN as constants, and $UV[x_t]$ in (2.5) is estimated with (2.7) by propagating the uncertainty through the DNN with UT, as illustrated in Fig. 2.2.b.

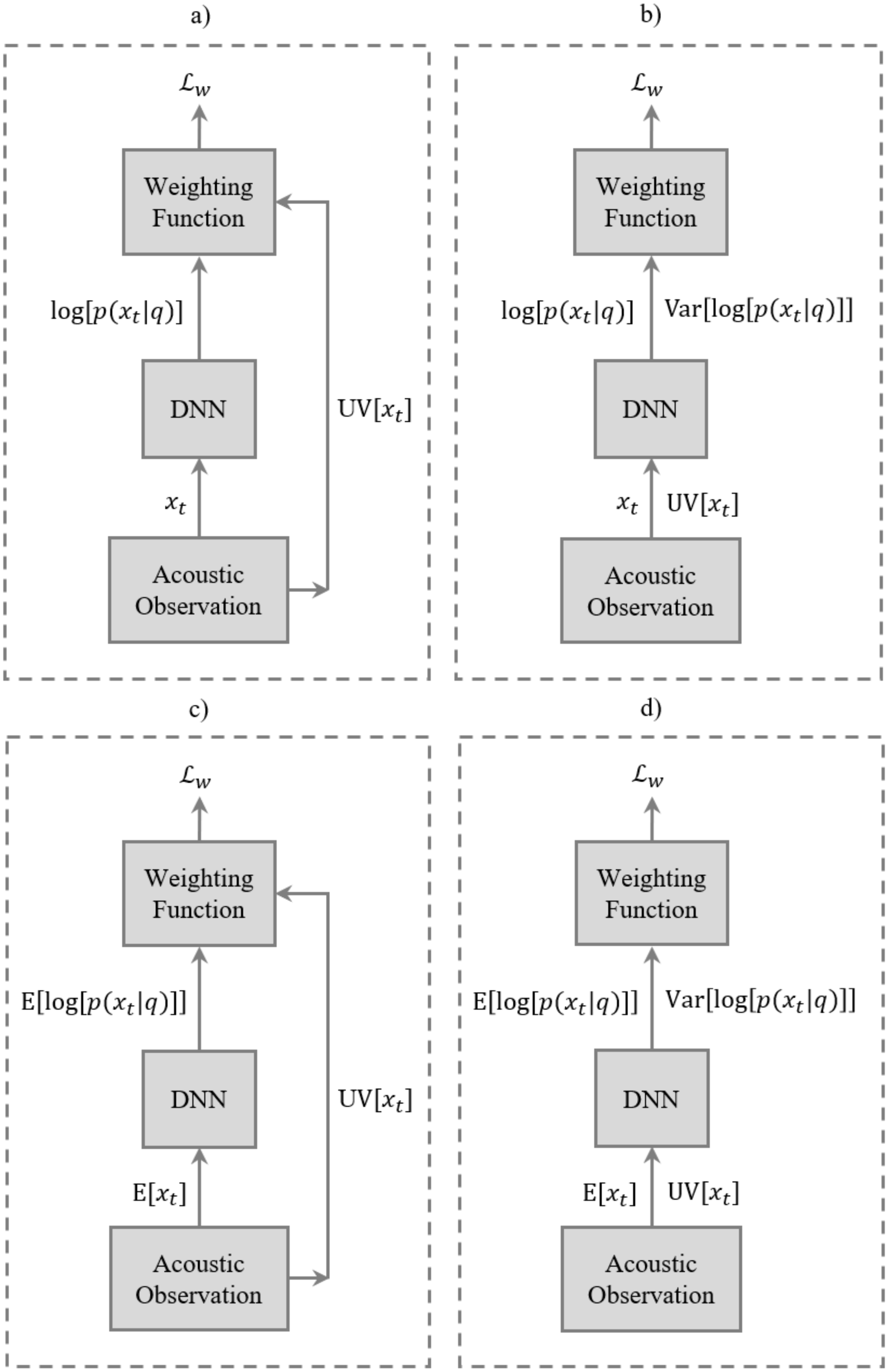


Figure 2.2 Proposed uncertainty weighting scheme for a) Configuration I, b) Configuration II, c) Configuration III and d) Configuration IV, as defined in Section 2.2.3. Observe that the weighted pseudo log-likelihoods, $\mathcal{L}_w$, corresponds to: $UW[x_t] \cdot \log[p(x_t|\mathrm{q})]$ in a) and b); and $UW[x_t] \cdot \mathrm{E}\{\log[p(x_t|q)]\}$ in c) and d).

Configuration III: The features of acoustic observation x_t are propagated through the DNN as random variables with UT, and $UV[x_t]$ is directly computed at the observation vector with (2.7), as illustrated in Fig. 2.2.c.

Configuration IV: The features of the acoustic observation x_t are propagated through the DNN as random variables with UT, and $UV[x_t]$ in (2.5) is estimated with (2.7) by using the propagated uncertainty, as illustrated in Fig. 2.2.d.

Table 2.1 Uncertainty weighting combined with uncertainty/feature propagation.

		Uncertainty variance	
		At the observation vector	**At the DNN output**
Features	**Propagated as usual**	Configuration I	Configuration II
	Propagated as a random variable	Configuration III	Configuration IV

2.3. Uncertainty weighting function estimation

The purpose of (2.4) is to increase the discriminability of the ASR system. To increase the recognition discriminability, it is necessary to estimate the parameters of the proposed model, i.e. the uncertainty weighting function parameters, K and Th in (2.5). In this chapter, two optimization schemes for K and Th in (2.5) are proposed: task-dependent and utterance-dependent estimation.

2.3.1. Task-dependent estimation of the weighting function

Observe that $UW[x_t]$ is a function of the uncertainty variances assigned to x_t and the DNN parameters. However, the DNN parameters are constant in the decoding process. They are defined by the task and estimated after a training procedure. Consequently, $UW[x_t]$ is a function of the uncertainty variances of x_t and the task itself. As a result, given a specific task, the UW function parameters, K and Th, can be estimated or tuned by increasing the discrimination ability of the DNN-HMM recognizer, e.g. by reducing the WER. A common strategy to perform this tuning is to employ a development database, which has different data from the testing data, to select the set of function parameters that results in the lowest WER. However, both the training and development databases should share similar language models with similar perplexities. Then, the selected parameters are applied to the testing database. As mentioned above, the proposed weighting scheme gives a higher weight to frames with high reliability in the recognition decision to improve the system word discriminability.

2.3.2. Utterance-dependent estimation of the weighting function

The weighting function parameters in (2.5) can be optimized (*i.e.* estimating K and Th) by minimizing the WER as in Section 2.3.1. Another strategy is to optimize the weighting function on an utterance-by-utterance basis by assessing the discriminability achieved by each pair (K, Th). To this end, a metric is defined that depends on the discrimination achieved by the decoding process on each utterance. Accordingly, the weighting function parameters can be optimized with respect to this metric by employing a grid search. The speech recognition engine employed in this research, Kaldi, uses weighted finite states transducers (WFST) representation, which provides a method to combine the acoustic and language models, leading to a simplified framework to handle the ASR decoding [29]. For each utterance, the final word sequence hypothesis is obtained from the lattice resulting from the decoding procedure. It is also possible to obtain an N-best list and the corresponding log-likelihoods from each utterance lattice. The likelihood differences between the most likely hypothesis, denoted as 1st-best, and the other hypotheses within the N-best list can be considered measures of discriminability in the hypothesis space. In addition, lattice density ($\delta_{Lattice}$), defined as the average number of arcs crossing a frame inside the lattice [30], is another parameter that characterizes the lattice obtained for each utterance. In fact, in [30] it was observed that the WER decays when $\delta_{Lattice}$ increases. As a result, $\delta_{Lattice}$ can also be used as a measure of discriminability.

To assess the discriminability resulting from the ASR decoding on each utterance, many metrics that combined N-best based analysis and $\delta_{Lattice}$ obtained from the corresponding lattice were tested. The best correlations with WER were achieved with the following metrics:

$$m_1 = \frac{llk_1}{\sigma_{N-best}} \quad , \tag{2.8}$$

$$m_2 = \frac{llk_1}{\sigma_{N-best}} \cdot \frac{1}{\delta_{Lattice}} \quad , \tag{2.9}$$

where llk_1 represents the log-likelihood (llk) of the 1st-best hypothesis, and σ_{N-best} denotes the standard deviation of llk within the N-best hypotheses. The proposed utterance-dependent estimation of the weighting function, i.e. K and Th, is summarized as follows:

```
Utterance dependent estimation of the weighting function

Choosing the uncertainty weighting/feature propagation configuration (Section 2.2.3);
Defining the search grid for $Th$ and $K$ in (2.5);
for each utterance
    for $K$ within the search grid
        for $Th$ within the search grid [11]
            ASR decoding according to (2.4);
            Obtaining resulting lattice($K$,$Th$) that depends on $K$ and $Th$;
            Obtaining N-best-list($K$,$Th$) out of lattice($K$,$Th$);
            Obtaining the log-likelihoods, $llk$, for N-best-list($K$,$Th$);
            Estimating features from the N-best-list($K$,$Th$) log-likelihoods such as $llk_1$ and $\sigma_{N-best}$;
            Obtaining lattice density $\delta_{Lattice}$ in lattice($K$,$Th$);
            Estimating discrimination metric $m(K,Th)$ for each pair ($K$,$Th$) in the search grid as a
            function of the N-best list-based features and $\delta_{Lattice}$ (e.g, $m_1$ and $m_2$ as in (2.8)
            and (2.9), respectively);
        end for
    end for
    Estimating optimal $K$ and $Th$, $\hat{K}$ and $\widehat{Th}$, respectively:
                    $\left(\hat{K},\widehat{Th}\right) = \underset{K,Th}{\arg\,\text{optimize}}\{m(K,Th)\}$
    Recognized word string = 1st-best hypothesis in N-best-list($\hat{K}$, $\widehat{Th}$);
end for
```

In the algorithm described above, the N-best hypotheses, corresponding log likelihoods, and lattice density were obtained by employing *lattice-to-nbest*, *lattice-to-post* and *lattice-depth* Kaldi tools, respectively.

2.4. Experiments

The ASR experiments were performed on the Aurora-4 corpus [31,32] by using the Kaldi Speech Recognition Toolkit [33]. Three training sets from Aurora-4 were employed: the clean, multi-noise, and multi-conditions. Each training set contains 7138 utterances from 83 speakers. The clean and multi-noise training set were recorded with a Sennheiser HMD-414 microphone. The clean training set contains only clean data. The multi-noise set contains clean (25%) and artificially-degraded utterances (75%) with one out of six noises added at SNRs between 10 and 20 dB [31]. Finally, half of the multi-condition training set was recorded with the Sennheiser HMD-414 microphone, while each utterance of the other half was recorded with one out of 18 different microphones, with noise added as in the multi-noise data [31]. The testing database was composed of 14 test sets clustered in four groups according to Table 2.2 [31]. Each noisy test set contains 330 artificially degraded utterances with one out of six noises added at SNRs between 5 and 15 dB. The development database was also composed of 14 sets with 330 utterances each, clustered in four groups [32]. The development and test sets are summarized in Table 2.2. The speakers and transcriptions in the development database are different from the testing ones. The development database was employed to avoid overfitting in the DNN training.

Table 2.2 Description of Aurora-4 development and testing data sets.

Data Set	Microphone	Noise	Group
1	Sennheiser HMD-414	Clean	A
2	Sennheiser HMD-414	Car	B
3		Babble	
4		Restaurant	
5		Street	
6		Airport	
7		Train	
8	Different Types	Clean	C
9	Different Types	Car	D
10		Babble	
11		Restaurant	
12		Street	
13		Airport	
14		Train	

Spectral subtraction (SS) [34] was applied on a frame-by-frame basis to multi-noise and multi-condition training sets, and to test data. The compensated Mel filter m is defined as:

$$FE_m{}^{SS} = \max\{\beta \cdot FE_m \,;\; FE_m - \alpha(SNR_m) \cdot \mathrm{E}[\overline{n_m^2}]\} \;, \tag{2.10}$$

where

$$\alpha(SNR_m)=\begin{cases}\alpha_0 & ,if\ SNR_m = 0\text{dB}\\ \alpha_0-(\alpha_0-1)\cdot\dfrac{SNR_m}{18} & ,if\ 0 < SNR_m < 18\text{dB}\\ 1 & ,if\ SNR_m \geq 18\text{dB}\end{cases}\quad , \tag{2.11}$$

where $\mathrm{E}\left[\overline{n_m^2}\right]$ is the noise energy estimated in non-speech intervals, as defined above; FE_m is the filter energy without SS; ${FE_m}^{SS}$ is the compensated filter energy obtained with SS; SNR corresponds to the segmental signal-to-noise ratio, where a segment corresponds to a frame; and β defines a positive lower bound to the compensated filter energy. In this chapter, α_0 and β are equal to 2.0 and 0.1, respectively. The uncertainty variances for the log energies of the Mel filters, and those for the corresponding delta and delta-delta features, were estimated according to [11] and as described in the Section 2.1. Constant c_m in (2.1) was made equal to 0.15 in all filters, as in [12]. Observe that c_m is merged in the weighting function, which in turn is defined by K and Th.

To compare the effectiveness of the proposed approach, results with Vector Taylor Series (VTS) [35] were obtained. This technique is much more complex than SS and compensates for additive noise and linear channel filtering by making use of a modified version of the EM algorithm. Given a GMM trained on clean speech, VTS obtains an MMSE estimation of the uncorrupted signal. VTS was implemented in Kaldi, and the GMM was composed of 256 components. It was trained with the whole clean training database from Aurora-4.

The DNN-HMM systems were trained using alignments from a GMM-HMM recognizer trained with the same data. In turn, the GMM-HMM systems were trained by using MFCC features, linear discriminant analysis (LDA) and maximum likelihood linear transforms (MLLT), according to the tri2b Kaldi Aurora-4 recipe: first, the monophone system is trained; second, the alignments from that system are employed to generate an initial triphone system; and finally, the triphone alignments are employed to train the final triphone system. The number of units of the output DNN layer is equal to the number of pdfs in the corresponding GMM-HMM system. For decoding stage, the standard 5K lexicon and trigram language model were used. Each DNN in the DNN-HMM system was composed of seven hidden layers and 2048 units per layer, and each was trained with the cross-entropy criterion. The feature vector was composed of 40 MelFB features, and delta and delta-delta dynamic features, with consideration of an 11-frame context window. In a previous optimization step, the DNN-HMM baseline system with multi-condition training was tested with 24, 32, 40 and 56 MelFB filters. The lowest WER, 10.9%, was found with 40 filters. This baseline WER is competitive with those published in the literature for the same task [36,37,38,39].

The feature and uncertainty variance propagation through DNN was performed with the UT method, which was implemented in C++ for compatibility with Kaldi. The following systems were evaluated: the baseline system without SS or uncertainty weighting and uncertainty propagation, *baseline*; the baseline system with SS as explained above, *baseline+SS*; feature-uncertainty propagation with UT and combined with SS, *SS+UT Prop*; uncertainty weighting according to Configuration I in Section 2.2.3, *SS+UW-config I*; uncertainty weighting according to Configuration II in Section 2.2.3, *SS+UW-config II*; uncertainty weighting according to Configuration III in Section 2.2.3, *SS+UW-config III*; and, uncertainty weighting according to Configuration IV in Section 2.2.3, *SS+UW-config IV*. Finally, statistical significance analysis was performed with the NIST matched-pair sentence-segment word error test (MAPSSWE) [40].

2.5. Discussion

The SS and VTS can reduce the average WER in 17.4% and 26.7%, respectively (statistically significant with $p < 0.001$) with clean training (see Table 2.3). With multi-noise training, SS and VTS lead to reductions in the average WER equals to 4.5% and 13%, respectively (statistically significant with $p < 0.001$) according to Table 2.4. Observe that VTS is a much more sophisticated method than SS and attempts to remove the channel mismatch as well as the additive noise. With multi-condition training, SS and VTS have no effect on recognition accuracy, according to Table 2.5. The experiments to assess the uncertainty propagation, uncertainty weighting and weighting parameter estimation are discussed as follows.

2.5.1. Feature and uncertainty variance propagation using UT

The features were propagated with UT by considering them as random variables, *SS+UT Prop*. It is an approach found in the literature that employs the uncertainty of features in DNN-based ASR without using the uncertainty weighting. In ordinary uncertainty propagation, the uncertainty

Table 2.3 WERs obtained with clean training for the Aurora-4 test groups.

System	B	C	D	AVG.
baseline	29.61	22.21	48.97	35.42
baseline + SS	21.70	22.64	42.39	29.26
baseline + VTS	20.27	14.61	37.38	25.96
SS + UT prop.	21.78	22.73	42.44	29.33
SS+ UW-config I	19.28	20.83	39.85	27.01
SS + UW-config II	17.05	22.55	37.53	25.18
SS+ UW-config III	19.35	20.81	39.95	27.09
SS + UW-config IV	17.09	22.57	37.59	25.23

Table 2.4 WERs obtained with multi-noise training for the Aurora-4 test groups.

System	B	C	D	AVG.
baseline	7.43	16.33	26.65	16.00
baseline + SS	6.99	15.13	25.66	15.28
baseline + VTS	6.95	10.84	23.26	13.92
SS + UT prop.	6.98	15.09	25.66	15.28
SS+ UW-config I	7.13	15.17	25.46	15.26
SS + UW-config II	7.16	15.06	24.95	15.04
SS+ UW-config III	7.13	15.15	25.44	15.25
SS + UW-config IV	7.19	15.08	24.95	15.06

Table 2.5 WERs obtained with multi-condition training for the Aurora-4 test groups.

System	B	C	D	AVG.
baseline	6.54	7.83	17.02	10.90
baseline + SS	6.67	7.38	17.21	11.00
baseline + VTS	7.01	6.74	17.23	11.10
SS + UT prop.	6.66	7.42	17.20	11.00
SS+ UW-config I	6.75	7.40	17.06	10.97
SS + UW-config II	6.83	7.47	16.98	10.98
SS+ UW-config III	6.76	7.38	17.09	10.99
SS + UW-config IV	6.83	7.57	16.96	10.98

variance at the DNN output is useless. On average, practically the same WER than *baseline+SS* was obtained for clean, multi-noise and multi-condition training, respectively, according to Tables 2.3, 2.4 and 2.5. These results are considered consistent with those presented by other authors in which the use of UT for uncertainty propagation in DNN-HMM-based systems led to minimal improvements in recognition accuracy [24,41].

2.5.2. Task dependent UW function estimation

The task-dependent optimization of K and Th in (2.5) was carried out by means of a grid search with group B of the development database provided by Aurora-4 (see Table 2.2): K was made equal to 1, 5, 10, 50 and 100; and Th was made equal to 0.02, 0.04, 0.06, 0.08, 0.10, 0.12, 0.14, and 0.16. All 5*8=40 configurations (K, Th) were tested. As mentioned above, group B of the development data differs from group B of the testing data. Because the language model in Aurora-4 task does not completely cover the development sets [42], a new language model was generated by adding the transcriptions of the development sets to the original training corpus. The resulting WER curves for each uncertainty weighting/feature propagation configuration (Section 2.2.3) are shown in

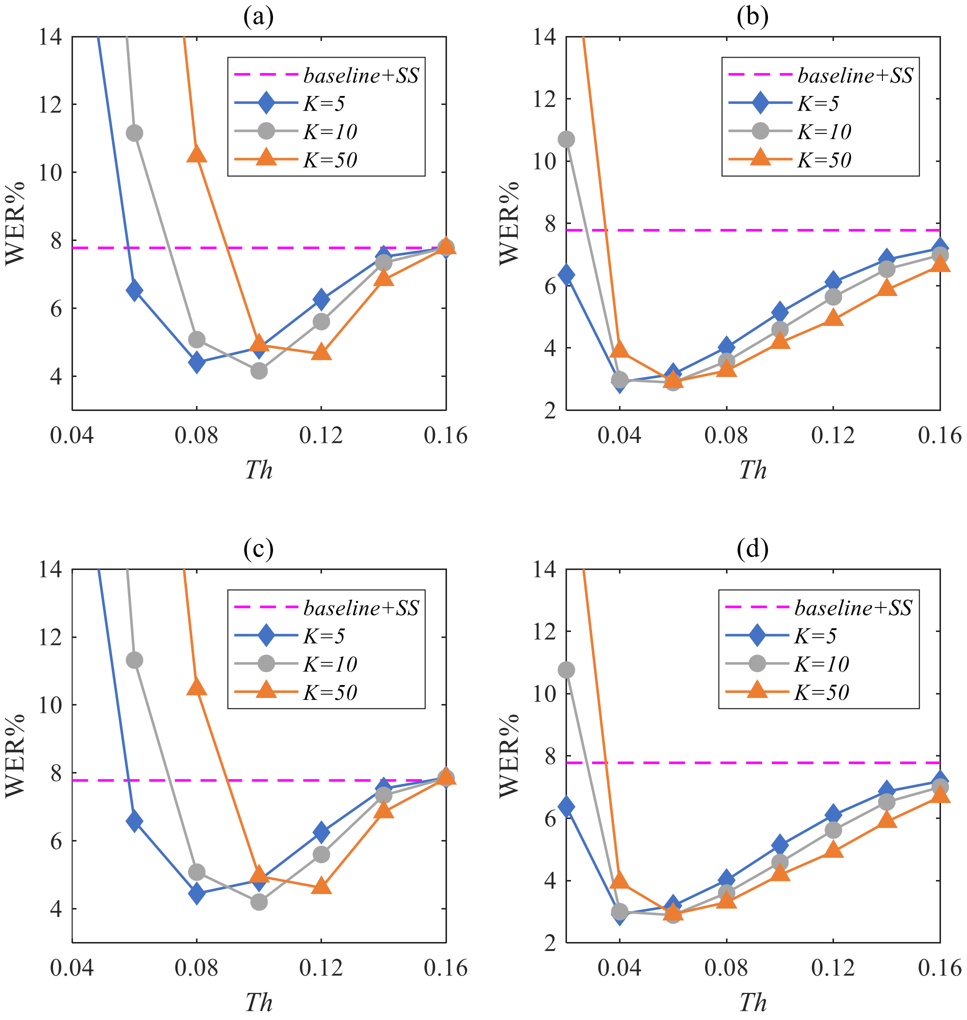


Figure 2.3 WER v/s Th with K equal to 5, 10, and 50 obtained with the development databases in the task-dependent estimation of the weighting function parameters with clean training: a) *UW-config I*, b) *UW-config II*, c) *UW-config III* and d) *UW-config IV*.

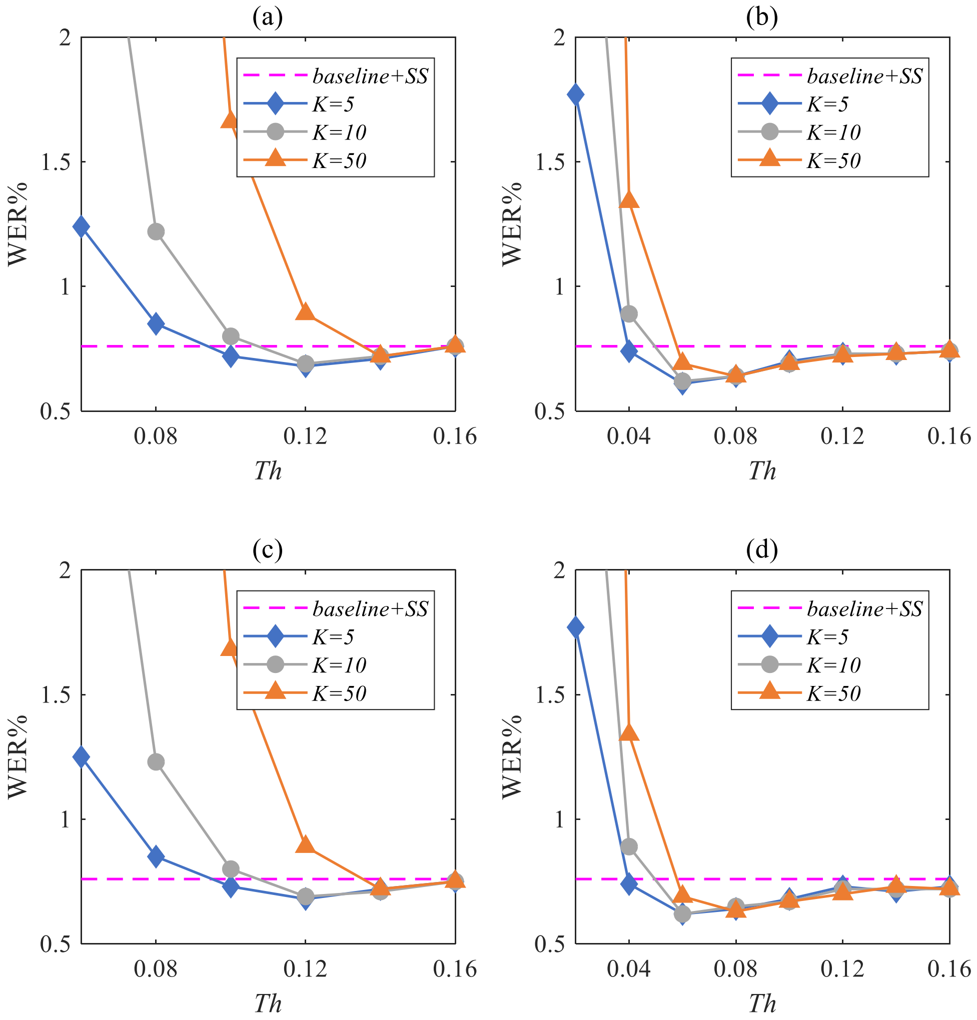


Figure 2.4 WER v/s Th with K equal to 5, 10, and 50 obtained with the development databases in the task-dependent estimation of the weighting function parameters with multi-noise training: a) *UW-config I*, b) *UW-config II*, c) *UW-config III* and d) *UW-config IV*.

Figs. 2.3, 2.4, and 2.5 when the DNN-HMM- based ASR training corresponds to clean, multi-noise and multi-condition, respectively. The optimal K and Th found for each uncertainty weighting/feature propagation configuration (Section 2.2.3) and DNN-HMM training condition are listed in Table 2.6. It is worth noting that the classic multi-noise and multi-condition training assume that the training conditions are similar to the testing ones. If there is a mismatch between training and testing, the accuracy of the ASR system should be degraded. In contrast, the optimization of the weighting function (*i.e.* K and Th) does not require use of the same types of noise in training and testing. Moreover, as shown in Figs. 2.3, 2.4, and 2.5, there is a wide range of values of K and Th where the maximum discrimination or the lowest WERs are observed. The WER obtained in the grid search optimization of K and Th with group B of the development database are summarized in Tables 2.3, 2.4 and 2.5 for clean, multi-noise and multi-condition training, respectively. As shown in Table 2.3, a significant reduction in WER is achieved using the optimal K and Th in (2.5) with clean training. This result strongly validates the uncertainty weighting model proposed here. Nevertheless, a minimally significant reduction in WER is observed with multi-noise and multi-condition training with all the uncertainty weighting/feature propagation configurations according to Tables 2.4 and 2.5. This result is consistent with the one

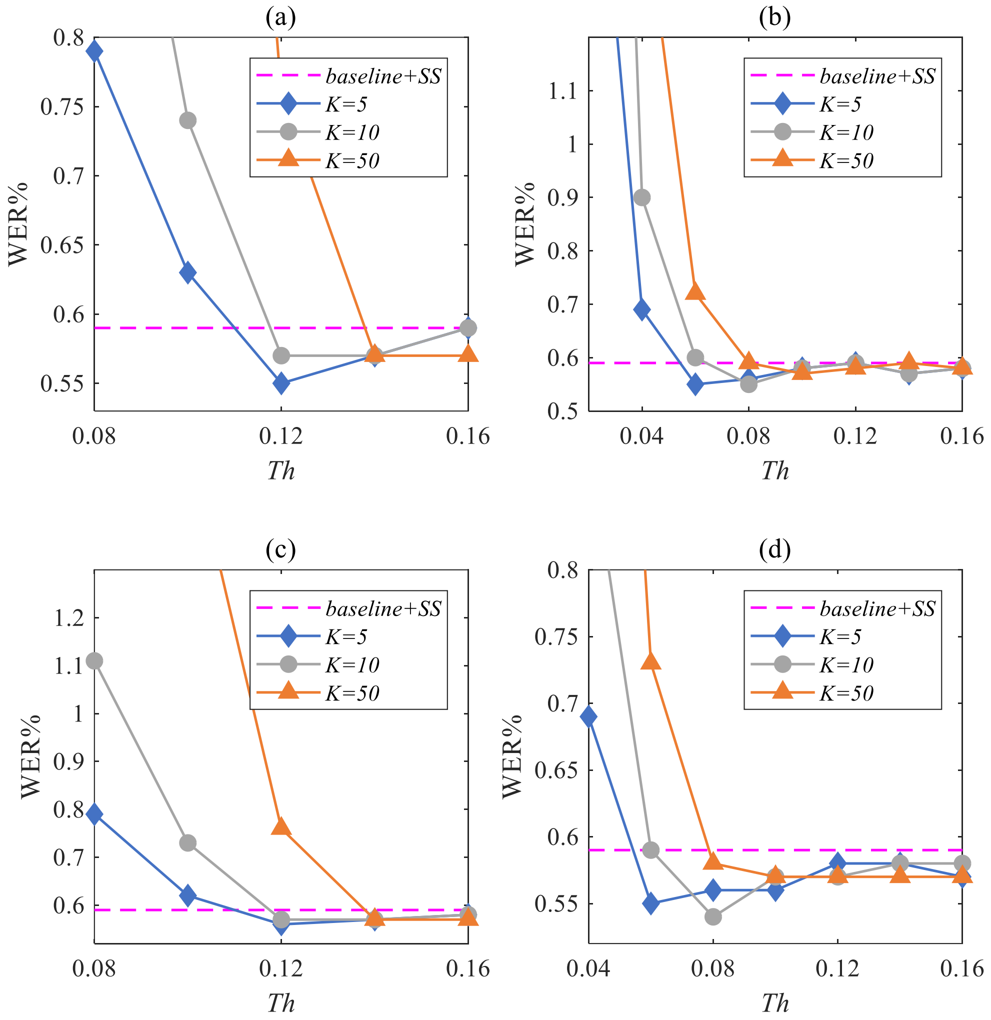


Figure 2.5 WER v/s Th with K equal to 5, 10, and 50 obtained with the development databases in the task-dependent estimation of the weighting function parameters with multi-condition training: a) *UW-config I*, b) *UW-config II*, c) *UW-config III* and d) *UW-config IV*.

obtained with uncertainty variance propagation mentioned above, whereby no improvement in recognition accuracy was observed with multi-noise and multi-condition training. The results achieved with the optimum K and Th are detailed as follows.

Table 2.6 Task-dependent optimum K and Th defined in (2.5) estimated with development database B.

Training	System	K	Th
Clean	UW-config I	10	0.10
	UW-config II	10	0.06
	UW-config III	10	0.10
	UW-config IV	10	0.06
Multi-noise	UW-config I	1	0.08
	UW-config II	1	0.04
	UW-config III	1	0.08
	UW-config IV	1	0.04
Multi-condition	UW-config I	5	0.12
	UW-config II	10	0.08
	UW-config III	5	0.12
	UW-config IV	10	0.08

With clean training, the proposed uncertainty weighting scheme with *SS+UW-config I, SS+UW-config II, SS+UW-config III* and *SS+UW-config IV* leads to average relative improvements in WER that are equal to 7.7%, 14.0%, 7.4% and 13.8%, respectively, when compared with *baseline+SS*, as shown in Table 2.3. The greatest improvements occur with group B (*i.e.* additive noise mismatch only), where the highest reductions in WER compared with *baseline+SS* are 21.4% and 21.2% for *SS+UW-config II* and *SS+UW-config IV*, respectively. It should be noted that SS attempts to cancel only additive noise. The combined effect of SS and uncertainty weighting lead to average improvements of 23.7%, 28.9%, 23.5% and 28.8% for *SS+UW-config I, SS+UW-config II, SS+UW-config III* and *SS+UW-config IV*, respectively, when compared with *baseline*. When compared with the uncertainty propagation with UT, the proposed uncertainty weighting scheme also resulted in reductions equal to 7.9%, 14.2%, 7.6% and 14.0% with *SS+UW-config I, SS+UW-config II, SS+UW-config III* and *SS+UW-config IV*, respectively. All these improvements are statistically significant with $p < 0.001$. When compared with VTS, the uncertainty weighting method with SS leads to reductions in the average WER equal to 3.0% (statistically significant with $p < 0.001$) and 2.8% (statistically significant with $p < 0.01$) with *SS+UW-config II* and *SS+UW-config IV*, respectively. Moreover, the WER reduction from *SS+UW-config II* and *SS+UW-config IV* is higher than 15% with group B (see Table 2.2). These results strongly validate the proposed decoding method according to (2.4).

According to Tables 2.4 and 2.5, the improvement in recognition accuracy from uncertainty weighting and uncertainty propagation with multi-noise and multi-condition training is dramatically lower than with clean training. This result is consistent with those obtained in the previous tuning of the weighting function with the development data (see Figs. 2.4 and 2.5). Group D with multi-noise training is the only case where some reduction in WER is observed when compared with *baseline+SS*: 2.8% with both *SS+UW-config II* and *SS+UW-config IV*. This reduction is statistically significant with $p < 0.001$ in both cases. The poorer performance with multi-noise and multi-condition training must be due to the fact that the accuracy of the DNN response does not depend on the uncertainty in noise cancelling when the same additive noise employed in testing is included in training.

Multi-noise training is the only condition in which VTS provides a higher reduction in WER than the uncertainty weighting scheme with SS. This is a result of the fact that VTS attempts to remove both additive noise and the channel mismatch (*i.e.* groups C and D in Table 2.2). In contrast, SS attempts to remove only additive noise. With multi-condition training, no significant difference was found between VTS and any of the configurations of the uncertainty weighting decoding, i.e. *SS+UW-config I, SS+UW-config II SS+UW-config III* and *SS+UW-config IV*.

In the experiments described above, the optimization of K and Th in (2.5) was carried out by means of a grid search with group B of the development database provided by Aurora-4 with the same types of additive noise employed in testing. In order to evaluate the independence of the optimal K and Th with respect to additive noise, Fig. 2.3 was reproduced (clean training) by using the leave-one-out strategy with the six types of additive noise in the development set, by tuning Th with K equal to 5, 10 and 50 in all the possible combinations of five-out-of-six types of additive noise. As a result, an extremely high coincidence of the tuning curves with all the possible combinations of five-out-of-six types of additive noise was observed in *SS+UW-config I, SS+UW-config II, SS+UW-config III* and *SS+UW-config IV*. In fact, the resulting optimal Th for each K is exactly the same with all combinations of five-out-of-six types of additive noise, in all the configurations of feature and uncertainty propagation. This result is consistent with Fig. 2.3,

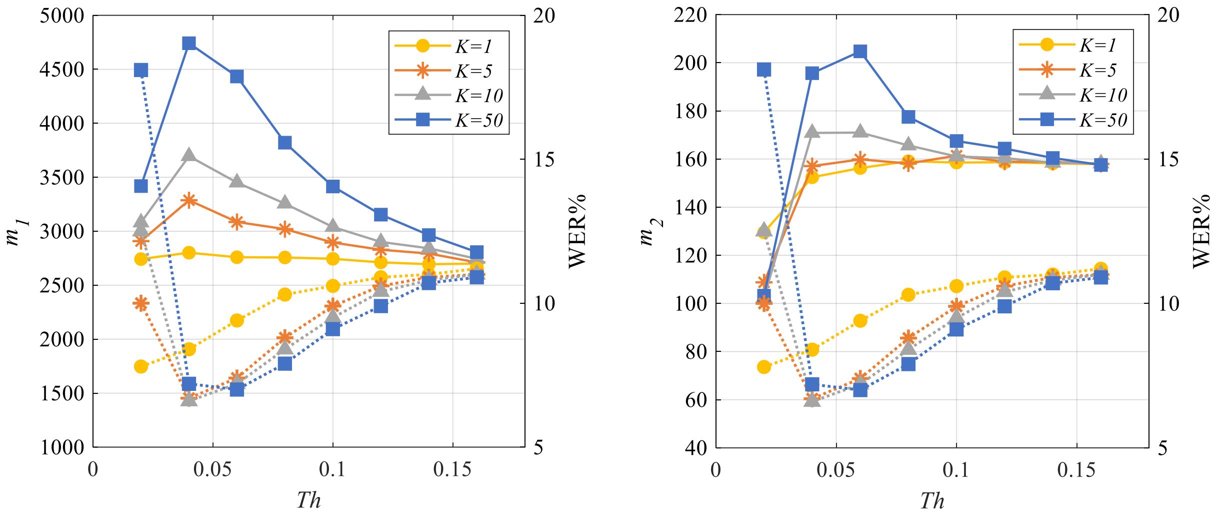


Figure 2.6 Discriminability metrics m_1 and m_2 v/s Th (solid line), and WER v/s Th (dotted line), with K equal to 1, 5, 10, and 50. Metrics m_1 and m_2 are defined in (2.8) and (2.9). K and Th are the weighting function parameters according to (2.5). The DNN-HMM system corresponds to *UW-config IV*.

where it is shown that there is a wide range of values for K and Th that minimize the WER. In other words, removing one noise from the development data does not affect the optimal weighting function. Moreover, according to (2.1), the uncertainty in noise cancelling depends only on local SNR estimated on the corresponding filter-band. Consequently, given a frame, the averaged uncertainty depends mostly on the local SNR. Finally, DNNs typically exhibit good generalization, which may also contribute to the consistency observed in the estimation of K and Th.

2.5.3. Utterance dependent UW function estimation

In this chapter, the estimation of the weighting function, *i.e.* K and Th in (2.5), is also proposed on an utterance-per-utterance basis, as explained in Section 2.3.2, by employing grid search optimization with metrics m_1 (2.8) and m_2 (2.9). Each metric is evaluated within the N-best hypotheses in the lattice resulting from the ASR decoding procedure for a given set of values for K and Th. Figure 2.6 shows WER vs. Th when K equal to 1, 5, 10, and 50 with the car noise testing sub-set and clean training, for metrics m_1 and m_2. The N-best-list analysis according to Section 2.3.2 was performed with N equal to 100 hypotheses. The pseudo log-likelihood of each one of the N-best hypotheses and the lattice densities to estimate metrics m_1 and m_2 according to (2.8) and (2.9) were obtained with Kaldi tools. As shown in Fig. 2.6, the maximum values of metrics m_1 and m_2 coincide with the lowest WER. This result strongly suggests that the weighting function parameters K and Th in (2.5) could be optimized on an utterance-by-utterance basis with respect to the maximization of metrics such as m_1 and m_2. The results obtained with the estimation of K and Th based on the utterance dependent maximization of metrics m_1 and m_2 are shown in Table 2.7. With clean training, *SS+UW-config IV* leads to average relative reductions in WER that

Table 2.7 WERs obtained with the utterance-dependent estimation of K and Th defined in (2.5) based on the maximization of discriminability metrics m_1 and m_2 defined in (2.8) and (2.9) for the Aurora-4 test groups.

System	B	C	D	AVG.
baseline+SS	21.7	22.64	42.39	29.26
UW-config IV, m_1	19.50	22.36	39.62	27.18
UW-config IV, m_2	18.65	21.93	39.33	26.61

are equal to 7.1% and 9.0% when compared with *baseline+SS* with metrics m_1 and m_2, respectively. This reduction is statistically significant at the level of $p < 0.001$ in both cases. These results suggest that the utterance dependent optimization of the uncertainty weighting function is possible. Nevertheless, the reductions in WER that are achieved are lower than those obtained by using a representative development data to tune K and Th. The use of new metrics that could be more finely correlated with the discriminability in the decoding process should overcome this limitation.

2.6. Conclusions

In this chapter, an uncertainty weighting scheme for DNN-HMM-based speech recognition is proposed. The motivation was to increase the discriminability in the decoding process by weighting the DNN pseudo-log-likelihoods according to the uncertainty variance assigned to the acoustic observation. This scheme was exhaustively tested and combined with uncertainty-propagation based schemes for computing the pseudo-log-likelihoods and uncertainty variance at the DNN output. It is worth highlighting that, in contrast to traditional uncertainty propagation schemes, the proposed uncertainty weighting enables use of the uncertainty variance at the DNN output. Special attention was focused on optimizing the DNN-HMM baseline system, which in turn produced a baseline WER that is competitive with those published elsewhere. The parameters of the weighting function can be optimized by making use of a grid search on a development database that is representative of the addressed task or on each utterance based on discrimination metrics. Experiments with Aurora-4 task and clean training showed that the proposed weighting scheme combined with spectral subtraction provided a reduction in WER as high as 21% when compared with the baseline system with spectral subtraction and uncertainty propagation with the unscented transform, when there was only an additive noise mismatch. In the same conditions, the reductions in WER compared to VTS, a much more sophisticated method than SS, were higher than 15%. It is worth highlighting that, in principle, the proposed uncertainty weighting scheme can be defined and applied to any front end or distortion removal technique.

With multi-noise training a low reduction in WER was observed. Nonetheless, in the case of multi-condition training no recognition accuracy increase was observed. If the same additive noise employed in testing is included in training, the accuracy of the DNN response would not depend on the uncertainty in noise cancelling, and the proposed weighted decoding would lose its effectiveness. Moreover, in this chapter, a method was proposed to resolve an interesting problem that has not been exhaustively addressed in the literature: the optimization of the acoustic/phonetic model and language model combination. In this context, if the accuracy of the DNN response is modelled with multi-noise and multi-condition training, this information can be employed in combination with the scheme proposed herein. On the other hand, bounds were established in this chapter to the uncertainty weighting schemes that are not found in the literature. It is important to emphasize that the results on the use of uncertainty reported in this chapter are competitive with those on uncertainty published elsewhere using the same database that was presently employed. Furthermore, these results were achieved with a baseline system that favorably compared with others used in the literature, which validated the improvements in accuracy reported here.

In addition, the uncertainty weighting method is a means to reducing the gap between clean and multi-noise/multi-condition training. This can be useful when it is not easy to train a DNN-HMM system in conditions that are similar to the testing ones. The proposed scheme can thus be applied to any network topology that delivers log-likelihood-like scores, it can be combined with any distortion removal technique or front end, and it requires a very low additional computational cost in the configurations where the propagation of uncertainty is not required due to the small number of operations necessary to obtain an apply the weighting factor. Finally, the combination of the uncertainty weighting scheme with other noise cancellation methods, and the modelling of the DNN response accuracy with multi-noise and multi-condition training, are proposed for future research.

Chapter 3
DNN-HMM based automatic speech recognition for HRI scenarios

3.1. Introduction

If social robotics is a reality, then the appropriate social integration between humans and robots could greatly improve the cooperation between users and machines. There are several applications in defense, hostile environments, mining, industry, forestry, education and natural disasters where some integration and collaboration between humans and robots will be required [43]. HRI is especially relevant in those situations when robots are not fully autonomous and require interaction with humans to receive instructions or information in decision-making applications [44,45,46,47]. In this context, human like communication between people and robots is essential for a successful human-robot collaborative symbiosis [48,49]. Additionally, speech is the most straightforward and natural way that humans employ to communicate [50,51,52]. As a consequence, voice-based HRI should be the most natural way to facilitate a collaborative human-robot synergy. Hence, speech technology, especially ASR, should play an important role in social robotics.

Furthermore, it is well known that computer vision is an important research topic in robotics. Recent challenges such as DARPA Robotics Challenge [53] and Robocup [54] have led to great improvements in computer vision [55,56,57,58]. On the other hand, there has also been a significant progress in ASR, but this advancement has taken place outside the HRI field. ASR has gained relevance in robotics in the last years, but its status is still far from the one enjoyed by computer vision in the robotic research. This is still somehow surprising, considering that both technologies make use of similar signal processing and deep learning methods, and may explain partly the lower penetration of ASR in the robotic community.

In this chapter, it is proposed that ASR technology should also be investigated, designed and developed to address HRI applications. Subsequently, the ASR engine should take into consideration the environment, and robot and user states and contexts. Following this strategy, this chapter focuses on the environment representation and modeling by training the ASR engine with the combination of clean utterances with the acoustic-channel responses and noise that were estimated and recorded, respectively, with an HRI testbed. This testbed represents the generic problem of HRI in mobile robotics. The resulting ASR accuracy outperforms publicly available ASR APIs with a limited amount of training data.

3.2. Related work

3.2.1. An introduction to ASR technology

Automatic speech recognition is the process and related technology for transcribing human speech into words. By using Bayes's rule, the ASR problem can be formulated as follows [59]:

$$\widehat{W} = \underset{W}{\operatorname{argmax}} P(W|X) = \underset{W}{\operatorname{argmax}} p(X|W) \cdot p(W) \ , \tag{3.1}$$

where $\widehat{W}$ is the optimal label (word or phone) sequence; X is the input speech observation sequence that represents a given speech utterance; $p(W)$ denotes the language model describing the probabilities of word combinations; and, $p(X|W)$ indicates the acoustic model. Consequently, the task of an ASR system is to find (by means of a process called decoding, performed with the Viterbi algorithm [60]) the most likely label sequence $\widehat{W}$ given an observed sequence of feature vectors that corresponds to the speech utterance. The language model can be represented with [61]: statistical models; stochastic context-free grammars (SCFG); or, stochastic finite-state models. In the case of statistical models, which are widely employed in research, the prior probability of a word sequence $W = w_1, \dots, w_L$ in (1) can be approximated with N-grams:

$$p(W) \cong \prod_{l=1}^{L} p(w_l | w_{l-1}, w_{l-2}, \dots, w_{l-N+1}) \ , \tag{3.2}$$

where N is typically between 2 and 4. The language model defines the transition probability from one N-gram to the next word to guide the search for an interpretation of the acoustic input. Additionally, the size of the vocabulary and perplexity [62] are critical for the ASR accuracy. Basically, perplexity measures the uncertainty about the words that may follow a given N-gram. A low-perplexity language model defined by a given task or context will constrain the decoding and perform better than a high-perplexity one.

Acoustic modeling defines the statistical representations for the sequence of acoustic feature vectors X obtained from the speech waveform. The utterances are divided into 20 or 30 ms windows with overlap (*e.g.* 50%). The set of acoustic features are usually obtained from the short-term fast Fourier transform (FFT) within each window [60,63,64]. Speed and acceleration coefficients (also called delta and delta-delta coefficients) are also typically used, and the final feature vector is composed of the static features plus the delta and delta-delta coefficients [65]. Mean and variance normalization of the coefficients can also be employed. Until a few years ago, most speech recognition systems adopted hidden Markov models (HMMs), to deal with the temporal variability of speech, and Gaussian mixture models (GMMs) to represent $p(X|W)$. Given a set of speech feature vectors $\boldsymbol{X} = \{\boldsymbol{x}_t\}_{t=1}^{T}$, the state observation probability density function of feature vector $\boldsymbol{x}_t$ at frame t in state s_i is expressed by [60]:

$$p(\boldsymbol{x}_t | \mathrm{s}_i) = \sum_{m=1}^{M} c_{i,m} \cdot \mathcal{N}(\boldsymbol{x}_t; \mu_{i,m}, \Sigma_{i,m}) \ , \tag{3.3}$$

where $c_{i,m}$, $\mu_{i,m}$, and $\Sigma_{i,m}$ correspond to the mixture weights, mean vectors, and covariance matrices, respectively, for M Gaussian mixture components. In the last few years, artificial neural networks (ANNs), *e.g.* DNNs, have shown significant performance improvement over GMM based models. In a DNN-HMM system, the DNN provides a pseudo-log-likelihood defined as:

$$\log[p(\boldsymbol{x}_t | s_j)] = \log[p(s_j | \boldsymbol{x}_t)] - \log[p(s_j)] \ , \tag{3.4}$$

where s_j denotes one of the states or senones; and the state priors $\log[p(s_j)]$ can be trained using the state alignments obtained with the training speech data. The final decoded word string, $\widehat{W}$, is determined by:

$$\widehat{W} = \underset{W}{\operatorname{argmax}} \{\log[p(X|\mathrm{W})] + \lambda \cdot \log[p(W)]\} \ , \tag{3.5}$$

where the acoustic model probability $p(X|W)$ depends on the pseudo log-likelihood $\log[p(\boldsymbol{x}_t|s)]$ delivered by the DNN, and λ is the constant that is employed to balance the acoustic model and language model scores [66]. The results reported in [67] showed that the DNN-HMM ASR can lead to a word error rate reduction of 32% relative when compared to the ordinary GMM-HMM system with the Switchboard task [68]. However, training a DNN is not an easy task. The objective function can be highly non-convex, and the training algorithm can easily converge to a suboptimal local minimum. This problem can be minimized by making use of a pre-training strategy [69]. Also, ANNs need more training data than GMM-HMM systems [70]. It is worth mentioning that public ANN based ASR APIs employ at least tens of thousands of hours of speech for training, if not millions of hours. Other ANN architectures have also been applied to ASR: LSTM [71]; CNN [72]; and, RNN [73]. The results obtained using DNN-HMM systems are competitive when compared to those reported with others ANN architectures [74,75,76,77,78]. In some cases, systems employing combinations of ANN architectures, very deep CNN [79] or fCNN [80] have outperformed DNN, LSTM, or the ordinary CNN approaches. However, the higher the number of the ANN parameters, the higher the required amount of training data.

In matched conditions between training and testing data, ASR shows large performance gain. In contrast, models will have difficulties recognizing test samples if they differ from data used in training. For this reason, noise robustness of ANN based systems can be achieved by using multi-style training. For instance, a DNN trained with several types of noise and SNR levels can lead to a high accuracy improvement in real applications [81].

3.2.2. Black box-based integration of ASR technology

Most of the research that considers ASR in HRI scenarios use ASR toolkits or APIs as black boxes. A non-exhaustive list of available options that support ASR includes systems such as HTK [82], SPHINX [83,84], JULIUS [85], KALDI [33] and BAVIECA [86], and general purpose ASR APIs provided by, for instance, Google, Microsoft and IBM. These toolkits and APIs have been employed in HRI applications to incorporate ASR capabilities to a robot on a plug-and-play fashion [87,88,89,90,91], *i.e.* a speech signal is input to the ASR to obtain a text transcription (see Fig. 3.1) without taking into consideration operation conditions such as noise, relative movement between the speaker and the robot, microphones directivity and response, or user or robot context.

Acoustic input
ASR
Recognition output

Figure 3.1 Ordinary black box-based ASR integration in HRI scenarios.

In [87], a project that integrates smart home technology and a socially assistive robot to extend independent living for elderly people is described. A Nao robot plays the role of communication interface between the elderly, the smart home, and the external world. The robot can recognize simple answers from the user such as “yes” and “no” by using Sphinx 4.0 from Carnegie-Mellon University. Despite the fact that the Nao robot has a built-in microphone, its quality is too low for practical indoor applications, and a ceiling-mounted microphone was used to capture user speech. CMU Sphinx engine was also employed in [90], as part of a voice control system for a robotic endoscope holder during minimally invasive surgery. In [89], a general framework for multimodal human-robot communication is proposed. This framework allows users to interact with robots using speech and gestures. The Google Speech API was chosen because it offered speaker and vocabulary independency, which in turn could allow a natural speech interaction with no constraints. Google Speech API was also employed in [91] to provide ASR capabilities to a robot that needed to understand the intentions of users without requiring specialized user training. It comprises a recognition model that combines language, gestures, and visual attributes. In [92], four ASR engines were compared by making use of different grammars: the Google Speech API; the Microsoft Speech APIM; Pocket Sphinx from CMU; and, the NAO-embedded Nuance VoCon 4.7 engine. Experimental results showed that the Google Speech API led to the highest accuracy.

The integration of ASR technology on a black box basis can lead to poor performance because the chosen ASR system is not designed necessarily to comply with specific scenarios or tasks. In [88], an evaluation with children aged from 4 to 10 years old playing versions of a language-based game hosted by an animated character is described. Speech recognition results using Sphinx3 on children utterance showed a poor performance, partially due to the mismatch between the children's voices and the adult acoustic model of the ASR engine. General purpose speech toolkits or APIs have been widely used as an easy solution to integrate ASR to some platforms. However, while those ASR engines provide good results in several scenarios, they may not provide an optimal solution to specific tasks because they are not considered in the training procedure, or the technology simply does not compensate for unexpected distortions. As an example, in [93], it was investigated whether the open-source speech recognizer Sphinx can be tuned to outperform Google cloud-based speech recognition API in a spoken dialog system task. By training a domain-specific language and making adjustments, Sphinx could outperform the Google API by 3.3%.

3.2.3. Simulating ASR with WoZ evaluations

One of the challenges in HRI interaction that may require an ad-hoc solution instead of a multipurpose API, is the speech recognition with relative movements between the speaker and the robot. In scenarios where ASR is performed by moving robots, the corruption of speech produced by the additive noise of the robot's motors should be taken into consideration. Speech recognition experiments with moving robots in [94] led the authors to recommend that the robot should pause its actions as soon as it realizes that it is being talked to, which in some applications is unacceptable. They also suggest that the only reliable speech recognition engine for HRI is another human being. Given the fragility of ASR technology that was unveiled in HRI environments, many researchers have adopted interaction mechanisms that do not rely on speech recognition technology such as Wizard of Oz (WoZ) based approaches [3,4,5,95,96,97,98,99], *i.e.* the response of a speech recognition engine is simulated to evaluate other factors before it is implemented.

3.2.4. Evaluation of optimal physical set up and operating conditions

There is an alternative strategy, which instead of making the ASR technology more suitable to target operating conditions or adopting WoZ schemes, attempts to find the optimal operating environment that maximizes the ASR accuracy. In [92], the following variables were evaluated: different noise scenarios; different distances and angles of the speaker with respect to the microphones; three types of microphones, *i.e.* desktop, studio and the robot-mounted microphone. According to the experimental results, the authors provide recommendations regarding how the speech-based HRI with children should be deployed so as to achieve a smoother interaction. Some of the recommendations are: using additional input/output devices, even replacing verbal language input with a touchscreen; and, to place the user in an optimal location with respect to the microphones. Although these recommendations are based on evaluations with children, the authors suggest that they are applicable to HRI in general.

A speech recognition friendly artificial language (ROILA) was compared to English spoken language when talking to a Nao robot in [94]. The experiment considered: three microphone types (the ones built-in in the robot, a headset, and a desktop microphone); two conditions of head movement (static and moving) for the Nao robot; and, the two types of spoken languages (English and ROILA). The authors concluded that ROILA does not provide a significant improvement when compared to ordinary spoken English. However, the type of microphone and the robot's head movement are critical for the ASR accuracy.

If ideal operating conditions are not met, one strategy is to try to cancel the corrupting environments. For instance, in [100] and [101] the external noise sources or ego-noise caused by motors and fans of the robot are removed with enhancement methods.

3.2.5. Locally normalized features

A novel set of speech features for robust ASR called Locally-Normalized Cepstral Coefficients (LNCC) was proposed in [102]. LNCC are inspired by Seneff's Generalized Synchrony Detector (GSD) [103] which perform a local normalization in the frequency domain in each auditory channel, and hence are relatively invariant to changes in the frequency response of the transmission channel. LNCC are an extremely simple but effective way to instantaneously normalize speech features. Its effectiveness has been tested in speaker verification tasks where results demonstrate that the proposed LNCC features are far more effective compensating for spectral tilt [102] and more robust to additive noise than ordinary MFCC coefficients [104].

Given the effectiveness of LNCC and motivated by the fact that performance of DNN-HMM ASR systems is typically better when spectrogram-like features are used, rather than features in a pseudo-cepstral domain, Locally Normalized Filter Bank (LNFB) features are presented in [105]. LNFB features correspond essentially to LNCC features before the cepstral transform. The local normalization is achieved in the filter-bank space by dividing the output of a triangular frequency-weighting filter (which is similar to the triangular filter in conventional MFCC coefficients) by the output of a second frequency-weighting filter [102]. This normalization removes very coarse variations in the spectral shape that can be considered constant within both filters, such as overall tilt, which arise mostly from channel variability. These two filters are called "numerator filter" and "denominator filter", and their shape is an approximation to the frequency response of the numerator and denominator of the Seneff GSD operator:

$$Num_m(f)=\begin{cases} -\frac{2}{B}\left|f-f_m^C\right| & \text{, if } \left|f-f_m^C\right|\leq\frac{B}{2} \\ 0 & \text{, otherwise} \end{cases} \quad , \tag{3.6}$$

$$Den_m(f)=\begin{cases} \frac{2}{B}(1-d_{min})\left|f-f_m^C\right|+d_{min} & \text{, if } \left|f-f_m^C\right|\leq\frac{B}{2} \\ 0 & \text{, otherwise} \end{cases} \quad , \tag{3.7}$$

The shapes of these filters are shown in Fig. 3.2.

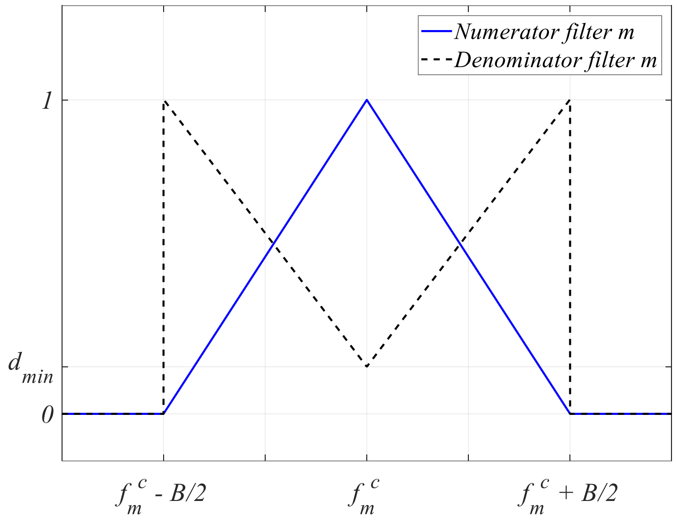


Figure 3.2 Graphical representation of the *m*th numerator filter (solid line) and the *m*th denominator filter (dashed line).

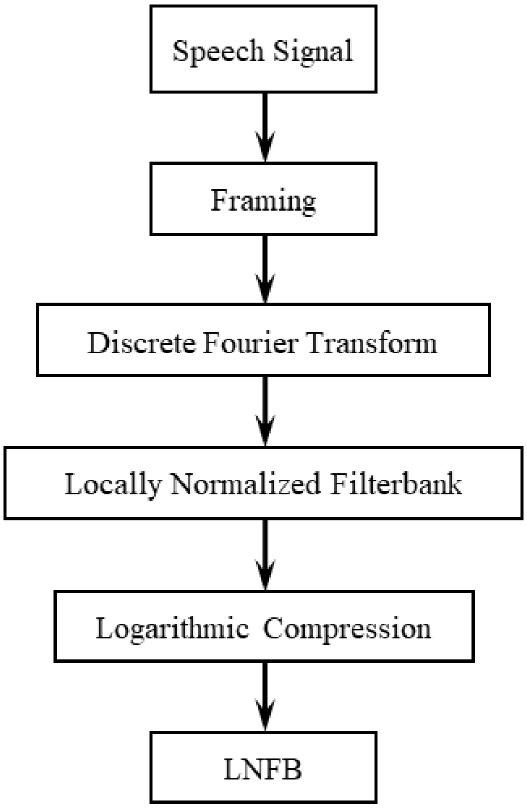


Figure 3.3 Processing sequence to obtain LNFB features.

Given a channel m with central frequency f_m^c and bandwidth B, the LNFB feature m is defined as the log of the locally-normalized energy for channel m, LN_m:

$$LNFB_m = \log(LN_m) = \log(LNNum_m/LNDen_m) \quad , \tag{3.8}$$

where $LNNum_m$ is the numerator filter energy, and $LNDen_m$ is the denominator filter energy.

The sequence of processing stages to obtain the LNFB is shown in Fig. 3.3.

In [105], LNFB was applied to the Aurora-4 robust DNN-HMM-based speech recognition task. It is shown that mean and variance normalization is more effective than mean normalization for the LNFB features and the relative global WER over all conditions for LNFB features was 7.4% smaller than the average WER obtained using MelFB features. These results indicate that LNFB features provide better recognition accuracy for DNN-HMM ASR systems compared to MelFB features. Furthermore, results suggest that LNFB are especially helpful in providing robustness to channel mismatches between training and testing data. The use of LNFB has not been tested with reverberant data yet and the use of these features could be an alternative to tackle the possible spectral variations produced by reverberation.

One of the motivations behind the LNCC or LNFB features was to provide a set of parameters that were robust to time-varying channels such as those found in HRI environments. In these cases, temporal-trajectory filtering techniques, such as RASTA or CMN, are not applicable. In [104] LNCC was shown to reduce time-varying spectral tilt in a speaker verification task. In [105], the use of LNFB features provided significant reductions in WER in a DNN-HMM ASR system with channel mismatch.

3.3. Generic ASR test bed for HRI

In contrast to the ASR integration on a black box basis as discussed above, this chapter proposes to consider not only the acoustic signal but also the operation conditions such as the environment, and robot and user state and contexts (see Fig. 3.4).

By environment it is understood basically the acoustic channel, reverberation conditions and the additive noise caused by the robot movement. Robot state and context denote all the information about current variables and operating conditions of the machine to generate a list of feasible or acceptable commands or information that could be input by the user. Finally, user state and context designate, among others, the user’s attitude, emotional conditions, and task completion status that

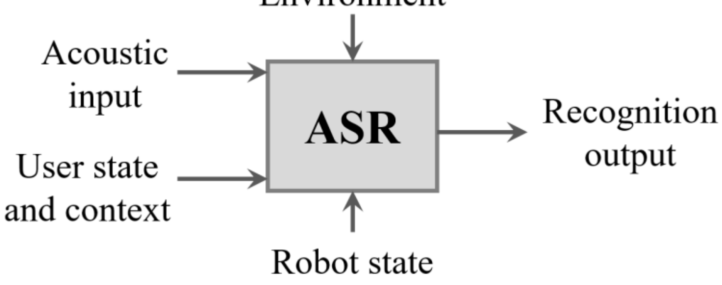


Figure 3.4 Proposed ASR integration in HRI scenarios.

can also predict user's command and info input to the robot. The full accomplishment of this kind of integration is far beyond the scope of this chapter, for that reason this section focusses on the environment representation and modeling by training the ASR engine with clean utterances combined with the acoustic-channel responses and noise that were estimated and recorded, respectively, with an HRI testbed. This testbed attempts to represent the generic acoustic environment of HRI in mobile robotics from the ASR point of view.

First of all, for instance, consider some real human social scenarios where robots could be very useful: a museum guide giving a tour, a student in a classroom asking the teacher a question, a rescue team helping a survivor and a team of chefs working in a restaurant. All these situations have something in common: a person talks to somebody else who is busy accomplishing a task and is not looking to who is talking to him/her. Also, the two individuals may be moving one with respect to the other.

As shown in Fig. 3.4, the proposed strategy considers the information related to the acoustic environment as one of the inputs of the ASR engine. In this chapter the acoustic environment is represented with the impulse responses that characterize the time-varying acoustic channel (TVAC) and the additive noise generated by the robot movement. The main advantage of this strategy is the fact that it is much more efficient than recording the training database in all the possible operating conditions. To record the testing speech data in a real mobile robot scenario, to estimate the channel impulse responses and to record the robot noise, a testbed that employs a loudspeaker and human speakers as sources plus a moving robot as a receiver was implemented.

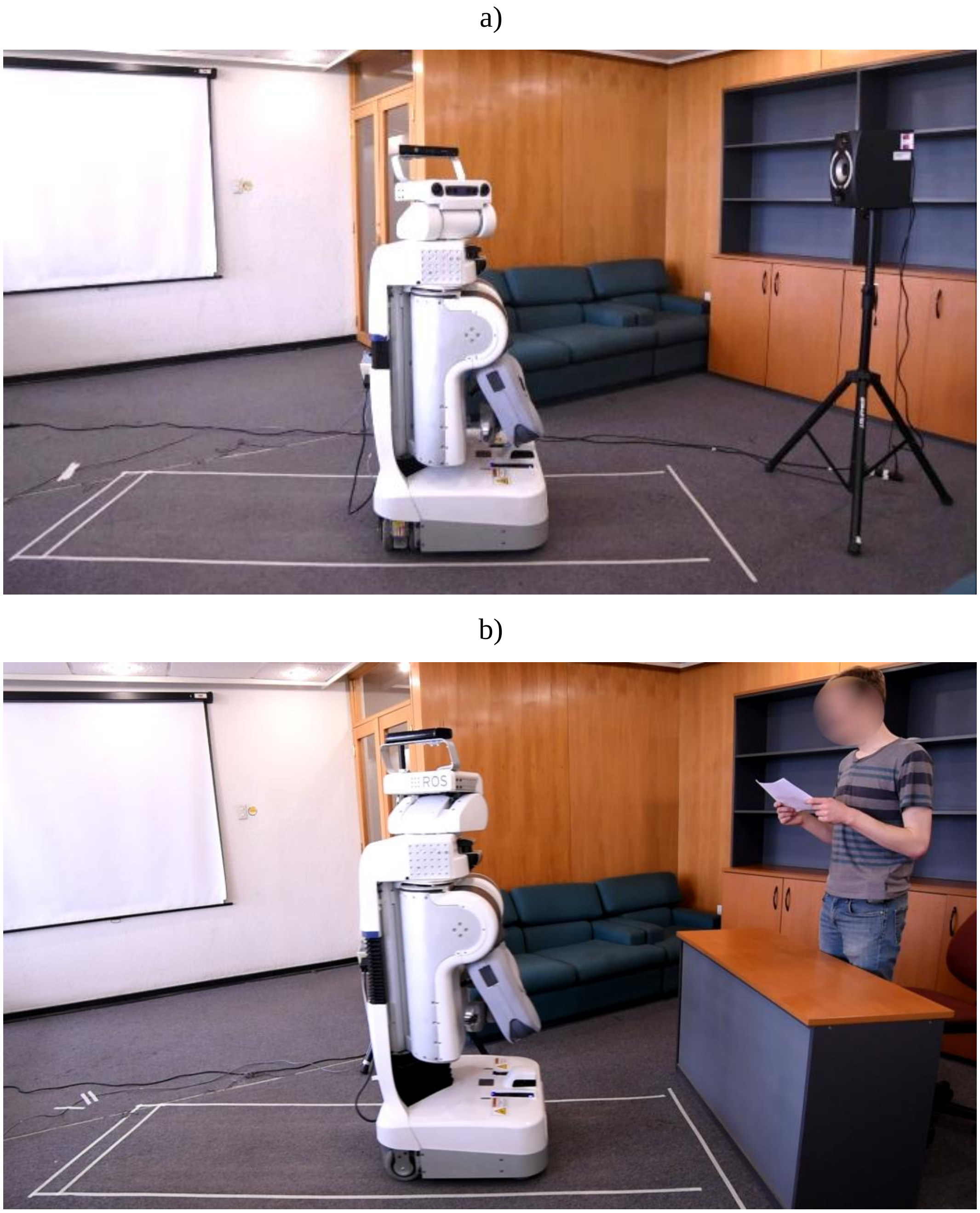
A preliminary version of this testbed was described in [106] where pilot experiments were reported. Because of the high relevance to the HRI community, a more complete version of this type of HRI scenario is proposed and described in the following sections. Particularly, different types of robot noise were recorded and included in the training procedure to represent more accurately the robot movement-conditions and the acoustic environment. Also, additional test sets were recorded by replacing the loudspeaker with human speakers in the same context.

3.3.1. Robotic platform and database recording

The experimental platform makes use of the PR2 (Personal Robot 2) shown in Fig. 3.5. The PR2 robot is equipped with a Microsoft Xbox 360 Kinect sensor mounted on top of its head. 330 clean testing utterances of the Aurora-4 database were re-recorded with the HRI testbed located in a meeting room (Fig. 3.6) including different specifications of the relative motion between the robot and the sources. Note that when the source and the robot are static one with respect to the other is a special case in relation to the more general situation (see Fig. 3.5). The two audio sources corresponded to a studio loudspeaker and four native American English speakers (two males and two females). The recording was performed by the PR2 Microsoft Kinect sensor, which contains a four-microphone array. The four signals received were summed to obtain a single channel signal. The recording procedure considered the relative movements of the robot microphones with respect to the sources by simultaneously applying translational movement to the robot body and angular rotation to the robot head.

3.3.1.1. Robot displacement

The robot moved towards and away from the source (*i.e.* the loudspeaker or the human speakers) between positions P1 and P3 (see Fig. 3.6). Three maximum robot displacement velocities were


a)

b)

Figure 3.5 PR2 robot equipped with a Microsoft Kinect that was used to record the database: a) the source corresponds to a studio loudspeaker that was employed to reproduce clean utterances from a database; and, b) the source is a human speaker reading sentences from the same corpus.

defined: $Vmax_1 = 0.30\,m/s$, $Vmax_2 = 0.45\,m/s$ and $Vmax_3 = 0.60\,m/s$. Those velocities were inspired by the discussions in [107], where a robot approached to a seated person at $0.2\,m/s$ and $0.4\,m/s$. In those conditions, none of the human participants found these robot speeds were too fast. Then, the maximum velocities mentioned above were multiplied by an acceleration and deceleration function.

Additionally, the recording of the test database was also performed with the robot in a static condition with respect to the source at position P1.

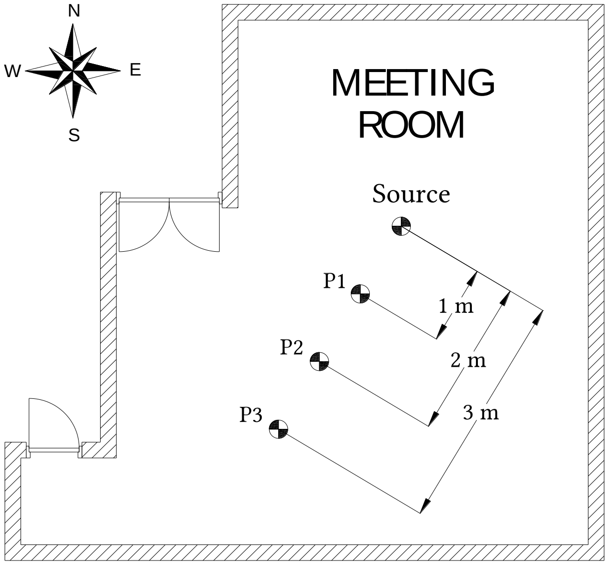


Figure 3.6 Meeting room where the HRI scenario was implemented. The robot moved towards and away from the source (*i.e.* the loudspeaker or the human speakers) between positions P1 and P3.

Consequently, four robot displacement scenarios were considered for the test data recording: three translational movements between P1 and P3 with maximum velocities $Vmax_1$, $Vmax_2$ and $Vmax_3$; and, a static position at P1.

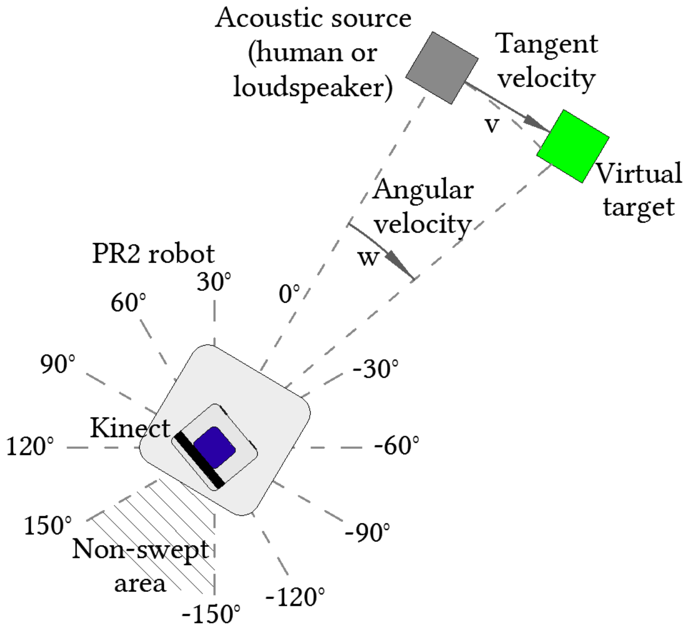


Figure 3.7 Movement of the PR2 robot head during the utterances recording. The head moves periodically from -150º to 150º and back at angular velocities equal to 0.28, 0.42 and 0.56 rad/s. Recordings with static head are performed at 0º. The selected angular velocities for the robot head emulates the situations where the robot follows with the head a target located two meters away and moving with linear velocities of 2, 3, and 4 km/h, respectively. The acoustic sources can be a loudspeaker or a human speaker. In both cases the sources were located at 0° with respect to the robot.

3.3.1.2. Robot head rotation

The robot makes turns with the head as shown in Fig. 3.7 for each of the four displacement conditions described above. The robot head moves periodically from –150º to 150º and back at three angular velocities. The sources are located at 0°. The three angular velocities ω_i for the robot head were made equal to: 0.28, 0.42 and 0.56 rad/s. The chosen angular velocities correspond to the angular speed of the head rotation necessary for the robot to follow a target with its head movement. The target is located two meters away from the robot and it is moving with tangential velocities of 2, 3 and 4 km/h, respectively, as shown in Fig. 3.7. A fourth angular motion condition was zero, fixing the robot's head at 0° (i.e., oriented towards the source) for each robot displacement described above.

3.3.1.3. HRI scenario testing databases

The combination of four conditions for robot displacement and four robot's head angular movements produces 16 test database recording conditions. Consequently, the total number of Aurora-4 clean testing utterances reproduced with the studio loudspeaker is equal to 330 utterances/robot-movement-conditions x 16 robot-movement-conditions = 5280 utterances. On the other hand, each of the four native American English speakers pronounced ten sentences from the Aurora-4 corpus per robot-movement-conditions. Those sentences were the same for the all the four speakers. As a result, the human speakers recorded 4 x 10 utterances/robot-movement-conditions x 16 robot-movement-conditions = 640 utterances. The average number of words per utterances is equal to 16.2 words. The vocabulary size in the testing data is 1270 words. It is important to mention that background noise was kept under control and measured before recording the test database at each robot movement condition. The equivalent sound pressure level over ten minutes was equal to 39 dBA. Instructions for requesting the HRI playback testing database are available at http://www.lptv.cl/en/hri-asr/. Further information about the testing database recording can be found in [108].

3.3.2. Representing time varying acoustic channel

TVAC in this HRI scenario can be modeled using a set of samples of the acoustic channel impulse responses. In this chapter 33 four-channel impulse responses (IRs) were computed with the robot placed at P1, P2 and P3 (Fig. 3.6), and for each robot position the head was oriented at 11 different angles with respect to the source. The head angle was varied from –150º to 150º in steps of 30º. Angle 0º corresponds to the Microsoft Kinect microphones oriented towards the sources in Fig. 3.7. The impulse responses were estimated using the Farina's sine sweep method [109]. An exponential sine sweep signal was generated from 64 Hz to 8 kHz and reproduced with a studio loudspeaker. The sweep audio was recorded with the four channel Microsoft Kinect sensor. An impulse response was estimated for each channel by convolving the corresponding recorded signal with the time-reversal of the original exponential sine sweep.

3.3.3. Noise recording

To incorporate additional information about the acoustic environment in the HRI scenario, different robot noise levels were recorded by the Kinect microphone array in the 16 robot movement conditions. The recorded noise was included in the ASR training procedure. The robot noise is generated by its internal fans and electrical motors operating at different translational and angular velocities. Finally, the four Kinect channels were summed to obtain a single channel signal.

3.4. Time-varying acoustic channels in ASR-based HRI

Speech recognition experiments were performed using the Kaldi Speech Recognition Toolkit [33]. Three training sets were employed, referred to as Clean, 1-IR, and 33-IR. The Clean training dataset consisted of the original utterances of the Aurora-4 database. The Clean training set consists of 7138 utterances from 83 speakers and contains only clean data recorded with a Sennheiser HMD-414 microphone.

The 1-IR training set was generated by convolving the 7138 utterances from the clean training set of the Aurora-4 database with the IRs, corresponding to the four Kinect channels, estimated when the robot-source distance was equal to 1 m and the angle between the robot head and source was 0º. For creating the 33-IR training set, 25% of the clean training set of the Aurora-4 database was convolved with the IRs, corresponding to the four Kinect channels, estimated when the robot-source distance was equal to 1 m and the angle between the robot head and source was 0º. The remaining 75% of the clean training set was convolved with the remaining 32 four-channel IRs in such a way that the 32 IRs were evenly distributed across the signals.

In this section, results obtained using the MelFB and LNFB feature vectors are compared. The DNN-HMM system is composed of DNNs with seven hidden layers and 2048 units per layer each, using a context window of 11 frames. The DNN-HMM systems were trained using alignments from a GMM-HMM recognizer trained with the same data. The GMM-HMM systems were trained using MFCC features, LDA, and MLLT, according to the tri2b Kaldi Aurora-4 recipe. First, a monophone system was trained; second, the alignments from that system were employed to generate an initial triphone system; and finally, the triphone alignments were employed to train the final triphone system. The number of units of the output DNN layer was equal to the number of Gaussians in the corresponding GMM-HMM system. The standard 5K lexicon and trigram language model were used.

3.5. Environment-based ASR training

The speech recognition experiments reported here made use of Aurora-4 database, which in turn was generated with the 5000-word closed-loop vocabulary task based on the DARPA Wall Street Journal (WSJ0) Corpus [31]. To generate the Environment-based Training (EbT) set, 25% of the clean training utterances of the Aurora-4 database, which consists of 7138 utterances (*i.e.* 15.2 hours) from 83 native English speakers and contains only data recorded with a high-quality microphone (*i.e.* Sennheiser HMD-414), was convolved with the IRs, corresponding to the four Kinect channels, estimated when the robot-source distance was equal to 1 meter and the angle between the robot head and source was 0º. Then, the four convolution results were summed to obtain a single channel signal. The remaining 75% of the clean training set was convolved with the remaining 32 four-channel IRs by employing the same procedure described above, in such a way that the IRs were evenly distributed across the signals. The recorded noise was added to this 75% of utterances using the Filtering and Noise Adding Tool FaNT [110] at SNR between 10 and 20 dB. It is worth highlighting that this training data is completely different from the testing databases described above, *i.e.* different speakers and different utterances.

In this section, the experiments were performed with a DNN-HMM ASR using the Kaldi Speech Recognition Toolkit [33], which is a state of the art and competitive ASR technology as mentioned above. To build a DNN-HMM system with Kaldi, first a GMM-HMM is trained with the EbT training data, using the tri2b Kaldi recipe for the Aurora-4 database. In this recipe, a monophone system is trained; then, the alignments from that system are employed to generate an initial triphone system; finally, the triphone alignments are employed to train the final triphone system. Also, MFCC parametrization of speech, LDA, and MLLT are part of the recipe. Once the GMM-HMM system is trained, the GMM is replaced with a DNN. The DNN is composed of seven hidden layers and 2048 units per layer each, and the input considers a context window of 11 frames. The number of units of the output DNN layer is equal to the number of Gaussians in the corresponding GMM-HMM system. The reference for the DNN training is the alignment obtained with the clean version of the whole training data and the GMM-HMM trained with the same clean data. This leads to a better reference for the DNN than using the noisy or corrupted speech data directly [111,112]. The DNN is trained firstly using the Cross-Entropy criterion. Then, the final system is obtained by re-training the DNN with the state-level minimum Bayes risk (sMBR) discriminative training [113]. The final ASR system is referred as EbT. For comparison reasons, a DNN-HMM system was trained with the clean database without any information regarding the HRI testbed scenario. Additionally, statistical significance analysis was performed using the NIST matched-pair sentence-segment word error test (MAPSSWE) [40].

For decoding, the standard 5K lexicon and trigram language model from WSJ were used [114]. As a result, the language model is tuned to the task, *i.e.* it is task dependent. The required files and scripts to generate the EbT training data and the detailed Kaldi recipe to train the DNN-HMM based ASR system employed here are available at http://www.lptv.cl/en/hri-asr/.

3.6. Results and discussions

3.6.1. Time-varying acoustic channels in HRI scenarios

Results were obtained for a total of 96 experimental conditions consisting of all permutations of the four displacement velocities: v equal to 0, 0.3, 0.45, and 0.6 m/s, four head angular velocities: ω equal to 0, 0.28, 0.42, and 0.56 rad/s, two types of feature extraction procedures (MelFB and LNFB), and three sets of training data (Clean, 1-IR, and 33-IR). Table 3.1 describes the WER obtained for each experimental condition. As can be seen in Table 3.1, the best results are observed for LNFB in all cases for each training condition. Note that 1-IR training achieves the best WER only for the case of a static robot, where test and training conditions match perfectly. Otherwise, the use of 33-IR training condition with LNFB features leads to a WER reduction greater than 54% when compared with a baseline system with MelFB features and Clean training.

On average, LNFB features outperform MelFB over all training conditions. The WER for LNFB is 19% (relative) less than for MelFB, in the Clean training condition, and 23% less in the 1-IR and 33-IR training conditions. A comparison of training conditions reveals that the use of 1-IR training leads to 35% and 32% WER reductions for LNFB and MelFB, respectively, compared to Clean training. This improvement most likely reflects the incorporation of the room and Kinect microphones responses in the training data. For the 33-IR training conditions, the WER is reduced by 56% and 53% with respect to Clean training, for LNFB and MelFB, respectively. These greater reductions in WER are due to additionally incorporating into the training data the three source-microphone distances and 11 head angles for each distance. In this way, the DNN-HMM system can also compensate for the channel variability caused by the robot movements.

As can be seen, the WER obtained when the robot is in motion is worse than when the robot is static at 1 meter from the source. This degradation increases linearly with the displacement

Table 3.1 WERs obtained using MelFB and LNFB features with different training conditions and different velocities of robot displacement and head rotation.

		Training condition					
Testing condition		Clean		1-IR		33-IR	
v (m/s)	ω (rad/s)	MelFB	LNFB	MelFB	LNFB	MelFB	LNFB
0	0	9.3	8.6	5.5	5.4	6.2	6.2
	0.28	52.4	43.8	29.0	22.2	14.8	12.9
	0.42	53.2	41.4	28.7	19.4	14.6	11.8
	0.56	54.5	42.1	28.1	19.9	14.3	11.9
0.3	0	36.2	25.9	18.4	12.9	15.9	10.6
	0.28	77.6	66.6	52.8	42.4	32.6	27.5
	0.42	77.0	65.8	52.4	43.9	33.2	27.2
	0.56	79.7	67.1	56.1	44.9	34.8	27.5
0.45	0	45.1	30.3	21.3	15.9	17.9	12.3
	0.28	83.3	68.6	62.3	49.2	42.2	33.0
	0.42	83.8	68.7	62.4	49.5	43.1	33.0
	0.56	84.4	70.5	65.5	49.8	43.5	33.0
0.6	0	55.3	33.4	28.5	19.5	26.5	15.5
	0.28	86.4	73.4	69.1	53.5	50.3	37.5
	0.42	85.8	69.4	66.6	50.8	48.5	36.5
	0.56	86.9	73.1	68.7	54.0	51.1	39.5

velocity and can be as high as 202% and 253% for LNFB and MelFB, respectively, for the greatest velocity. The results show that part of the degradation is caused by the robot motors noise, which was found to increase linearly with velocity. The effect of channel variability given by the robot movement towards and away from the source also increases with the displacement velocity, leading to an additional degradation. It is worth mentioning that the use of LNFB features reduces the WER respect to the WER obtained with conventional MelFB, confirming the natural robustness of the LNFB features with channel variability and channel mismatch.

WER is also worse when the robot head is undergoing rotational motion compared to when it is static. Nevertheless, this degradation is relatively independent of angular velocity, and can be as high as 151% and 116% for LNFB and MelFB, respectively, for the greatest velocity. It is worth mentioning that the percentage of occluded frames in each testing condition, *i.e.* frames for which the path from the source to the Kinect microphones is blocked by the Kinect encasement, is the same for each head angular velocity, except for the static head condition which does not produce any occluded frames. Moreover, the noise power of the head motors was found to be independent of the head angular velocity, except for the static head condition which produces no head motor noise.

3.6.2. EbT and comparisons with APIs

The average WER obtained with the 5280 utterances recorded in the HRI scenario (Section 3.3.1.3) with the loudspeaker, was equal to 65.0% using the ASR system trained with clean data. When only the IRs were incorporated in the training procedure, the average WER was dramatically reduced to 31.4%. Moreover, the EbT system (*i.e.* that includes both IRs and robot noise) provided a much lower WER: 11.6%. This dramatic increase of the ASR accuracy strongly supports the proposed approach to model the acoustic environment of an HRI scenario with channel impulse responses and robot additive noise. Observe that this WER was achieved with only 15 hours of training data. This result was corroborated by making use of the testing data set that was recorded with the four native American English speakers: 73.5% and 20.1% with clean training and EbT, respectively. These WERs are higher than those obtained with the playback testing data. This must be due to the fact that the human speakers pronounced the utterances with a lower volume resulting in a lower signal-to-noise ratio. Actually, the average SNRs were equal to 11 and 18 dB for human speakers and loudspeaker data, respectively.

For comparison reasons, ASR experiments with three publicly available APIs by using the "Speech Recognition" Python library (Version 3.7) [115]: the Google Web Speech API (Google API); the IBM Speech to Text API (IBM API); and, the Bing Voice Recognition API (Bing API) were performed. Fig. 3.8 shows the WER obtained with the EbT system and the three API mentioned above with the 330 clean utterances from Aurora-4. As it can be seen in Fig. 3.8, the EbT system provided the lowest WER that is 34% lower than the second best (statistically significant with $p < 0.001$), *i.e.* IBM API. This result suggests that adopting a better tuned language model, as done in the EbT ASR system, provides a clear advantage over a flatter or more general-purpose language model.

In the HRI test sets, it was observed that in challenging scenarios the APIs evaluated here delivered empty strings as the result of the ASR queries. Given this situation, the WERs were estimated with the non-empty returned text strings. Table 3.2 presents the ASR results obtained with the EbT system, Google API and IBM API, in all the robot motion conditions, with the playback loudspeaker testing database (Section 3.3.1.3). In the case of Bing API, the number of empty strings per each test set or empty string rate (ESR), increased dramatically and prevented us from showing a representative WER. All the ASR results with the APIs shown in Table 3.2 were carried out between September 6th and 12th, 2017. According to Table 3.2 the lowest and highest WERs were achieved with the static condition (*i.e.* translation and angular velocities equal to zero), and with the highest displacement and rotational velocities, respectively, with EbT, Google API and IBM API. Also, the lowest WER for each robot movement condition is achieved with EbT.

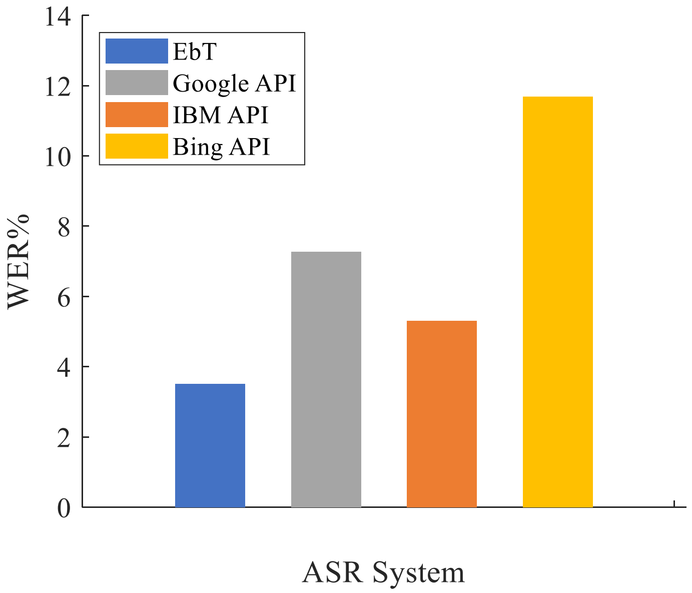


Figure 3.8 WERs obtained with the EbT system and the publicly available ASR APIs. The testing data corresponds to the original clean test set from Aurora-4 database (330 utterances).

Table 3.2 WERs obtained with the EbT system, Google API and IBM API. The testing sets correspond to the playback loudspeaker sub databases recorded at each combination of robot displacement and robot head angular velocities.

		ASR System			
v (m/s)	ω (rad/s)	Clean training	EbT	Google API	IBM API
0	0	8.63	4.11	7.94	9.27
	0.28	52.79	7.73	9.94	27.88
	0.42	53.09	7.68	10.91	25.94
	0.56	54.06	7.83	11.32	26.57
0.3	0	36.04	5.66	9.32	22.71
	0.28	76.42	12.22	16.31	49.23
	0.42	76.59	12.33	17.39	51.68
	0.56	78.80	12.82	17.88	51.3
0.45	0	43.36	6.16	9.04	22.55
	0.28	82.94	16.74	19.30	53.03
	0.42	83.49	15.28	20.73	54.69
	0.56	83.73	15.45	22.31	55.93
0.6	0	53.30	7.49	10.78	29.57
	0.28	85.80	18.27	22.06	56.9
	0.42	85.07	17.47	22.91	56.37
	0.56	86.46	18.68	24.42	58.19
	AVG.	65.04	11.62	15.79	40.74

Fig. 3.9 summarizes the WERs obtained with EbT, Google API and IBM API, in all the robot movement conditions shown in Table 3.2, with the playback loudspeaker testing database (Section 3.3.1.3). As can be seen in Table 3.2 and Fig. 3.9, the lowest WER correspond to the EbT system. The average WER achieved with the EbT system is 26% lower than the second best (statistically significant with $p < 0.001$), *i.e.* Google API. According to Fig. 3.9, the EbT system and Google API provided the lowest WER dispersion. Also, the observed average ESRs were equal to 0%, 0.3% and 6.5% with EbT, Google API and IBM API, respectively. If the empty strings were

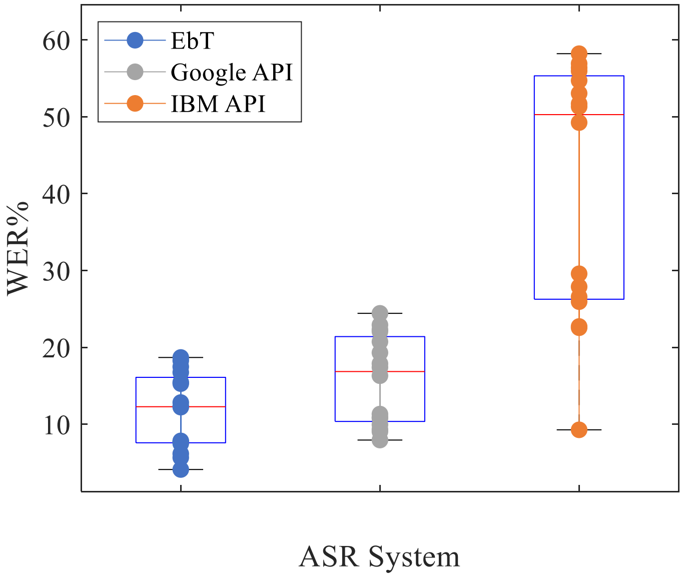


Figure 3.9 WERs obtained with the EbT system, Google API and IBM API in all the robot movement conditions shown in Table 3.2, with the playback loudspeaker testing database (Section 3.1.3).

included in the computation of the error rates, the WERs increased to 15.9% and 42.6% with Google API and IBM API, respectively. With EbT the WER was not modified because ESR is equal to zero in this case.

For validation purposes, Fig. 3.10 summarizes the WERs obtained with EbT, Google API and IBM API, in all the robot movement conditions shown in Table 3.2, with the native American English speaker testing database (Section 3.3.1.3). According to Fig. 3.10, the lowest value and dispersion for WER also corresponds to the EbT system. The average WER achieved with system EbT is 38% lower than the second best, i.e. Google API. The average ESRs with the human speaker testing data set are equal to 0%, 5.8% and 5.6%. If the empty strings were included in the computation of the error rates, the WERs increased to 35.0% and 57.1% with Google API and IBM API, respectively. The results with the ASR APIs using the native American English speaker testing database were obtained between September 25th and October 5th, 2017.

By comparing Fig. 3.8 with Fig. 3.9, it can be observed that the lowest WER is achieved with the EbT system. However, the EbT system also provides the highest relative increase in average WER, *i.e.* 231%, from the clean testing data to the playback loudspeaker testing database (Section 3.3.1.3) in the HRI scenario. In contrast, Google API, for instance, shows a relative increase in average WER equal to 117%. This result can be explained according to [116], [117] and [118], where it is said that the ASR engines that support the APIs evaluated here could have been trained with at least thousands of hours of speech covering a wide diversity of acoustic conditions. In contrast, the EbT system was trained with only 15.2 hours of clean speech utterances that were convolved with channel impulse responses and had noise added (Section 3.5). The proposed procedure is applicable to any HRI environment, being only necessary the capture of the robot noise and the estimation of the acoustic impulse responses to get a new EbT system. Implementing this new system is very practical and simple to make, because this procedure requires just a couple of days and a few hours of training data. At this point it is worth highlighting that the

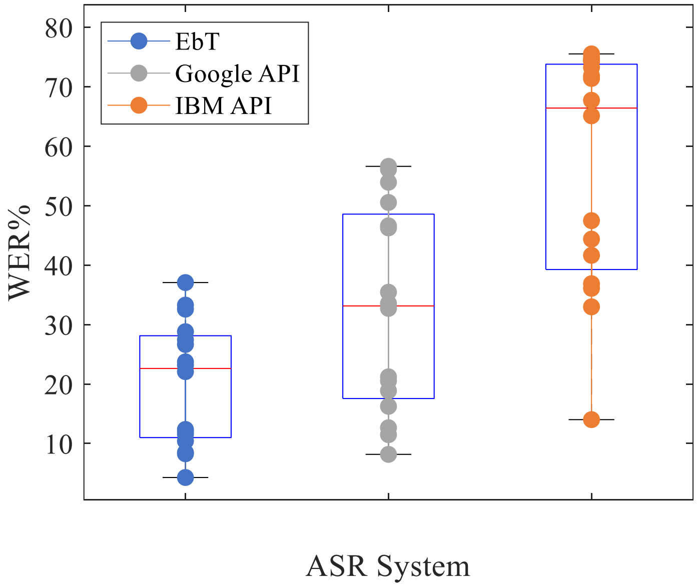


Figure 3.10 WERs obtained with the EbT system, Google API and IBM API in all the robot movement conditions shown in Table 3.2, with the native American English speakers testing database (Section 3.1.3). The average WERs were 20.1%, 32.6% and 56.4% with EbT, Google API and IBM API, respectively.

adequate use of user and robot states and contexts can reduce the language model perplexity, and lead to further improvements in recognition accuracy.

3.7. Conclusions

Locally-Normalized Filter-Bank features and DNN-HMM training strategies were employed to address the problem of time-varying channels in speech recognition-based human-robot interaction. Time-varying channels were generated by performing displacement movements and head rotations at different speeds with respect to a source location that remained fixed. The use of 33-IR training produced reductions in WER greater than 50% compared to Clean training with both LNFB and MelFB. However, LNFB provided a WER 23% lower than MelFB with 33-IR. When compared with Clean training and MelFB, 33-IR and LNFB led to a reduction in WER equal to 64%.

It is proposed to replace the popular black box integration of automatic speech recognition technology in HRI applications with the addition of the HRI environment representation and modeling, and the robot and user states and contexts. Then, as a consequence of this strategy, this section was focused on the environment representation and modeling by training a DNN-HMM model based automatic speech recognition engine with the combination of clean utterances with the acoustic-channel responses and noise that were estimated and recorded, respectively, with an HRI testbed built with a PR2 robot. The proposed procedure is much more effective and efficient than recording a training database in all the possible acoustic environments, given an HRI scenario. Also, different speech recognition testing conditions were generated by recording two types of acoustic sources, *i.e.* a loudspeaker and human speakers, using the PR2 robot, which has a Microsoft Kinect sensor mounted on top, while performing head rotations and movements towards and away from the fixed sources. This testbed models the generic problem of HRI in mobile robotics, and the resulting automatic speech recognition accuracy outperformed publicly available speech recognition APIs. The word error rate achieved by the EbT system is at least 26% and 38% lower than the evaluated APIs with the loudspeaker and human testing databases, respectively, with a limited amount of training data. Other factor in HRI scenarios is that the user speech may be stressed in noisy conditions, *i.e.* Lombard effect. This problem, and the incorporation of user and robot states and contexts are proposed for future research.

Chapter 4
Combining DNN-based acoustic models improves speech recognition accuracy in reverberant environments

4.1. Introduction

Many speech applications require that the user is not tethered to a close-talking microphone. Examples of such applications include automatic meeting transcription, voice dialogue systems for devices in smart homes, and interaction with humanoid robots, which are more intuitive, comfortable and effective if the user could interact with microphones on the device or in some third location, independent of the user. In many of these scenarios, the talker could be located several meters away from the microphone, and the received signal could be corrupted by interfering sounds, such as background noise and interfering speakers. In addition, speech in rooms is corrupted by the effects of reverberation caused by reflections of the speech from the surfaces of the room and the objects that are in it [119]. The effects of reverberation are a major problem in distant-talking ASR.

Reverberation and background noise decreases speech intelligibility and speech quality. This especially affects the performance of ASR systems, which are not as robust to reverberation as the human auditory system [120]. These performance degradations depend on the nature of the environment and make such systems less effective [121]; therefore, far-field speech recognition remains a challenge. One frequently-used measure of reverberation is the reverberation time (RT), which is defined as the time required for sound pressure level to decay by 60 dB. Offices and home environments typically have an RT from around 0.5 to 1.0 seconds, and longer RTs lead to greater reverberation distortion and greater degradation in ASR accuracy. Although RT is the most commonly-used general descriptor of reverberation, there are many other aspects of rooms that impact on speech intelligibility and ASR accuracy [122].

4.1.1. Reverberation model

Reverberant speech is usually modeled as the convolution of clean speech $x(t)$ with a room impulse response (RIR) $h(t)$ [7,123,124,125,126] according to:

$$y(n) = x(t) * h(t)\,. \tag{4.1}$$

The RIR $h(t)$ reflects the reverberation properties of the room and depends on the acoustic absorption of the surfaces in the room, the layout of the room, and the location of the speaker, microphone, and other objects in the room. The RIR is frequently described in three parts: the direct speech signal, the early reflections, and the late reflections. Early reflections consist of the initial discrete reflections that arrive at the microphone, which frequently occur within 50 ms of the arrival of the direct component. These reflections vary with the relative positions of the speaker and microphone and when combined with the direct signal can improve the perception of the human auditory system [127]. Late reflections correspond to the response that develops when the components arrive so frequently that $h(t)$ approximates a continuous function of time. The later reflections are typically modeled as an exponential decay and are independent of speaker and microphone position. They are believed by some to be the primary source of degradation in ASR systems (*e.g.* [128]).

While reverberation could be treated as a convolutional distortion for clean speech, in real rooms reverberation times are generally much longer than the 20-35 ms of a typical analysis frame for ASR and similar tasks. Furthermore, the properties of the reverberant environments vary in time and space. For these reasons, traditional methods proposed to reduce convolutional distortion, such as cepstral mean normalization (CMN) [129] and relative spectral (RASTA) filtering [130,131], do not achieve good results in reducing the reverberation effect for ASR as they can only remove or compensate for the short-term convolutional distortions of systems that have short impulse responses [123,132].

Several methods have been proposed to address reverberation distortion, and they are usually divided in three classes according to the stage in which they are implemented: ASR front-end, ASR back-end, and speech preprocessing [133,134,135]. Methods implemented in the ASR front-end aim to improve the feature robustness to reverberation. Examples of these methods includes: algorithms to handle missing or unreliable speech regions that are dominated by noise [135,136,137] and feature-extraction methods based on the normalization of sub-band temporal modulation envelopes [132,138]. Methods that are implemented in the ASR back-end aim to improve the robustness of the ASR system by adapting the acoustic model, as in [124,139,140]. Finally, there are speech signal preprocessing methods that are applied before the derivation of the feature vectors, such as speech reverberation suppression [133].

Many recent proposed methods [7,141,142,143] have been developed and tested using the data distributed through the REVERB challenge [144]. This challenge was organized to address the lack of common data sets with which to evaluate progress in the field of reverberant ASR and speech enhancement (SE). A common speech corpus was proposed to evaluate SE and ASR developments in reverberant conditions. Various acoustic environments and different levels of reverberation and stationary noise were considered. The challenge data set also included real recordings carried out in a meeting room.

In [141], different robust features are explored for use in a convolutional DNN (CDNN) based acoustic model for recognizing continuous speech in a reverberant condition. The features were motivated by human auditory perception and speech production for their experiments. Damped Oscillator Coefficients (DOC) [145], Normalized Modulation Coefficients (NMC) [146] and Gammatone Filter Coefficients (GFCs) (a linear approximation of the auditory filterbank performed in the human ear), are some of the features that were employed. The authors obtain WERs as low as 30.40% using NMC, and 28.65% when applying combined GFC and NMC to the single channel data of the REVERB 2014 challenge.

In [147], the authors address the problem of distant speech recognition for reverberant noisy environments. They propose a double-stream architecture combining a state-of-the art GMM system with a deep long short-term memory (LSTM) recurrent neural network (RNN) trained to predict frame-wise phoneme estimates, which are converted into observation likelihoods to be used as an acoustic model. LSTM-RNNs can learn long-range temporal context, by making use of memory cells in the hidden units. This capability leads to an increase of the robustness against noise and reverberation. They employed a double-stream HMM system that, in every time frame, has access to two independent information sources, the acoustic likelihoods of the GMM and the LSTM predictions. They show that the LSTM system can improve a robust state-of-the-art GMM system. Experiments were carried out on the medium-vocabulary task of the 2nd ‘CHiME’ Speech Separation and Recognition Challenge, which includes reverberation and highly variable noise.

The challenge baseline WER was reduced by 64% relative on average. Additionally, results show that speech enhancement using NMF, although it leads to improvement in the GMM system, it does not improve the results when combining GMM with LSTM.

4.1.2. Speech dereverberation techniques

Several enhancement algorithms have been proposed to address the effects of reverberation, SSF [148], NMF [149], and WPE [150,151]. The SSF algorithm is motivated by the precedence effect, which is the observation that the human auditory system appears to emphasize the first-arriving components of sounds in reverberant environments. SSF performs an onset enhancement at the peripheral level and steady-state suppression on a band-by-band basis [148]. In contrast, NMF accomplishes blind deconvolution of the response to a reverberated signal in the frequency domain [149]. It is easy to observe that the presence of reverberation causes a representation like a spectrogram to become blurred or smeared along the time axis, caused by convolution of the response representing clean speech with the sample response of the room acoustics, as represented in the frequency domain. Because phase information is lost in the spectrogram, blind deconvolution cannot be accomplished exactly, but a good approximation can be achieved by exploiting the facts that the matrix representing the sample response in the frequency domain would be non-negative and sparse. Finally, the WPE algorithm is focused on robust blind deconvolution based on long-term linear prediction, which aims at late reverberation reduction. In this way, the algorithm receives a single-channel speech signal which may contain multiple speakers, background noise, and reverberation. The de-reverberation is performed based on long-term linear prediction in the short-time Fourier transform (STFT) domain and provides low speech distortion.

4.1.3. Signal representation and pre-processing

In recent years, ASR system has experienced a large improvement by the replacement of conventional GMM models with DNNs for acoustic modeling. With this change, some researchers have found that the engineered features used for training GMMs are not optimal for DNNs: while diagonal GMMs are better trained with de-correlated features such as MFCC, deep models can better learn from correlated features [152,153].

Several works have focused on identify the best features to train deep models [154,155]. One of the approaches to learn new features is to combine engineered features using neural networks [156,157,158]. In particular, Tüske et al. trained several ASR systems using raw time signals, the FFT and engineered features such as MFCC, PLP and Gammatone (GT) [157]. They found that the best system was obtained using a combination of MFCC, PLP and GT, using either 50h and 250h of training data. These results suggest that: the differences between engineered features benefit the DNN training, and the improvement due to combination does not depend on the amount of training data.

4.1.4. Classifier fusion

Multiple Classifier Systems (MCS) is a powerful method for increasing classification rates in pattern recognition problems. MCS has been successfully applied in many and different fields such as finance, medical diagnosis, security, remote sensing, pattern recognition, between others [159,160,161,162]. In fact, MCS has shown good performance in almost any field in which pattern classifiers are used [163]. Currently, MCS still a very active area with many researches in machine learning and pattern recognition, and several approaches are currently used to construct an MCS [164,165]. However, there is no fusion method that can obtain the optimal classification performance for all the applications. Therefore, the study of multiple classifier fusion is still an open problem [166].

Particularly, in ASR many research groups have noted that the performance of ASR systems can be improved by the combination of information from multiple parallel feature streams. In fact, the best systems in most international evaluations of speech processing technologies are based on the combination of multiple contrasting systems, which are ultimately combined to produce a final hypothesis.

The algorithms currently used to combine information from parallel feature sets can he broadly viewed as belonging to one of three classes that operate at three different levels of information processing. The first type of combination approach, which is sometimes referred to as feature combination or input combination, typically concatenates different independent or correlated feature vectors to form a larger feature vector, performing recognition based on the values of the combined features, sometimes using a dimensionality-reduction algorithm such as LDA or principal components analysis (PCA) for dimensionality reduction of the feature vector and/or feature decorrelation (*e.g.* [167]). The last paragraph of Section 4.1.3 provides some specific examples of feature combination. The second type of combination approach, which has been called state combination, probability combination, and middle combination, refers to methods that combine information from parallel streams at the stage at which probabilities are evaluated in the search process using one of several combination methods (*e.g.* [168,169]). The third (and perhaps most popular) type of combination approach has been called score combination, hypothesis combination, output combination, or lattice combination, refers to methods that combine parallel sources of information after the search procedure is completed. These methods include the well-known ROVER method [170], confusion network combination (CNC) [171], the Hypothesis Combination method [172], as well as various lattice-combination methods (*e.g.* [173,174]).

4.1.4.1. Lattice combination and minimum Bayes risk-based decoding

One approach to combining systems is the lattices combination and using the minimum Bayes risk (MBR) decoding [175]. This lattice combination and MBR-based decoding (LC/MBR) method represents an alternative to ROVER [170] and CNC [171].

4.1.4.2. Linear combination of scores

Another approach to combine multiple systems is to make a weighted sum of the scores delivered by the different systems. The linear combination of systems is a widely used approach in the literature. However, finding the weighting parameters is a subject that is far from being solved.

In general, the degree of improvement in recognition accuracy obtained from combination of parallel feature streams depends in large part on the extent to which the information that is being combined is complementary (*e.g.* [176]). In any particular application which uses parallel information sources, the choice of the specific type of combination method to be used typically depends on the type of application, classifier architecture, and available computation resources, among other factors.

This chapter explores multi-DNN and multi-lattice combinations to obtain more robust systems for reverberant environment. Also, the optimization of the DNN linear combination is exhaustively explored. For the most part, the information combination methods considered in this chapter involve various forms probability combination.

4.2. Engineered features and reverberation

The comparison of different robust features in combination with enhancement techniques in a controlled highly-variable real reverberant environment is not found in the literature. In this section the robustness of LNFB and MelFB features in combination with NMF, SSF and WPE enhancement methods is discussed and evaluated regarding RT and speaker-microphone distance with clean and reverberated training. For this purpose, preliminary experiments were carried out to explore the aforementioned variables.

4.2.1. Preliminary training data

Speech recognition experiments were performed using the Kaldi Speech Recognition Toolkit [33]. The Clean training set from the Aurora-4 database was employed. This set contains 7138 utterances from 83 speakers recorded with a Sennheiser HMD-414 microphone. Additionally, a reverberant training set was developed, referred to as “Reverb.”

For Reverb training, simulations were made with the simulation program Room Impulse Response Generator [177], which uses the image method assuming a rectangular room [178]. In order to avoid potential artifacts in training because of potential standing wave patterns that may develop in rectangular rooms, the Reverb training database consists of 5353 utterances that were passed through 5353 different randomly-generated room impulse responses (RIRs). The dimensions of the simulated rooms varied from RIR to RIR with an average of 7.95 meters length, 5.68 meters width and 4.5 meters height, approximating the dimensions of the larger-sized reverberation chamber of the Acoustic Institute. The dimensions for each individual RIR were drawn from uniform distributions over the range of plus or minus 20 percent of the nominal values stated above. A nominal RT was then selected by sampling a random variable over the range of 0.45 to 1.87 seconds, and the nominal average absorption and reflection coefficients that would provide the selected nominal RT were calculated using the Sabine equation [179]. Six separate reflection coefficients, one for each room surface, were drawn from a uniform distribution between plus and minus 10 percent of the nominal reflection coefficient calculated from the Sabine equation, resulting in a room with a reverberation that was random, but close to the intended nominal value. The distance between speaker and microphone was drawn from a uniform distribution between 0.144 and 2.816 meters. The speaker and microphone were placed in random locations at the room, using the distance that was selected for a particular trial, with the constraints that both speaker and microphone are at least 1 meter from any wall and between 1 and 2 meters from the floor.

4.2.2. Preliminary system training

Two types of feature vectors were compared in this section, the MelFB and LNFB features, in both cases considering a context window of 11 frames, including 5 frames before and 5 frames after the current frame. Each DNN in the DNN-HMM system consists of seven hidden layers and 2048 units per layer. The DNN-HMM systems were trained using alignments from an GMM-HMM recognizer trained with the same data. In turn, the GMM-HMM systems were trained by using MFCC features, LDA, and MLLT, according to the tri2b Kaldi Aurora-4 recipe. First, a monophone system was trained; second, the alignments from that system were employed to generate an initial triphone system; and finally, the triphone alignments were employed to train the final triphone system. The number of units of the output DNN layer was equal to the number of Gaussians in the corresponding GMM-HMM system. For decoding stage, the standard 5K lexicon and trigram language model were used.

4.2.3. Preliminary results and discussion

The results were obtained for a total of 330 testing utterances for each one of the 20 reverberation conditions (four RTs and five microphone-speaker distances) available in the highly-reverberant real environments (HRRE) database [180]: RTs equal to 0.47, 0.84, 1.27, and 1.77 seconds; and, microphone-speaker distances equal to 0.16, 0.32, 0.64, 1.28, and 2.56 meters. Two types of feature extraction procedures (MelFB and LNFB), two sets of training data (Clean and Reverb) and four types of environmental compensation (none, NMF, SSF, and WPE) were combined.

Table 4.1 describes the WERs obtained for each speaker-microphone distance averaged across the four RTs that were available in the reverberation chamber. The lowest WER for each column is highlighted in bold in Table 4.1. As can be seen in Table 4.1, the best results are observed for Reverb training with MelFB combined with WPE in most cases. The best MelFB features perform better than the best LNFB features (in conjunction with Reverb training) averaged over all RTs. Compared with the baseline system with MelFB and Clean training condition, the optimal reductions in Table 4.1 are higher than 70% with all the speaker-microphone distances.

4.2.3.1. Training procedure

According to what has been mentioned about multi-style training, the best results are achieved with Reverb training in most test conditions. However, as can be seen in Fig. 4.1, Clean training in combination with WPE achieves better performance than Reverb training in four of the twenty conditions: RT equal to 0.84 and 1.27 seconds at 2.56 meters using LNFB; and, with RT equal 0.47 seconds in the shortest distances (*i.e.* 0.16 and 0.32 meters) using MelFB.

Table 4.1 WERs averaged across all RTs values using MelFB and LNFB for different training conditions and pre-processing techniques in the HRRE database.

	Training	Feature	Speaker-microphone distance (m)				
			0.16	0.32	0.64	1.28	2.56
Baseline	Clean	MelFB	34.1	55.5	70.2	78.9	84.7
		LNFB	18.7	32.6	53.0	69.1	79.5
	Reverb	MelFB	13.3	16.3	21.7	31.1	36.4
		LNFB	14.0	17.7	22.2	30.1	34.8
NMF	Clean	MelFB	16.4	25.5	38.9	56.3	67.8
		LNFB	14.3	20.8	30.6	49.6	62.5
	Reverb	MelFB	11.9	14.3	17.6	26.2	31.9
		LNFB	12.6	15.1	17.9	26.0	32.0
SSF	Clean	MelFB	14.9	22.0	34.5	53.7	65.7
		LNFB	12.3	18.0	27.2	46.2	59.8
	Reverb	MelFB	11.0	12.6	15.0	21.9	26.2
		LNFB	11.5	12.6	15.2	21.2	25.0
WPE	Clean	MelFB	9.8	19.1	39.9	61.1	72.8
		LNFB	7.9	13.9	29.0	53.2	67.3
	Reverb	MelFB	8.7	10.0	13.1	20.0	25.5
		LNFB	9.8	11.4	14.2	21.0	26.3

4.2.3.2. Effect of enhancement techniques

As discussed above, the NMF, SSF and WPE techniques were designed to reduce the mismatch between training and testing conditions. As seen in Table 4.1, the application of this techniques is always helpful no matter which training data are used. Additionally, SSF always outperforms NMF for the examined conditions. On the other hand, WPE surpasses SSF in all distances only with Reverb training.

The use of WPE in combination with MelFB and Reverb training, and averaging across all RTs, produces the best system for speaker-microphone distances greater than 0.32 meter. For the speaker-microphone distance of 0.16 meter, the best result is obtained with WPE with Clean training and using the LNFB features. The use of WPE in combination with MelFB and LNFB provides the best results for almost all test conditions, except for the greatest RTs at the longest distances, *i.e.* RT equal to 1.27 and 1.77 seconds at a speaker-microphone distance equal to 2.56 meters, where SSF combined with LNFB and Reverb training leads to greater accuracies (see Fig. 4.1).

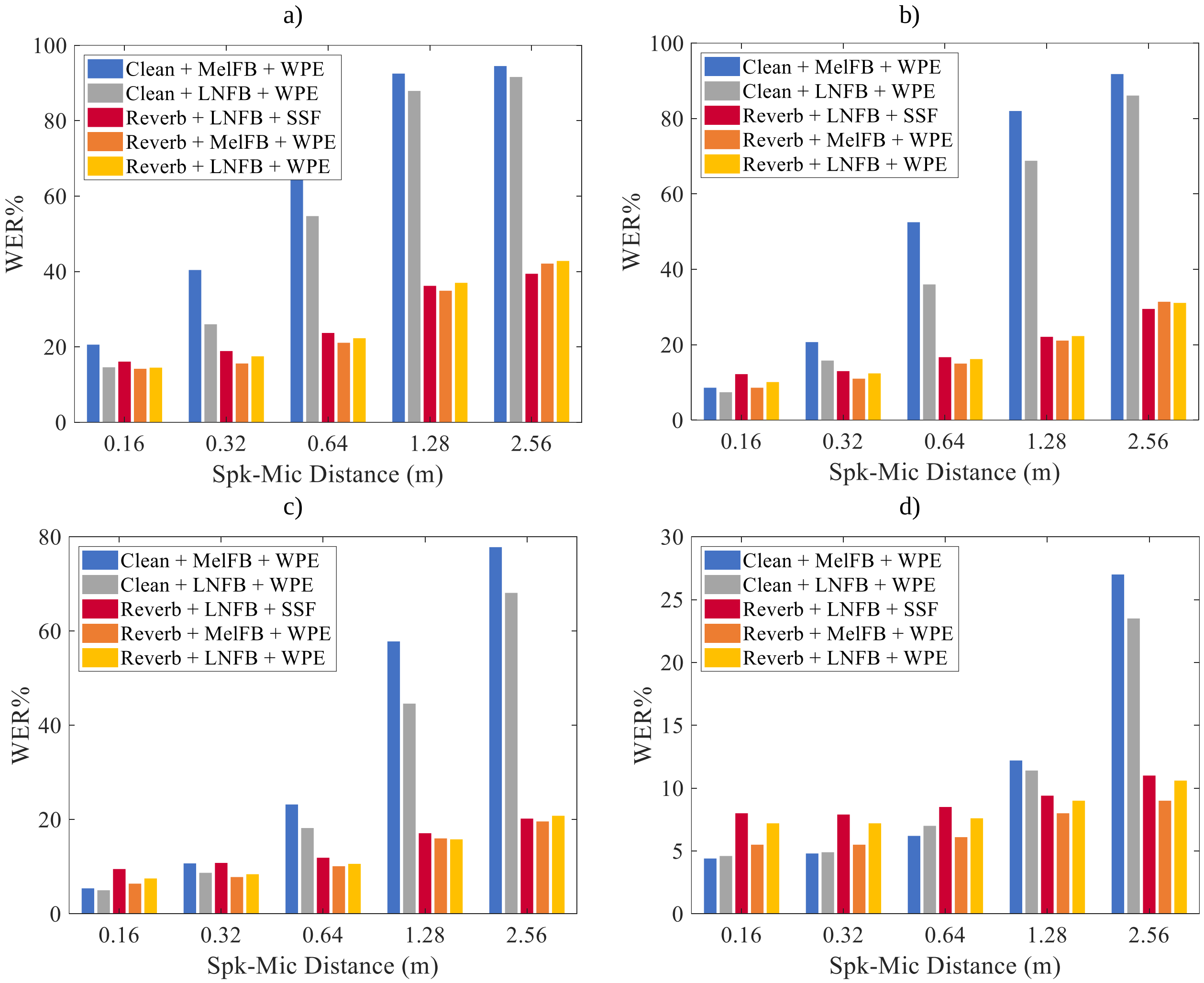


Figure 4.1 Results for the best ASR systems for a) RT=1.77 s, b) RT=1.27 s, c) RT=0.84 s and d) RT=0.47s.

4.2.3.3. Performance of MelFB versus LNFB features

Figure 4.1 compare directly the best systems obtained using the MelFB and LNFB features. MelFB achieve the best WER in several cases. Nevertheless, as can be seen in Fig. 4.1, LNFB exhibits better accuracy in some critical RTs and distances, *i.e.* with RT equal to 1.27 and 1.77 seconds at a distance of 2.56 meters. On the other hand, LNFB worked better in the shortest distance, *i.e.* 0.16 meter, for RT equal to 0.84 and 1.27 seconds.

4.2.3.4. Complementarity between ASR systems

Despite the fact that on average the use of MelFB in combination with WPE and Reverb training provided the lowest WER, different combinations of features, training data and enhancement techniques could address more effectively some testing conditions. The results shown in the Fig. 4.1 suggest that there is some degree of complementarity between systems trained with different data, enhancement, and parametrization methods. Although always the best system can be selected, also the best engines can be combined to obtain a new system that could be even more accurate in different test conditions.

4.2.4. Preliminary findings

Two training conditions were evaluated: Clean and Reverb. The comparisons also included the NMF, SSF, and WPE environmental compensation algorithms. The results presented here show that the lowest average WER is achieved using Reverb training and MelFB features combined with WPE. With Clean training, *i.e.* significant mismatch between testing-training conditions, LNFB features clearly outperform MelFB parameters.

Generally, the use of the NMF, SSF and WPE compensation techniques improves WER for LNFB and MelFB features, for both training styles. Specifically, with Reverb training the use of WPE and LNFB provides WERs that are 3% and 20% lower in average than SSF and NMF, respectively. WPE and MelFB provides WERs that are 11% and 24% lower in average than SSF and NMF, respectively.

It is worth highlighting that for some test conditions some systems led to higher accuracies than MelFB/WPE. These results strongly suggest that there is complementarity among the different engines tested here.

4.3. DNN linear combination

4.3.1. DNN complementarity

It is well known that DNN acoustic modelling outperforms ordinary GMM models. This may be the result of the fact that DNN can learn robust speech representations in the deeper layers. These representations would be jointly learned when DNN is trained as a classifier. As a result, using raw signal as an input has become a real option and many authors have suggested that engineered features like PLP or cepstral features are not essential to achieve high ASR accuracy [81,153,181]. This has consolidated the notion that only a large amount of data is sufficient to train a deep learning-based ASR independently on how the input signal is pre-processed. In contrast, some authors have argued that it is still useful to combine different features to learn new speech representations [156,157].

In the DNN training, the backpropagation algorithm is applied after pre-training. The backpropagation algorithm relies on the computation of local gradients, which are used to compute the weight corrections $\Delta w_{ji} = \eta \delta_j y_i$ [182], where: Δw_{ji} is the correction applied to the weight that connects neuron i to neuron j; δ_j is the local gradient associated with neuron j; and, y_i is one of the input signals to neuron j and the output from neuron i. If neuron j belongs to the first hidden layer of the network, then the above equation can be written as: $\Delta w_{ji} = \eta \delta_j x_i$, where x_i is the ith input feature. Consequently, if the speech enhancement or parametrization change, the DNN training process will achieve different solutions for weights and bias $\{\boldsymbol{W}, \boldsymbol{b}\}$ independently of the training data size, despite the fact that the same network parameter initialization and training data are used. As a result, complementary DNNs could be obtained by just modifying the input signal representation. Observe that the increase of the training data size should provide a reduction in WER independently of the speech enhancement or parametrization methods. However, according to what discussed here, two DNNs trained with different speech representation methods could provide complementary information although they may lead to the same ASR accuracy individually. On other words, the complementarity of DNNs should not depend on the training data size. This is the main motivation to explore DNN combination to address the problem of robust ASR, particularly in reverberant environments.

There are several methods for classifier fusion such as the maximum rule, minimum rule, mean rule, product rule, weighted majority vote rule and linear combination [183,184,185]. This chapter will focus on the latter one. Given R DNNs, the linear combination of scores is defined as:

$$\widehat{m}(s,n) = \sum_{r=1}^{R} \omega_{r,s,n} \cdot m_r(s,n) \ , \tag{4.2}$$

where $\widehat{m}$ is the combined score (*e.g.* LLK or log-posterior probabilities) for the DNN s^{th} output (also state or senone) in input frame n^{th}; $m_r(s,n)$ is the score provided by DNN r^{th}; and, $\omega_{r,s,n}$ is the corresponding weight. These weights can be considered dependent or independent of output s^{th} or frame n^{th}.

One of the key assumptions for combining two classifiers is the degree of complementarity that could exist between them. A measure of complementarity could be the correlation between the outputs of two classifiers. Table 4.2 shows the average correlation coefficient for the whole test

Table 4.2 Average correlation coefficient for 20 subsets of HRRE database with the different classifiers output.

Classifiers	Avg. Corr. Coeff.	Classifiers	Avg. Corr. Coeff.
MelFB-LNFB	0.80	MelFB^1-MelFB^2	0.85
MelFB-PNCC	0.79	MelFB^1-MelFB^3	0.87
MelFB-RPLP	0.74	MelFB^1-MelFB^4	0.85
LNFB-PNCC	0.79	MelFB^2-MelFB^3	0.87
LNFB-RPLP	0.75	MelFB^2-MelFB^4	0.85
PNCC-RPLP	0.75	MelFB^3-MelFB^4	0.87
Avg.	0.77	Avg.	0.86

HRRE database calculated for the outputs of different pairs of DNNs. In the Table 4.2 it is observed that the outputs of DNNs trained with different initializations but with the same features vector are more correlated than the outputs of the DNNs trained with different features. This results strongly suggests that there could be a greater complementarity between DNNs trained with engineered features than with DNNs trained from the same feature vector.

4.3.2. Estimation of the DNN linear combination

4.3.2.1. Flat-weight based combination

A natural approach to performing the linear combination of DNNs is to employ flat-weights in (4.2) that are state and frame independent, *i.e.* ω_r. By definition, $\sum_{r=1}^{R} \omega_r = 1$. A special case corresponds to uniformly distributed weights across all the DNNs:

$$\omega_r = {}^{1}\!/_{R} \ . \tag{4.3}$$

4.3.2.2. Frame-by-frame based DNN combination

To combine DNNs on a frame-by-frame basis one method based on the minimization of the mutual information and another one defined according to the maximization of a discrimination metric were evaluated. In the first case, the mutual information is estimated between the score of the best classifier and the pdf of the combined score generated by the linear combination of the first classifier and a second one [185,186]. Accordingly, consider that: $m_1(s,n)$ and $m_2(s,n)$ in (4.4) denote the pseudo LLKs provided by two DNNs; $m_1(s,n)$ corresponds to the LLKs score of the most accurate DNN; and, the combined LLK scores is given by:

$$\widehat{m}(s,n) = \omega_{1,n} \cdot m_1(s,n) + \omega_{2,n} \cdot m_2(s,n) \ , \tag{4.4}$$

where $\omega_{1,n} = 1 - \omega_{2,n}$. Observe that in (4.4) weights $\omega_{1,n}$ and $\omega_{2,n}$ depend on frame n but are state independent. The pdf of the combined score $\widehat{m}$ at a given frame n is denoted by $f\left[\widehat{m}(s,n)|n,\omega_{2,n}\right]$. As a result, the mutual information between $m_1(s,n)$ and $f\left[\widehat{m}(s,n)|n,\omega_{2,n}\right]$ given frame n and weight $\omega_{2,n}$ can be expressed as:

$$\begin{aligned} &I\left\{m_1(s,n); f\left[\widehat{m}(s,n)|n,\omega_{2,n}\right]\middle|n\right\} \\ &\qquad = H\{m_1(s,n)|n\} - H\left\{m_1(s,n)\middle|f\left[\widehat{m}(s,n)|n,\omega_{2,n}\right],n\right\}. \end{aligned} \tag{4.5}$$

Weight $\omega_{2,n}$ could be obtained by maximizing the additional information provided by $f\left[\widehat{m}(s,n)|n,\omega_{2,n}\right]$ to $m_1(s,n)$, which is equivalent to minimizing the mutual information between $f\left[\widehat{m}(s,n)|n,\omega_{2,n}\right]$ and $m_1(s,n)$, i.e. $I\left\{m_1(s,n); f\left[\widehat{m}(s,n)|n,\omega_{2,n}\right]\middle|n\right\}$. Note that $H\{m_1(s,n)|n\}$ in (4.11) does not depend on $m_1(s,n)$ and to minimize the mutual information is equivalent to maximizing conditional entropy $H\left\{m_1(s,n)\middle|f\left[\widehat{m}(s,n)|n,\omega_{2,n}\right],n\right\}$. Consequently, the optimum weight in (4.5), $\widehat{\omega}_{2,n}$, can be estimated according to:

$$\widehat{\omega}_{2,n} = \arg\max_{\omega_{2,n}} \left\langle H\left\{m_1(s,n)\middle|f\left[\widehat{m}(s,n)|n,\omega_{2,n}\right],n\right\}\right\rangle \ , \tag{4.6}$$

where

$$H\{m_1(s,n)|f[\widehat{m}(s,n)|n,\omega_{2,n}],n\}$$

$$= -\sum_{i=1}^{Q} \Pr\{m_1(s,n)\big|f[\widehat{m}(s,n)|n,\omega_{2,n}],n\} \tag{4.7}$$

$$\cdot \log\langle \Pr\{m_1(s,n)\big|f[\widehat{m}(s,n)|n,\omega_{2,n}],n\}\rangle \ \ ,$$

and $\Pr\{m_1(s,n)\big|f[\widehat{m}(s,n)|n,\omega_{2,n}]\}$ is estimated by evaluating $m_1(s,n)$ in $f[\widehat{m}(s,n)|n,\omega_{2,n}]$.

Given frame n, the pdf´s of $m_1(s,n)$, $m_2(s,n)$ and $\widehat{m}(s,n)$ can be approximated with Gaussian distributions, whose parameters (i.e. mean and variance) are, respectively, $(\mu_{m_1},\sigma^2_{m_1})$, $(\mu_{m_2},\sigma^2_{m_2})$ and $(\mu_{\widehat{m}},\sigma^2_{\widehat{m}})$. Accordingly, $(\mu_{\widehat{m}},\sigma^2_{\widehat{m}})$ can be estimated as follows:

$$\mu_{\widehat{m}}(\omega_{2,n}) = (1-\omega_{2,n})\cdot\mu_{m_1} + \omega_{2,n}\cdot\mu_{m_2} \ \ , \tag{4.8}$$

$$\begin{aligned}\sigma^2_{\widehat{m}}(\omega_{2,n}) = \omega^2_{2,n}\cdot\left[\sigma^2_{m_1}+\sigma^2_{m_2}-2\cdot E[m_1(s,n)\cdot m_2(s,n)]+2\cdot\mu_{m_1}\cdot\mu_{m_2}\right] \\ -2\cdot\omega_{2,n}\cdot\left[\sigma^2_{m_1}-E[m_1(s,n)\cdot m_2(s,n)]+\mu_{m_1}\cdot\mu_{m_2}\right]+\sigma^2_{m_1} \ \ .\end{aligned} \tag{4.9}$$

Due to the fact that (4.7) does not provide an analytical solution, $\widehat{\omega}_{2,n}$ in (4.6) can be obtained by grid search. Note that optimal weight $\widehat{\omega}_{2,n}$ is found on a frame-by-frame basis within each testing utterance. Accordingly, given utterance u, it is also possible to define an average optimum weight:

$$\overline{\widehat{\omega}_{2,n}} = \frac{\sum_{n=1}^{N_u}\widehat{\omega}_{2,n}}{N_u} \ \ , \tag{4.10}$$

where N_u, is the number of frames in utterance u.

The second criterion explored to determine the optimal weights on a frame-by-frame basis attempts to increase the discriminability in the decoding process [187]. To achieve this purpose, the following metric was employed:

$$D(x_n) = \frac{\max\{\widehat{m}(s,n)\} - \mathrm{mean}\{\widehat{m}(s,n)\}}{\mathrm{SD}\{\widehat{m}(s,n)\}} \ \ , \tag{4.11}$$

where: $\widehat{m}(s,n)$ is defined as in (4.4) given a frame n; SD denotes standard deviation; and, $\max\{\widehat{m}(s,n)\}$, $\mathrm{mean}\{\widehat{m}(s,n)\}$ and $\mathrm{SD}\{\widehat{m}(s,n)\}$ are obtained over all the states of the combined DNNs at frame n. Then, the optimal weight $\widehat{\omega}_{2,n}$ can be achieved by maximizing $D(x_n)$:

$$\widehat{\omega}_{2,n} = \arg\max_{\omega_{2,n}}\{D(x_n)\} \ \ . \tag{4.12}$$

4.4. DNN combination and LC/MBR

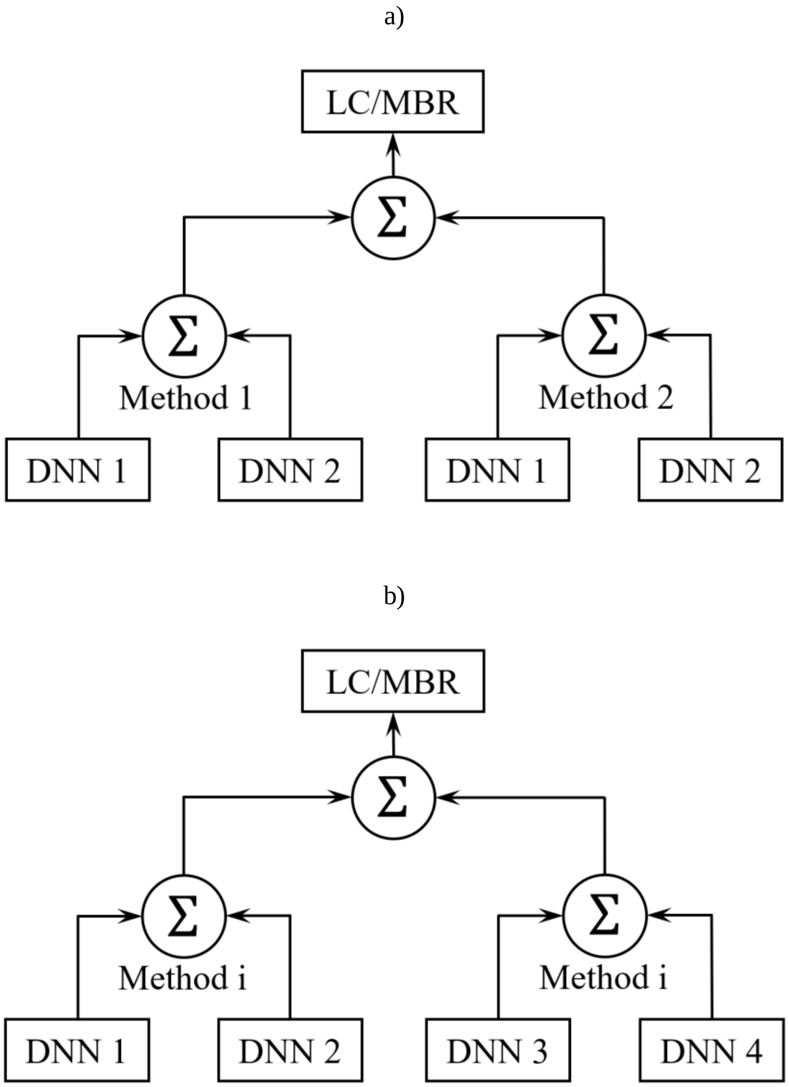
In this section, two combination schemes are presented that represent an alternative or complement to the DNNs combination methods proposed in the previous section.

4.4.1. Scheme with two systems

Another alternative to perform the information fusion of two classifiers is the proposed combination scheme shown in Fig. 4.2.a. This scheme considers two of the DNN combination methods proposed above as well as the LC/MBR.

4.4.2. Scheme with four systems

Finally, it is proposed to combine four systems at the same time using the combination scheme shown in Fig. 4.2.b. This scheme, unlike the scheme proposed for two systems, uses only one of the DNNs combination method proposed here and the LC/MBR.


a)

b)

Figure 4.2 Combination schemes using the proposed DNN combination methods and the LC/MBR used to combine a) two DNNs, as in Table 4.6, and b) four DNNs, as in Table 4.7.

4.5. Experiments

4.5.1. Training and testing data

For these experiments the test data from publicly-available real recordings of speech in a reverberant chamber with controllable RTs between 0.47 and 1.77 seconds was used. The training data were developed by subjecting clean speech to simulated reverberation. The databases are described in this section.

4.5.1.1. Training data

A multi-condition training database was generated based on the WSJ0 SI-284 corpus, which consists of about 81 hours of clean speech. The multi-condition database was derived from the entire clean SI-284 dataset. Each utterance of the clean database was convolved with three different simulated RIRs selected randomly from a list of 30,000 RIRs. The RIRs were simulated using the Room Impulse Response Generator [177]. The RT values of the generated RIRs varied between 0.4 and 2.4 seconds with an overall distribution of RTs that is shown in Fig. 4.3. The dimensions for each individual RIR were drawn from uniform distributions over the range of plus or minus 20 percent of the nominal values of 7.95 meters length, 5.68 meters width and 4.5 meters height. The motivation was to not match exactly the dimensions of the reverberation chamber used to collect the test data. The speaker-to-microphone distance was drawn from a uniform distribution between 0.144 and 2.816 meters. The speaker and microphone were placed in random locations at the room, using the distance that was selected for a particular trial, with the constraints that both speaker and microphone are at least 1 meter from any wall and between 1 and 2 meters from the floor. This randomization of the simulation parameters was implemented to reduce potential effects of artifacts caused by standing-wave phenomena in the rectangular shoebox-shaped room that RIR and other similar simulations based on the image method [178]. The resulting multi-condition database has a duration of 325 hours of which 25% are clean utterances. The size of the database is comparable to the amount of data in the Switchboard task [188]. In pilot experiments, also the CHiME-2 - Track 2 database was used [189]. This database is also based on the WSJ0 SI-284 dataset and was also obtained by convolving clean speech with binaural RIRs and adding background noise.

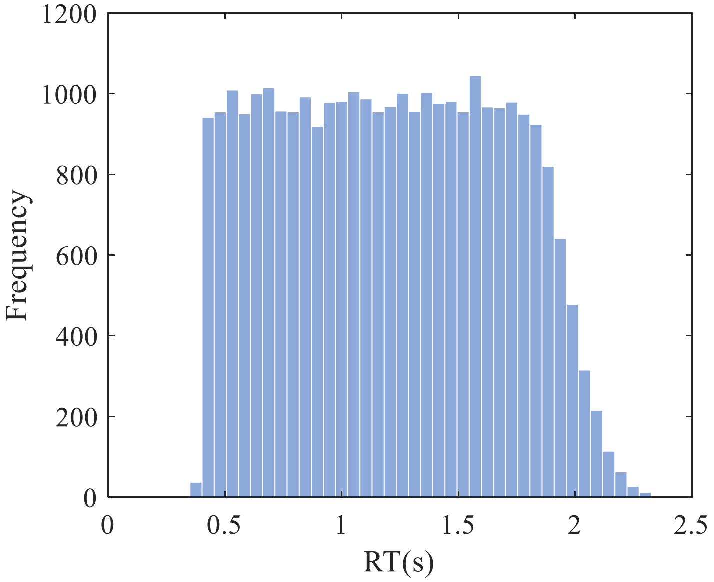


Figure 4.3 Histogram of the reverberation times in the generated room impulse responses used to obtain the reverberated training data.

4.5.1.2. Testing data

The HRRE database was used for system testing. This database is described in detail in [180] and is publicly available for research purposes. The database is composed of re-recorded utterances from the Aurora-4 clean evaluation set. The recording was performed in a real reverberant chamber, varying the speaker-to-microphone distance and the reverberation time. The speaker-to-microphone distances are 0.16, 0.32, 0.64, 1.28 and 2.56 meters. The resulting room RTs are 0.47, 0.84, 1.27 and 1.77 seconds. Altogether, there are 20 combinations of speaker-to-microphone distances and RTs. Each combination is composed of 330 testing utterances.

4.5.2. ASR systems

Four DNN-HMM based ASR engines were trained in parallel using four different types of initial features: conventional MelFB, LNFB [105], PNCC [138], and RASTA-PLP (RPLP) [190]. Each classifier was trained using the Kaldi Speech Recognition Toolkit [33]. As usual, a GMM-HMM recognizer was trained on clean data using the tri2b Kaldi Aurora4 recipe. This recipe uses MFCC features and performs LDA and MLLT to train a triphone system. This GMM-HMM system is subsequently used to obtain clean forced alignments to the reverberant training data. The resultant alignments are employed as references to train the DNNs [111]. The DNN architecture is composed of seven hidden layers and 2048 units per layer. The input layer considers a context window of 11 frames, with 5 frames before and 5 frames after the current frame. Finally, the minimum Bayes risk (MBR) decoding was performed considering the standard 5K lexicon and trigram language model from the WSJ database.

4.6. Results and discussion

4.6.1. Pilot results comparing the CHiME-2 and HRRE databases

As an exploratory experiment, the real evaluation database, HRRE, was tested on a system trained with the CHiME-2 database using MelFB features as described in Section 4.5.2. The results are shown in Table 4.3, where a baseline WER% reported in the literature is also added for comparison purposes. As can be seen, the proposed system provides competitive accuracy on the CHiME-2 evaluation. Nevertheless, system performance is much worse for the HRRE test data than for the CHiME-2 data. This result suggests that CHiME-2 is not very representative of real reverberated data, which in turn justifies the use of the HRRE database. As will be shown, the WER for the HRRE database drops dramatically (77% relative) when the multi-condition training described in Section 4.5.1 is applied.

Table 4.3 WER obtained for the CHiME-2 and HRRE databases using MelFB features.

Train	Test	AVG.
CHiME-2 (Han [191])	CHiME-2	16.19
CHiME-2+WPE	CHiME-2+WPE	15.29
CHiME-2+WPE	HRRE+WPE	38.07
Multi-condition	HRRE	8.67

4.6.2. Baseline experiments

Baseline experiments were generated with the four feature sets (MelFB, LNFB, PNCC, and RPLP) using the HRRE testing database and multi-condition training as described in Section 4.5.1. WPE

Table 4.4 WER obtained using MelFB, LNFB, PNCC and RPLP features. Results were obtained using multi-condition training and testing using the HRRE data, as described in Section 4.5.1. WPE was applied to both training and testing data. WER is averaged over the speaker-to-microphone distances for each RT.

	RT (s)				
Feature	0.47	0.84	1.27	1.77	AVG.
MelFB	3.35	4.78	6.61	9.22	5.99
LNFB	3.68	5.15	6.70	9.64	6.29
PNCC	3.40	5.04	6.64	9.44	6.13
RPLP	4.33	6.78	8.62	11.92	7.91

Table 4.5 Same as Table 4.4 except that WER is averaged across the RTs for each speaker-to-microphone distance.

	Speaker-to-microphone distance (m)					
Feature	0.16	0.32	0.64	1.28	2.56	AVG.
MelFB	3.34	3.87	4.88	7.83	10.04	5.99
LNFB	3.13	4.10	5.37	8.28	10.58	6.29
PNCC	2.84	3.79	4.80	8.03	11.19	6.13
RPLP	3.73	4.63	6.28	10.23	14.70	7.91

was always applied in both the training and testing data for this experiment and all of the experiments that follow. Results are summarized in Tables 4.4 and 4.5, in which WERs were averaged across speaker-to-microphone distances and RTs, respectively. All tested features are spectral-based and almost all of them led to similar recognition accuracy, except for RPLP, which produced slightly worse results. It is worth highlighting that all feature sets provided lower WERs than the WER obtained with the multi-condition training using MelFB features without WPE enhancement (see Table 4.3).

4.6.3. ASR systems combination with two systems

This section give attention to experimental results obtained by combining the results of the DNN classifiers using two different input features. First results obtained using the combination methods discussed in Sections 4.3.2 and 4.4.1 were considered, with classifiers combined two at a time. Subsequently these data were compared to results obtained by combining all features streams at once. Statistical significance was estimated according to the NIST matched-pair sentence-segment word error test (MAPSSWE) [40].

4.6.3.1. LC/MBR

For comparison reasons, the results obtained with this system combination method are presented for two systems in Table 4.6. The LC/MBR provide a reduction in WER equal to 3.4% in average when compared with de best single system baseline, *i.e.* MelFB. It is important to mention that all the DNN combination methods presented in Section 4.3.2 and the combination schemes in Section 4.4.1 that are discussed in the following sections exceed the results obtained with this system combination method widely used in the literature.

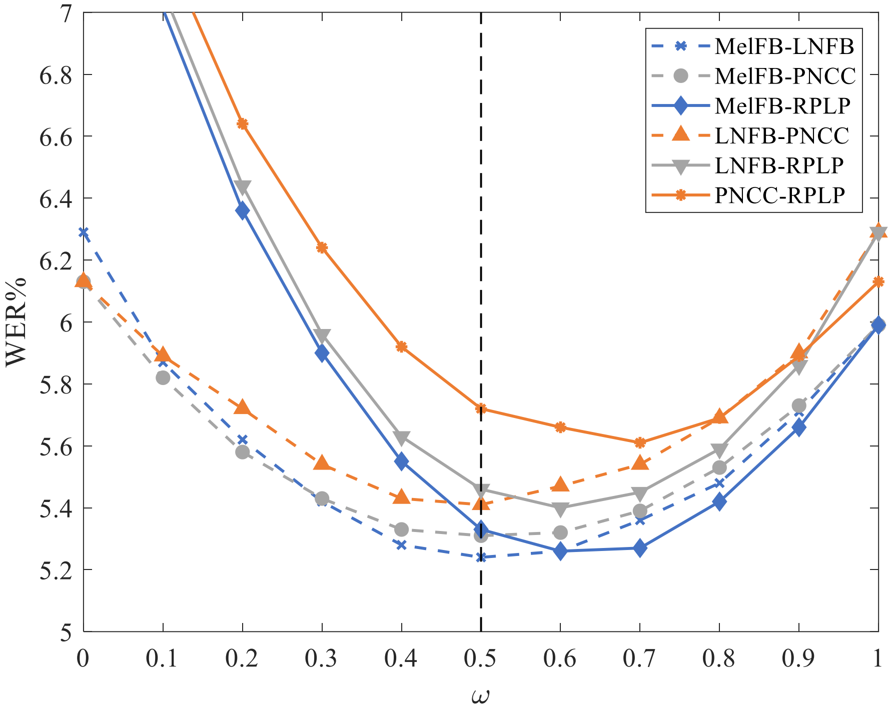


Figure 4.4 WERs obtained from grid search averaged across the 20 HRRE testing subsets vs the Flat-weight combination according to Section 4.2.2.

4.6.3.2. Flat-weight combination

Section 4.3.2.1 above discusses the application of a fixed linear score combination, or Flat-weight combination. Figure 4.4 depicts the actual average WER obtained when scores are combined from two classifiers using mixing parameter ω according to the equation $\hat{m} = \omega \cdot m_1 + (1-\omega) \cdot m_2$. It can be seen in Fig. 4.4 that the curve describing WER as a function of ω generally has a rather shallow minimum and the value $\omega = 0.5$ provides a WER that is close to optimal, which is between 0.5 and 0.7 depending on the DNN combination. In fact, almost all tested features led to similar recognition accuracy individually. Similarly, the boxplots of the optimal value ω obtained in the 20 HRRE testing subsets for each DNN pairs are shown in Fig. 4.5. As can be seen in Fig. 4.5, the optimal Flat-weight ω depends on the testing condition. When this subset-dependent value of ω is employed, the average reduction in the WER is about 1.7% relative to the WER obtained with $\omega = 0.5$ across all testing conditions. Compared to the MelFB baseline system (Tables 4.4 and 4.5), the relative improvement using subset-dependent weights is 11.2%, while the Flat-weight equal to 0.5 leads to an improvement of 9.7%. Recognizing that the best subsequent-dependent must be determined by exhaustive search for each condition, it is not considered that this additional improvement provided by subset-dependent weights to be particularly useful or practical. In most cases of DNN combination with two systems the best result was achieved by the combination using Flat-weight method, in just some cases other methods manage to overcome but not significantly.

4.6.3.3. Frame-based combination

The combination based on MMI described in Section 4.3.2.2 provided a relative improvement in WER equal to 7.4% when compared to the single MelFB-based classifier. When this weight is averaged across all the frames within the testing utterance as in (4.10), a slight increase in WER is

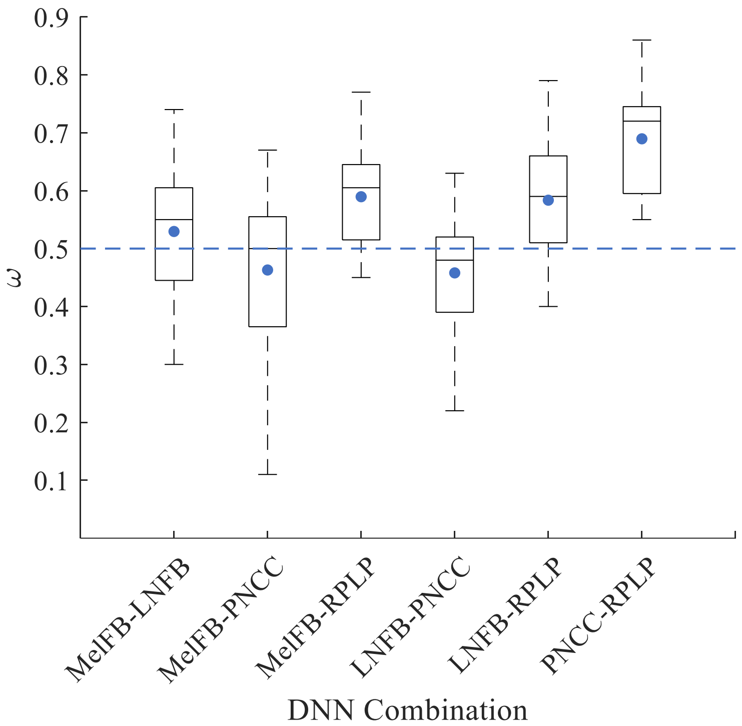


Figure 4.5 Optimal combination Flat-weight boxplots according to Section 4.3.2.1 for the 20 HRRE testing subsets for each pair of DNN.

observed. Similarly, the metric based on discriminability defined in (4.12) led to a relative reduction in WER equal to 6.6% when compared to the MelFB baseline in Table 4.4 or 4.5.

It is observed that the use of combination procedures based on optimal weighting and frame-based combination according to mutual information, however well motivated, do not provide better WER than simple averaging the scores of the individual DNN-based recognizers. It is worth

Table 4.6 WER averaged across all 20 HRRE testing data subsets. All possible combinations of two feature sets were tested. Multi-condition training was employed as described in Section 4.5.1. WPE was applied to both training and testing data.

Combination	MelFB -LNFB	MelFB -PNCC	MelFB -RPLP	LNFB-PNCC	LNFB-RPLP	PNCC-RPLP	AVG.
LC/MBR	5.57	5.60	5.90	5.57	5.88	6.21	5.79
Flat-weight (0.5)	5.24	5.31	5.33	5.41	5.46	5.72	5.41
Optimal weight	5.19	5.25	5.21	5.38	5.34	5.57	5.32
MMI	5.45	5.54	5.45	5.62	5.56	5.67	5.55
MMI avg.	5.47	5.57	5.46	5.70	5.59	5.66	5.57
Discriminability	5.40	5.54	5.51	5.60	5.54	5.98	5.60
Flat-weight/MMI + LC/MBR	5.26	5.34	5.23	5.42	5.35	5.62	5.37
Flat-weight/Discriminability + LC/MBR	5.26	5.34	5.29	5.45	5.41	5.75	5.41
MMI/Discriminability + LC/MBR	5.31	5.48	5.37	5.53	5.42	5.72	5.47

highlighting that the combining using frame-based methods do not require any tuning or prior information about the accuracies of the individual DNN-based classifiers.

4.6.3.4. DNN combination and LC/MBR scheme

When two DNNs were combined, the best result was obtained by the scheme with Flat-weight/MMI and LC/MBR leading to an improvement of 10.4% compared to the MelFB baseline (Tables III or IV) and 0.8% compared to the Flat-weight (significant at level $p < 0.002$) without any tuning or a priori information of the individual ASR accuracy.

The scheme Flat-weight/MMI+LC/MBR led to significant relative improvements of 1.9% ($p < 0.005$), 2.0% ($p < 0.002$) and 1.6% ($p < 0.011$) with MelFB-RPLP, LNFB-RPLP and PNCC-RPLP, respectively when compared to the Flat-weight DNNs combination method. In addition, on average this method led to a significant improvement of 0.8% ($p < 0.002$) when compared to Flat-weight method.

4.6.4. ASR systems combination with four systems

Table 4.7 describes the results obtained by combining the systems of all four different feature sets (MelFB, LNFB, PNCC, and RPLP) and four systems trained with the same feature set (MelFB) with different network initialization.

4.6.4.1. LC/MBR

The LC/MBR strategy by itself led to an improvement of 11.92% compared to the MelFB baseline. However, all proposed methods and schemes manage to overcome this combination method.

4.6.4.2. Flat-weight and UD (1/N)

For practical reasons the tuning of the DNNs combination for each testing subset was not performed. Combining the classifier outputs with uniform flat coefficients produced a relative improvement in WER of 17.5% relative to the best single-classifier baseline using MelFB features and 6.4% relative to the LC/MBR. However, the uniform distribution of weights may be a consequence of the fact that all the DNNs provide similar recognition accuracy.

4.6.4.3. DNN combination and LC/MBR

If the DNNs combination methods and LC/MBR are used together, in all cases improvements with respect to using only LC/MBR were achieved. In addition, the best results were obtained using the

Table 4.7 WER obtained when combining the four DNNs trained with different features and trained with 4 different initializations using the MelFB.

	MelFB–LNFB–PNCC–RPLP	MelFB with 4 DNN initializations
LC/MBR	5.28	5.88
Flat-weight	4.94	5.69
Flat-weight + LC/MBR	4.91	5.67
MMI + LC/MBR	5.07	5.71
MMI avg. + LC/MBR	5.06	5.72

Flat-weight+LC/MBR scheme leading to an improvement of 18% compared to the MelFB baseline (Tables III or IV). However, this combination scheme does not achieve significant improvements with respect to using Flat-weight method by itself.

4.6.4.4. Different initialization MelFB systems

Adequate reduction in WER could be obtained with a single set of features simply by combining systems that had been subject to different parameter initializations. To address this issue, four ASR systems using the best-performing single feature set (MelFB) with different sets of initial conditions for the DNN weights were trained, as was done in Section 4.5.2. The results of these experiments are summarized in the right column of Table 4.7. It is important to mention that for every single testing condition described in Table 4.7 substantially lower WERs were observed when ASR systems with different input features were combined (left column) than when ASR systems with different system initializations were combined (right column).

4.7. Conclusions

In this chapter, DNN and system combination is proposed to address the ASR robustness in highly-reverberant real environment. The experimental results were mainly obtained using a publicly available naturally-recorded highly reverberant speech data. The individual classifiers were trained with multi-condition fashion on about 330 hours of artificially-degraded speech and WPE was applied consistently to training and testing data. Furthermore, the complementarity of acoustics models trained with the same data but with different signal representation in reverberated speech data was discussed. DNN fusion methods based on flat-weight combination, the minimization of mutual information and the maximization of discrimination metrics were proposed and evaluated. Schemes that consider the combination of ASR systems with lattice combination and minimum Bayes risk decoding were also tested and combined with DNN fusion techniques. It was shown that significant improvements in WER can be achieved by combining the scores of state-of-the-art DNN-based ASR systems with different feature sets, obtaining relative improvements of 10.4% with two classifiers and 18.0% with four classifiers when compared to the best single-classifier baseline, without any tuning or a priori information of the ASR accuracy, on a difficult testing database of highly reverberated naturally-recorded speech data. It is worth highlighting that DNN combination with uniform flat weights provided reductions in WER equal to 9.7% and 17.5% using two and four classifiers, respectively, when compared to the best single-classifier baseline. This result must be due to the fact that all the single DNN-HMM systems except one led to similar accuracies. As a reference, the lowest WER was achieved when the flat-weight was tuned on each testing sub-data.

Chapter 5
Final conclusions and future work

In this thesis, the additive noise problem is addressed using uncertainty variance in noise cancelling in the decoding process of the state-of-the-art automatic speech recognition systems. Additionally, the problem of additive noise and the time-varying acoustic channel in human-robot interaction scenarios is tackled by a proposed an ASR training strategy that considers these disturbances produced by the acoustic environment. Moreover, the reverberation problem is also addressed by combining different acoustic models to achieve the robust recognition systems in highly-reverberant real environments.

According to the results obtained with the proposed uncertainty weighting scheme, underweighting frames with high uncertainty leads to significant improvements using clean training. The parameters of the proposed scheme can be optimized on a task-dependent or utterance dependent basis. Furthermore, the proposed scheme addresses the problem of acoustic/phonetic and language model combination that has not been exhaustively explored in the literature. Even though the uncertainty weighting scheme does not lead to significant improvements with multi-noise/multi-condition training, it can still reduce the gap with clean training. This is an important issue because it is not always possible or feasible to train a system in the same testing conditions. Also, the proposed scheme can thus be applied to any network topology that delivers log-likelihood-like scores, it can be combined with any distortion removal technique or front end, and it requires a very low additional computational cost with some configurations.

The popular black box integration of automatic speech recognition technology in HRI applications was improved with the addition of the HRI environment representation and modeling. Also, it was proposed that the robot and user states and contexts should be included in the voice-based HRI. Accordingly, this thesis focused on the environment representation and modeling by training a deep neural network-hidden Markov model based automatic speech recognition engine combining clean utterances with the acoustic-channel responses and noise that were obtained from an HRI testbed built with a PR2 mobile manipulation robot. This method avoids recording a training database in all the possible acoustic environments given an HRI scenario. Furthermore, different speech recognition testing conditions were produced by recording two types of acoustics sources, *i.e.* a loudspeaker and human speakers, using a Microsoft Kinect mounted on top of the PR2 robot, while performing head rotations and movements towards and away from the fixed sources. In this generic HRI scenario, the resulting automatic speech recognition engine provided a word error rate that is at least 26% and 38% lower than publicly available speech recognition APIs with the playback (*i.e.* loudspeaker) and human testing databases, respectively, with a limited amount of training data.

This thesis also addressed the combination of complementary parallel speech recognition systems to reduce the error rate of speech recognition systems operating in real highly-reverberant environments. The systems considered used four different feature sets and were trained using DNN-based techniques on 330 hours of data. The testing environment consists of recordings of speech in a calibrated real room with reverberation times from 0.47 to 1.77 seconds and speaker-to-microphone distances of 0.16 to 2.56 meters. The systems were combined both at the level of the DNN outputs and at the level of the final ASR outputs. The use of system combination provided up to 18.0% relative improvement in WER, compared to the best individual system. The greatest improvements in WER were obtained when systems were combined both at the outputs of the DNNs and at the final hypothesis level. It was also observed that system combination at the level of the outputs of the DNNs alone is more effective than system combination at the level of the output hypotheses alone, consistent with earlier findings in other domains. On average, a simple uniform weighting of DNN outputs provides the best results of all approaches examined at the DNN-output level, considering all the data on average. Nevertheless, the optimum linear combination weights depend on the experimental conditions such as RT and speaker-microphone distance, as well as which pair of systems is being combined.

Improving the robustness of voice-based HRI with the user and robot context is proposed for future research. Also, the use of beamforming schemes in combination with the proposed time-varying acoustic channel representation and modeling should be explored. Finally, it is worth highlighting that the results reported in this thesis were achieved with experiments on English databases for comparison purposes with methods published elsewhere. However, the implementation of the techniques presented here on Spanish language is straightforward.

Glossary of acronyms and abbreviations

ASR	Automatic Speech Recognition
API	Application Programming Interface
ANN	Artificial Neural Network
BP	Backpropagation
CMVN	Ceptral Mean and Variance Normalization
CMN	Ceptral Mean Normalization
CNC	Confusion Network Combination
CDNN	Convolutional Deep Neural Network
CNN	Convolutional Neural Network
CE	Cross Entropy
DOC	Damped Oscillator Coefficients
DBN	Deep Belief Network
DNN	Deep Neural Network
DTW	Dynamic Time Warping
ESR	Empty String Rate
EbT	Environment-based Training
fCNN	fused-CNN
GT	Gammatone
GFC	Gammatone Filter Coefficients
GMM	Gaussian Mixture Model
GSD	Generalized Synchrony Detector
HMM	Hidden Markov Model

HRRE	Highly-Reverberant Real Environments
HRI	Human Robot Interaction
IR	Impulse Response
LDA	Linear Discriminant Analysis
LNCC	Locally Normalized Cepstral Coefficient
LNFB	Locally Normalized Filter Bank
LLK	Log-Likelihood
LSTM	Long Short-Term Memory
MAPSSWE	Matched-Pair Sentence-Segment Word Error
MBR	Minimum Bayes Risk
LC/MBR	Lattice Combination and Minimum Bayes Risk Decoding
MCS	Multiple Classifier Systems
MLLT	Maximum Likelihood Linear Transformation
MVN	Mean and Variance Normalization
MN	Mean Normalization
MSE	Mean Square Error
MelFB	Mel Filter Bank
MFCC	Mel Frequency Cepstral Coefficients
MMSE	Minimum Mean Square Error
MLP	Multilayer Perceptron
NMF	Non-Negative Matrix Factorization
NMC	Normalized Modulation Coefficients
PLP	Perceptual Linear Predictive
PR2	Personal Robot 2

PNCC	Power Normalized Cepstral Coefficients
PCA	Principal Components Analysis
RNN	Recurrent Neural Network
RASTA	Relative Spectra
RPLP	Relative Spectra Perceptual Linear Predictive
RT	Reverberation Time
RIR	Room Impulse Response
STFT	Short-Time Fourier Transform
SS	Spectral Subtraction
SE	Speech Enhancement
SWV	Stochastic Weighted Viterbi
SSF	Suppression of Slowly-varying components and the Falling edge
TVAC	Time-Varying Acoustic Channel
UW	Uncertainty Weighting
UoO	Uncertainty-of-Observation
UT	Unscented Transform
WFST	Weighted Finite State Transducer
WPE	Weighted Predicted Error
WoZ	Wizard-of-Oz
WER	Word Error Rate

Bibliography

[1] S. J. Julier and J. K. Uhlmann, "A New Extension of the Kalman Filter to Nonlinear Systems," in *Proceedings of AeroSense*, Orlando, FL, USA, 1997, pp. 182-193.

[2] V. Beiu, J. A. Peperstraete, J. Vandewalle, and R. Lauwereins, "VLSI Complexity Reduction by Piece-Wise Approximation of the Sigmoid Function," in *Proceedings of ESANN*, Brussels, Belgium, 1994, pp. 181-186.

[3] M. Marge et al., "Applying the Wizard-of-Oz technique to multimodal human-robot dialogue," *arXiv preprint arXiv:1703.03714*, 2017. [Online]. https://arxiv.org/abs/1703.03714

[4] P. Sequeira et al., "Discovering social interaction strategies for robots from restricted-perception Wizard-of-Oz studies," in *Proceedings of HRI*, Christchurch, New Zealand, 2016, pp. 197-204.

[5] K. Hensby et al., "Hand in hand: Tools and techniques for understanding childrenś touch with a social robot," in *Proceedings of HRI*, Christchurch, New Zealand, March 2016, pp. 437-438.

[6] J. Li, L. Deng, R. Haeb-Umbach, and Y. Gong, "Processing in the feature and model domains," in *Robust Automatic Speech Recognition: A Bridge to Practical Applications*. Paris, France: Academic Press, 2015, pp. 65-106.

[7] R. Giri, M. L. Seltzer, J. Droppo, and D. Yu, "Improving speech recognition in reverberation using a room-aware deep neural network and multi-task learning," in *Proceedings of ICASSP*, Brisbane, QLD, Australia, 2015, pp. 5014-5018.

[8] N. Becerra Yoma, F. R. McInnes, and M. A. Jack, "Improving Performance of Spectral Subtraction in Speech Recognition Using a Model for Additive Noise," *IEEE Transactions on Speech and Audio Processing*, vol. 6, no. 6, pp. 579-582, 1998.

[9] N. Becerra Yoma, F. R. McInnes, and M. A. Jack, "Spectral Subtraction and Mean Normalization in the context of Weighted Matching Algorithms," in *Proceedings of EUROSPEECH*, Rhodes, Grecia, 1997, pp. 1411-1414.

[10] N. Becerra Yoma, F. R. McInnes, and M. A. Jack, "Weighted Matching Algorithms and Reliability in Noise Canceling by Spectral Subtraction," in *Proceedings of ICASSP*, vol. 2, Munich, Germany, 1997, pp. 1171-1174.

[11] N. Becerra Yoma and M. Villar, "Speaker Verification in Noise Using a Stochastic Version of the Weighted Viterbi Algorithm," *IEEE Transactions on Speech and Audio Processing*, vol. 10, no. 3, pp. 158-166, 2002.

[12] N. Becerra Yoma, I. Brito, and J. Silva, "Language Model Accuracy and Uncertainty in Noise Cancelling in the Stochastic Weighted Viterbi Algorithm," in *Proceedings of EUROSPEECH*, Geneva, Switzerland, 2003, pp. 2193-2196.

[13] N. Becerra Yoma, I. Brito, and C. Molina, "The Stochastic Weighted Viterbi Algorithm: A Frame Work to Compensate Additive Noise and Low - Bit Rate Coding Distortion," in *Proceedings of INTERSPEECH*, Jeju Island, Korea, 2004, pp. 2821-2824.

[14] J. Droppo, A. Acero, and L. Deng, "Uncertainty Decoding With SPLICE for Noise Robust Speech Recognition," in *Proceedings of ICASSP*, Orlando, FL, USA, 2002, pp. I-57-I-60.

[15] L. Deng, J. Droppo, and A. Acero, "Dynamic Compensation of HMM Variances Using the Feature Enhancement Uncertainty Computed From a Parametric Model of Speech Distortion," *IEEE Transactions on Speech and Audio Processing*, vol. 13, no. 3, pp. 412-421, 2005.

[16] T. T. Kristjansson and B. J. Frey, "Accounting for Uncertainity in Observations: A New Paradigm for Robust Automatic Speech Recognition," in *Proceedings of ICASSP*, Orlando, USA, 2002, pp. I-61-I-64.

[17] H. Liao and M. J. F. Gales, "Joint Uncertainty Decoding for Noise Robust Speech Recognition," in *Proceedings of INTERSPEECH*, Lisbon, Portugal, 2005, pp. 3129-3132.

[18] M. C. Benítez, J. C. Segura, A. de la Torre, J. Ramirez, and A. Rubio, "Including Uncertainty of Speech Observations in Robust Speech Recognition," in *Proceedings of ICSLP*, Jeju Island, Korea, 2004, pp. 137-140.

[19] D. T. Tran, E. Vincent, and D. Jouvet, "Fusion of Multiple Uncertainty Estimators and Propagators for Noise Robust ASR," in *Proceedings of ICASSP*, Florence, Italy, 2014, pp. 5512-5516.

[20] R. F. Astudillo and R. Orglmeister, "Computing MMSE Estimates and Residual Uncertainty Directly in the Feature Domain of ASR using STFT Domain Speech Distortion Models," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 5, pp. 1023-1034, 2013.

[21] R. F. Astudillo, D. Kolossa, P. Mandelartz, and R. Orglmeister, "An Uncertainty Propagation Approach to Robust ASR Using the ETSI Advanced Front-End," *IEEE Journal of Selected Topics in Signal Processing*, vol. 4, no. 5, pp. 824-833, 2010.

[22] Y. Tachioka and S. Watanabe, "Uncertainty Training and Decoding Methods of Deep Neural Networks Based on Stochastic Representation of Enhanced Features," in *Proceedings of INTERSPEECH*, Dresden, Germany, 2015, pp. 3541-3545.

[23] R. F. Astudillo, J. Correia, and I. Trancoso, "Integration of DNN based Speech Enhancement and ASR," in *Proceedings of INTERSPEECH*, Dresden, Germany, 2015, pp. 3576-3580.

[24] A. H. Abdelaziz, S. Watanabe, J. Hershey, E. Vincent, and D. Kolossa, "Uncertainty Propagation Through Deep Neural Networks," in *Proceedings of INTERSPEECH,* Dresden, Germany, 2015, pp. 3561-3565.

[25] S. J. Julier, "The Scaled Unscented Transformation," in *Proceedings of ACC*, vol. 6, Anchorage, AK, USA, 2002, pp. 4555-4559.

[26] H. Bourlard, H. Hermansky, and N. Morgan, "Towards Increasing Speech Recognition Error Rates," *Speech Communication*, vol. 18, no. 3, pp. 205-231, 1996.

[27] G. Stemmer, V. Zeissler, E. Nöth, and H. Niemann, "Towards a Dynamic Adjustment of the Language Weight," in *Proceedings of TSD,* Železná Ruda, Czech Republic, 2001, pp. 323-328.

[28] D. Yu and L. Deng, "Deep Neural Network-Hidden Markov Model Hybrid Systems," in *Automatic Speech Recognition, A Deep Learning Approach*.: Springer London, 2015, pp. 99-116.

[29] T. Hori and A. Nakamura, "Generalized Fast On-the-fly Composition Algorithm for WFST-Based Speech Recognition," in *Proceeding of INTERSPEECH 2005 - EUROSPEECH 2005*, Lisbon, Portugal, 2005, pp. 557-560.

[30] D. Povey et al., "Generating Exact Lattices in the WFST Framework," in *Proceedings of ICASSP,* Kyoto, Japan, 2012, pp. 4213-4216.

[31] G. Hirsch, "Experimental Framework for the Performance Evaluation of Speech Recognition Front-ends on a Large Vocabulary Task, Version 2.0, AU/417/02," *ETSI STQ Aurora DSR Working Group,* 2002.

[32] N. Parihar and J. Picone, "Aurora Working Group: DSR Front End LVCSR Evaluation AU/384/02," *Institute for Signal and Information Processing, Mississippi State University, Tech. Rep.*, 2002.

[33] D. Povey et al., "The Kaldi Speech Recognition Toolkit," in *Proceedings of ASRU,* Hawaii, USA, 2011, No. EPFL-CONF-192584.

[34] S. V. Vaseghi and B. P. Milner, "Noise Compensation Methods for Hidden Markov Model Speech Recognition in Adverse Environments," *IEEE Transactions on Speech and Audio Processing,* vol. 5, no. 1, pp. 11-21, 1997.

[35] P. J. Moreno, B. Raj, and R. M. Stern, "A Vector Taylor Series Approach for Environment-Independent Speech Recognition," in *Proceedings of ICASSP*, vol. 2, Atlanta, GA, USA, 1996, pp. 733-736.

[36] A. Narayanan and D. Wang, "Investigation of Speech Separation as a Front-end for Noise Robust Speech Recognition," *IEEE/ACM Transactions on Audio, Speech and Language Processing*, vol. 22, no. 4, pp. 826-835, 2014.

[37] M. Seltzer, D. Yu, and Y. Wang, "An Investigation of Deep Neural Networks for Noise Robust Speech Recognition," in *Proceedings of ICASSP*, Vancouver, Canada, 2013, pp. 7398-7402.

[38] B. Li and K. C. Sim, "A Spectral Masking Approach to Noise-robust Speech Recognition Using Deep Neural Networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 8, pp. 1296-1305, 2014.

[39] T. Drugman, Y. Stylianou, L. Chen, X. Chen, and M. J. Gales, "Robust Excitation-based Features for Automatic Speech Recognition," in *Proceedings of ICASSP*, Brisbane, QLD, Australia, 2015, pp. 4664-4668.

[40] D. S. Pallett, W. M. Fisher, and J. G. Fiscus, "Tools for the Analysis of Benchmark Speech Recognition Tests," in *Proceedings of ICASSP*, vol. 1, Albuquerque, NM, USA, 1990, pp. 97-100.

[41] R. F. Astudillo and J. P. D. S. Neto, "Propagation of Uncertainty Through Multilayer Perceptrons for Robust Automatic Speech Recognition," in *Proceedings of INTERSPEECH*, Florence, Italy, 2011, pp. 461-464.

[42] D. B. Paul and J. M. Baker, "The Design for the Wall Street Journal-based CSR Corpus," in *Proceedings of the DARPA SNL Workshop*, Harriman, NY, USA, 1992, pp. 357-362.

[43] M. A. Goodrich and A. C. Schultz, "Human-Robot Interaction: A Survey," *Foundations and Trends in Human-Computer Interaction*, vol. 1, no. 3, pp. 203–275, 2007.

[44] L. S. Lopes and A. Teixeira, "Human-robot interaction through spoken language dialogue," in *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, Takamatsu, Japan, 2000, pp. 528-534.

[45] G. Hoffman and K. Vanunu, "Effects of robotic companionship on music enjoyment and agent perception," in *Proceedings of HRI*, Tokyo, Japan, 2013, pp. 317-324.

[46] C. Y. Lin et al., "User identification design by fusion of face recognition and speaker recognition," in *Proceedings of 12th International Conference on Control, Automation and Systems*, JeJu Island, South Korea, 2012, pp. 1480-1485.

[47] K. Zheng, D. F. Glas, T. Kanda, H. Ishiguro, and N. Hagita, "Designing and Implementing a Human-Robot Team for Social Interactions," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 13, no. 4, pp. 843-859, 2013.

[48] Y. Kondo, K. Takemura, J. Takamatsu, and T. Ogasawara, "A gesture-centric android system for multi-party human-robot interaction," *Journal of Human-Robot Interaction*, vol. 2, no. 1, pp. 133-151, 2013.

[49] D. Wang, H. Leung, A. P. Kurian, H. J. Kim, and H. Yoon, "A Deconvolutive Neural Network for Speech Classification With Applications to Home Service Robot," *IEEE Transactions on Instrumentation and Measurement*, vol. 59, no. 12, pp. 3237 - 3243, 2010.

[50] E. L. Meszaros, M. Chandarana, A. Trujillo, and B. D. Allen, "Compensating for Limitations in Speech-Based Natural Language Processing with Multimodal Interfaces in UAV Operation," in *Proceedings of AHFE*, vol. 595, California, LA, USA, 2017, pp. 183-194.

[51] S. Han, J. Hong, S. Jeong, and M. Hahn, "Robust GSC-based speech enhancement for human machine interface," *IEEE Transactions on Consumer Electronics*, vol. 56, no. 2, pp. 965-970, 2010.

[52] M. Staudte and Crocker M. W., "Investigating joint attention mechanisms through spoken human-robot interaction," *Cognition*, vol. 120, no. 2, pp. 268-291, 2011.

[53] H. Polido, "DARPA Robotics Challenge," 2014.

[54] H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, and E. Osawa, "Robocup: The robot world cup initiative," in *Proceedings of the first international conference on Autonomous agents*, Marina del Rey, CA, USA, 1997, pp. 340-347.

[55] L. Zhang, L. Zhang, and B. Du, "Deep Learning for Remote Sensing Data: A Technical Tutorial on the State of the Art," *IEEE Geoscience and Remote Sensing Magazine*, vol. 4, no. 2, pp. 22-40, 2016.

[56] S. E. Umbaugh, *Digital image processing and analysis: Human and Computer Vision Applications with CVIPtools*, 2nd ed.: CRC Press, 2011.

[57] W. Burger and M. J. Burge, *Digital image processing: an algorithmic introduction using Java*.: Springer, 2016.

[58] J. Nakamura, *Image sensors and signal processing for digital still cameras*.: CRC press, 2016.

[59] S. Young, "HMMs and Related Speech Recognition Technologies," in *Springer Handbook of Speech Processing*.: Springer, 2008, pp. 539-558.

[60] X. D. Huang, Y. Ariki, and M. A. Jack, "Hidden Markov models for speech recognition," *Edinburgh University Press*, vol. 2004, 1990.

[61] R. Justo and M. I. Torres, "Integration of complex language models in ASR and LU systems," *Pattern Analysis and Applications*, vol. 18, no. 3, pp. 493-505, 2015.

[62] S. F. Chen, D. Beeferman, and R. Rosenfeld, "Evaluation metrics for language models," in *Proceedings of DARPA Broadcast News Transcription and Understanding Workshop*, Lansdowne, VA, USA, 1998, pp. 275-280.

[63] M. Chetouani, B. Gas, and J. Zarader, "Discriminative Training for Neural Predictive Coding Applied to Speech Features Extraction," in *Proceedings of the 2002 International Joint Conference on Neural Networks*, vol. 1, Honolulu, HI, USA, 2002, pp. 852-857.

[64] N. Dave, "Feature Extraction Methods LPC, PLP and MFCC in Speech Recognition," *International Journal for Advance Research in Engineering and Technology*, vol. 1, no. 6, pp. 1-5, 2013.

[65] S. Furui, "Speaker independent isolated word recognition using dynamic features of speech spectrum," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 34, no. 1, pp. 52–59, 1986.

[66] L. Bahl, R. Bakis, E. Jelinek, and R. Mercer, "Language-model/acoustic channel balance mechanism," *IBM Technical Disclosure Bulletin*, vol. 23, no. 7B, pp. 3464-3465, 1980.

[67] G. Hinton 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.

[68] J. Godfrey and E. Holliman, "Switchboard-1 Release 2," Philadelphia, Catalog No.: LDC97S62, 1997.

[69] G. E. Hinton, S. Osindero, and Y. -W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527-1554, 2006.

[70] J. Schröder, J. Anemüller, and S Goetze, "Performance comparison of GMM, HMM and DNN based approaches for acoustic event detection within Task 3 of the DCASE 2016 challenge," in *Proceedings of Workshop on Detection and Classification of Acoustic Scenes and Events,* Budapest, Hungary, 2016, pp. 80-84.

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

[72] O. Abdel-Hamid et al., "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on audio, speech, and language processing*, vol. 22, no. 10, pp. 1533-1545, 2014.

[73] A. Graves, A. R. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proceedings of ICASSP*, Vancouver, BC, Canada, 2013, pp. 6645-6649.

[74] Z. Tang, D. Wang, and Z. Zhang, "Recurrent neural network training with dark knowledge transfer," in *Proceedings of ICASSP*, Shanghai, China, 2016, pp. 5900-5904.

[75] J. Li, A. Mohamed, G. Zweig, and Y. Gong, "LSTM time and frequency recurrence for automatic speech recognition," in *Proceedings of ASRU*, Scottsdale, AZ, USA, 2015, pp. 187-191.

[76] T. N. Sainath and B. Li, "Modeling Time-Frequency Patterns with LSTM vs. Convolutional Architectures for LVCSR Tasks," in *Proceedings of INTERSPEECH*, San Francisco, USA, 2016, pp. 813-817.

[77] Y. Liu and K. Kirchhoff, "Novel Front-End Features Based on Neural Graph Embeddings for DNN-HMM and LSTM-CTC Acoustic Modeling," in *Proceedings of INTERSPEECH*, San Francisco, USA, 2016, pp. 793-797.

[78] D. Yu et al., "Deep convolutional neural networks with layer-wise context expansion and attention," in *Proceedings of INTERSPEECH*, San Francisco, USA, 2016, pp. 17-21.

[79] Y. Qian, M. Bi, T. Tan, and K. Yu, "Very Deep Convolutional Neural Networks for Noise Robust Speech Recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 12, pp. 2263-2276, 2016.

[80] V. Mitra and H. Franco, "Coping with Unseen Data Conditions: Investigating Neural Net Architectures, Robust Features, and Information Fusion for Robust Speech Recognition," in *Proceedings of INTERSPEECH*, San Francisco, USA, 2016, pp. 3783-3787.

[81] C. Weng, D. Yu, M. L. Seltzer, and J. Droppo, "Single-channel mixed speech recognition using deep neural networks," in *Proceedings of ICASSP*, Florence, Italy, 2014, pp. 5632-5636.

[82] S. Young et al., "The HTK book," *Cambridge university engineering department*, vol. 3, p. 175, 2006.

[83] K. F. Lee, H. W. Hon, and R. Reddy, "An overview of the SPHINX speech recognition system," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 1, pp. 35-45, 1990.

[84] W. Walker et al., "Sphinx-4: A flexible open source framework for speech recognition," Sun Microsystems, Inc., Mountain View, CA, USA, SMLI TR-2004-139, 2004.

[85] A. Lee, T. Kawahara, and K. Shikano, "JULIUS - an open source real-time large vocabulary recognition engine," in *Proceeding of INTERSPEECH*, Aalborg, Denmark, 2001, pp. 1691-1694.

[86] D. Bolaños, "The Bavieca open-source speech recognition toolkit," in *Proceedings of SLT*, Miami, FL, USA, 2012, pp. 354-359.

[87] D. O. Johnson et al., "Socially Assistive Robots: A Comprehensive Approach to Extending Independent Living," *International Journal of Social Robotics*, vol. 6, no. 2, pp. 195–211, 2014.

[88] J. F. Lehman, "Robo fashion world: a multimodal corpus of multi-child human-computer interaction," in *Proceedings of the 2014 Workshop on Understanding and Modeling Multiparty, Multimodal Interactions*, Istanbul, Turkey, 2014, pp. 15-20.

[89] F. Cutugno, A. Finzi, M. Fiore, E. Leone, and S. Rossi, "Interacting with robots via speech and gestures, an integrated architecture," in *Proceedings of INTERSPEECH*, Lyon, France, 2013, pp. 3727-3731.

[90] K. Zinchenko, C. Y. Wu, and K. T. Song, "A Study on Speech Recognition Control for a Surgical Robot," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 2, pp. 607-615, 2017.

[91] C. Matuszek, L. Bo, L. Zettlemoyer, and D. Fox, "Learning from Unscripted Deictic Gesture and Language for Human-Robot Interactions," in *Proceedings of the 28th National Conference on Artificial Intelligence*, Québec City, Quebec, Canada, 2014, pp. 2556-2563.

[92] J. Kennedy et al., "Child speech recognition in human-robot interaction: evaluations and recommendations," in *Proceedings of HRI*, Vienna, Austria, 2017, pp. 82-90.

[93] P. Lange and D. Suendermann-Oeft, "Tuning Sphinx to Outperform Google’s Speech Recognition API," in *Proceedings of ESSV*, Dresden, Germany, 2014, pp. 1-10.

[94] O. Mubin, J. Henderson, and C. Bartneck, "You just do not understand me! Speech Recognition in Human Robot Interaction," in *Proceedings of RO-MAN*, Edinburgh, Scotland, 2014, pp. 637-642.

[95] G. Hoffman, "OpenWoZ: A Runtime-Configurable Wizard-of-Oz Framework for Human-Robot Interaction," in *Proceedings of AAAI Spring Symposium Series*, Palo Alto, CA, USA, 2016.

[96] N. Martelaro, "Wizard-of-Oz Interfaces as a Step Towards Autonomous HRI," in *Proceedings of AAAI Spring Symposium Series*, Palo Alto, CA, USA, 2016.

[97] S. Pourmehr, J. Thomas, and R. Vaughan, "What untrained people do when asked "make the robot come to you"," in *Proceedings of HRI*, Christchurch, New Zealand, March 2016, pp. 495-496.

[98] E. Senft, P. Baxter, J. Kennedy, S. Lemaignan, and T. Belpaeme, "Providing a robot with learning abilities improves its perception by users," in *Proceedings of HRI*, Christchurch, New Zealand, March 2016, pp. 513-514.

[99] J. M. K. Westlund and C. Breazeal, "Transparency, teleoperation, and childrenś understanding of social robots," in *Proceedings of HRI*, Christchurch, New Zealand, March 2016, pp. 625-626.

[100] H. W. Löllmann et al., "Microphone array signal processing for robot audition," in *Proceedings of Hands-free Speech Communications and Microphone Arrays*, San Francisco, CA, USA, 2017, pp. 51-55.

[101] A. Deleforge and W. Kellermann, "Phase-optimized K-SVD for signal extraction from underdetermined multichannel sparse mixtures," in *Proceedings of ICASSP*, Brisbane, QLD, Australia, 2015, pp. 355-359.

[102] V. Poblete et al., "A Perceptually-Motivated Low-Complexity Instantaneous Linear Channel Normalization Technique Applied to Speaker Verification," *Computer Speech & Language*, vol. 31, no. 1, pp. 1-27, 2015.

[103] S. Seneff, "A joint synchrony/mean-rate model of auditory speech processing," *J. Phonetics*, vol. 16, no. 1, pp. 55-76, 1988.

[104] J. Fredes, J. Novoa, S. King, R. M. Stern, and N. Becerra Yoma, "Robustness to Additive Noise of Locally-Normalized Cepstral Coefficients in Speaker Verification," in *Proceedings of INTERSPEECH*, Dresden, Germany, 2015, pp. 3011-3015.

[105] J. Fredes, J. Novoa, S. King, R. M. Stern, and N. Becerra Yoma, "Locally Normalized Filter Banks Applied to Deep Neural-Network-Based Robust Speech Recognition," *IEEE Signal Processing Letters*, vol. 24, no. 4, pp. 377-381, 2017.

[106] J. Novoa et al., "Robustness over time-varying channels in DNN-HMM ASR based human-robot interaction," in *Proceedings of INTERSPEECH*, Stockholm, Sweden, 2017, pp. 839-843.

[107] Kerstin Dautenhahn et al., "How may I serve you?: a robot companion approaching a seated person in a helping context," in *Proceedings of HRI*, Salt Lake City, UT, USA, 2006, pp. 172-179.

[108] J. Novoa et al., "Multichannel Robot Speech Recognition Database: MChRSR," *arXiv preprint arXiv:1801.00061*, pp. 1-5, 2017. [Online]. https://arxiv.org/abs/1801.00061

[109] A. Farina, "Simultaneous measurement of impulse response and distortion with a swept-sine technique," in *Proceedings of AES Convention 108*, Paris, France, 2000, pp. 1-23.

[110] G. Hirsch. (2005) FaNT filtering and noise adding tool. Software. [Online]. http://dnt.kr.hs-niederrhein.de/

[111] S. Sivasankaran, E. Vincent, and I. Illina, "A combined evaluation of established and new approaches for speech recognition in varied reverberation conditions," *Computer Speech & Language*, vol. 46, no. Supplement C, pp. 444-460, 2017.

[112] P. Lin, D.-C. Lyu, F. Chen, S.-S. Wang, and Y. Tsao, "Multi-style learning with denoising autoencoders for acoustic modeling in the internet of things (IoT)," *Computer Speech & Language*, vol. 46, no. Supplement C, pp. 481-495, 2017.

[113] K. Veselý, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks," in *Proceeding of INTERSPEECH*, Lyon, France, 2013, pp. 2345-2349.

[114] J.-L. Gauvain, L. Lamel, and M. Adda-Decker, "Developments in continuous speech dictation using the ARPA WSJ task," in *Proceedings of ICASSP*, vol. 1, Detroit, MI, USA, 1995, pp. 65-68.

[115] A. Zhang. (2017) Speech Recognition (Version 3.7). Software. [Online]. https://github.com/Uberi/speech_recognition#readme

[116] B. Li et al., "Acoustic Modeling for Google Home," in *Proceedings of INTERSPEECH*, Stockholm, Sweden, 2017, pp. 399-403.

[117] G. Saon, H.-K. J. Kuo, S. Rennie, and M. Picheny, "The IBM 2015 English Conversational Telephone Speech Recognition System," in *Proceedings of INTERSPEECH*, Dresden, Germany, 2015, pp. 3140-3144.

[118] W. Xiong et al., "The microsoft 2016 conversational speech recognition system," in *Proceedings of ICASSP*, New Orleans, LA, USA, 2017, pp. 5255-5259.

[119] C. W. Han, S. J. Kang, and N. S. Kim, "Reverberation and noise robust feature compensation based on IMM," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 8, pp. 1598-1611, 2013.

[120] R. P. Lippmann, "Speech recognition by machines and humans," *Speech Communication*, vol. 22, no. 1, pp. 1-15, 1997.

[121] S. Furui, "History and development of speech recognition," in *Speech Technology*.: Springer, 2010, pp. 1-18.

[122] J. Cowan, "Building Acoustics," in *Springer Handbook of Acoustics*. New York: Springer, 2007, pp. 387-425.

[123] M. Delcroix, T. Nakatani, and S. Watanabe, "Static and Dynamic Variance Compensation for Recognition of Reverberant Speech With Dereverberation Preprocessing," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 17, no. 2, pp. 324 - 334, 2009.

[124] A. Sehr, R. Maas, and W. Kellermann, "Reverberation model-based decoding in the logmelspec domain for robust distant-talking speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1676-1691, 2010.

[125] V. Leutnant, A. Krueger, and R. Haeb-Umbach, "Bayesian feature enhancement for reverberation and noise robust speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 8, pp. 1640-1652, 2013.

[126] H. Kuttruff, "Measuring techniques in room," in *Room Acoustics*.: CRC Press, 2009, pp. 251-293.

[127] J. Li, L. Deng, R. Haeb-Umbach, and Y. Gong, "Reverberant speech recognition," in *Robust Automatic Speech Recognition: A Bridge to Practical Applications*. Paris, France: Academic Press, 2015, pp. 203-238.

[128] R. Petrick, K. Lohde, M. Wolff, and R. Hoffmann, "The harming part of room acoustics in automatic speech recognition," in *Proceedings of INTERSPEECH*, Antwerp, Belgium, 2007, pp. 1094–1097.

[129] F.-H. Liu, R. M. Stern, X. Huang, and A. Acero, "Efficient cepstral normalization for robust speech recognition," in *Proceedings of HLT*, Princeton, NJ, USA, 1993, pp. 69-74.

[130] H. Hermansky, N. Morgan, and H.-G. Hirsch, "Recognition of speech in additive and convolutional noise based on RASTA spectral processing," in *Proceedings of ICASSP*, Minneapolis, MN, USA, 1993, pp. 83-86.

[131] H. Hermansky and N. Morgan, "RASTA processing of speech," *IEEE Transactions on Speech and Audio Processing*, vol. 2, no. 4, pp. 578-589, 1994.

[132] X. Lu, M. Unoki, and S. Nakamura, "Sub-band temporal modulation envelopes and their normalization for automatic speech recognition in reverberant environments," *Computer Speech & Language*, vol. 25, no. 3, pp. 571-584, 2011.

[133] A. Tsilfidis, I. Mporas, J. Mourjopoulos, and N. Fakotakis, "Automatic speech recognition performance in different room acoustic environments with and without dereverberation preprocessing," *Computer Speech & Language*, vol. 27, no. 1, pp. 380-395, 2013.

[134] A. Krueger and R. Haeb-Umbach, "Model-Based Feature Enhancement for Reverberant Speech Recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1692-1707, 2010.

[135] M. Cooke, P. Green, L. Josifovski, and A. Vizinho, "Robust automatic speech recognition with missing and unreliable acoustic data," *Speech Communication*, vol. 34, no. 3, pp. 267-285, 2001.

[136] K. J. Palomäki, G. J. Brown, and J. P. Barker, "Techniques for handling convolutional distortion with 'missing data' automatic speech recognition," *Speech Communication*, vol. 43, no. 1-2, pp. 123-142, 2004.

[137] B. Raj, M. L. Seltzer, and R. M. Stern, "Reconstruction of Missing Features for Robust Speech Recognition," *Speech Communication*, vol. 43, no. 4, pp. 275-296, 2004.

[138] C. Kim and R. M. Stern, "Power-Normalized Cepstral Coefficients (PNCC) for Robust Speech Recognition," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 24, no. 7, pp. 1315-1329, July 2016.

[139] 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.

[140] R. Gomez and T. Kawahara, "Robust Speech Recognition Based on Dereverberation Parameter Optimization Using Acoustic Model Likelihood," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 7, pp. 1708-1716, 2010.

[141] V. Mitra, W. Wang, and H. Franco, "Deep convolutional nets and robust features for reverberation-robust speech recognition," in *Proceedings of SLT*, South Lake Tahoe, NV, USA, 2014, pp. 548-553.

[142] F. Weninger, S. Watanabe, Y. Tachioka, and B. Schuller, "Deep recurrent de-noising auto-encoder and blind de-reverberation for reverberated speech recognition," in *Proceedings of ICASSP*, Florence, Italy, 2014, pp. 4623-4627.

[143] B. Cauchi et al., "Joint dereverberation and noise reduction using beamforming and a single-channel speech enhancement scheme," in *Proceedings of Reverb Challenge*, Florence, Italy, 2014, pp. 1-8.

[144] K. Kinoshita et al., "A summary of the REVERB challenge: state-of-the-art and remaining challenges in reverberant speech processing research," *EURASIP Journal on Advances in Signal Processing*, vol. 2016, no. 1, pp. 1-19, 2016.

[145] V. Mitra, H. Franco, and M. Graciarena, "Damped oscillator cepstral coefficients for robust speech recognition," in *Proceedings of INTERSPEECH*, Lyon, France, 2013, pp. 886-890.

[146] V. Mitra et al., "Robust features in Deep Learning based Speech Recognition," in *New Era for Robust Speech Recognition: Exploiting Deep Learning*.: Springer, 2017, pp. 187-217.

[147] J. T. Geiger et al., "Memory-enhanced neural networks and NMF for robust ASR," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 6, pp. 1037-1046, 2014.

[148] C. Kim and R. M. Stern, "Nonlinear enhancement of onset for robust speech recognition," in *Proceedings of INTERSPEECH*, Makuhari, Chiba, Japan, 2010, pp. 2058-2061.

[149] K. Kumar, R. Singh, B. Raj, and R. Stern, "Gammatone sub-band magnitude-domain dereverberation for ASR," in *Proceedings of ICASSP*, Prague, Czech Republic, 2011, pp. 4604-4607.

[150] T. Yoshioka, T. Nakatani, M. Miyoshi, and H. Okuno, "Blind separation and dereverberation of speech mixtures by joint optimization," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 1, pp. 69-84, 2011.

[151] T. Yoshika, X. Chen, and M. Gales, "Impact of single-microphone dereverberation on dnn-based meeting transcription systems," in *Proceedings of ICASSP*, Florence, Italy, 2014, pp. 5527-5531.

[152] A. Mohamed, G. Hinton, and G. Penn, "Understanding how deep belief networks perform acoustic modelling," in *Proceedings of ICASSP*, Kyoto, Japan, 2012, pp. 4273-4276.

[153] L. Deng et al., "Recent advances in deep learning for speech research at Microsoft," in *Proceedings of ICASSP*, Vancouver, Canada, 2013, pp. 8604-8608.

[154] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798-1828, 2013.

[155] T. N. Sainath, A. Mohamed, B. Kingsbury, and B. Ramabhadran, "Deep convolutional neural networks for LVCSR," in *Proceedings of ICASSP*, Vancouver, Canada, 2013, pp. 8614-8618.

[156] C. Plahl, R. Schlüter, and H. Ney, "Improved acoustic feature combination for LVCSR by neural networks," in *Proceedings of INTERSPEECH*, Florence, Italy, 2011, pp. 1237-1240.

[157] Z. Tüske, P. Golik, R. Schlüter, and H. Ney, "Acoustic modeling with deep neural networks using raw time signal for LVCSR," in *Proceedings of INTERSPEECH,* Singapore, 2014, pp. 890-894.

[158] A. Tjandra et al., "Combination of two-dimensional cochleogram and spectrogram features for deep learning-based ASR," in *Proceedings of ICASSP*, Brisbane, QLD, Australia, 2015, pp. 4525-4529.

[159] J. M. Moyano, E. L. Gibaja, K. J. Cios, and S. Ventura, "Review of ensembles of multi-label classifiers: Models, experimental study and prospects," *Information Fusion*, vol. 44, pp. 33-45, 2018.

[160] W. Chen et al., "Rail crack recognition based on Adaptive Weighting Multi-classifier Fusion Decision," *Measurement*, vol. 123, pp. 102-114, 2018.

[161] X. Zhang, "Interactive patent classification based on multi-classifier fusion and active learning," *Neurocomputing*, vol. 127, pp. 200-205, 2014.

[162] A. A. Aburomman and M. B. I. Reaz, "A survey of intrusion detection systems based on ensemble and hybrid classifiers," *Computers & Security*, vol. 65, pp. 135-152, 2017.

[163] J. Fierrez, A. Morales, R. Vera-Rodriguez, and D. Camacho, "Multiple classifiers in biometrics. Part 1: Fundamentals and review," *Information Fusion*, vol. 44, pp. 57-64, 2018.

[164] R. M. O. Cruz, R. Sabourin, and G. D. C. Cavalcanti, "Dynamic classifier selection: Recent advances and perspectives," *Information Fusion*, vol. 41, pp. 195-216, 2018.

[165] J. Fierrez, A. Morales, R. Vera-Rodriguez, and D. Camacho, "Multiple classifiers in biometrics. Part 2: Trends and challenges," *Information Fusion*, vol. 44, pp. 103-112, 2018.

[166] A. Mi, L. Wang, and J. Qi, "A multiple classifier fusion algorithm using weighted decision templates," *Scientific Programming*, vol. 2016, pp. 1-10, 2016.

[167] A. Adjoudani and C. Benoit, "On the integration of auditory and visual parameters in an HMM-based ASR," in *Speechreading by Humans and Machines*. Berlin, Germany: Springer, 1996, pp. 461-471.

[168] H. Bourlard and S. Dupont, "A new ASR approach based on independent processing and recombination of partial frequency bands," in *Proceedings of ICSLP*, Philadelphia, PA, USA, 1996, pp. 426-429.

[169] S. Dupont and J. Luettin, "Audio-visual speech modeling for continuous speech recognition," *IEEE Transactions on Multimedia*, vol. 2, no. 3, pp. 141-151, 2000.

[170] J.G Fiscus, "A post-processing system to yield reduced word error rates: Recognizer Output Voting Error Reduction (ROVER)," in *Proceedings of ASRU*, Santa Barbara, CA, USA, 1997, pp. 347-354.

[171] L. Mangu, E. Brill, and A. Stolcke, "Finding consensus in speech recognition: word error minimization and other applications of confusion networks," *Computer Speech and Language*, vol. 14, no. 4, pp. 373-400, 2000.

[172] R. Singh, M. L. Seltzer, B. Raj, and R. M. Stern, "Speech in Noisy Environments: robust automatic segmentation, feature extraction, and hypothesis combination," in *Proceedings of ICASSP*, Salt Lake City, UT, USA, 2001, pp. 273-276.

[173] X. Li, R. Singh, and R. M. Stern, "Lattice Combination for Improved Speech Recognition," in *Proceedings of ICSLP*, Denver, CO, USA, 2002.

[174] H. Xu, D. Povey, L. Mangu, and J. Zhu, "An improved consensus-like method for Minimum Bayes Risk decoding and lattice combination," in *Proceedings of ICASSP*, Dallas, TX, USA, 2010, pp. 4938-4941.

[175] H. Xu, D. Povey, L. Mangu, and J. Zhu, "Minimum Bayes risk decoding and system combination based on a recursion for edit distance," *Computer Speech & Language*, vol. 25, no. 4, pp. 802-828, 2011.

[176] F. Kubala, "Broadcast News is Good News," in *Proceedings of DARPA Broadcast News Workshop*, Herndon, VA, USA, 1999, pp. 83-87.

[177] E. Habets. (2010) Room Impulse Response Generator. Software. [Online]. https://github.com/ehabets/RIR-Generator

[178] J. B. Allen and D. A. Berkley, "Image method for efficiently simulating small-room acoustics," *The Journal of the Acoustical Society of America*, vol. 65, no. 4, pp. 943-950, 1979.

[179] H. Kuttruff, "Reverberation and steady-state energy density," in *Room Acoustics*.: CRC Press, 2009, pp. 127-159.

[180] J. P. Escudero et al., "Highly-Reverberant Real Environment database: HRRE," *arXiv preprint arXiv:1801.09651*, 2018. [Online]. https://arxiv.org/abs/1801.09651

[181] D. Yu and L. Deng, "Feature Representation Learning in Deep Neural Networks," in *Automatic speech recognition: A deep learning approach*.: Springer, 2014, pp. 157-175.

[182] S. Haykin, "Multilayer Perceptrons," in *Neural networks and learning machines*.: Pearson Education, 2009, pp. 122-229.

[183] R. P. Duin and D. M. Tax, "Experiments with classifier combining rules," in *Proceedings of MCS*, Cagliari, Italy, 2000, pp. 16-29.

[184] X. Li and R. M. Stern, "Training of stream weights for the decoding of speech using parallel feature streams," in *Proceedings of ICASSP*, Hong Kong, China, 2003, pp. I-832-I-835.

[185] F. Huenupán, N. Becerra Yoma, C. Garretón, and C. Molina, "On-line linear combination of classifiers based on incremental information in speaker verification," *ETRI Journal*, vol. 32, no. 3, pp. 395-405, 2010.

[186] C. Molina, N. Becerra Yoma, F. Huenupán, C. Garretón, and J. Wuth, "Maximum entropy-based reinforcement learning using a confidence measure in speech recognition for telephone speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 18, no. 5, pp. 1041-1052, 2010.

[187] J. Novoa, J. Fredes, V. Poblete, and N. Becerra Yoma, "Uncertainty weighting and propagation in DNN-HMM-based speech recognition," *Computer Speech & Language*, vol. 47, pp. 30-46, 2018.

[188] V. Peddinti, D. Povey, and S. Khudanpur, "A time delay neural network architecture for efficient modeling of long temporal contexts," in *Proceedings of INTERSPEECH*, Dresden, Germany, 2015, pp. 3214-3218.

[189] E. Vincent et al., "The second ‘CHiME’ speech separation and recognition challenge: Datasets, tasks and baselines," in *Proceedings of ICASSP*, Vancouver, BC, Canada, 2013, pp. 126-130.

[190] H. Hermansky, N. Morgan, A. Bayya, and P. Kohn, "RASTA-PLP speech analysis technique," in *Proceedings of ICASSP*, San Francisco, CA, USA, 1992, pp. 121-124.

[191] K. Han, Y. He, D. Bagchi, E. Fosler-Lussier, and D. Wang, "Deep neural network based spectral feature mapping for robust speech recognition," in *Proceedings of INTERSPEECH*, Dresden, Germany, 2015, pp. 2484-2488.

Appendix A - Publications as first author

Institute for Scientific Information (ISI) journal publications

2018 J. Novoa, J. Fredes, V. Poblete, and N. Becerra Yoma, "Uncertainty weighting and propagation in DNN-HMM-based speech recognition," *Computer Speech & Language*, vol. 47, pp. 30-46, 2018. [Online]. https://doi.org/10.1016/j.csl.2017.06.005 (Q2)

2018 J. Novoa, J. Fredes, J. Wuth, F. Huenupán, R. Stern and N. Becerra Yoma, "System combination for robust speech recognition in reverberant environments." (Submitted to *Speech Communication*).

Conference publications

2018 J. Novoa, J. Wuth, J. P. Escudero, J. Fredes, R. Mahu and N. Becerra Yoma, "DNN-HMM Based Automatic Speech Recognition for HRI Scenarios," in *Proceedings of HRI*, Chicago, IL, USA, 2018, pp. 150-159. [Online]. http://doi.acm.org/10.1145/3171221.3171280 (23% acceptance rate)

2017 J. Novoa, J. Wuth, J. P. Escudero, J. Fredes, R. Mahu, R. Stern and N. Becerra Yoma, "Robustness over time-varying channels in DNN-HMM ASR based human-robot interaction," in *Proceedings of INTERSPEECH*, Stockholm, Sweden, 2017, pp. 839-843. [Online]. http://dx.doi.org/10.21437/Interspeech.2017-1308 (50% acceptance rate)

Preprint publications

2018 J. Novoa, J. P. Escudero, J. Wuth, V. Poblete, S. King, R. Stern and N. Becerra Yoma, "Exploring the robustness of features and enhancement on speech recognition systems in highly-reverberant real environments," *arXiv preprint arXiv:1803.09013*, 2018. [Online]. https://arxiv.org/abs/1803.09013

2017 J. Novoa, J. Fredes and N. Becerra Yoma, "DNN-based uncertainty estimation for weighted DNN-HMM ASR," *arXiv preprint arXiv:1705.10368*, 2017. [Online]. https://arxiv.org/abs/1705.10368

2017 J. Novoa, J. Wuth, J. P. Escudero, J. Fredes, R. Mahu and N. Becerra Yoma, "Multichannel Robot Speech Recognition Database: MChRSR," *arXiv preprint arXiv:1801.00061*, 2017. [Online]. https://arxiv.org/abs/1801.00061

Appendix B - Publications as co-author

Institute for Scientific Information (ISI) journal publications

2017 J. Fredes, J. Novoa, S. King, R. M. Stern, and N. Becerra Yoma, "Locally Normalized Filter Banks Applied to Deep Neural-Network-Based Robust Speech Recognition," *IEEE Signal Processing Letters*, vol. 24, no. 4, pp. 377-381, 2017. (Q2)

Conference publications

2016 V. Poblete, J. P. Escudero, J. Fredes, J. Novoa, R. M. Stern, S. King and N. B. Yoma, "The Use of Locally Normalized Cepstral Coefficients (LNCC) to Improve Speaker Recognition Accuracy in Highly Reverberant Rooms," in *Proceedings of INTERSPEECH*, San Francisco, CA, USA, pp. 2373-2377, 2016.

2015 J. Fredes, J. Novoa, V. Poblete, S. King, R. M. Stern and N. B. Yoma, "Robustness to additive noise of locally-normalized cepstral coefficients in speaker verification," in *Proceedings of INTERSPEECH*, Dresden, Germany, pp. 3011-3015, 2015.

Preprint and non-ISI publications

2018 J. P. Escudero, V. Poblete, J. Novoa, J. Wuth, J. Fredes, R. Mahu R. Stern and N. Becerra Yoma, "Highly-Reverberant Real Environment database: HRRE," *arXiv preprint arXiv:1801.09651*, 2018. [Online]. https://arxiv.org/abs/1801.09651

2018 J. P. Escudero, J. Novoa, R. Mahu, J. Wuth, R. Stern and N. Becerra Yoma, "An improved DNN-based spectral feature mapping that removes noise and reverberation for robust automatic speech recognition," *arXiv preprint arXiv:1803.09016*, 2018. [Online]. https://arxiv.org/abs/1803.09016

2016 V. Poblete, J. Fredes, J. Novoa, S. King, R. M. Stern and N. B. Yoma, "Coeficientes Cepstrales Sub-Banda Localmente-Normalizados," in *Síntesis Tecnológica*, Universidad Austral, 2016.