Hannes Unterholzner, BSc

Channel Selection for Distant Automatic Speech Recognition

on the CHiME-5 dataset

MASTER'S THESIS

to achieve the university degree of
Diplom-Ingenieur
Master's degree programme: Information and Computer Engineering

submitted to

Graz University of Technology

Supervisor

Assoc.Prof. Dipl.-Ing. Dr. Franz Pernkopf

Signal Processing and Speech Communication Laboratory

Graz, January 2019

AFFIDAVIT

I declare that I have authored this thesis independently, that I have not used other than the declared sources/resources, and that I have explicitly indicated all material which has been quoted either literally or by content from the sources used. The text document uploaded to TUGRAZonline is identical to the present master's thesis.

Date

Signature

Acknowledgements

First and foremost I would like to gratefully thank my thesis advisor Franz Pernkopf for his patient guidance, enthusiastic encouragement and constructive feedback on this research work. The door to Prof. Pernkopf's office was always open whenever I encountered an issue or trouble spot, or simply had a question on my research and writing. I owe my deepest gratitude to the SpeechTek group at Fondazione Bruno Kessler for their assistance, constant support and comments, in particular to Marco Matassoni, Giuseppe Daniele Falavigna and Alessio Brutti. Finally, I must express my profound gratitude to my family and friends, for their endless support, inspiring discussions and encouragement throughout my years of study and the entire process of researching and writing this thesis. Thank you.

Abstract

Current automatic speech recognition systems already show remarkable results in constrained scenarios with close-talk recordings. However, the performance is affected when recordings are taken from the far-field due to both noise and reverberations. In the presence of multiple distant-talking microphones we can assume that some decoded channels deliver better transcriptions than others. The objective of this thesis is to investigate a DNN-based classifier for channel selection trained on signal-based and/or decoder-based features. The CHiME-5 dataset, a novel dataset for distant multiple-microphone conversational speech recognition, is used to conduct the experiments. A promising performance gain of 18% is provided from an oracle analysis. Actual experimental results reveal the limitation of the extracted features and DNN classifiers to correlate well with the oracle results, i.e. the classification results and the DNNs generalisation performance is weak. The problem is traced back to a high proportion of simultaneous and spontaneous speech, different acoustic scenarios and background noises, as well as the variation in speakers among the sessions of the dataset. Moreover, based on the obtain classifier rankings we apply hypothesis combination with ROVER on different channel subsets for error reduction, based on average confidence scores.

Kurzfassung

Für Anwendungen mit Nahfeldmikrofonen erzielen automatische Spracherkennungssysteme bereits heute sehr gute Ergebnisse. Die Genauigkeit der Transkriptionen kann sich jedoch verschlechtern, falls die Aufnahmen mit Distanz zum Sprecher erfolgen, da Einflussfaktoren wie Hintergrundrauschen als auch Nachhall das Signal zunehmend stören. Falls Mikrofonsignale von unterschiedlichen Positionen zur Verfügung stehen, ist anzunehmen, dass einige Signale bessere Transkriptionen liefern als andere. Die Arbeit baut auf dieser Annahme auf und untersucht die Möglichkeit der Kanalklassifizierung mithilfe eines tiefen neuralen Netzes, welches mittels signal-bezogener und oder auch decoder-bezogener Merkmale trainiert wird. Der CHiME-5 Korpus dient als Datensatz für die Experimente. Durchgeführte Orakel Untersuchungen zeigen potenzielle Erkennungsverbesserung um 18%. Die experimentellen Resultate verdeutlichen die Schwierigkeit, geeignete Merkmale zu extrahieren und Klassifizierer zu lernen, die möglichst stark mit den Orakel-Labels korrelieren. Das Problem ist hauptsächlich auf den hohen Anteil an überlappender und spontaner Sprache, den verschiedenen akustischen Szenarien und Hintergrundgeräuschen, als auch auf die Variation in den Sprechern zurückzuführen. In einem letzten Schritt wird auf Basis der erhaltenen Klassifizierungsrangliste eine Hypothesenkombination mittels ROVER durchgeführt.

Contents

1

Introduction

Automatic speech recognition (ASR) systems are increasingly present across a wide range of modern applications already demonstrating their functionality, high usability and advantage. These applications mainly include voice user interfaces and speech-to-text engines for home automation and security assess control, medical assistance, industrial robotics, telecommunication industry and also for learning and educational purposes. However, in most cases remarkable ASR results are only obtained for restricted conditions and well known application areas. Most of the systems are limited to one active speaker at a time, one language, a restricted vocabulary and the accuracy depends on a high signal to noise ratio (SNR), usually obtained by a short distance between the microphone(s) and the speaker(s) mouth. For example, in a car voice control unit the required vocabulary to control the system is typically small and also clear assumptions can be made on the speaker positions and the types of background noise (e.g. engine noise, rolling noise, entertainment noise) present within the car interior. This prior knowledge can be efficiently used to tailor the overall ASR system to the specific scope of application, from the microphone placements, signal enhancement, feature extraction and transformation all the way to the acoustic and language modelling. Considering a more indefinite scenario with an unknown number of freely moving speakers within an environment and multiple conversations taking place at the same time, it is very hard, if not impossible, to design a proper system that delivers a good performance in form of accurate transcriptions. Solely an arbitrary placement of the microphones with an increased distance to the speaker can have a serious impact on the SNR and consequently a degradation in the ASR performance. This is due to a variety of effects including background noises, reverberation in form of early reflections and diffuse sound, and as well as simultaneous speech. Distant ASR is a current matter that focuses on investigating in different techniques for noise reduction, dereverberation and speaker separation to mitigate these disturbing effects of the far-field recordings.

When multiple microphones are used to record the speech signal from different angles, it can be assumed that there is one microphone among the set that captures the source signal the best; in other words the decoded utterance from this channel gives the lowest word error rate (WER), i.e. the best transcription. A very simple idea would be to learn a mapping between channel-based features and the final WER or good channels using a deep neural network (DNN). The difficulties lie in the design of proper features that are able to capture signal properties directly related to its quality and more importantly to the recognition accuracy.

1.1 CHiME-5 Challenge

The CHiME-5 challenge [1] deals with the problem of *distant multiple-microphone conversational speech recognition in everyday home environments*. The training and development sets comprise audio data recorded from both array and binaural microphones, and the corresponding manual transcriptions. Recordings were conducted during dinner parties with each consisting of four participants. Three baseline systems for array synchronization, speech enhancement and conventional or end-to-end ASR were provided by the challenge organisers. The challenge features two main tracks for reporting results on both the development set and an evaluation set, where no transcriptions were provided. For recognition, in the single-array track only one reference array can be used, whereby in the multiple-array track recordings from all six arrays can be used. In addition to this, each track is split up into two sub-rankings, one for conventional acoustic modelling and official language modelling and a second sub-ranking without constrains on the used system (i.e. end-to-end system and modified language model).

1.2 Aim and contributions

The aim of this thesis is to evaluate the applicability of a DNN based channel selection method for distant conversational speech recognition on the CHiME-5 dataset. We investigate both signal-based and decoder-based features in order to train a DNN to predict the best performing channels in a supervised manner. Labels are provided from computational expensive decodings of the available channels using the conventional baseline system. We execute several oracle experiments on a subset or on all the available channels and can demonstrate a remarkable gain for improvement, if a proper channel selection technique is applied. Moreover we address hypotheses combination using recognition output voting error reduction (ROVER) to increase the recognition accuracy even further using the classifier ranked channels to provide the best quality channel as a skeleton for construction of a comprised word transition network (WTN). Furthermore we perform several analysis on the provided data and show the difficulty of the underlying dataset for ASR, due to noise, far-field recordings, as well as spontaneous and simultaneous speech.

1.3 Outline

The thesis is organised in five chapters, where Chapter 2 introduces the conventional baseline ASR system to provide the reader with the necessary background information on how the recorded utterances are processed within the ASR baseline chain. The audio data acquisition with multiple microphones is covered in Chapter 3. This chapter also highlights relevant analysis on the recorded data that will play a crucial role in the experimental results. Following this, the theoretical aspects of the channel selection method as well as the system combination technique using ROVER are covered in Chapter 4. Chapter 5 documents the conducted experiments on channel selection and combination and discussed on the obtained results. The work is then concluded in Chapter 6.

2 Baseline System

In automatic speech recognition (ASR) we distinguish between two main types of recognition architectures, a conventional and an end-to-end system. A conventional ASR system comprises a handcrafted acoustic model (AM), language model (LM) and a pronunciation dictionary, which are often trained separately from each other, using different datasets. A LM assigns a probability to a certain word sequence and the pronunciation dictionary is used to represent this word sequence by a sequence of smaller linguistic units or phones. The AM then predicts the likelihood of a sequence of acoustic features, given this set of phones. Despite the fact that conventional ASR systems are the standard approach in performing transcriptions from speech, the decomposed training of the three models is complex and often requires careful engineering work. Mismatches of these components impair the accuracy of the overall system and can be avoided when training the models in a joint way, using end-to-end methods. In doing so, usually a deep neural network (DNN) is trained in a sequence-to-sequence fashion to find the corresponding character sequence from a sequence of acoustic feature vectors. Nowadays large amount of computational resources and the accessibility of huge datasets makes this training feasible and even more important the mutual training delivers competitive results.

The CHiME-5 challenge comes with an ASR baseline system for both conventional and end-to-end speech recognition, hence offering participating groups a range of possibilities in the design of a competitive system. The conventional baseline is implemented within the Kaldi framework [2][1], a state of the art speech recognition toolkit written in C++ that provides full implementations for each single stage within the ASR pipeline, including data preparation, feature extraction, model training, decoding and scoring. The provided end-to-end baseline system is an attention/connectionist temporal classification (CTC) hybrid architecture [3], implemented using the ESPnet[2] toolkit. Moreover, both systems make use of two preprocessing stages, one for headworn and array synchronization[3] and a second one for speech enhancement using a weighted delay-and-sum beamformer based on the BeamformIt toolkit [4][4].

The following parts of this section describe the conventional baseline system, whereby the emphasis is to give an overview rather than a detailed description of all the incorporated components, for which the reader is referred to the specified references in the following paragraphs and also to [5], [6] and [7] for a more general insight into ASR. In Section 2.1 and Section 2.2 the two preprocessing stages for array synchronisation and speech enhancement are introduced followed by a description of the conventional baseline system in Section 2.3. Section 2.4 states the official baseline results. Part of the information is taken from the challenges website[5].

[1] Kaldi: speech recognition toolkit repository available under https://github.com/kaldi-asr/kaldi

[2] ESPnet: End-to-end speech processing toolkit repository can be copied from https://github.com/espnet/espnet

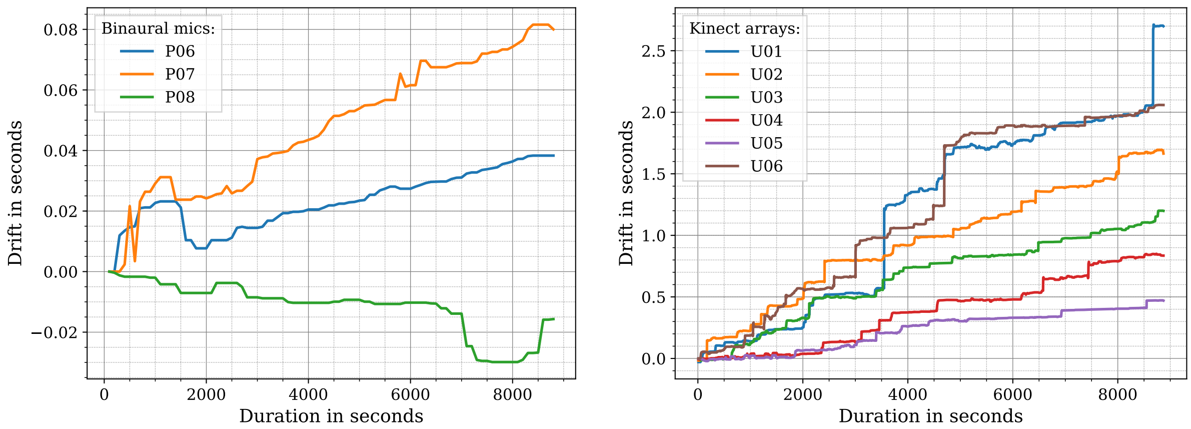
[3] Source code of the array synchronization available from https://github.com/chimechallenge/chime5-synchronisation

[4] BeamformIt: implementation of the weighted delay-and-sum beamformer can be found on https://github.com/xanguera/BeamformIt

[5] CHiME-5 challenge website: http://spandh.dcs.shef.ac.uk/chime_challenge

2.1 Array synchronisation

As we will see in Chapter 3, each dinner party is recorded with a set of six Kinect arrays and four binaural microphones, one for each of the four speakers, onto separate recorders over the full duration of the session (∼ 2 h). The ten recorders suffer from time misalignments due to a combination of clock drifts (on all devices) and occasional frame-dropping on the Kinect arrays, whose impact becomes larger with the duration of the recorded session. Especially the signals from the Kinect arrays reach time misalignments in the range of seconds, the ones from the head-worn recordings around 100 ms. Hence, device synchronisation is crucial in order to extract equal signal snippets from the different recordings for later processing, likewise it is needed when training an AM on utterances. Figure 2.1 illustrates the applied device alignments for the development session S02 with respect to the reference binaural device P05 for both the remaining binaural devices (see Figure 2.1(a)) and for the Kinect arrays (see Figure 2.1(b)). The effect of frame dropping can be observed in the right figure for Kinect U01 (in blue) around a recording time of 3500 seconds when the time drift jumps from approximately 0.6 to 1.3 seconds. In order to compensate for the misalignment the transcription files are equipped with device dependent aligned start and end times for each utterance, obtained from the array synchronisation baseline system[3]. The synchronisation chain itself comprises two main stages, starting from computing coarse time alignments and followed by subsequent refined alignments which are then applied to the transcription files. Synchronisation for the microphone arrays is applied on device level, since there are no drifts among the four single channels of each array. Before the synchronisation is explained in more detail, we introduce the following five terms:



(a) Alignment correction for the three target binaural microphones.

(b) Alignment correction for the Kinect arrays.

Figure 2.1: Estimated misalignment for training session S02 with respect to the binaural reference channel P05 to compensate for the synchronisation drift between channels.

Reference channel: Binaural device that serves as the reference channel for the time lag estimation. This is always the first binaural channel of a specific session in an alphabetical and numerical order. The Kinect is not suitable as reference channel due to the risk of frame dropping.

Target channel: Channel whose alignment is compared with the one from the reference channel.

Time resolution: The time interval for which time lags are computed between the target and the reference device, to get equidistant separated points in time over the full recording duration.

Search duration: Expected maximum time drift between reference and target channel for which the cross-correlation is computed.

Template duration: Duration of the target channel audio snippet actually used to calculate the cross-correlation. The length of the reference channel audio snippet is composed by the template duration plus twice the search duration. Often this is also known as the analysis window.

The estimation of the time lag between the *target channel* and the *reference channel* is based on a cross-correlation principle[6]. Imagine two signals with a peak at different positions in time. The position of the peak in its cross-correlation result may serve as an estimate of the time lag between the peaks and the magnitude as a measure of confidence of this estimated quantity. In this manner, misalignments between the two channels are then computed in equidistant time intervals over the recording length, as given by the applied *time resolution*. It would be pointless and computational demanding to compute each time lag with a cross-correlation over the total session recording, hence small chunks in the time region of interest with a length defined by the *template duration* are used instead. In addition to speed up the whole process the correlation is computed over the *search duration* rather then the full chunk length, assuming that the maximal time drift does not exceed this observation window. Time lags are computed for both binaural reference channels (left and right) and at the end the lag with the higher confidence score (correlation peak) is chosen. For the Kinect arrays the recordings from channel 1 are always used to compare against the *reference channel*.

In the first coarse alignment pass the misalignment between two binaural channels is estimated every 100 seconds, while this is done every 10 seconds when aligning a Kinect channel. The *template duration* is set to 5 seconds for both device types, and the *search duration* to 5 seconds for Kinect devices and 0.5 seconds for binaural devices. The refined alignment pass is build upon the previous stage and primary deals with large differences in time lags of consecutive time intervals for the Kinect arrays. If the lag threshold of 50 ms is exceeded, time lags are recomputed with a refined *time resolution* of 1 second rather then 10 seconds. Moreover smoothing of adjacent lags is performed with the help of a median filter on all channels. Afterwards the transcriptions are aligned by altering the start and end times of the utterances according to the estimated misalignments. A transcription snippet of an aligned utterance from session S02 is shown in Listing 3.1.

2.2 Speech enhancement

Before executing the actual speech recognition step, the raw recorded signals are enhanced with appropriate processing techniques in order to increase the signal to noise ratio (SNR) of the speech signal of interest. When multiple-microphones are available, beamforming is a common technique to emphasise the signal coming from the direction of the speaker. The basic principle behind beamforming is to make use of recordings from different spacial positions with a delayed summation of the signals to achieve a constructive interference of the signal from a steering direction (i.e. direction of the speaker) while simultaneously attenuating signals arriving from all the other directions. Usually, a prior weighting W_m of the signals is applied before the summation step so that the beamformed output is defined as follows [4]:

$$y[n] = \sum_{m=1}^{M} W_m[n] x_m[n - \mathrm{TDOA}^{(m,ref)}[n]], \tag{2.1}$$

[6] The cross-correlation is a measure of similarity between two signals. For a more mathematical explanation see also `https://en.wikipedia.org/wiki/Cross-correlation`.

where m is one of M microphone channels, $x_m[n]$ is the signal of the m-th channel at time n and the TDOA$^{(m,ref)}$ is the time delay of arrival (further explained in Section 4.3.2) between the m-th channel and a reference channel. Equation (2.1) describes weighted-delay and sum beamforming which is also applied as front-end enhancement method within the CHiME-5 baseline system (see Figure 2.3).

In the particular domain under investigation, recordings are made with Kinect arrays, like the one shown in Figure 2.2. The board comprises four microphones that are non-uniformly spaced as line array, with three microphones on the left side below the camera and a fourth microphone on the very right-hand side. Always the four microphone signals of one device are processed together so that one enhanced output signal is provided for each Kinect. The exact position of the six Kinects within the apartment is unknown and the position of the speakers varies over time. Under these conditions it is necessary to adapt the weights and to facilitate an automatic selection of the reference channel for the TDOA estimation that provides best enhancement results for a certain point in time. The weighted-delay and sum beamformer is based on the BeamformIt toolkit[4] and implements some superior techniques originally elaborated in order to face the problem of beamforming in typical meeting scenarios with a number of microphones arbitrarely placed within the meeting room. The main components and properties of BeamformIt are: (the interest reader is referred to [4] for a more details):

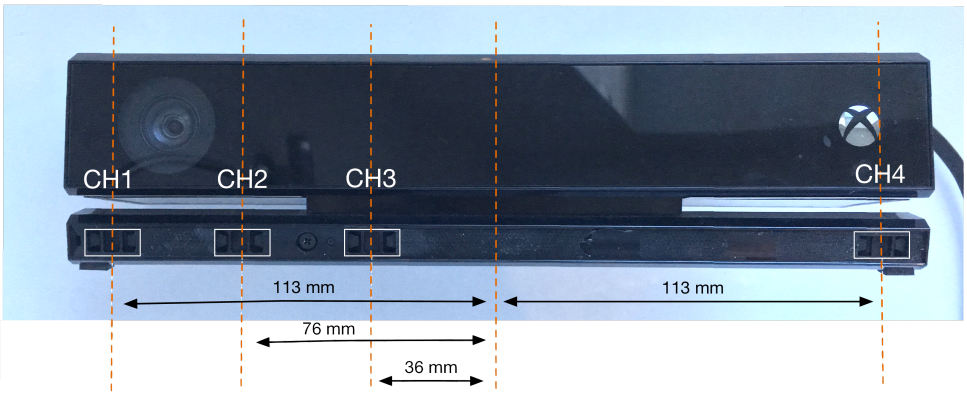


Figure 2.2: Microsoft Kinect device[3] with indicated microphone positions and the distance to each other. Since the four microphones are placed in line without an uniform spacing, the array device falls into the category of non-uniform linear arrays with ambiguities in the estimation of the source position along azimuth and elevation.

- Skew estimation to fix synchronisation errors between channels.
- Dynamic range expansion by applying a common weighting factor for the amplitude, computed over the full recording length of a channel.
- TDOA computation between channels of a Kinect device based on GCC-PHAT[7] (explained in Section 4.3.2).
- The accuracy of the TDOA estimation among channels highly depends on the quality of the chosen reference channel. The time-average of the cross-correlation between each channel is computed and the one providing the highest cross-correlation on average is nominated as the reference channel.
- Optimal TDOA estimation by computing N relative maxima of the cross-correlation (instead of only between one arbitrary channel and the reference channel). This should minimize false delay computations, which can be caused by spurious noises or overlapping speakers that overshoots the speakers local maximum in the cross-correlation.

[7] GCC-PHAT - generalized cross-correlation phase transform

- Postprocessing of the computed delays using noise thresholding and dual-pass viterbi decoding to discard unreliable TDOA values (e.g. obtained from silence regions) and to avoid delay switching in order to maximize speaker continuity.

2.3 Conventional ASR baseline system

In the 1970's the researchers James Baker and Janet M. Baker and a few years later also Fred Jelinek set a milestone in speech recognition by using hidden Markov models (HMMs) as a statistical way to model the time evolution of speech with states and their transitions. At the same time also the n-gram language model was introduced and has become a standard in the statistical description of word patterns within a language[8]. Only little has been changed so far and nowadays we often speak from a conventional ASR system when referring to a recognition system based on these ideas.

Considering a sequence of spectral compact acoustic vectors $\boldsymbol{Y} = \boldsymbol{y}_1, \boldsymbol{y}_2, ..., \boldsymbol{y}_T$ extracted from segments of the recorded speech signal by using an adequate signal processing technique, the problem in speech recognition is to find the most probable word sequence $\boldsymbol{w}^* = w_1, w_2, ..., w_N$ given $\boldsymbol{Y}$, i.e finding $\boldsymbol{w}$ that maximizes the conditional probability:

$$\boldsymbol{w}^* = \underset{\boldsymbol{w}}{\operatorname{argmax}} P(\boldsymbol{w}|\boldsymbol{Y}). \tag{2.2}$$

Solving this maximization problem is infeasible taking into account the sound diversity in the human speech and a large language vocabulary with more than 10000 words. Thus, thanks to Bayes' theorem the search for $\boldsymbol{w}^*$ can be split up into a product of two independent probabilities, namely the one of computing the *a-priori* probability of the word sequence $\boldsymbol{w}$ and secondly the calculation of the *likelihood* of the observation sequence $\boldsymbol{Y}$ given $\boldsymbol{w}$. Thus, we can write

$$\boldsymbol{w}^* = \underset{\boldsymbol{w}}{\operatorname{argmax}} P(\boldsymbol{Y}|\boldsymbol{w})P(\boldsymbol{w}), \tag{2.3}$$

where the *a-priori* probability is determined by a language model (LM) and the *likelihood* is provided from an acoustic model (AM). Please also note that the marginal probability of the observation $P(\boldsymbol{Y}) = \sum_{\boldsymbol{w}} P(\boldsymbol{Y}, \boldsymbol{w}) = \sum_{\boldsymbol{w}} P(\boldsymbol{Y}|\boldsymbol{w})P(\boldsymbol{w})$, part of the denominator in the Bayes' theorem, does not influence the result of the maximization itself and is therefore neglected in Equation (2.3). However, when comparing results among different ASR systems it is necessary to reintroduce this scaling factor since it depends on the nature of the acoustic model.

A LM can be defined from strict grammatic rules of the language under investigation, like context free grammars (CFG) or context sensitive grammars (CSG). However, these models require a careful design (usually linguist experts are needed), have problems in robustness when handling a large vocabulary and are ambiguous in their outcome. Hence, a statistical n-gram model trained on a large text corpus is the preferred method to make statements on the probability of words in a sequence. To handle the large number of possible words in a language, it is inpractical to gather observation statistics on word level and it makes sense to decompose each word into smaller distinctive sounds called phones. For example the english language incorporates about 40 individual phone sounds, all produced by the speaker while varying the position of the acoustic actors (e.g. lips, tongue, velar) in combination with an either voiced or unvoiced excitation signal controlled with the function of the vocal folds (see also source filter model in [8] for a more mathematical view on this). To execute the mapping from words to a decomposition in phones, usually a handcrafted (by human experts) or an auto-generated pronunciation dictionary is used.

[8] More on the history of ASR can be found on `https://en.wikipedia.org/wiki/Speech_recognition#History`

The acoustic model is a set of statistical hidden Markov models (HMMs), one for each individual phone, which are concatenated according to the given word sequence to form a composite HMM in order to compute the probability of that model given the acoustic observation sequence $\boldsymbol{Y}$. Each HMM is defined by states, state initial probabilities, transition probabilities among those states and emission probabilities. For monophone acoustic models, each phone is represented by one state only, in contrast a triphone model has three different states to allow context modelling with neighboring phones. For a deeper view into HMMs see also [9]. The emission probabilities of the HMM are typically modelled using a mixture of Gaussian components,i.e. a Gaussian mixture model (GMM), but recently also hybrid systems using deep neural networks (DNNs) have been widely used and lead to a better recognition performance. For example the work in [10] has made effective use of a time-delayed neural network (TDNN) for modelling long temporal contexts in a hybrid system.

Both the HMM and the GMM parameters need to be learned from data. For a HMM the Baum-Welch method is used to estimate the state and transition probabilities. Typically to learn the parameters for the GMMs an iterative algorithm called expectation maximization (EM) is employed. DNN training is performed with the backpropagation algorithm. For a more in depth description of the conventional approach in speech recognition the reader is referred to the explanations in [5] and [7].

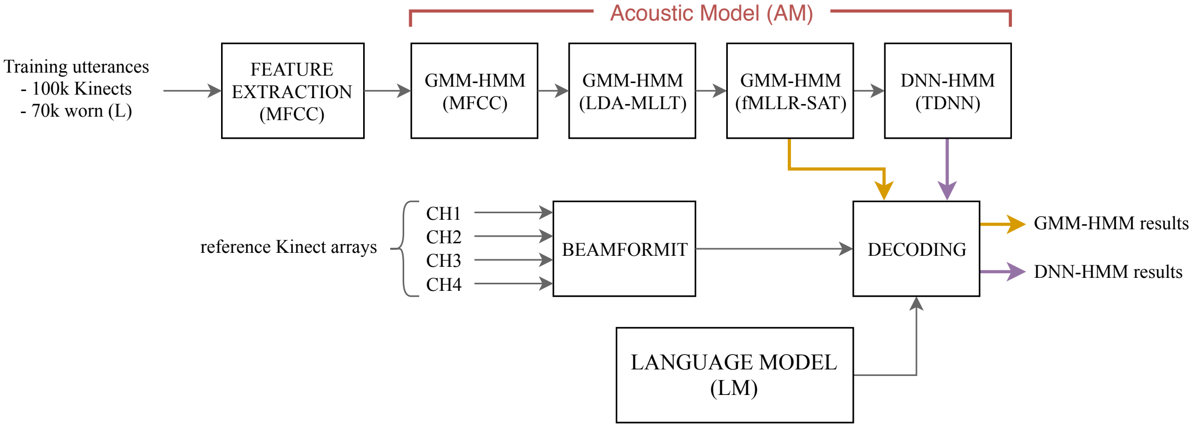


Figure 2.3: Main components of the conventional baselines system provided by the challenge organisers. The ASR system comprises a in-domain LM and an AM trained on a random subset of Kinect utterances together with all binaural utterances over various refinement training stages starting from a simple monophone model up to a hybrid DNN-HMM model.

Figure 2.3 depicts the conventional baseline system (be aware that the very first processing stage for array synchronisation is missing within this sketch). The acoustic model is trained on a subset of the recorded utterances from the training sessions, comprising of 100000 utterances randomly selected from the Kinect arrays and 70000 utterances from the speakers left channel recordings of the binaural microphones. Speech enhancement is only applied on the development reference Kinect recordings. The SRILM[9] toolkit is used to train an in-domain LM and the pronunciation lexicon is also constructed from the transcriptions with the help of Phonetisaurus[10]. The following parts describe the acoustic model of the conventional baseline system in more detail.

[9] SRILM: SRI language modelling toolkit; for code see `http://www.speech.sri.com/projects/srilm`

[10] Phonetisaurus: Lexicon construction using grapheme to phoneme conversion

2.3.1 Acoustic model

The baseline AM is trained over several stages, where a former trained GMM-HMM model provides the phone-level alignments of the feature data in order to train a hybrid DNN-HMM model in a supervised manner. First of all a monophone GMM-HMM model is trained from a flat start (i.e. without any provided alignments) on MFCC features extracted from a subset of the training utterances. The obtained alignments are then used to train a triphone model. This follows two refinement stages that make use of feature-space and model-space transformation in order to reduce the feature dimensionality, to enhance the phone class separability and also to adapt the model to the speakers. Finally, lattice free maximal mutual information (LF-MMI) is used to train the DNN of the hybrid system in a sequence discriminative manner.

Features are transformed using both, feature-space or model-space methods. Feature-space approaches convert a feature vector $\boldsymbol{X}$ into a vector $\boldsymbol{Y}$ via a linear or non-linear transformation operation $\mathcal{T}$ [11], i.e.

$$\boldsymbol{Y} = \mathcal{T}(\boldsymbol{X}). \tag{2.4}$$

This way the transformation solely applies to the feature vector and no model specific parameters are changed. Linear discriminant analysis (LDA) and maximum likelihood linear transformation (MLLT) are both feature-space methods and used in the baseline system to enhance the class separability. Another way to fit the system to data distortions and noise is achieved through model-domain methods. Hereby the transformation applies to the acoustic model parameters Λ which is often more effective because the optimization process is directly linked with the ASR objective, i.e.

$$\hat{\Lambda} = \mathcal{T}(\Lambda, \boldsymbol{X}). \tag{2.5}$$

This usually leads to a significant gain in ASR performance while it requires more computational time. The baseline uses feature space maximum likelihood linear regression (fMLLR) with speaker-adaptive training (SAT) for model-domain transformation.

MFCC

Mel frequency cepstral coefficients (MFCCs) are the dominant features in speech recognition to represent speech in a compact form. As illustrated in Figure 2.4 during feature extraction the speech signal goes through several stages that are designed upon perceptual and statistical properties of speech. In a first step the raw speech signal is processed with a pre-emphasis filter to balance the amplitude spectrum, usually for speech the lower frequencies contain most of the energy, while the upper part of the spectrum is low in amplitude. The filtered signal is then split into overlapping (i.e. 50%) frames of length typically between 10 ms - 40 ms for

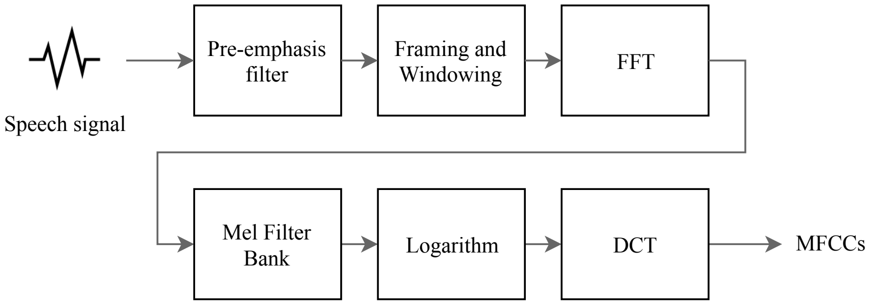


Figure 2.4: Processing stages of the MFCC feature extraction.

which speech is assumed to be stationary. Applying a window to each of the frames (a hamming window is typically used) helps to reduce spectral leakage. The Fourier transformation is used to transform the time frames into the spectral domain. In practice, the fast Fourier transformation (FFT) is applied in order to reduce the computational time. Following this, typically between $20-40$ Mel-filterbank features are computed from the periodogram power spectral estimate applying triangular filters. Important here is the applied non-linear spectral distribution of the filters according to the Mel-scale that mimics the non-linear human ear perception of sound. By taking the logarithm the log-filterbank energies are obtained. In a very last step a discrete cosine transformation (DCT) decorrelates the obtained coefficients from the previous filter banks[11]. Usually for ASR only the first $12-13$ coefficients are used. Table 2.1 summarizes the most important MFCC extraction settings used in the baseline system.

Setting	**Value**
pre-emphsis coeff.	$\alpha = 0.97$
window-type	"povey"
frame length	25 ms
frame shift	10 ms
mel-frequency bins	23
MFCC coefficients	13

Table 2.1: Essential MFCC extraction settings used in the baseline system. The "povey" window is similar to the hamming window type, but with the property that it goes to zero at both edges.

LDA

Even though MFCCs are well established features in the field of ASR they lack from sparse time context of the signal due to the applied short time window. In practice MFCC features are therefore augmented by their differential and acceleration coefficients (first and second order delta feature) to introduce more information on the speech dynamics. This consequently spans the feature vector to maximal 100 ms. However, it has been shown by Yang et. al [12], that the best accuracy in phone recognition is achieved for an observation window of about 200 ms. One way to mitigate this drawback can be achieved by a former splicing of consecutive MFCC feature frames with an additional transformation to reduce the obtained dimensionality. A common method is called linear discriminant analysis (LDA), which tries to find a projection of the original feature to a subspace by maximizing the between-class and minimizing the within-class scatter matrix [13]. In this sense classes can be modelled by phones or even by HMM states. The transformation in general allows a better discrimination of the classes and has been shown to be beneficial for ASR [14].

LDA is a generalisation of Fisher's linear discriminant [15] and applies to an arbitrary number of classes. It performs a dimensionality reduction from the D-dimensional observation to the P-dimensional feature space using a linear projection matrix $\boldsymbol{W}$. Since projection onto a lower dimension may lead to significant loss of information, $\boldsymbol{W}$ underlies a careful design aiming to maximize the between-class variation while simultaneously minimizing the within-class variation to prevent classes from overlapping. With J number of classes and N_j data samples per class

[11] See also the explanation at `https://haythamfayek.com/2016/04/21/speech-processing-for-machine-learning.html`

the mean $\boldsymbol{\mu}_j$ and the covariance matrix $\boldsymbol{S}^j$ of a normal distributed class j is given with

$$\boldsymbol{\mu}_j = \frac{1}{N_j} \sum_{i=1}^{N_j} \boldsymbol{x}_{ji} \tag{2.6}$$

$$\boldsymbol{S}^j = \frac{1}{N_j} \sum_{i=1}^{N_j} \left(\boldsymbol{x}_{ji} - \boldsymbol{\mu}_j\right) \left(\boldsymbol{x}_{ji} - \boldsymbol{\mu}_j\right)^T, \tag{2.7}$$

where $\boldsymbol{x}_{ji}$ is the ith sample from class j. With the assumption of identical class covariances $\left(\boldsymbol{S}^j = \boldsymbol{S}, j \in \{1, ..., J\}\right)$ the within-class covariance $\boldsymbol{S}_W$ and the between-class covariance $\boldsymbol{S}_B$ can be defined as

$$\boldsymbol{S}_W = \sum_{j=1}^{J} \boldsymbol{S}^j \tag{2.8}$$

$$\boldsymbol{S}_B = \sum_{j=1}^{J} N_j \left(\boldsymbol{\mu}_j - \boldsymbol{\mu}\right) \left(\boldsymbol{\mu}_j - \boldsymbol{\mu}\right)^T, \tag{2.9}$$

where $\boldsymbol{\mu} = \frac{1}{N} \sum_{i=1}^{N} \boldsymbol{x}_i$ is the global mean and $N = \sum_{j=1}^{J} N_j$ is the total number of samples. The linear transformation from the observation into the subspace is accomplished with

$$\boldsymbol{y} = \boldsymbol{W}^T \boldsymbol{x}. \tag{2.10}$$

Applying this transformation to the mean and the class covariance defined in Equation 2.6 and 2.7 respectively, we can write the corresponding expressions in the subspace:

$$\tilde{\boldsymbol{\mu}}_j = \frac{1}{N_j} \sum_{i=1}^{N_j} \boldsymbol{W}^T \boldsymbol{x}_{\boldsymbol{ji}} = \boldsymbol{W}^T \boldsymbol{\mu}_j \tag{2.11}$$

$$\tilde{\boldsymbol{S}}^j = \frac{1}{N_j} \sum_{i=1}^{N_j} \left(\boldsymbol{W}^T \boldsymbol{x}_{ji} - \boldsymbol{W}^T \boldsymbol{\mu}_j\right) \left(\boldsymbol{W}^T \boldsymbol{x}_{ji} - \boldsymbol{W}^T \boldsymbol{\mu}_j\right)^T = \boldsymbol{W}^T \boldsymbol{S}^j \boldsymbol{W}. \tag{2.12}$$

With this we can write the corresponding within-class and between-class covariances after the applied transformation, i.e.

$$\tilde{\boldsymbol{S}}_W = \boldsymbol{W}^T \boldsymbol{S}_W \boldsymbol{W} \qquad \text{and} \tag{2.13}$$

$$\tilde{\boldsymbol{S}}_B = \boldsymbol{W}^T \boldsymbol{S}_B \boldsymbol{W}. \tag{2.14}$$

The optimal solution to this problem is given by the transformation matrix $\boldsymbol{W}$ that maximizes the quotient

$$J(\boldsymbol{W}) = \frac{|\tilde{\boldsymbol{S}}_B|}{|\tilde{\boldsymbol{S}}_W|} = \frac{|\boldsymbol{W}^T \boldsymbol{S}_B \boldsymbol{W}|}{|\boldsymbol{W}^T \boldsymbol{S}_W \boldsymbol{W}|}. \tag{2.15}$$

A solution can be found by taking the first p eigenvectors of the matrix $\boldsymbol{S}_W^{-1} \boldsymbol{S}_B$ for a p-dimensional projection, as derived in [16].

MLLT

In a GMM-HMM model, the observation probabilities of the HMM states are modelled with Gaussian mixture models (GMMs). A standard GMM with N components and parameters

$\Theta = \{w_i, \boldsymbol{\mu}_i \boldsymbol{\Sigma}_j\}_{i=1}^N$ has the form

$$p(\boldsymbol{x}|\Theta) = \sum_{i=1}^{N} w_i \mathcal{N}(\boldsymbol{x}|\boldsymbol{\mu}_i, \boldsymbol{\Sigma}_i). \tag{2.16}$$

Usually a constrained diagonal covariance matrix $\boldsymbol{\Sigma}$ is used under the assumption of statistical independent feature components, mainly to reduce the number of parameters to learn but also to overcome noisy parameter estimates due to data sparsity. Maximum likelihood linear transformation (MLLT) is a technique for linear transforming the features so that the structural diagonal constraint in the covariance matrix is more valid by means of maximum likelihood [17].

MLLR

Maximum likelihood linear regression (MLLR) is an acoustic model adaptation technique that transforms the means and the covariances of class separated GMM distributions using independently estimated transformation matrices. A common usecase for this kind of transformation is the adaptation of the HMM states emission distributions of the acoustic model towards a channel, an acoustic scene or a specific speaker. Unlike in maximum a-posteriori estimation (MAP), where only model parameters seen in the adaptation data are updated, in MLLR the same transformation applies to a regression class (i.e multiple states or a set of phonemes) and therefore a respectively smaller set of adaptation data is required. A regression class is a set of Gaussian distributions and can be obtained by building a regression class tree. Hence, even if the available adaptation data is tiny, a single transformation can still be applied to a class incorporating all emission distributions. In doing so, the transformation itself maintains the weights of the individual Gaussians of the speaker independent model [6,18,19]. Figure 2.5 demonstrates the MLLR transformation of a group of four two-dimensional Gaussian distributions. Note that each individual distribution is moved in the same direction. With D being the dimension of the observation space, the general transformation of the mean vector and the covariance matrices is given as

$$\hat{\boldsymbol{\mu}} = \boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{b} \qquad \text{and} \tag{2.17}$$

$$\hat{\boldsymbol{\Sigma}} = \boldsymbol{H}\boldsymbol{\Sigma}\boldsymbol{H}^T, \tag{2.18}$$

where $\boldsymbol{A}$ and $\boldsymbol{H}$ are both $D \times D$ dimensional transformation matrices and $\boldsymbol{b}$ an additive D-dimensional vector to be obtained from the data using maximum likelihood. A solution to the problem and additional implementation issues are provided in [6].

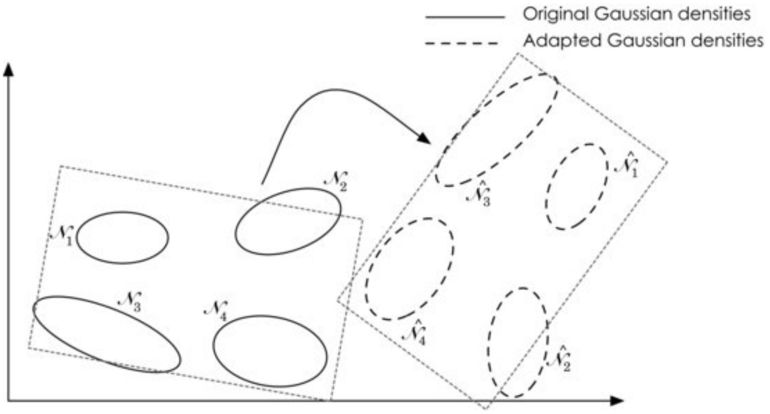


Figure 2.5: Example of MLLR adaptation in a two-dimensional feature space, taken from [19]. Four Gaussians are grouped into one class, as illustrated by the rectangle frame, and equally transformed.

fMLLR

Constrained maximum likelihood linear regression (CMLLR) is a constrained version of the more general MLLR method described above. It is still a technique to adapt the model parameters, however due to a single applied adaptation matrix for both the mean and the variance the number of parameters to estimate is smaller, hence less adaptation data is required. The general mean and variance transformation is given with [6]:

$$\hat{\boldsymbol{\mu}} = \boldsymbol{A}\boldsymbol{\mu} + \boldsymbol{b} \tag{2.19}$$

$$\hat{\boldsymbol{\Sigma}} = \boldsymbol{A}^T\boldsymbol{\Sigma}\boldsymbol{A} \tag{2.20}$$

Due to this constraints a feature transformation can be found that behaves equivalent as adapting the model itself, simply by taking the inverse of the CMLLR feature matrix. The modification of the feature vector $\boldsymbol{x}_t$ at time t can be written as

$$\hat{\boldsymbol{x}}_t = \boldsymbol{A}^{-1}\boldsymbol{x}_t + \boldsymbol{A}^{-1}\boldsymbol{b}. \tag{2.21}$$

Therefore, this method is often called feature space maximum likelihood linear regression (fMLLR). This is actually very powerful in a way that given an acoustic model, the fMLLR matrix can be estimated separately for each speaker to transform the features towards a uniform speaker[12].

SAT

Typically speaker adaptation starts from a speaker independent model trained on data from multiple speakers which gets adapted towards a speaker present in the test data. Authors in [20] proposed speaker adaptive training (SAT) that already uses either MLLR or CMLLR/fMLLR transformations during model training in order to reduce the speaker variation and the variations between channels or acoustic conditions. The baseline makes use of fMLLR/SAT adaptive training. In fMLLR/SAT the training data is grouped by speakers. After an initial model training on the full training data fMLLR transformations are computed from the data corresponding to a specific speaker. Each of the transformations is then applied to the acoustic feature vectors, which are used to reestimate the model parameters. The whole process can be repeated until convergence (see also the explanations in [19]).

TDNN

The nature of speech is sequential with long time dependencies among its acoustic features. Modelling of these temporal relations within a neural network can be achieved either by concatenating contiguous input feature vectors to construct a certain context or by using network architectures that are capable to exploit time dependencies such as recurrent neural networks (RNNs). However, the recurrent connections in a RNN structure make parallelized training with multiple computational resources, like CPUs or GPUs, not expoitable in the same way how it can be achieved for feed-forward neural networks. In order to make use of parallelized training while modelling a long temporal context at the same time, the baseline system employes a time delayed neural network (TDNN) [10].

In a feed-forward neural network the activation $a_i^{(k)}$ of neuron i at layer k is typically computed by a weighted summation over the J previous layer outputs $\boldsymbol{v}^{k-1}$ as follows

$$a_i^{(k)} = \sum_{j=1}^{J} w_{ij}^{(k)} v_j^{(k-1)}. \tag{2.22}$$

[12] For an more detailed overview on fMLLR see also http://data.cstr.ed.ac.uk/asr/2017-18/asr-lec11.html

For a TDNN the activation of the neural unit not only depends on the current, but the delayed inputs as illustrated in Figure 2.6. The temporal width is defined by the number of delays N. Hence, we can formulate the activation value $a_i^{(k)}$ as an element wise sum over the weighted current and delayed inputs

$$a_i^{(k)} = \sum_{j=1}^{J} \sum_{n=0}^{N} w_{ij+n}^{(k)} v_{j+n}^{(k-1)}. \tag{2.23}$$

In this way the inherent structure of a feed-forward network is maintained and a temporal context is constructed with the introduced delayed input frames. However, training of this network can be expensive, as the number of multiplications increases exponentially with the number of introduced delays N. Therefore, often sub-sampling is applied as described in [10]. The basic idea of sub-sampling is to skip the calculation for a majority of hidden activations in each layer in a structured way, to allow information from all input time steps to influence the network output while keeping the number of hidden layer activations to be evaluated low. An example how this method works is illustrated in Figure 2.7 for a network with four layers, where each rectangle represents a data frame at time t. Using an input context of $[-2,+2]$ with a subsequent context of $[-1,2]$, $[-3,3]$, $[-7,2]$ for the three upper layers, requires to compute activations for each time step. The sub-network in red splices together the input frames $[-2,2]$ and $\{-1,2\}$, $\{-3,3\}$, $\{-7,2\}$ for the upper layers. With this the computational cost is reduced by a factor of ~ 10. For acoustic modelling in the baseline system, a TDNN with 8 layers, rectified linear unit (ReLU) activations, and context specifications as listed Table 2.2 is employed.

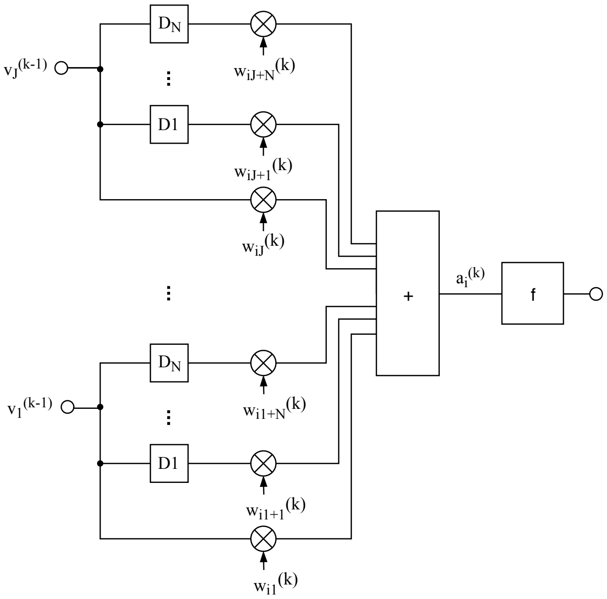


Figure 2.6: TDNN neural unit redrawn from [21]. The input to the activation function f is the sum over the weighted delayed inputs.

LF-MMI

A neural network with multiple classes is typically trained in a discriminative manner. The sigmoid function at the last layer forces the sum of the class probabilities to 1, i.e. a higher probability for a specific label results in a probability decrease of the other outputs. In speech recognition the output labels are frame tight (i.e. phones) and the weights are adjusted in order to minimize the cross-entropy error. Sequence discriminative training with maximum mutual information (MMI) can be performed, although it requires an initial cross-entropy pass

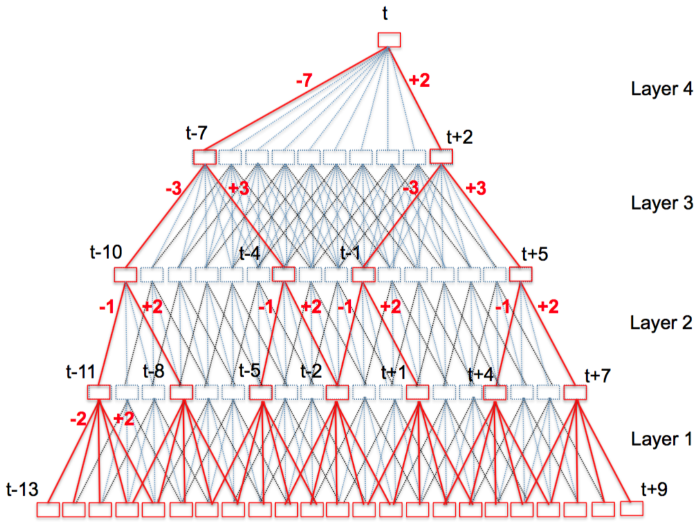


Figure 2.7: Computation of TDNN network output with sub-sampling (red) and without sub-sampling (blue + red) taken from [10].

Layer	**Input context**
1	$\{0\}$
2	$\{-1, 0, 1\}$
3	$\{0\}$
4	$\{-1, 0, 1\}$
5	$\{0\}$
6	$\{-3, 0, 3\}$
7	$\{-3, 0, 3\}$
8	$\{-6, -3, 0\}$

Table 2.2: Input context (frames) for each layer of the baseline TDNN architecture.

in order to generate the alignments, the lattices and also to initialize the networks weights. MMI improves the recognition accuracy as demonstarted in [22]. However, the training is quite expensive due to the required initial cross-entropy training and because of the time and resources needed to build and store the lattices. Lattice free maximum mutual information (LF-MMI) is a more state-of-the-art discriminative training procedure for neural networks that does not require an initial cross-entropy model and pre-generated lattices. Instead the total likelihood in the denominator is calculated directly using forward-backward computations. To speed up the overall training, serveral tricks are used as reported in the original paper [23].

2.4 Baseline results

The official baseline results on the development set for both the conventional and the end-to-end baseline system are presented in Table 2.3. The hybrid model (DNN-HMM) introduces a great improvement in recognition accuracy. The end-to-end (E2E) ASR system does not perform well with this amount of training data. Compared with the binaural microphone results in Table 2.4 we can conclude that the performance lacks mainly from the distance of the speakers to the microphone arrays, as the recognition accuracy obtained using the close-talk recordings is much higher.

Session	**S02**			**S09**			**Overall**
Location	**Kit**	**Din**	**Liv**	**Kit**	**Din**	**Liv**	
GMM-HMM	93.9	91.4	90.8	90.8	90.9	88.1	91.7
DNN-HMM	86.6	79.1	78.4	82.6	81.7	77.9	81.1
E2E	-	-	-	-	-	-	94.7

Table 2.3: Published WER [%] results on the development set using the reference Kinect array with beamforming (single-array track) for the conventional (GMM-HMM), hybrid (DNN-HMM) and end-to-end (E2E) system.

baseline	**Dev**
GMM-HMM	72.8
DNN-HMM	47.9
E2E	67.2

Table 2.4: Baseline WERs [%] for the binaural close-talk microphones. The actual recording from the speakers microphone was used for decoding (oracle).

3 Dataset

The 5$^{\text{th}}$ CHiME challenge [1] considers the problem of *distant multiple-microphone conversational speech recognition in everyday home environments*. The title itself describes the issue of a very realistic dataset and problem. In the current chapter the data is described in more detail. The information was taken from the challenge's website[13], onto which the reader is referred for a full description on the challenge data collection and preparation.

3.1 CHiME-5 corpus

3.1.1 Background

In general data was elicited from 20 dinner party scenarios taking place in real homes with a mixed group of 4 participants over an average duration of 2 hours. Each session was hosted in a different home with different speakers. The concept of *conversational speech* describes the scenario of multiple speakers behaving naturally without any script, i.e a causal and informal talk. This is often characterized by a remarkable degree of spontaneous and overlapping speech, which makes especially the applicability of conventional multiple-microphone speech enhancement techniques challenging. The term part of *everyday home environments* depicts the specific surrounding in which the speakers are located and move around. Specifically, it addresses the scenario of a dinner party, with all kind of noise sources imaginable to fit this type scenario. One can think of stationary noise signals coming from the air conditioning system or the oven, but also high energy and short time noises emitted by kitchen utensils which distract the speech and mitigate the SNR. However, we do not need to deal with other speech sources from radios and television devices, since they were switched off during recording time due to licensing issues. Furthermore, the participants were encouraged to change room every 30 minutes between kitchen, dining room and living room. This naturally involves a switch of the acoustic environment, hence an additional degree of freedom to consider and probably to put to good use while designing the overall ASR system. All this is recorded using *multiple microphones*, some placed within the rooms in *distance* to the speakers and others worn by the speakers itself. As close-talk microphones, a set of Soundman OKM II Classic Studio binaural microphones are used and the audio from these is recorded via Soundman A3 adapter onto Tascam DR-05 recorders. The close-talk recordings were required to facilitate the manual transcriptions.

3.1.2 Audio data

The recordings from the 20 different dinner parties are split up into three sets for training, development and evaluation. The training set comprises 16 sessions and 2 sessions are assigned for each development and evaluation set. Each session was recorded by six distant microphone arrays and additional binaural microphones, one for each speaker. Tables 3.3, 3.1, 3.2 list details on the recorded sessions for the training, development and evaluation set, respectively. The

[13] CHiME-5 challenge website: `http://spandh.dcs.shef.ac.uk/chime_challenge/`

Session ID	Participants (Bold=Male)	Duration	#Utts	Notes
S02	p05, **P06, P07**, p08	2 : 28 : 24	3, 822	
S09	p25, p26, p27, p28	1 : 59 : 21	3, 618	U05 missing

Table 3.1: Development sessions.

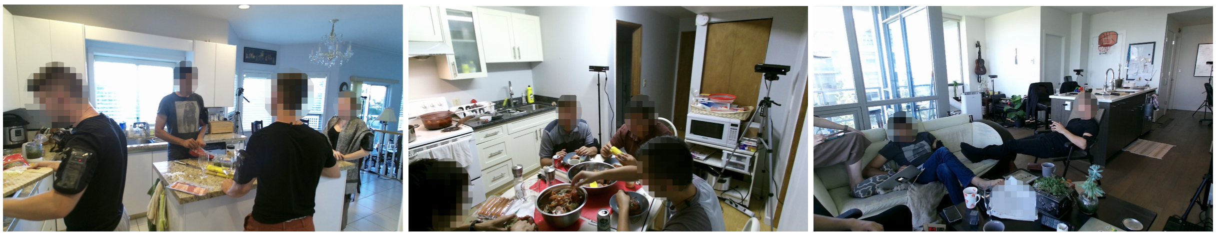
Session ID	Participants (Bold=Male)	Duration	#Utts	Notes
S01	p01, p02, **P03**, p04	2 : 39 : 04	5, 797	No registration tone
S21	**P45**, p46, **P47**, p48	2 : 33 : 20	5, 231	

Table 3.2: Evaluation sessions.

Session ID	Participants (Bold=Male)	Duration	#Utts	Notes
S03	**P09, P10, P11, P12**	2 : 11 : 22	4, 090	P11 dropped from min 15 to 30
S04	**P09, P10, P11, P12**	2 : 29 : 36	5, 563	
S05	**P13**, p14, p15, **P16**	2 : 31 : 44	4, 939	U03 missing (crashed)
S06	**P13**, p14, p15, **P16**	2 : 20 : 06	5, 097	
S07	p17, **P18**, p19, **P20**	2 : 26 : 53	3, 656	
S17	p17, **P18**, p19, **P20**	2 : 32 : 16	5, 892	
S08	**P21, P22, P23, P24**	2 : 31 : 35	6, 175	
S16	**P21, P22, P23, P24**	2 : 32 : 19	5, 004	
S12	**P33, P34, P35**, p36	2 : 29 : 24	3, 300	Last 15 minutes of U05 missing (accidentally turned off)
S13	**P33, P34, P35**, p36	2 : 30 : 11	4, 193	
S19	p49, **P50, P51**, p52	2 : 32 : 38	4, 292	
S20	p49, **P50, P51**, p52	2 : 18 : 04	5, 365	
S18	p41, **P42**, p43, p44	2 : 42 : 23	4, 907	
S22	p41, **P42**, p43, p44	2 : 35 : 44	4, 758	U03 missing
S23	p53, **P54, P55**, p56	2 : 58 : 43	7, 054	Neighbour interrupts
S24	p53, **P54, P55**, p56	2 : 37 : 09	5, 695	P54 mic unreliable, P53 disconnects for bathroom

Table 3.3: Training sessions.

session is tagged with a unique session id, it has four participants and a minimal duration of two hours. What makes this dataset interesting and also challenging for the speech recognition system design, are its few restrictions in how the data is collected. Between sessions we can find large differences in the acoustical environment, the gender of the participants and in the position of the array microphones. Particularly speaking, each session was held in a different flat with different floor plans, from tiny apartments with a traditional room separation up to large open plan spaces with few walls (see Figure 3.2 for two example floor plans). This creates a dataset with a wide range of acoustical scenarios that makes each session unique. Therefore also the placement of the Kinect arrays is not prescribed and it is only important that at least two arrays cover one room (kitchen, dining room, living room). For pictures of example rooms see Figure 3.1. Besides the environment, different participants for most of the sessions introduce a further versatility among sessions. Hereby it is important to point out the variance



(a) kitchen (b) dining room (c) living room

Figure 3.1: Example rooms in which the recording of the dataset took place.

in the proportion of men and women among training, development and evaluation set. In the training set we can count a mean of 2.5 men per session and an overall men to woman ratio of 40/24, whereby in the development and evaluation set the presence of female speakers is more dominant with 3 women and 2.5 women, respectively. The amount of utterances per session count from 3300 in session S12 to a maximum of 7054 in session S23. The comparable few numbers of utterances in the development set is of remark. Furthermore individual sessions leak on recording data, mostly due to unpredictable behaviors of the recording equipment, details on this are listed in the the last line of the tables. Concluding, a first glance on the CHiME-5 dataset at abstract level already reveals interesting information on its content, structure and it is somehow evident that work needs to be done in different stages of the recognition system in order to deal with the high variability and the sparse development set that might make system validation and comparison difficult.

3.1.3 Transcriptions

The transcriptions of the recorded material have been manually annotated with the help of the binaural recordings and stored within json files, one file per session. Apart from the transcriptions, each entry of the json file contains additional information on the recorded data (e.g speaker id and location) and the array alignments (see Section 2.1). A transcription example for an utterance from session S02 of the development set is depicted in Listing 3.1. All other utterance transcriptions are structured in the same way containing manually annotated start and end times for the binaural recordings, its corresponding aligned times for the Kinect arrays ”U01-U06”, as well as speaker and session ids. ”location” and ”ref” array are only provided for development and evaluation set (not for the training set). The tags listed in Table 3.4 are applied in cases of non verbal utterances.

```
{
    ”end_time”: {
        ”original”: ”0:00:43.82”,
        ”U01”: ”0:00:43.85”,
        ”U02”: ”0:00:43.84”,
        ”U03”: ”0:00:43.83”,
        ”U04”: ”0:00:43.83”,
        ”U05”: ”0:00:43.82”,
        ”U06”: ”0:00:43.82”,
        ”P05”: ”0:00:43.82”,
        ”P06”: ”0:00:43.82”,
        ”P07”: ”0:00:43.82”,
        ”P08”: ”0:00:43.82”
    },
    ”start_time”: {
        ”original”: ”0:00:40.60”,
        ”U01”: ”0:00:40.63”,
        ”U02”: ”0:00:40.62”,
            ”U03”: ”0:00:40.61”,
            ”U04”: ”0:00:40.61”,
            ”U05”: ”0:00:40.60”,
            ”U06”: ”0:00:40.60”,
            ”P05”: ”0:00:40.60”,
            ”P06”: ”0:00:40.60”,
            ”P07”: ”0:00:40.60”,
            ”P08”: ”0:00:40.60”
        },
        ”words”: ”[laughs] It’s the blue, I think. I think.”,
        ”speaker”: ”P05”,
        ”ref”: ”U02”,
        ”location”: ”kitchen”,
        ”session_id”: ”S02”
    },
```

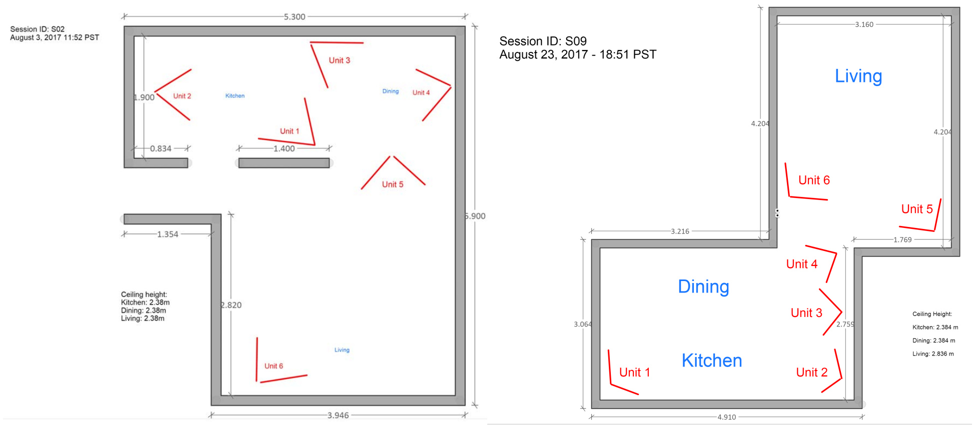
Listing 3.1: First utterance belonging to the transcription file of session S02 of the development set. The aligned utterance start and end times are denoted for the binaural microphone and the Kinect arrays. Additional to the transcription itself, information on the speaker, the reference Kinect, the location and the session id is provided (available information varies among training, development and evaluation set).

Tag	**Meaning**
[*noise*]	denotes any non-language noise made by the speaker (ex: grunts, coughing, loud, chewing, loud lip smacking etc.)
[*inaudible*]	denotes speech that is audible but not clear enough to be transcribed
[*laughts*]	denotes an instance where the participant laughs
[*redacted*]	are parts of the signals that have been zeroed out for privacy reasons (speaker id not given)

Table 3.4: Transcription tags used to name non speech and unintelligible utterance parts of the recordings.

3.1.4 Floor plans

Floor plans for all sessions are provided. Figure 3.2 illustrates the floor plans from both sessions of the developments set; the ones for the training set can be found in Section C.1 of the Appendix. It is worth to notice that the room partitioning highly varies among apartments from conventional flats with wall separated rooms to open space designs where sounds can propagate freely within the apartment. Especially for the open spaced scenarios each Kinect array can be a candidate to deliver a good recording, even when it is positioned far away from the speakers.

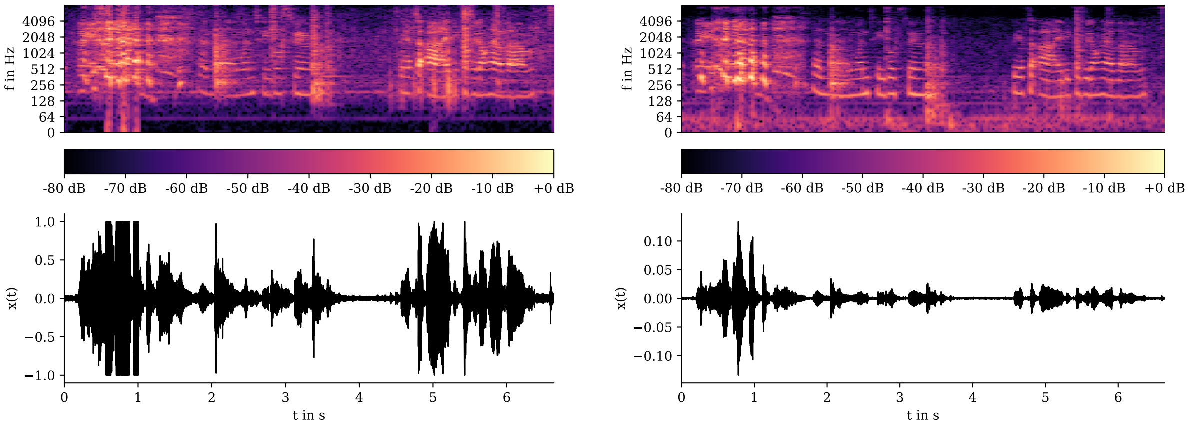


(a) Session: S02 (b) Session: S09

Figure 3.2: Floorplans of both development sessions. The location and orientation of the six Kinect arrays are marked in red.

3.2 Recordings

Multiple microphones, including 6 Kinect arrays and 4 binaural microphones ($6 \times 4 + 4 \times 2 = 32$) were used to collect conversation material from 20 dinner parties. Listening to randomly parts of the recordings and comparing different channels is a good way to get a first glance of the data. Figure 3.3 illustrates the spectrograms and the time domain representations of an utterance from session S02 for an arbitrary selected Kinect microphone channel and its corresponding recording from the left channel of the binaural microphone. This recording slice contains laughter followed by a female utterance with background noises from the kitchen, like clanking pots.



(a) Recording of channel CH1 from Kinect device U01. ID: P05_S02_U01_CH1_KITCHEN-0015986-0016650

(b) Left channel of the binaural recorded audio from microphone with speaker id P08. ID: P05_S02_KITCHEN-0015991-0016655

Figure 3.3: Spectrogram and time domain representation of an utterance from speaker P05 extracted from the development session S02.

Presented very well in both figures is the difference in the dynamic range of the signals. The Kinect recordings are usually much louder as observable in the time domain in Figure 3.3. They clip very often, not only during parts with laughter. This is due to the far distance of the array to the speakers, which demands for a higher microphone gain to capture each utterance at an adequate level. Consequently, background noises are more prominent and the overall SNR is low. On the other hand the binaural recordings present small amplitudes in most of the recorded sessions, but nevertheless transient noises still disturb the speech signal. Moreover, at least from a listening point of view, no major signal quality differences are audible among Kinect channels.

3.3 Simultaneous speech

A key point what makes a proper recognition of the recorded utterances very challenging, is the amount of overlapping or simultaneous speech as an integral part of conversational speech. Using the time information from the transcriptions, a rough analysis on the occurrence of cross-talk situations for each session can be made. The obtained results are depicted in Figure 3.4. For each session a histogram presents the percentage of non-speech duration as well as overall duration on single, double, triple and quadruple talk over the course of a session. Even if the share of single talk dominates, most of the sessions incorporate also a remarkable amount of double talk. An interesting fact is the difference in simultaneous speech between sessions from the training and the development set. Both development sessions show a smaller amount of sections without speech and more simultaneous speech from two or three speakers. This might be one reason to justify the poor baseline recognition results on the development set in Table 2.3.

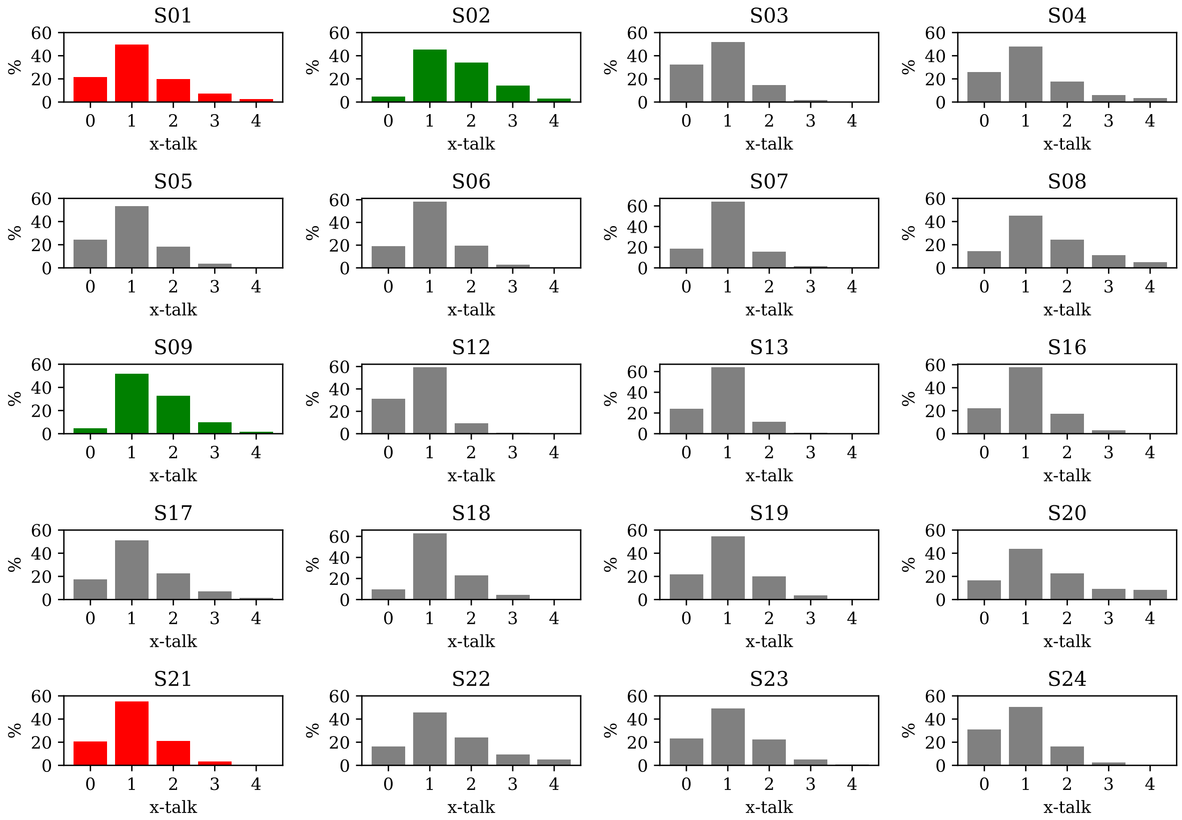


Figure 3.4: Percentage of no talk, single, double, triple and quadruple talk for each session. Statistics for the sessions part of the training set are grey colored, the ones from the development set are presented in green color and the remaining two sessions from the evaluation set in red.

4

Channel Selection and Fusion

The following section introduces a channel selection method to increase the recognition accuracy on the development set. Therefore we start the discussion with the main idea behind channel selection in Section 4.1. Training of a channel classifier requires the ground truth, which is obtained by an oracle. The definition and further introduction to the oracle analysis is going to be explained in Section 4.2. The major part of this chapter deals with appropriate feature extractions directly from the signal or the ASR decoder in Section 4.3. Finally, we introduce ROVER in Section 4.3 as a way to combine different hypotheses.

4.1 Channel selection

The setup how the challenge data was recorded comprises a large amount of microphone channels to gather the conversations from different locations within the apartments. Due to this distributed setup each microphone is placed with a different distance and angle towards the speaker(s). Utterances are even recorded from rooms where the speakers not reside. In this way each channel somehow records an individual representation of the spoken utterance within the environment, that in particular depends on the distance to the speaker, the reverberation and the noise distortion present at the microphones position at a certain point in time. Consequently, utterance recordings may vary strongly between channels and we can assume that there is one channel or a set of several channels that gather the spoken utterance with less distortions, or that fit the properties of the acoustic model better.

The idea behind channel selection is to select the channel that provides the highest recognition accuracy for subsequent decoding. Since the WER information is unknown for unseen data, it is required to find proper features to allow a correct ranking and successive selection of the channels. Suitable for this are either signal-based features, directly extracted from the recordings with the help of signal processing techniques, or features derived from the ASR decoding stage, denoted as decoder-based features.

A simple form of channel selection is already employed in the single-array track of the challenge. Here the additional visual information recorded from the onto the Kinect mounted video cameras is used to select a reference Kinect device accordingly to the estimated speaker distance. As this is a very rough selection process, there is hope to do better using more sophisticated features and methods as we propose in this work.

We approach channel selection in two ways. First of all the extracted features can directly be used to rank the channels; this method is titled as *feature direct classification*. The second method employs the features to train a deep neural network (DNN) to be capable of classifying and ranking the channels quality. In order to train this network in a supervised manner, target labels are required that are obtained from an oracle analysis as described in the following section.

4.2 Oracle

We may come across various definitions when searching for the word "oracle" on the internet. On wikipedia the first appearing sentences explain the word as *"[...] a person or agency considered to provide wise and insightful counsel or prophetic predictions or precognition of the future, inspired by the gods. As such it is a form of divination."*[14]

In the field of speech recognition we can define the oracle as a fully informed tool used to check the possible improvement in the recognition accuracy when employing a channel selection technique, exploiting the knowledge how each individual hypothesis scores. Hence, it executes a helpful analysis to estimate the ASR performance when picking always the hypothesis that provides the lowest WER and can furthermore be used to rank the available channels. Of course this is only possible when the reference transcriptions are provided, since they are necessary in order to compute the WER for each candidate hypothesis, which is not the case for unseen data. The main steps consist of decoding each utterance with a fully trained ASR system, followed by hypotheses ranking. The main drawback of this analysis is the huge computational power required for the decoding steps in order to generate all the hypotheses.

4.3 Features

Appropriate features are required that contain valuable information on the quality of the channel recording with strong relations to the oracle results and the outcome of the ASR system. The features are key to make this idea of channel classification even possible and therefore require great attention. In general we can distinguish between two main types of features, signal-based and decoder-based features. The former ones are directly obtained from the time signals of the recorded utterances, whereby for the latter ones information is extracted from the ASR model (i.e. DNN posterior probabilities). Decoder-based features are therefore closer to the striving goal, since they are obtained from behind the ASR chain. In the following we discuss on four signal-based and two decoder-based features, used to train the channel classifier as described in Chapter 5.

4.3.1 Signal energy

We compute the normalized energy of the recorded utterance u as

$$x_m^u[n] = \frac{1}{N_e - N_s + 1} \sum_{n=N_s}^{N_e} |s_m^u[n]|^2, \tag{4.1}$$

where s_m^u is the signal from microphone m and N_s and N_e the start and end time of the utterance u in samples, respectively. The scaling ensures that energies among channels are independent from the signal length. The final feature vector for utterance u is then composed by the signal energies from M different channels $\boldsymbol{x}^u = [x_1^u, x_2^u, \ldots, x_M^u]^T$.

4.3.2 Time Delay of Arrival (peak)

The time a signal requires to get from the transmitter (sender) to the antenna of the receiver is denoted as the time of arrival (TOA) or the time of flight (TOF). Likewise in telecommunications the same value is used to describe the time duration of an acoustic signal (e.g. speech) from its origin to a microphone. Once several microphones are used for recording, we can furthermore

[14] https://en.wikipedia.org/wiki/Oracle

determine the delay of arrival among the different microphones, that will obviously depend on the position of the microphones and the position of the source itself. This delay in time is known as the time delay of arrival (TDOA) and describes the time difference in the arrival of a signal between a target microphone and a reference microphone. Knowing the position of both microphones and the propagation speed of the signal, the TDOA can be used to estimate the source position but also to track a speaker over time if the quantity is computed in an online-fashion. Here a degree of freedom is the length of the analysis window, that influences the tracking resolution and the robustness of the estimate. Compared to the TOA, for the TDOA no information on the exact start time of transmission is needed, since the location is inferred from the time delays. The obtained steering direction is often used in a beamformer for enhancement purposes. A classical way to estimate the TDOA between a signal $x_i[n]$ and a reference signal $x_{ref}[n]$ uses the delay d that maximizes the cross-correlation between the two segments:

$$R_{i,ref}(d) = \sum_{n=0}^{N} x_i[n] x_{ref}[n+d]. \tag{4.2}$$

A main drawback of this method is the high sensitivity of the outcome on the noise and the reverberations. This is of even more concern when speech is recorded from far-field, since with the distance to the speaker usually the acoustics has greater impact on the overall recorded signal and might affect the correctness of the outcome (see also [4]). Compared to equation (4.2) the GCC-PHAT[15] is often a preferred method to estimate this time delay. It computes the cross-correlation and performs an amplitude normalization in the frequency domain that results in a sharper peak in the final time-domain representation. The normalization in the computation is important especially when comparing peak values of the cross-correlation from different microphones. The GCC-PHAT can be written as:

$$\hat{R}_{i,ref}(d) = \mathcal{F}^{-1}\left(\frac{X_i(f)X^*_{ref}(f)}{|X_i(f)X^*_{ref}(f)|}\right), \tag{4.3}$$

where $X_i(f)$ and $X_{ref}(f)$ are Fourier transformed signals, $\mathcal{F}^{-1}$ denotes the inverse Fourier transform and $*$ is the hermitian. Following the results obtained from Equation (4.3) the TDOA is the delay d that maximizes the estimate:

$$TDOA_{i,ref} = \underset{d}{\operatorname{argmax}}\left(\hat{R}_{i,ref}(d)\right) \tag{4.4}$$

We can use the GCC-PHAT results directly to estimate the Kinect device closest to the speaker under the assumption that the peak in the correlation is more distinctive and has a larger value if the device is closer to the sound source. This is motivated by the fact that the ratio of direct sound to early reflections and reverberation decreases with the distance to the speaker.

4.3.3 Envelope variance

It has been shown in [24] that room reverberation distorts the speech signal and increases the WER of the speech recognition result. Hence, it could be possible to estimate the WER of a recorded utterance with the help of a proper feature that contains information on the amount of this distortion within a signal. As reported in [25] a potential approach is to make use of the fact that reverberation smooths the time sequence of the speech energy values and consequently has an impact on the dynamic range of the time envelope. From this we can infer that the speech signal with the largest dynamic range must be the one obtained from a microphone that was

[15] GCC-PHAT - generalized cross-correlation phase transform

either positioned very close to the speaker or within a good acoustic environment. Usually the reverberation depends on the frequency, which we can exploit to further increase the informative value of this feature. In the following the feature is derived as reported in [26].

In order to construct the envelope variance feature vector we first compute the Mel-scaled filterbank energies from the speech signal $x_m[n]$ of microphone m. This is the same proceeding like computing MFCC features, but without computing the cepstral coefficients. From this we get one vector for each frame l that contains k energy values, one for each Mel-sub-band. One element of this vector is denoted as $x_m[k,l]$. Next, the mean of each sub-band is subtracted in the log domain.

$$\hat{x}_m[k,l] = e^{\log(x_m[k,l]) - \mu_{\log(x_m[k]))}} \tag{4.5}$$

This should compensate for any inequalities among microphones, like different gain settings. In the following the spectral energies are compressed using a non-linear cubic root function and the variance is computed for each sub-band as

$$V_m[k] = var(\hat{x}_m[k,l]^{\frac{1}{3}}). \tag{4.6}$$

With this we get a representation of the dynamic energy range for each sub-band and channel. The envelope variance over the frequency range can then be computed by summing over the weighted normalized variances. The normalization is required in order to obtain a final channel variance value in the range between 0 and 1. The weighting of each channel $w_m[k]$ is used to compensate for higher distortions in individual sub-bands and should be learned from data. Channel C with the highest variance is the one that has the lowest reverberation distortions, i.e.

$$C^* = \underset{m}{\operatorname{argmax}} \sum_k w_m[k] \frac{V_m[k]}{\underset{m}{\max}(V_m[k])}. \tag{4.7}$$

The weighting should accumulate to 1 and can be either manually set or learned from data.

4.3.4 Mel-filterbank

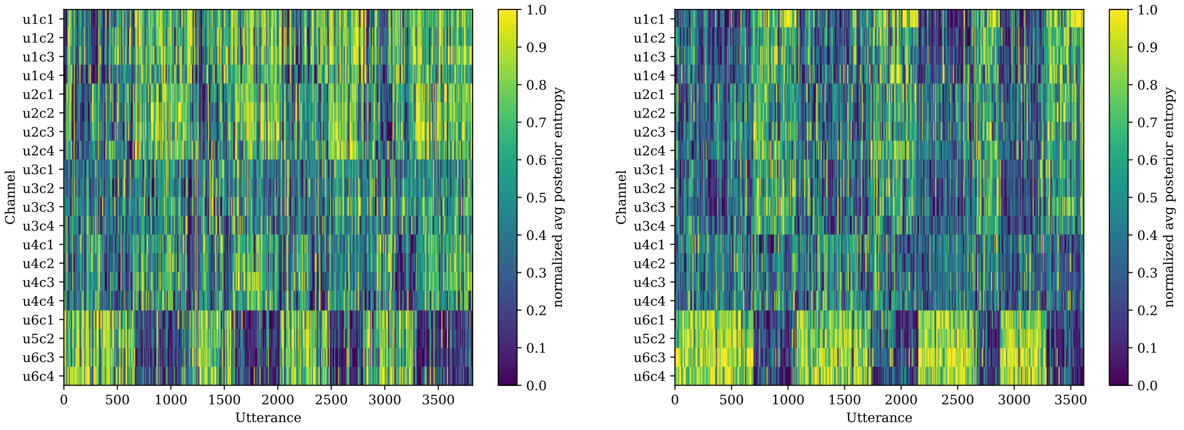
The Mel filterbank features are likewise to the MFCC feature extraction depicted in Figure 2.4, without the last two steps of computing the logarithm and the discrete cosine transformation (DCT). With usually 40 different bands for each channel and frame, this is by far the most data expensive feature among the other presented features.

4.3.5 Average Posterior Entropy

So far, the features were derived directly from the channel signals with the help of simple calculations and standard transformations. Another way to estimate the quality of a channel can be achieved from the recognizer's point of view by considering the output confidence of the acoustic model. In general, for each acoustic input feature vector $\boldsymbol{x}_t^m$ of channel m at time instant t, the AM results a stream of posterior probabilties $\boldsymbol{p}_t^m = [p(s_1, \boldsymbol{x}_t^m), p(s_2, \boldsymbol{x}_t^m), \ldots, p(s_N, \boldsymbol{x}_t^m)]^T$, one value for each HMM state s_i. From this distribution the uncertainty of the model outcome can be determined using the entropy measure as follows:

$$H_t^m = -\sum_s \boldsymbol{p}_t^m \cdot log_2\left(\boldsymbol{p}_t^m\right). \tag{4.8}$$

The entropy is a measure of information. Hence, it can be used to evaluate the uncertainty in the AM phone posterior probability predictions, where a high entropy value stands for a high uncertainty in the model outcome and a lower value reflects more confidence in the predicted phone posteriors. In order to get a measure on utterance-level, entropies are averaged among frames. Figure 4.1 depicts the averaged posterior entropy feature for session S02 and S09 of the development set. For visualization purpose the entropy values are normalized within a session to lie in the range between 0 and 1 (plots for the training sessions are located in Section B.1 of the Appendix).



(a) Session: S02

(b) Session: S02

Figure 4.1: Normalized average posterior entropy feature over utterances for both session of the development set.

4.3.6 Average posterior moments

In principle the DNN posterior probabilities are expected to be more fuzzy for uncertain predictions of the acoustic model. Apart from using the entropy as an estimate on the certainty of the predicted phone posteriors (as described in Section 4.3.5), we now make the rough assumption of a Gaussian posterior distribution and compute the first four moments, i.e. the mean (4.9), variance (4.10), skewness (4.11)and the kurtosis (4.12). This means, we collect all posterior probabilities $p(s_i, \boldsymbol{x}_t^m)$ for each state s_i and time frame t belonging to an utterance u of a certain channel m and compute the four statistical values with:

$$\mu_u^m = \frac{1}{N} \sum_{i \in S} \sum_{t \in T} p(s_i, \boldsymbol{x}_t^m), \tag{4.9}$$

$$\sigma_u^m = \sqrt{\frac{1}{N} \sum_{i \in S} \sum_{t \in T} (p(s_i, \boldsymbol{x}_t^m) - \mu_u^m)^2}, \tag{4.10}$$

$$\gamma_u^m = \frac{\frac{1}{N} \sum_{i \in S} \sum_{t \in T} (p(s_i, \boldsymbol{x}_t^m) - \mu_u^m)^3}{\sqrt{\frac{1}{N-1} \sum_{i \in S} \sum_{t \in T} (p(s_i, \boldsymbol{x}_t^m) - \mu_u^m)^3}}, \qquad \text{and} \tag{4.11}$$

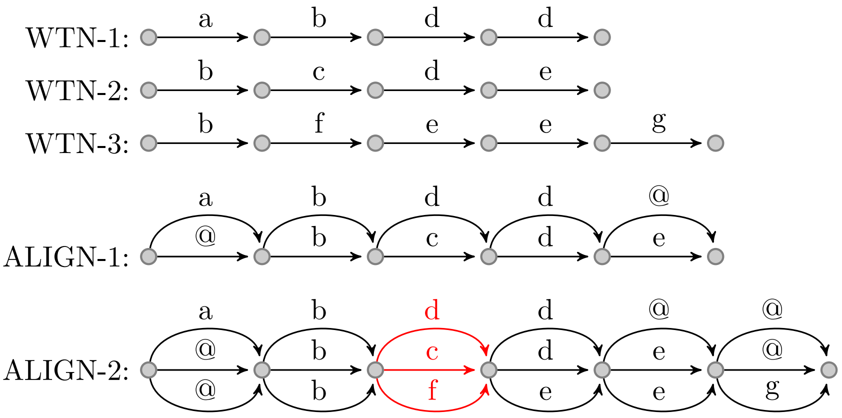
$$\kappa_u^m = \frac{\frac{1}{N} \sum_{i \in S} \sum_{t \in T} (p(s_i, \boldsymbol{x}_t^m) - \mu_u^m)^4}{\left(\frac{1}{N} \sum_{i \in S} \sum_{t \in T} (p(s_i, \boldsymbol{x}_t^m) - \mu_u^m)^2\right)^2} - 3, \tag{4.12}$$

where $N = |S| \cdot |T|$ is the number of posteriors, i.e. the number of phone states S times the number of frames T for utterance u.

4.4 ROVER

A very effective method to further reduce the transcription errors is the combination of hypotheses from different regonizers or channels using the recognizer uutput voting error reduction technique, in short ROVER [27]. The main idea behind ROVER is to make use of the differences in the transcription errors of each system (or decoded microphone channel) in order to generate a final best transcription from the combined word transition network (WTN) with respect to a certain voting scheme. A WTN is a linear graph that is made up of a sequence of nodes and multiple arcs connecting each pair of adjacent nodes. Each arc corresponds to a word or an empty symbol (i.e. "@"). The algorithm comprises two main stages, the creation of the composite WTN using dynamic programming and the Levensthein distance as well as the selection of the best word at each bin via majority voting.

An example of the procedure for three transcriptions (hypotheses) is depicted in Figure 4.2. Each of the three ASR outputs are wrapped into an initial linear WTN (WTN-1, WTN-2, WTN-3) that needs to be combined. For illustration purposes, the words of the hypotheses are represented with letters. First, WTN-1 is selected as the base network to which WTN-2 is aligned. From this the aligned WTN (ALIGN-1) is obtained, where the "@" symbol represent an insertion or a deletion. The aligned WTN (ALIGN-1) is then used as the base network to align WTN-3 to and we get the final composite WTN (ALIGN-2). Once the final composite WTN is ready, a voting scheme is used to select the letter (i.e. word) with the highest score for each branching point to generate the final hypothesis. The scores can be computed in various ways, for example, using the frequency of occurrence, average word confidence score, maximum confidence score, and also a combination of those. Here, the confidence measures are provided by the ASR system. It has been shown in [28] that the combined transcription results depend on the order how the hypotheses are feed into ROVER. In cases where different words receive the same vote (this is pointed out in red color in Figure 4.2), ROVER gives priority to the one in the top position of the column in the branch (i.e. "d"). This might not be the best solution, and can be mitigated by ranking the hypotheses according to their estimated quality before the actual combination is performed.


WTN-1: a b d d
WTN-2: b c d e
WTN-3: b f e e g
ALIGN-1: a @ b b d c d d @ e
ALIGN-2: a @ @ b b b d c f d d e @ e e @ @ g

Final hypothesis: @ b d d e @
Reference: a b c d e

Figure 4.2: Illustration of the ROVER procedure with three different initial transcriptions from [28].

5 Experimental Results

In the previous chapters, the CHiME-5 dataset, the baseline ASR system as well as the necessary blocks for the channel selection and the hypotheses fusion have been introduced. This chapter describes the conducted experiments and lists the obtained results (see also [29]).

In Section 5.1 we report our own results on the baseline system and in Section 5.2 the ones obtained from decoding individual channels. In Section 5.3 the theoretical performance gain of a proper channel selection technique is explored using the oracle. Section 5.5 reports the performance of the DNN channel classifier and in Section 5.6 we explore the performance gain when combining different hypotheses using ROVER. Finally, in Section 5.7 we draw a conclusion on the observations.

5.1 Local baseline results

All the experimental results reported within this chapter make use of the conventional baseline system, introduced in Section 2.3 of Chapter 2.

The acoustic model is trained on a subset of the available data, that comprises around 70000 utterances from the binaural left channel recordings and 100000 utterances randomly picked from the Kinect recordings. Due to this randomness in the selection of the training data, it is assumed that a fresh model training delivers a different model and therefore different decoding results. Furthermore, it is interesting to check if the baseline performance improves if the size of the training data is increased. Therefore we conducted several training runs with different amount of training data, one with slightly more utterances from the Kinect recordings and another run with all available data from the Kinect arrays. The WER results when model training was conducted at the SPSC machines at Graz University of Technology are provided in Table 5.1 and Table 5.2 lists the results when training was conducted on the FBK[16] machines. First of

System	Training set	Decoding channel	Dev
GMM-HMM	70k worn (L) +100k Kinects	ref. Kinects + BfIt worn	91.25 74.57
DNN-HMM	70k worn (L) +100k Kinects	ref. Kinects+ BfIt worn	80.86 49.86
GMM-HMM	70k worn (L) +200k Kinects	ref. Kinects + BF worn	91.35 74.31
DNN-HMM	70k worn (L) +200k Kinects	ref. Kinects+ BfIt worn	80.65 49.11

Table 5.1: WER [%] results of the conventional baseline system trained on the SPSC cluster at Graz University of Technology. Decoding was conducted for both the beamformed reference Kinect arrays and the worn binaural microphones.

[16] FBK - Fondazione Bruno Kessler

all we notice the big advantage of retraining a DNN upon the GMM-HMM model, from which we gain an absolute WER reduction of about 10%. As expected, different training runs lead to different performance outcomes. Moreover when using all the available Kinect data (around 1600000 utterances) only a slight improvement can be observed. Interesting are the results on the binaural close-talk microphones (worn). They show that a major performance impact is due to the microphone distance. The remaining experiments were conducted at FBK and we stick to the baseline performances obtained when trained on those machines.

System	Training set	Decoding channel	Dev
GMM-HMM	70k worn (L) +100k Kinects	ref. Kinects + BfIt worn	91.02 71.87
DNN-HMM	70k worn (L) +100k Kinects	ref. Kinects + BfIt worn	**82.52*** 48.92
DNN-HMM	all Kinect channels	ref. Kinects + BfIt worn	81.95 -
DNN-HMM	all worn	ref. Kinects + BfIt worn	91.39 48.67

Table 5.2: WER [%] performance of the ASR baseline system trained at FBK. (*) FBK baseline system used for benchmarking.

5.2 Channel decoding

We decoded the available channels individually by considering only utterances from one specific channel. This provides insight into the performance mismatch between channels and furthermore reveals the suitability of the proposed reference Kinect selection method as well as the contribution from the enhancement stage (see Section 2.2).

First we investigated the performance gain obtained from the speech enhancement stage. For this we decoded each microphone part of the reference Kinect separately to get four individual recognition results as listed in Table 5.3. The best performance is achieved for the second microphone channel with a WER of 82.36%, which is actually better then the baseline result of 82.52%. From this we can conclude that BeamformIt does not perform well within this dataset. This is explained particularly by the fact that the enhancement highly depends on the TDOA estimation, which is used for the direction of the steering vector. The distance of the speakers to the microphones, the reflections from the surrounding walls and objects, upon this the spontaneous and overlapping speech make this estimation not reliable, which results in a poor enhancement gain.

Next, we compared the performance among Kinect arrays. Identical to the baseline system, BeamformIt is applied to fuse the four single channels of each array into an "enhanced" channel which is then feed into the decoding stage. The results on five out of six devices are reported in Table 5.4. The lack of Kinect U05 is due to a recording failure in session S09 that has caused frame droppings and therefore several utterances of the recording from this devices are missing (see also Table 3.1). We will see in Section 5.5 that this also restricts the number of channels for the proposed channel selection method. Overall the device U03 delivers the best transcriptions on the development set. One possible explanation could be related to the position of the Kinects within the two apartments. In view of the floor plans of Figure 3.2, Kinect U03 is the one located in the living room, at a preferred place able to cover the full duration of the party session the best as the array is closest to the speakers over the course of the dinner party, on average.

Decoding channel	Dev
ref. Kinects CH1	82.54
ref. Kinects CH2	**82.36**
ref. Kinects CH3	82.53
ref. Kinects CH4	82.72

Table 5.3: WER [%] from decoding each microphone channel from the reference Kinect device, i.e. no BeamformIt speech enhancement is applied.

Decoding channel	Dev
Kinect U01 + BfIt	85.32
Kinect U02 + BfIt	84.91
Kinect U03 + BfIt	**82.61**
Kinect U04 + BfIt	83.27
Kinect U06 + BfIt	84.94

Table 5.4: Decoding results for all beamformed Kinect channels.

Decoding channel	Dev
U01.CH1	85.68
U01.CH2	85.35
U01.CH3	85.50
U01.CH4	85.64
U02.CH1	85.56
U02.CH2	85.21
U02.CH3	85.41
U02.CH4	85.37
U03.CH1	83.87
U03.CH2	**83.39**
U03.CH3	83.75
U03.CH4	83.76
U04.CH1	83.61
U04.CH3	84.17
U04.CH2	83.67
U04.CH4	83.74
U06.CH1	85.50
U06.CH2	85.22
U06.CH3	85.36
U06.CH4	85.35

Table 5.5: Comparison of WER [%] between the single microphone channels.

Following the investigations on device level, we decoded each single microphone channel and obtained the results listed in Table 5.5. The total difference in absolute WER is around 2.2% among channels. It is assumed that a refined channel selection on utterance level will enhance the performance even more. This is going to be investigated in the next section.

5.3 Oracle results

Several oracle experiments were conducted with different sets of hypotheses in order to investigate the possible performance gain when selecting among a set of decoded channels on utterance-level. Table 5.6 shows the oracle scores for the development set and its individual sessions. These results may serve as an indicator to which extend both the best selection and the set of channels impacts the final performance. Most of the experiment are conducted without using array five, since it is partially missing in the development set. However, information coming from this array is introduced when U_ref is part of the channel set. Using all the available hypotheses from 29 channels the oracle scores to a WER of 63.6% on the development set, which is a total of 18.9% in absolute word error rate reduction compared to the baseline system. Moreover importantly, remarkable results are also obtained when solely selecting among the 20 single array channels, without using any enhanced signal or information of the reference array. However, as illustrated in Figure 5.1, a good selection is crucial since the performance drastically decreases with increasing oracle rank. Further investigations on the oracle output show that the number of channels being oracle (i.e deliver the lowest WER) differs among utterances. This is visualized in Figure 5.2 separately for each session of the development set, where an explicit peak shows that for many utterances, only one channel delivers the best transcription with the lowest WER. The peak at the very left hand side in the histogram comes from the utterances

Channels	**Dev**		
	S02	S09	Overall
Baseline: U_ref + BfIt (1)	83.4	81.1	82.5
U_ref (4)	76.1	72.8	74.8
U + BfIt (5)	70.8	68.2	69.3
U (20)	66.3	63.3	65.1
U + BfIt, U (25)	65.5	62.3	64.3
U_ref, U + BfIt, U (29)	64.6	62.2	63.6

Table 5.6: WER [%] results of the oracle for different sets of decoded channels. U indicates a set of 20 single array channels while U_ref are the four channels from the reference array provided by the baseline system. The parenthesized number states the number of available channels in this specific setting. BfIt stands for the BeamformIt beamformer enhancement.

where no error rate can be computed, i.e. for an empty hypothesis. The peak at 20 states that actually for a high number of utterances all channels perform equally, which is supposed to be the case when the system outputs nonsense transcriptions. An additional observation in

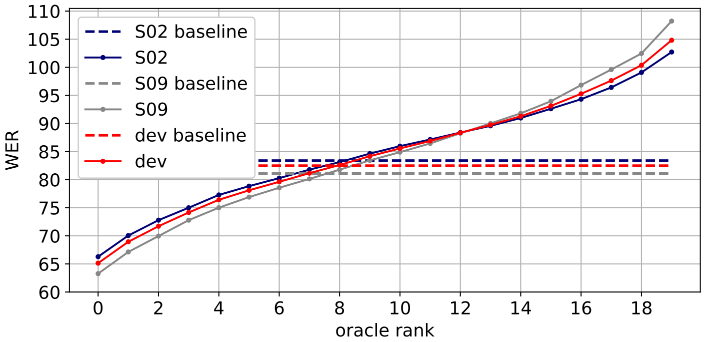
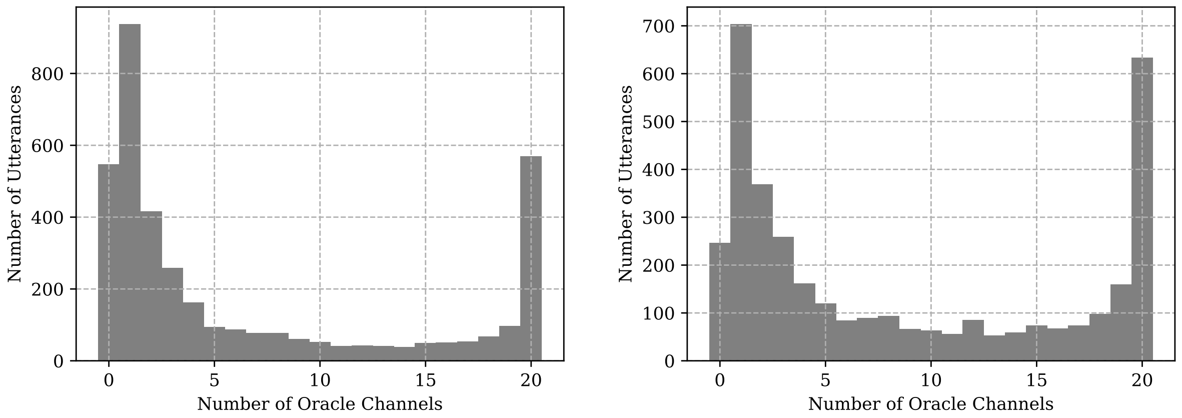


Figure 5.1: WER [%] results for the development set on the per utterance oracle informed channel ranks, considering the 20 array channels.

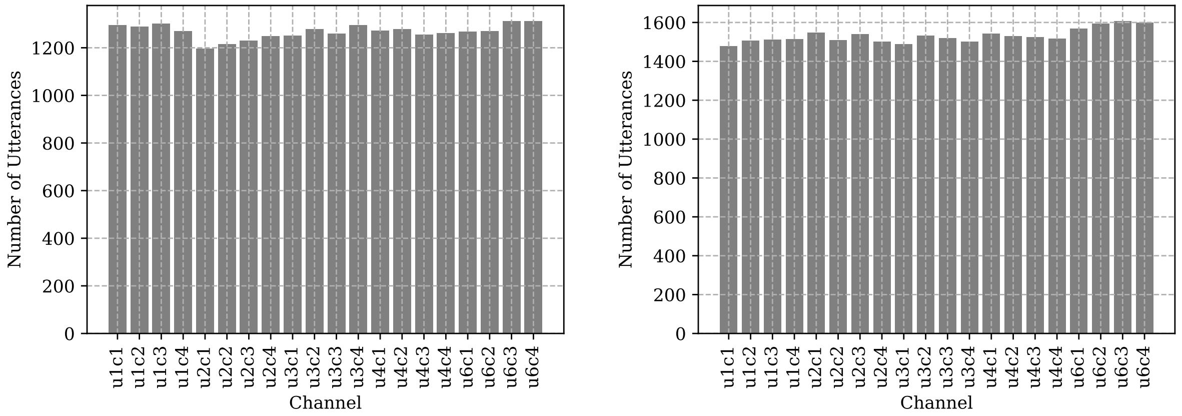
Figure 5.3 demonstrates that the chance to be oracle is equally distributed among channels. The findings that we get from the plot in Figure 5.4 is that no strict pattern over the utterances of a session is observed. For example one could expect that the appearance of oracle channels might be associated to the speakers location within the apartment. However, there is no strict evidence following this analysis.



(a) Session: S02

(b) Session: S09

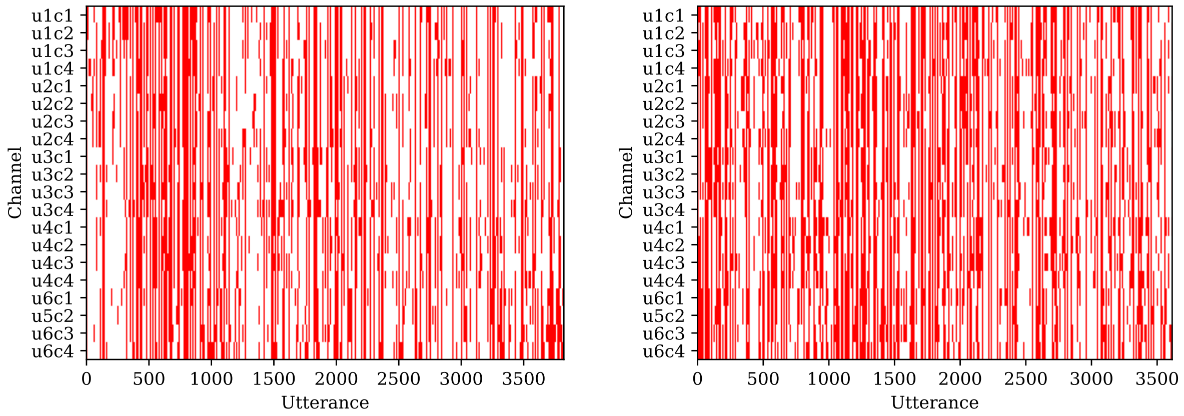
Figure 5.2: Histogram over the number of oracle channels per utterance. Zero oracle channels apply to the cases where the WER is infinity. This happens for empy transcriptions.



(a) Session: S02

(b) Session: S09

Figure 5.3: Oracle channel distribution among the available 20 *single channels of a session.*



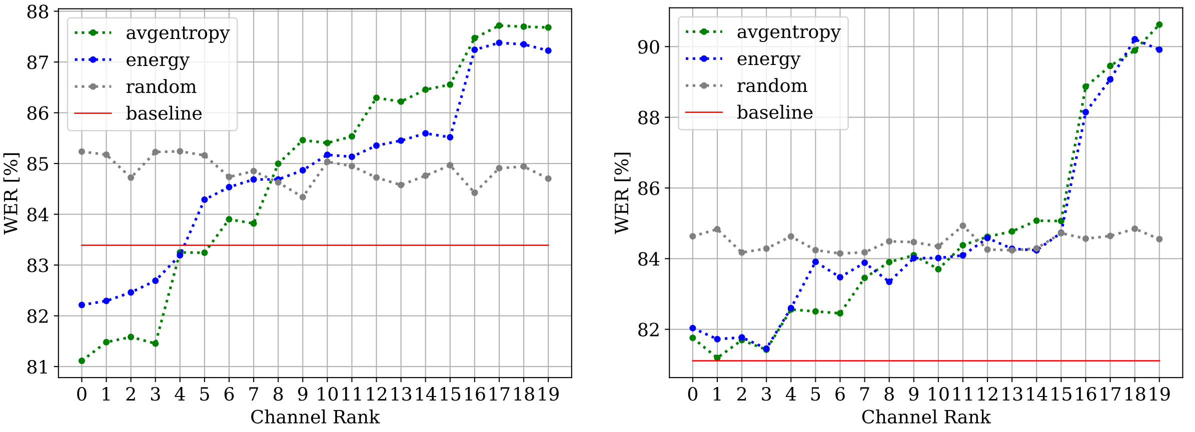
(a) Session: S02

(b) Session: S09

Figure 5.4: Oracle channel (in red) for each channel over the utterances of a session. No particular pattern can be observed. Figures for the training sessions are available in B.2 of the Appendix.

5.4 Feature direct classification

Some of the features introduced in Section 4.3 can be directly used for quality ranking of the 20 single channels at utterance level, without the actual need to train a classifier model. We ranked the channels with respect to the signal energy feature described in Section 4.3.1 and the average posterior entropy features from Section 4.3.5. For the energy ranking it is assumed that microphones closer to the speaker provide higher SNRs and signal energies, compared to the ones located in the far-field. The average posterior entropy states the confidence of the model on the phone state predictions and here it is assumed that the best hypothesis is the one with the smallest entropy value. The obtained ranks, as depicted in Figure 5.5, show a correlation with the channels WER, that proves our reflections in the sense that the features contain some significant information on the channels quality.



(a) Session: S02

(b) Session: S09

Figure 5.5: WER [%] obtained from channel ranking according to either the energy or the average posterior entropy features. Results from a random channel ranking are shown in grey, and the baseline result in red color. A clear trend between the WER and the features rank is observed. Similar results for training sessions can be found in Section B.3 of the Appendix.

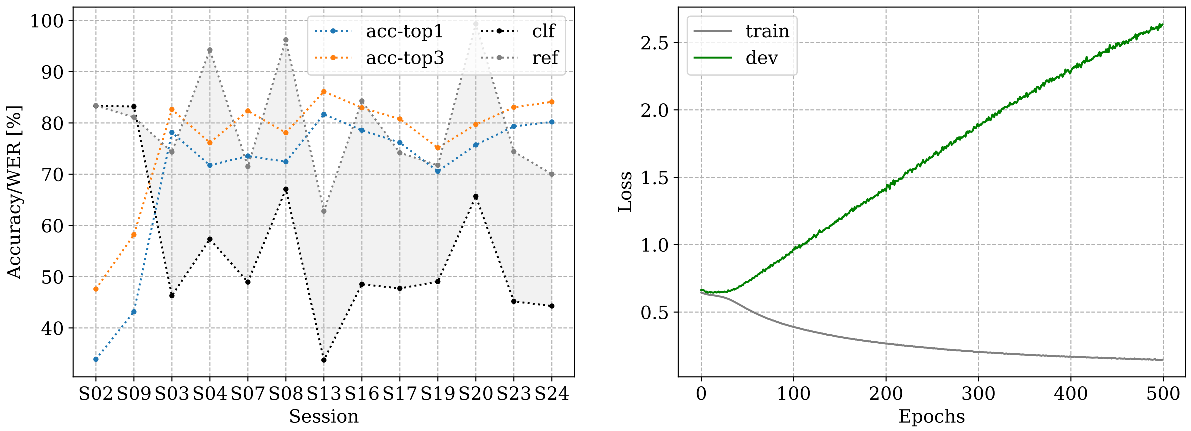
Ranking can also happen on Kinect level, with respect to the energy computed on the beamformed array channel or by using the cross-correlation peak from the GCC-PHAT as confidence measure, usually used to estimate the TDOA. For the spatial feature the average peak of the GCC-PHAT between the two most outstanding microphones (CH1 and CH4) is computed over the full length of the utterance, with an estimation time interval of 128 ms. The numerical results on the development set when always using the best channel according to rank 0 are listed in Table 5.7. Compared to the baseline results of 82.52% in Table 5.2, a slight increase in accuracy is obtained.

Channels	**Feature**	**Dev**		
		S02	S09	Overall
U+BfIt (5)	Energy	81.2	81.6	81.3
	GCC-PHAT	81.1	81.7	81.4
U (20)	Energy	82.2	82.0	82.1
	Avg. Entropy	81.1	81.8	81.4

Table 5.7: WER [%] results on the development set for feature based channel selection using either energy, average posterior entropy or GCC-PHAT features. The number in round brackets state the number of available channels. U indicates the set of available single channels and BfIt is an abbreviation for the BeamformIt beamformer.

5.5 DNN classifier

We address the problem of channel selection with a DNN-based classifier, trained on features either directly extracted from the recordings of the training sessions or the ASR decoder stage (see Section 4.3) to predict the oracle channels (see Section 4.2). Since multiple oracle channels may occur for a certain utterance, the classification is categorized as a multi-class multi-label problem. We employ a sigmoid activation in the last layer of the neural network, to allow ranking the channels according to the predicted scores. Following this ranking, we can simple decode the channel with lowest rank or combine the best N hypothesis using ROVER (see Section 5.6) in order to obtain a good recognition accuracy. As training objective the cross-entropy error is used and the final model is evaluated on both sessions of the development set.



(a) Recognition accuracy on unseen data and on the sessions used to train the classifier. "acc-top1" states the accuracy that the best scoring channel is an oracle channel. “acc-top3” is the accuracy that an oracle channel is within the best three scoring channels. To demonstrate the performance gain obtained from the classifier informed channel selection (clf), the points in grey show the reference baseline results (ref). These results are obtained by decoding the beamformed reference Kinect for both development sessions or Kinect U03 for the training sessions.

(b) Cross-entropy loss over the number of epochs on both training and development data.

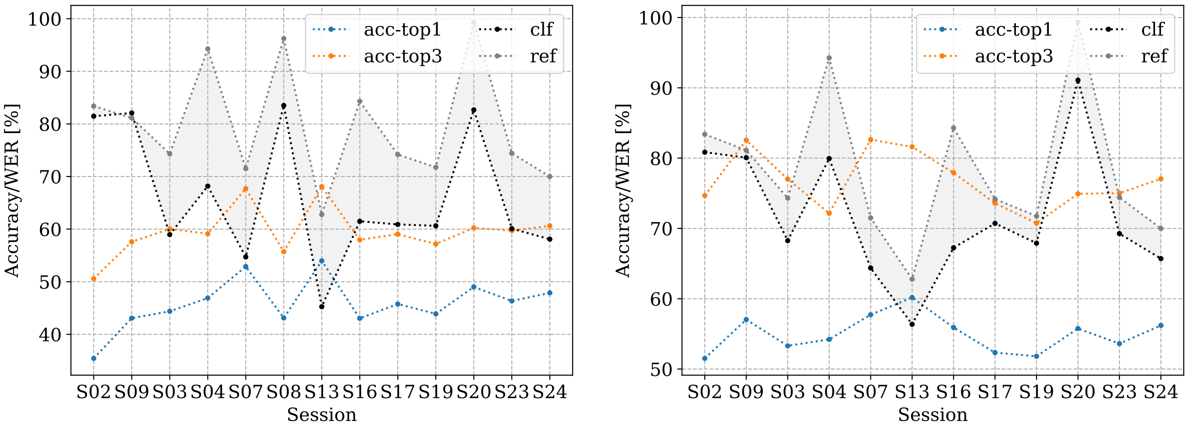
Figure 5.6: DNNs trained on average posterior entropy features from the development set. No regularization or dropout is applied during training. The model shows poor generalization.

For the envelope variance features (EV) we compute 12 sub-bands per channel. The sub-band variances are directly feed into the network without any prior sub-band weighting (as requested by Equation (4.7)), since it is assumed that the network can learn a reasonable weighting by its own. The Mel-filterbank features (Fbank) make use of 40 filter banks for each channel and frame, extracted from the signals with an applied window of 20 ms and a hop-size of 10 ms. To allow the network to learn the temporal context hidden in contiguous feature vectors, we introduce recurrent units in the classifier architecture. All other features, like the average posterior entropy (avg. entropy) and the energy have a dimensionality of one, i.e. one value for each channel and utterance. Moreover we stacked the energy, average posterior entropy and the average posterior moments to construct higher dimensional features while combining information from different analysis.

As we have seen in Figure 5.2, for many utterances either no oracle channel exists or all channels deliver the same WER. Since it is not expected that the network can extract meaningful information from this data, the utterances used for training are restricted to the ones where this is not the case. This reduces the total amount of training data by approximately 40%.

During experimental procedures we investigated several network architectures, incorporating different number of layers, units in each layer, activation functions, and furthermore played with regularizations and training algorithms. At the end we decided on two final architectures, a LSTM (1 recurrent layer followed by 2 dense layers) for frame based features (i.e. Mel-filterbank features) and a feed-forward network (MLP) with two hidden layers of 1024 units each when utterance-based features are used instead. ADAM is used as the learning algorithm with a L2-regularization and dropout of 20% in each layer.

Our findings from training without any regularization are that the network learns the training data by heart, i.e. the model performs well on the sessions from the training set, but bad on the development set. This observation has been obtained when the model was trained on average posterior entropy features from 11 training sessions over 500 epochs. The loss function, as well as the evaluation on the various sessions is depicted in Figure 5.6. Figure 5.6(b) shows the bad generalisation behavior of the model, as the cross-entropy loss on the development set increases strongly while simultaneously the training loss converges to a minimum.



(a) Classifier with 20 labels, one for each single channel.

(b) Classification with 5 outputs, one for each Kinect device.

Figure 5.7: WER [%] results obtained from the trained model when the best scoring channel is selected for each utterance. The network was trained using the average posterior entropy features with L2-regularization and dropout of 20% *over* 50 *epochs.*

Introducing a L2-regularization term and dropout mitigates this discrepancy, although it does not help the network to learn useful features. We set the L2-regularization factor to 0.00001 and applied 20% dropout in each layer. Training the model on 50 epochs has provided best overall results, independent on the type of features. Table 5.8 presents the WER of the best selected channel for the full development set for all models, trained on the different features. The success of the channel classifier is weak, as the performances are similar to the baseline results. The average posterior entropy features seem to incorporate more valid information as they perform best compared to the others. Figure 5.7(a) shows the results of the selection approach for this feature type in more detail for each session.

Additionally we also tried to rank among the beamformed Kinect channels (see the results in Figure 5.7(b)). In this experiment the classifier provides five different scores, one for each Kinect array without considering Kinect U05. Unfortunately, the obtained results are not better compared to the ones from the 20 single channel ranking.

Channels	**Feature**	**Dev**		
		S02	S09	Overall
U (20)	Energy	82.2	82.7	82.8
	EV	83.7	82.6	82.7
	Fbank	83.8	83.5	83.7
	Avg. Entropy	81.7	82.8	82.1
	Avg. Moments	82.8	81.3	82.3
	Stacked	82.3	82.3	82.3
U+BfIt (5)	Avg. Entropy	80.8	80.1	80.5
	Avg. Moments	81.1	80.7	81.0

Table 5.8: WER [%] results on the development set for classifier based channel selection with different features. The number in round brackets state the number of available channels.

5.6 ROVER

We applied hypotheses fusion using ROVER on a subset of best scored channels obtained from the DNN channel classifier experiments described in Section 5.5. To determine the final transcriptions we use average confidence scores for majority voting on the composite WTN. As listed in Table 5.9 the best result (78.10%) is achieved when combining the best 10 channels given by the network trained on the average posterior entropy features.

#Channels	3	5	10	20
Energy	82.00	81.08	79.96	79.65
EV	80.02	79.21	79.08	79.54
Avg. Entropy	79.36	78.25	**78.10**	79.40
Avg. Moments	79.73	78.53	78.17	79.51
Stacked	79.99	78.89	78.63	79.49
Fbank	81.71	80.41	79.56	79.52
Oracle	67.67	68.81	72.46	78.82
Random	81.92	80.90	79.88	79.67

Table 5.9: Numerical results of the conducted ROVER combinations upon classifier rankings with different features. The fusion is executed for the best 3, 5, 10, and for all the available 20 channels. The same results are also visualized in Figure 5.8.

With this an absolute WER reduction of 4.4% with respect to the baseline is achieved. The table also lists results on the combine N-best hypotheses obtained from both the oracle and a random channel ranking. This is mainly to set an upper and a lower bound to check whether the approach was successful or not.

The ROVER outcomes are also visualized in Figure 5.8. Our findings from these experiments are that the DNN classifier is not capable in ranking the channels accordingly. For example we would expect much more gain in performance when only combining the best three channels, if at least one of them is oracle.

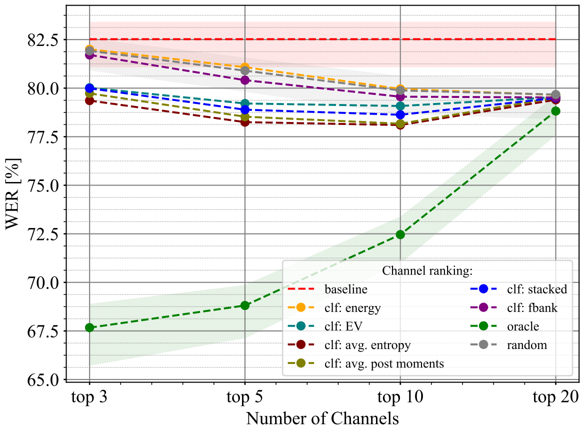


Figure 5.8: ROVER results from the combination of 3, 5, 10 or 20 best channels given by the classifier (clf) trained on different features. Transparency colored regions states the performance deviation among the two development sessions.

5.7 Discussion

The experimental results show clearly that the approach of channel selection does not deliver notable improvements in WER. One of the main concerns is the significance of the extracted features with respect to provide reliable information for the distinction of high- and low-performing channels. However, the energy and average posterior entropy features seem to incorporate data that correlates with the oracle channels, since we observed an increasing trend in WER with the number of channel ranks (see Figure 5.5). It is not evident, at least from this point of view, where the required information is hidden. Due to the difficulty of the dataset, standard features do simply not work and probably more reflections are required in order to extract significant features.

Furthermore, the results from the DNN classifier experiments demonstrate that a deep learning approach is capable in extracting the required information from the features. The model however generalises very badly in this scenario and by introducing an appropriate regularization it has been shown that the accuracy improvement is close to zero again. Perhaps besides the features itself, also the limitation on the available data may restrict the model training.

As the classifier did not work as expected, the ranking of the channels is not reliable. It is therefore no surprise that the performance gain from the ROVER combination is relatively small. For correctly ranked channels we would expect the lowest WER when combining the best 3 or 5 channels. In the ROVER results however, the number of combined channels is decisive for the final accuracy, which confirms the findings of an unstructured and poor classifier channel ranking.

Besides these rather weak results, the oracle analysis has shown a promising gap of improvement from channel selection and at the same time verified the poor selection of the reference Kinect provided in the baseline.

6 Conclusion

In this work, we addressed the problem of channel selection for distant automatic speech recognition on the CHiME-5 dataset. We started the discussion with an explanation of the individual blocks of the CHiME-5 conventional baseline system for array synchronisation, speech enhancement and ASR. Subsequently the dataset with a description and demonstration of its main properties is introduced. In Chapter 4 we elaborated the main idea of channel selection and fusion, as well as provided a description of the various features. The main experimental work of the thesis has been presented in Chapter 5 where our observations are discussed.

The difficulty of the channel selection approach has been demonstrated, at least concerning the CHiME-5 dataset. At this point of time and work it is not evident, how established features can be extracted from the signal or stages of the ASR system that incorporate information matching the oracle channels. Perhaps further in-depth analysis of the data is necessary, using advanced methods and approaches.

It was shown, that at least by using signal energy, spatial information, estimates of the reverberation content, and the DNN posterior entropy an accurate channel selection is not possible. The main obstacle is by far the difficulty of the dataset itself. The far-field recordings are dominated by noise, spontaneous and simultaneous speech, supplementary distinct by different acoustical environments in each session and room. This limits the application of speech enhancement techniques; e.g. we showed that multi-channel speech enhancement using BeamformIt only delivers minimal or no improvements in the final ASR accuracy.

Furthermore, I would like to address the limitations in time and computing power available in which the experiments were conducted, which may (unintentionally) influence the type of features investigated. Strictly speaking, the allocation of memory was limited to approximately 40 GB. Hence experiments with frame-based features, like the Mel-filterbank features, always required attention on the memory consumption.

In spite of these findings, the oracle experiments show a high possible theoretical performance gain if someone is capable to select the best channel on utterance-level, which provides motivation and encouragement to keep researching various and other approaches. Therefore this work raises several interesting questions for further work:

- First of all it would be very useful to investigate this approach on an easier data corpus to trace back the problem to the channel selection itself rather conflicting with the difficulty of the provided dataset. This would allow to check the method in general, as well as being able to compare features more accurately.

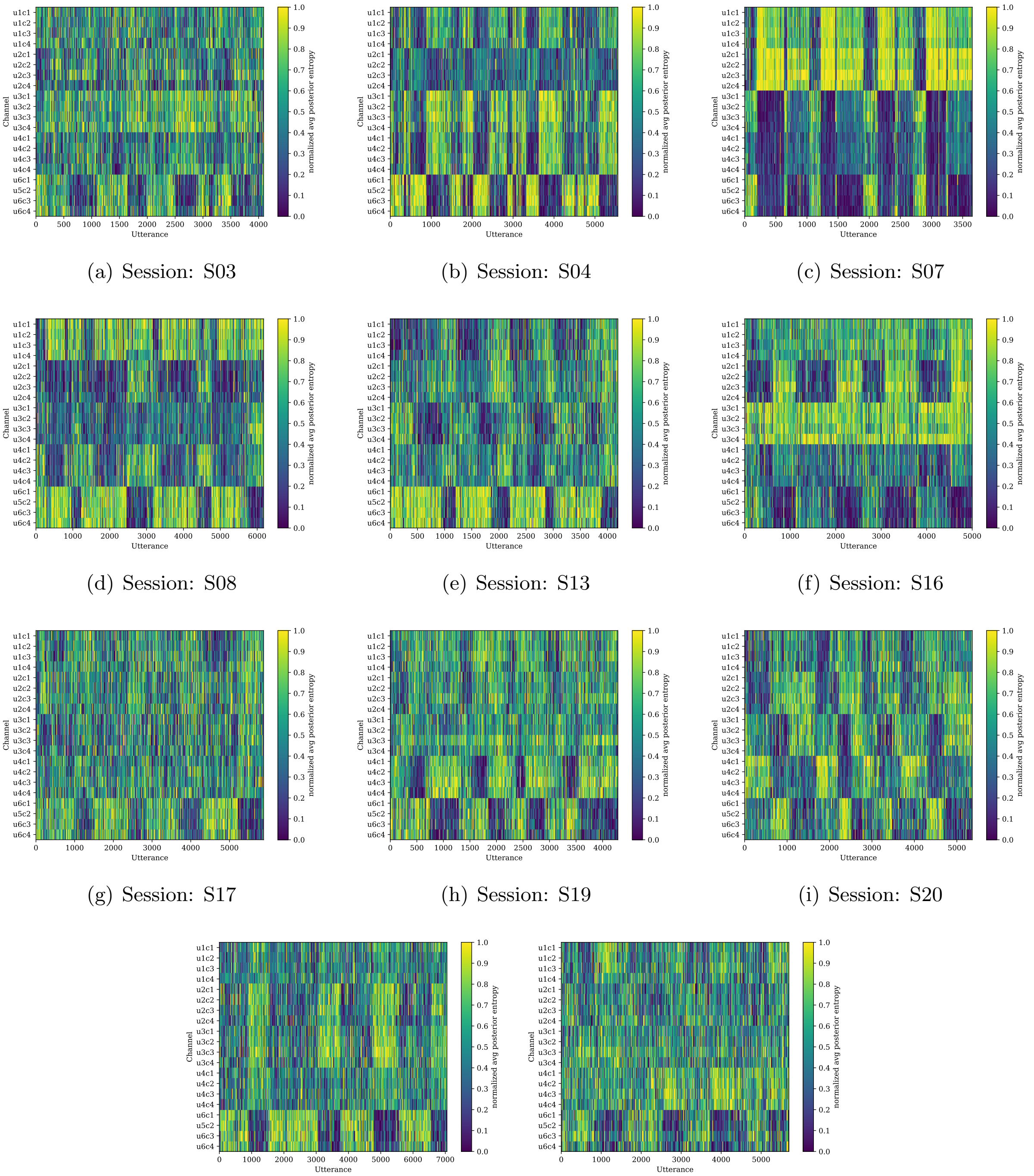
- Application of other features. This may require further in-depth analysis of the recorded data as well as the inner workings of the ASR system. The goal here is to provide information that correlates with the accuracy of the decoded channel. For expensive features an autoencoder could be used to reduce the dimensionality while preserving the information.

A
List of Abbreviations

AM	Acoustic Model
ASR	Automatic Speech Recognition
CHiME	Computational Hearing in Multisource Environments
CMLLR	Constrained Maximum Likelihood Linear Regression
CTC	Connectionist Temporal Classification
DNN	Deep Neural Network
EM	Expectation Maximization
FBK	Fondazione Bruno Kessler
FFT	Fast Fourier Transformation
fMLLR	Feature space Maximum Likelihood Linear Regression
G2P	Grapheme to Phoneme
GCC-PHAT	Generalized Cross-Correlation Phase Transform
GMM	Gaussian Mixture Model
HMM	Hidden Markov Model
LDA	Linear Discriminant Analysis
LM	Language Model
LSTM	Long Short-Term Memory
MAP	Maximum A Posteriori
MFCC	Mel Frequency Cepstral Coefficients
MLLR	Maximum Likelihood Linear Regression
MLP	Multi-Layer Perceptron
ReLU	Rectified Linear Unit
ROVER	Recognizer Output Voting Error Reduction
SAT	Speaker Adaptive Transformation
SRILM	SRI Language Modelling toolkit
SPSC	Signal Processing and Speech Communication
TDOA	Time Delay of Arrival
SNR	Signal to Noise Ratio
WER	Word Error Rate
WTN	Word Transition Network

B
Graphical results (training sessions)

B.1 Average posterior entropy features



(a) Session: S03

(b) Session: S04

(c) Session: S07

(d) Session: S08

(e) Session: S13

(f) Session: S16

(g) Session: S17

(h) Session: S19

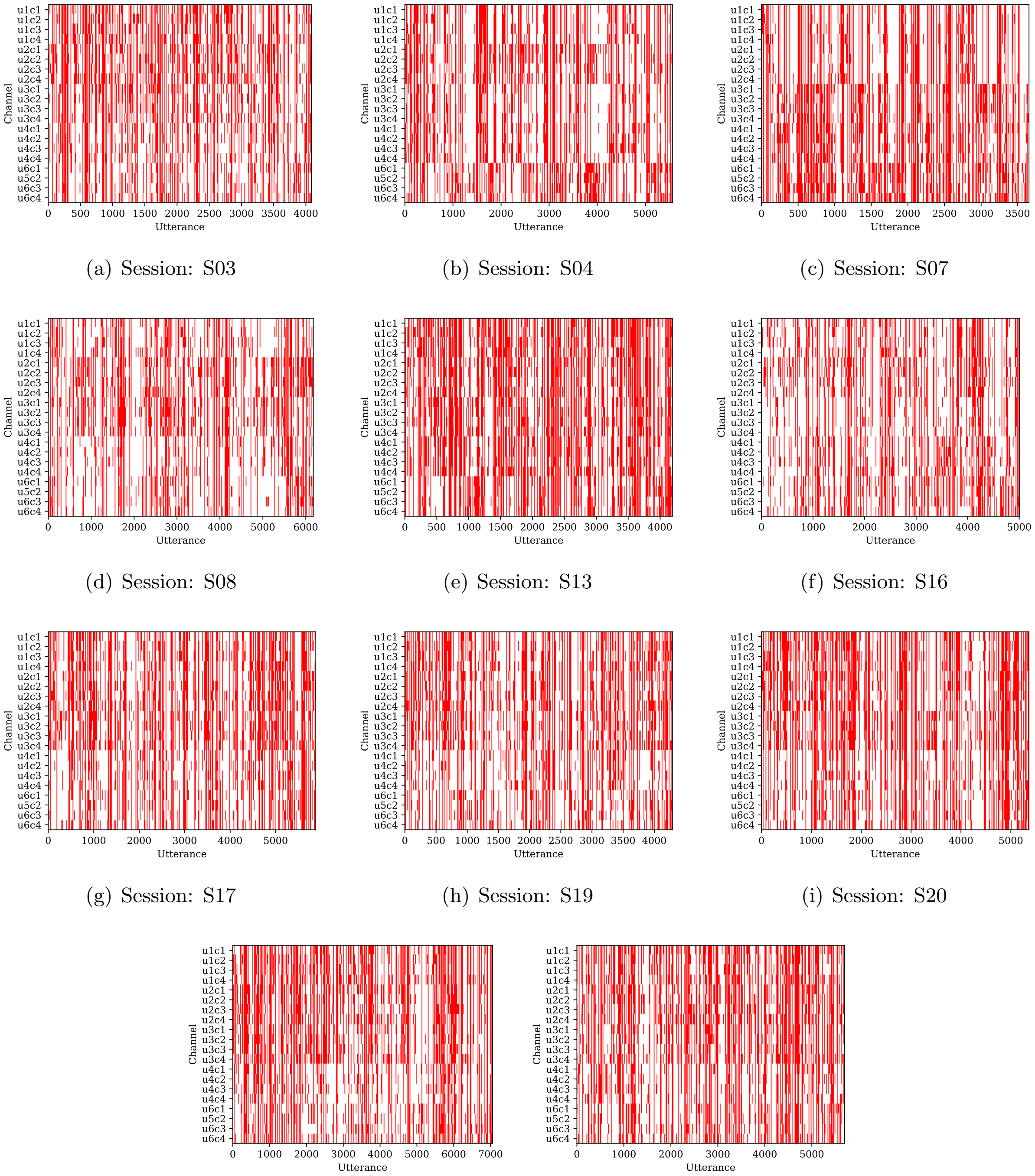
(i) Session: S20

(j) Session: S23

(k) Session: S24

Figure B.1: Normalized average posterior entropy for part of the training sessions.

B.2 Oracle channels



(a) Session: S03

(b) Session: S04

(c) Session: S07

(d) Session: S08

(e) Session: S13

(f) Session: S16

(g) Session: S17

(h) Session: S19

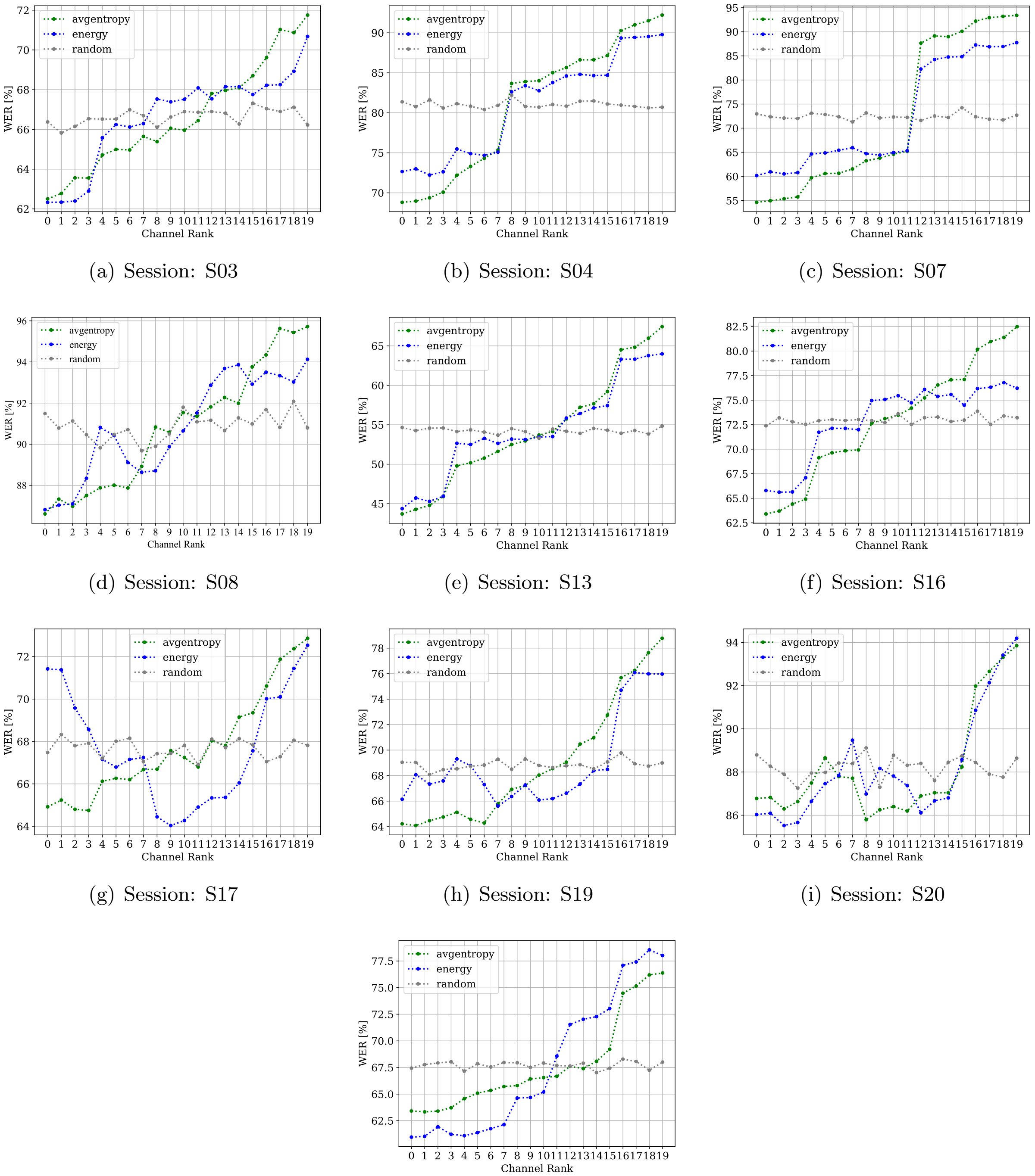
(i) Session: S20

(j) Session: S23

(k) Session: S24

Figure B.2: Oracle channels for sessions of the training set.

B.3 WER over feature ranks



(a) Session: S03

(b) Session: S04

(c) Session: S07

(d) Session: S08

(e) Session: S13

(f) Session: S16

(g) Session: S17

(h) Session: S19

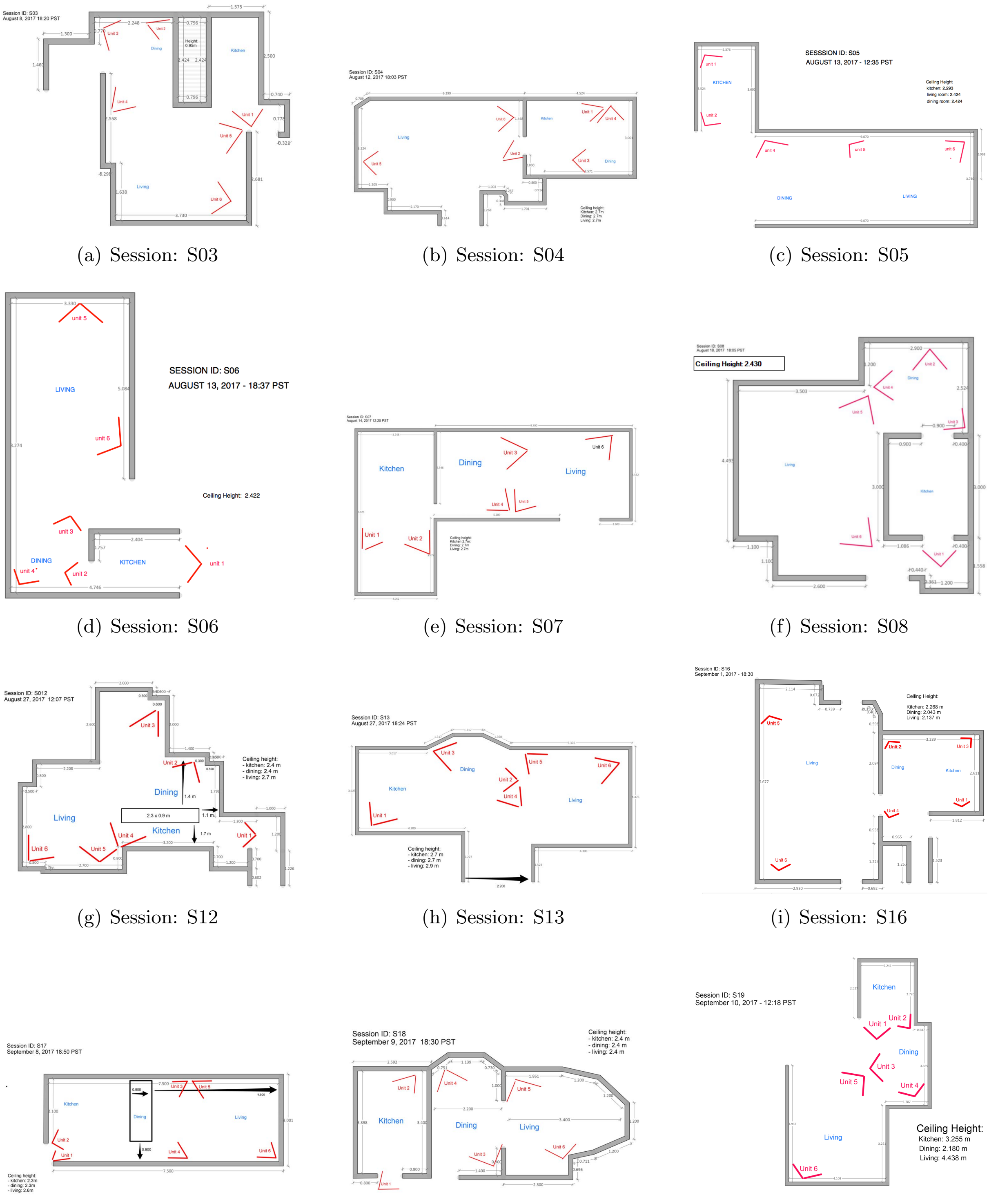
(i) Session: S20

(j) Session: S23

Figure B.3: WER over feature ranks for the sessions of the training set.

C
Floorplans (training sessions)

C.1 Floorplans



(a) Session: S03

(b) Session: S04

(c) Session: S05

(d) Session: S06

(e) Session: S07

(f) Session: S08

(g) Session: S12

(h) Session: S13

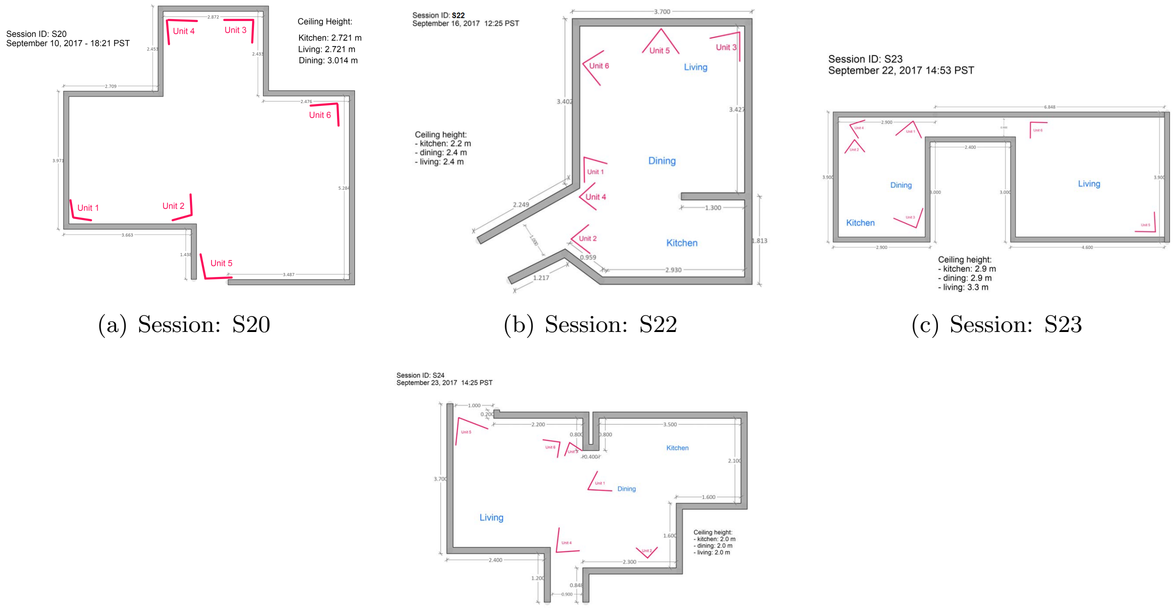
(i) Session: S16

(j) Session: S17

(k) Session: S18

(l) Session: S19

Figure C.1: Oracle channels for sessions of the training set.



(a) Session: S20

(b) Session: S22

(c) Session: S23

(d) Session: S24

Figure C.2: Oracle channels for sessions of the training set.

Bibliography

[1] J. Barker, S. Watanabe, E. Vincent, and J. Trmal, “The fifth ‘chime’ speech separation and recognition challenge: Dataset, task and baselines,” in *Proceedings of the 19th Annual Conference of the International Speech Communication Association (INTERSPEECH 2018)*, Hyderabad, India, Sep. 2018.

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

[3] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi, “Hybrid ctc/attention architecture for end-to-end speech recognition,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.

[4] X. Anguera, C. Wooters, and J. Hernando, “Acoustic beamforming for speaker diarization of meetings,” *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 7, pp. 2011–2022, 2007.

[5] D. Jurafsky and J. H. Martin, *Speech and language processing.* Pearson London, 2014, vol. 3.

[6] M. J. Gales *et al.*, “Maximum likelihood linear transformations for hmm-based speech recognition,” *Computer speech & language*, vol. 12, no. 2, pp. 75–98, 1998.

[7] S. Young, “A review of large-vocabulary continuous-speech,” *IEEE signal processing magazine*, vol. 13, no. 5, p. 45, 1996.

[8] L. R. Rabiner and R. W. Schafer, *Digital processing of speech signals.* Prentice-hall Englewood Cliffs, NJ, 1978, vol. 100.

[9] L. R. Rabiner and B.-H. Juang, “An introduction to hidden markov models,” *ieee assp magazine*, vol. 3, no. 1, pp. 4–16, 1986.

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

[11] J. Li, L. Deng, R. Haeb-Umbach, and Y. Gong, *Robust Automatic Speech Recognition: A Bridge to Practical Applications.* Academic Press, 2015.

[12] H. H. Yang, S. Van Vuuren, S. Sharma, and H. Hermansky, “Relevance of time–frequency features for phonetic and speaker-channel classification,” *Speech communication*, vol. 31, no. 1, pp. 35–50, 2000.

[13] X. Wang and D. O’Shaughnessy, “Improving the efficiency of automatic speech recognition by feature transformation and dimensionality reduction,” in *Eighth European Conference on Speech Communication and Technology*, 2003.

[14] R. Haeb-Umbach and H. Ney, “Linear discriminant analysis for improved large vocabulary continuous speech recognition,” in *Acoustics, Speech, and Signal Processing, 1992. ICASSP-92., 1992 IEEE International Conference on*, vol. 1. IEEE, 1992, pp. 13–16.

[15] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics).* Berlin, Heidelberg: Springer-Verlag, 2006.

[16] K. Fukunaga, *Introduction to statistical pattern recognition.* Elsevier, 2013.

[17] R. A. Gopinath, "Maximum likelihood modeling with gaussian distributions for classification," in *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, vol. 2. IEEE, 1998, pp. 661–664.

[18] C. J. Leggetter and P. C. Woodland, "Maximum likelihood linear regression for speaker adaptation of continuous density hidden markov models," *Computer speech & language*, vol. 9, no. 2, pp. 171–185, 1995.

[19] T. Virtanen, R. Singh, and B. Raj, *Techniques for noise robustness in automatic speech recognition.* John Wiley & Sons, 2012.

[20] T. Anastasakos, J. McDonough, R. Schwartz, and J. Makhoul, "A compact model for speaker-adaptive training," in *Spoken Language, 1996. ICSLP 96. Proceedings., Fourth International Conference on*, vol. 2. IEEE, 1996, pp. 1137–1140.

[21] A. Waibel, T. Hanazawa, G. Hinton, K. Shikano, and K. J. Lang, "Phoneme recognition using time-delay neural networks," in *Readings in speech recognition.* Elsevier, 1990, pp. 393–404.

[22] K. Veselỳ, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks." in *Interspeech*, 2013, pp. 2345–2349.

[23] D. Povey, V. Peddinti, D. Galvez, P. Ghahremani, V. Manohar, X. Na, Y. Wang, and S. Khudanpur, "Purely sequence-trained neural networks for asr based on lattice-free mmi." in *Interspeech*, 2016, pp. 2751–2755.

[24] R. Petrick, K. Lohde, M. Wolff, and R. Hoffmann, "The harming part of room acoustics in automatic speech recognition," in *Eighth Annual Conference of the International Speech Communication Association*, 2007.

[25] M. Wolf and C. Nadeu, "On the potential of channel selection for recognition of reverberated speech with multiple microphones," in *Eleventh Annual Conference of the International Speech Communication Association*, 2010.

[26] M. Wolf, "Channel selection and reverberation-robust automatic speech recognition," 2013.

[27] J. G. Fiscus, "A post-processing system to yield reduced word error rates: Recognizer output voting error reduction (rover)," in *Automatic Speech Recognition and Understanding, 1997. Proceedings., 1997 IEEE Workshop on.* IEEE, 1997, pp. 347–354.

[28] S. Jalalvand, M. Negri, D. Falavigna, M. Matassoni, and M. Turchi, "Automatic quality estimation for asr system combination," *Computer Speech & Language*, vol. 47, pp. 214–239, 2018.

[29] H. Unterholzner, L. Pfeifenberger, F. Pernkopf, M. Matassoni, A. Brutti, and D. Falavigna, "Channel-selection for distant-speech recognition on chime-5 dataset," *Energy*, vol. 81, no. 81.6, pp. 81–3, 2018.