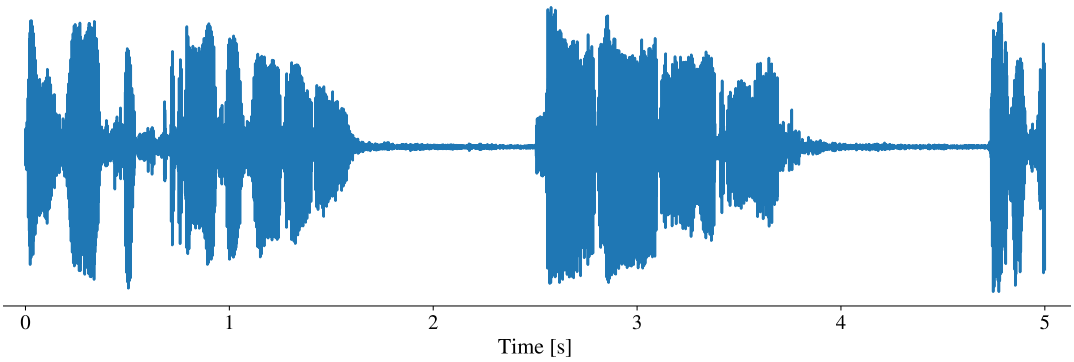
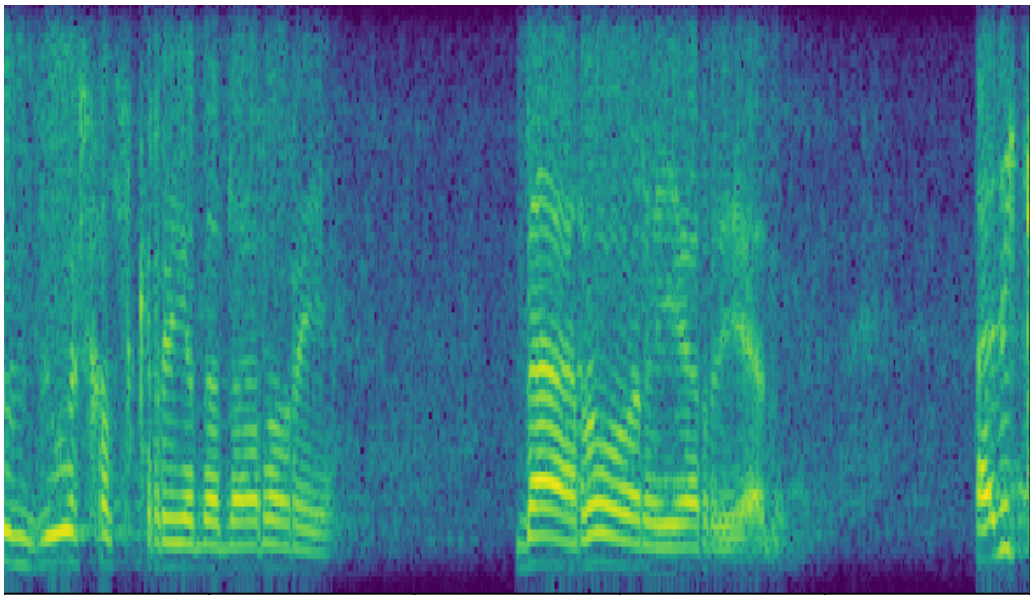
Single-Channel BLSTM Enhancement for Language Identification

Master's Thesis

Peter Sibbern Frederiksen





Aalborg University
Mathematical Engineering

Preface

This master's thesis presents the work of Peter Sibbern Frederiksen as part of the a Master of Science (MSc) degree in Mathematical Engineering at Aalborg University. The thesis is based on a project proposal by Professor Zheng-Hua Tan from the Department of Electronic Systems, Aalborg University, tailored to fit with an eight month visit at the lab of Assistant Professor Najim Dehak at Johns Hopkins University, at the Center for Language and Speech Processing. Official supervision was given by Professor Zheng-Hua Tan, with additional supervision from Associate Research Professor Shinji Watanabe, Assistant Research Professor Jesús Villalba, and Assistant Professor Najim Dehak from the Center for Language and Speech Processing at Johns Hopkins University.

The scope of the thesis was to investigate the use of deep neural network based single-channel speech enhancement on the noisy language identification problem. Speech enhancement has been applied to remove noise for human listening and automatic speech recognition, but not on language recognition to the author's knowledge. A paper version of the results that have been accepted to the Interspeech 2018 conference. The paper is included in the appendix.

The project period was approximately 9 months, from September 2017 to June 2018. The models have been implemented partially in the Kaldi framework and in Python with the PyTorch framework. The code is available for replication of results, and can be found at, "`github.com/psf0/9-jhu`".

Aalborg University, June 7, 2018

Peter Sibbern Frederiksen
<psf@ieee.org>

Copyright © Aalborg University 2016

Mathematical Engineering
Aalborg University
http://www.aau.dk

AALBORG UNIVERSITY
STUDENT REPORT

Title:
Single-Channel BLSTM Enhancement for Language Identification

Project Period:
Spring Semester 2017 and Fall Semester 2018

Project Group:

Participants:
Peter Sibbern Frederiksen

Supervisors:
Zheng-Hua Tan

Copies: 1

Page Numbers: 48

Date of Completion:
June 7, 2018

Abstract:

This project applies deep neural network (DNN)-based single-channel speech enhancement (SE) to language identification. The 2017 language recognition evaluation (LRE17) introduced noisy audio from videos, in addition to the telephone conversations from past challenges. Because of that, adapting models from telephone speech to noisy speech from the video domain was required to obtain optimum performance. Such adaptation requires knowledge of the audio domain. Instead we propose to use speech enhancement preprocessing to clean up the noisy audio. We used a BLSTM DNN model to predict a spectral mask. The noisy spectrogram is enhanced when multiplied by the mask, and it is transformed back into the time domain by using the noisy speech phase. The experiments show significant improvement to language identification of noisy speech, for systems with and without domain adaptation, while preserving performance in the telephone audio domain. In the best adapted state-of-the-art bottleneck i-vector system the relative improvement is 11.3% for noisy speech.

The content of this report is freely available, but publication (with reference) may only be pursued due to agreement with the author.

Matematik-Teknologi
Aalborg University
http://www.aau.dk

AALBORG UNIVERSITY
STUDENT REPORT

Titel:
Single-Channel BLSTM Enhancement for Language Identification

Tema:
Spring Semester 2017 and Fall Semester 2018

Projektperiode:

Projektgruppe:
Peter Sibbern Frederiksen

Deltager(e):
Zheng-Hua Tan

Vejleder(e):
1

Oplagstal: June 7, 2018

Sidetal: 48

Afleveringsdato:

Abstract:

Dette projekt foreslår at anvende dybe neurale netværk (DNN)-baseret enkelt kanal taleforstærkning (SE) på sprog identificering. 2017 language recognition evaluation (LRE17) introducerede støjfyldt lyd fra videoer udover telefon samtalerne fra tidligere evalueringer. Derfor var der et behov for at adaptere modeller fra telefon samtaler til støjfyldt lyd fra video domænet, for at opnå optimal ydeevne. Adapteringen kræver viden om lyd domænet. I stedet foreslår vi et forbehandlings trin der renser den støjfyldte lyd med taleforstærkning. Vi brugte en BLSTM DNN model til at estimere en spektral maske. Det støjfyldte spectrogram bliver taleforstærket når det er multipliceret med masken, og bliver derefter transformeret tilbage til tids domænet ved at bruge den uændrede støjfyldte tales fase. Experimenterne viser en betydelig forbedring til sprog genkendelse af støjfyldt tale, for systemer med og uden domæne adaptering, samtidigt med at den bevare ydeevne i telefonlyds domænet. I det bedste adapterede nyeste flaskehals i-vector system er den relative forbedring 11.3 for støjfyldt tale.

The content of this report is freely available, but publication (with reference) may only be pursued due to agreement with the author.

Contents

Chapter 1
Introduction

Language recognition refers to the process of automatically detecting the language spoken in a speech utterance. Its applications range across customized speech recognition, multi-language translation, service customization and forensics (Bahari et al. 2014). Language recognition is a similar task to speaker recognition. The focus of research in the field has been on developing recognition methods to improve the performance of general systems, while little attention has been given to improving the noise-robustness of language recognition systems.

NIST Language recognition evaluations (LRE) has played an instrumental role in driving language recognition research over the years and LRE constantly increases the challenge level of its evaluations. The most recent LRE 2017 evaluation (*NIST 2017 Language Recognition Evaluation Plan* 2017a) presents a new scenario with a significant mismatch between training and evaluation data. The training dataset consists of a large amount of narrow-band telephone speech, which is in line with previous evaluations. However, the evaluation dataset consists of a combination of narrow-band telephone data and wide-band data from Internet videos. Furthermore, the LRE 2017 organizers provide a limited amount of in-domain development data for model adaptation and calibration purposes. While telephone speech contains low levels of noise and reverberation, it was observed that the video audio data is severely degraded by babble noise, music and reverberation. The LRE 2017 has both a fixed and an open training condition. Fixed training conditions will be offered to allow cross-system comparisons and open training conditions to understand the effect of additional and unconstrained amount of training data on system performance (*NIST 2017 Language Recognition Evaluation Plan* 2017b). This project will focus on the fixed training condition.

There is generally three approaches to make a system noise robust. First train the system using noisy data, and second use data augmented with noise, and thirdly remove the noise beforehand. Noisy training data is not available, which leaves two choices of either augmenting the training data with noise or removing the noise. The advantage of making a model that removes the noise is it could be used by multiple systems. And it would, if it was perfect, spare the following system from having to model noise. This project will focus on removing the noise. Single-channel speech enhancement (SE) can be used as preprocessing to mitigate the aforementioned degradation and reduce the mismatch between training and

evaluation data. SE has been widely used as preprocessing for speech applications, such as automatic speech recognition (ASR) (Weninger et al. 2015), speaker verification (Michelsanti and Tan 2017), mobile communications and hearing aids (Kolbæk, Tan, and Jensen 2017). This project uses using single-channel SE because the data is limited to a single-channel. The research questions posed on this project is

1. What is the effectiveness of utilizing single-channel speech enhancement to improve the noise-robustness of a language recognition system.

2. How can a speech enhancement system be train with only limited noise examples in the in-domain noisy video audio development data?

It has been experimentally shown that applying ideal binary mask in the time-frequency domain is able to improve speech intelligibility of noisy speech signals for both normal hearing and hearing impaired listeners with various noise types (Wang et al. 2009). Various ideal ratio masks have become preferable over ideal binary mask in recent studies (Kolbæk, Tan, and Jensen 2017; Wang, Narayanan, and Wang 2014; Erdogan et al. 2015). In (Lu et al. 2013; Xu et al. 2015) a DNN is trained to predict clean speech from noisy speech without the use of a mask by casting it as a regression problem. A long short-term memory (LSTM) network has shown to outperform feed-forward DNN methods, when used as preprocessing for noise robust ASR (Weninger et al. 2015), and the bidirectional extension of LSTM (BLSTM) achieves further improvement (Erdogan et al. 2015). This project follows the success of the BLSTM SE method, and applies it to a language recognition system. The BLSTM SE is processed in the time-frequency domain, but only deals with the magnitude while the phase component remains corrupted, similar to the other DNN-based SE methods. The method internally predicts a mask from BLSTM, and the predicted mask is multiplied by the noisy speech magnitude, which yields the enhanced magnitude. The network is trained with the mean square error criterion between the clean and enhanced magnitudes. In BLSTM SE (and other DNN-based enhancement), only additive noise is considered, where the noise source is extracted from in-domain data with limited size in our setup. The effectiveness of BLSTM SE on the language identification is evaluated by a state-of-the-art bottleneck i-vector LRE system, where BLSTM SE is used as preprocessing of the LRE system (Richardson et al. 2018).

To validate the effectiveness of the BLSTM SE methods, it is compared with the optimally-modified log-spectral amplitude (OM-LSA) speech estimator with the improved minima controlled recursive averaging (IMCRA) noise estimator (Cohen and Berdugo 2001), (Cohen 2003). OM-LSA is a well-known signal processing method that does not require data-driven training and adaption stages.

The outline of the report is as such: Chapter 2 introduces the language recognition system. Chapter 3 introduces the speech enhancement system. Chapter 4 contains the experimental setup and results. Chapter 5 is the conclusion and future work.

Chapter 2
Language Recognition System

The purpose of this chapter is to describe the language recognition system used for the experiments. The language recognition system is a state-of-the-art bottleneck i-vector system (Richardson et al. 2018). The system is implemented in Kaldi (Povey et al. 2011). The pipeline of the system can be seen in figure 3.1 with the proposed preprocessing speech enhancement step added.

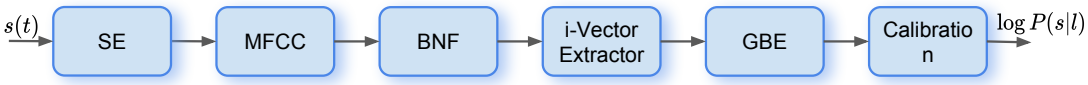


Figure 2.1: Proposed i-Vector language recognition system with single-channel enhancement. The language recognition system takes the time domain speech signal as input and returns for each language a log probability of the speech signal given the language. The system consists of the following blocks: Speech Enhancement, Mel Frequency Cepstral Coefficient, BottleNeck Features, i-Vector Extractor, Gaussian classifier Back End, Calibration

The overall function of each step in the system can be described as:

- Speech enhancement removed noise from the input speech signal $s(t)$.
- MFCC are features engineered for human speech time domain signals.
- BNF extracts features using a DNN utilizing a large time context window.
- i-Vector extractor extracts fixed length features for language recognition.
- The Gaussian classifier classifies languages from high level i-vector features.
- The Calibration is an affine transformation that ensures proper log probabilities.

The steps MFCC, BNF, and i-vector extractor should be considered the front end of the baseline language recognition system and GBE and calibration should be the back end. The front end is responsible for extracting features and the back end is responsible for the classification. The remaining section will explain the individual steps outlined above.

2.1 Speech Enhancement

The purpose of the speech enhancement system is to make the language recognition system robust to noisy speech signals by removing the noise. This speech enhancement step is not part of the baseline language recognition system. It takes the speech signal $s(t)$ in the time domain and returns the enhanced speech signal $\hat{s}(t)$ in the time domain. The speech enhancement block is an independent preprocessing step to the language recognition. This enables the use of an out of the box speech enhancement system which eliminates the long training process or training a speech enhancement system tailored to the language recognition task, languages, and noise domain. The latter will be attempted in chapter 3 with a BLSTM model.

A classical speech enhancement signal processing algorithm should be used for comparison with the BLSTM which is a DNN model. The baseline speech enhancement system is the optimally-modified log-spectral amplitude (OM-LSA) speech estimator with the improved minima controlled recursive averaging (IMCRA) noise estimator (Cohen and Berdugo 2001), (Cohen 2003). OM-LSA is a well-known signal processing method that does not require data-driven training and adaptation stages. It was recently used as a baseline in(Chazan, Goldberger, and Gannot 2017).

2.2 Mel-Frequency Cepstral Coefficients

The purpose of the Mel-Frequency Cepstral Coefficients (MFCC) block is to extract features from time domain speech. These features describe the overall shape of the power spectral density, which is the energy as a function of the frequency, in a time-frequency domain. The time-frequency domain consists of 25 millisecond segments or frames with a 10 millisecond shift to the next segment. The MFCC has many parameters, but most importantly is how many coefficients are kept in the end corresponding to how much information is discarded. The MFCC extracting features from human speech signals have been hand engineered by subject matter experts (O'shaughnessy 1987). It relies on assumptions such as the power spectral density is smooth and the dynamic range of the higher frequencies are higher than the low frequencies, but not more important. Additionally, the energy of the higher frequencies is lower, but less important because of it. The overall step of the MFCC transform in the Kaldi implementation is as follows (Povey et al. 2011):

1. Optional: dithering, preemphasis, dc offset removal.
2. Windowing
3. Compute the power spectrum with the FFT.
4. Compute the energy in each bin of the mel filter bank.
5. Take the cosine transform of the log energy.
6. Keep the specified first number of coefficients.

The parameters used are: dithering, preemphasis, dc offset removal, windowing with 25 millisecond frames shifted by 10 millisecond each time, 23 mel filters, and 20 MFCC kept.

MFCC has a history of use in the hidden Markov model automatic speech recognition. The idea being these features are useful in predicting the phoneme of each segment. The MFCC are used as input to a deep neural network. Deep neural networks are not dependent on human engineered features. The idea is that they learn the features they use from data in addition to the original task. With every layer in a deep neural network it could learn higher level features or more complicated deep features. The DNN could learn its deep features directly from time domain frames input features.

This raises the question of why keep using it? Assuming MFCC is an optimal input feature and a deep neural network is trying to learn the MFCC transform, it might be a difficult transform to learn. One should also expect that it would require more data and training time to learn features directly from lower level features such as the time domain, than higher level MFCC features. If this is true and if the MFCC is optimal or close to it then using MFCC features would be beneficial to the model. On the other hand if the MFCC throws away vital information it should be expected that it would be beneficial to dropping the MFCC in favor of time domain frames or a more general time-frequency domain feature like the spectrogram. The point being that these alternatives makes less assumption about human speech signals. In any case the human engineered MFCC features are still in use in this state-of-the-art system, which should give credit to the MFCC assumptions. The different elements of the extraction of the MFCC will now be described in the following section.

2.2.1 Human Speech Signal

Sound is variations in air pressure over time. Human speech sound can be divided into three types of sound or phonemes, which is voiced, unvoiced, and plosive (Deller Jr., Hansen, and Proakis 2000). Voiced sounds are made by pushing air past relaxed vocal cords, such that they vibrate in a repeating pattern of opening and closing. This produces quasi-periodic longitudinal air pressure waves in the vocal tract. Quasi-periodic mean that the periods are irregular or flawed. Examples of voiced sounds are the vowels and some consonants like '/b/', '/d/', '/g/' and '/v/'. Unvoiced sounds are stochastic in nature, unlike the quasi-periodic voiced sounds that are close to being deterministic. They are formed when the vocal tract is constantly open. The process is analogous to white noise being filtered into colored noise. Examples of unvoiced sound are '/s/' and '/f/'. Plosive sounds like '/p/', '/t/' and '/k/' are formed by closing the vocal tract with the lips, tongue, etc. and building up pressure before releasing it quickly.

A microphone is used to convert the sound wave signals to an analog electric voltage signal. The Sound wave signals can be digitally represented with a wavefile '.wav' in the time domain, where the analog electric voltage signal is sampled with a sampling frequency f_s and a quantization of the amplitude. The waves fluctuate

around zero, representing silence, with more extreme positive and negative values corresponding to higher wave amplitudes and in turn louder sounds. It is possible to see the change in volume over time, and recognize sound types, but it is difficult to distinguish sounds within the same type from each other, in the time domain. At the 5 second time scale of a human speech signal in figure 2.2 it is possible to see the changes in volume and two long segments of silence. This sample will be reused throughout this section for illustration purposes. At a much smaller time scale of 25 milliseconds the quasi-periodic pattern of the voiced sound '/b/' can be seen in figure 2.3 with its corresponding frequency content in figure 2.4. At the same time scale of 25 milliseconds the colored noise of the unvoiced sound '/s/' can be seen in figure 2.5 with its corresponding frequency content in figure 2.6. It has a much lower energy than the voiced sound, There is however a noticeable offset away from zero, which has led to a noticeable energy at the zero Hz frequency. It is however similar to the '/b/' in figure 2.4. Such offset can be an artifact of the microphone recording. Things such as wind or breath can disturb the microphone during recording. At the time scale of 100 milliseconds the word 'the' with the plosive sounds in the beginning can be seen in figure 2.7 with its corresponding frequency content the entire word in figure 2.8. They are all clearly different from each other.

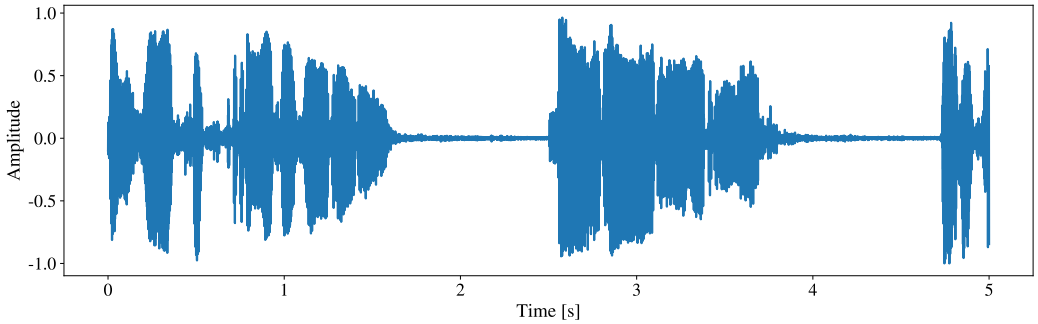


Figure 2.2: This illustrates a 5 second speech segment in the time domain using a 16-bit and 8000 kHz. The signal has been decoded from the 8-bit μ-law encoding. The signal has several spoken words with pauses in between. The x-axis is time and the y-axis is amplitude of the sound wave signal.

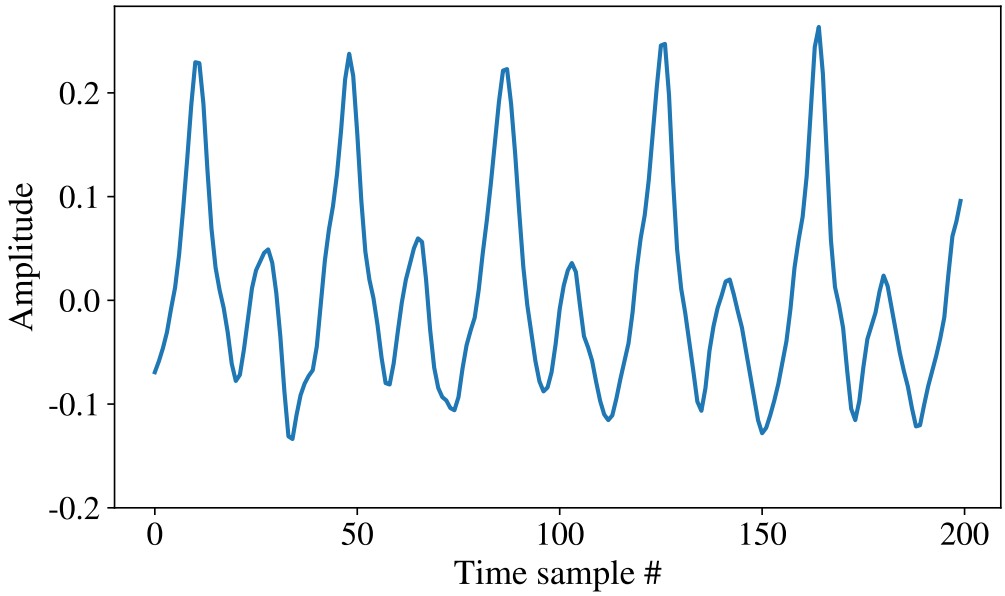


Figure 2.3: This is a 25 millisecond sample of the voiced sound ′/b/′ in the time domain using a 16-bit linear PCM and 8000 Hz sampling. Note the quasi-periodic pattern. There is close to 39 samples per quasi-period corresponding to a frequency of 205 Hz. Some of the imperfections are the highest of the positive alternating peaks increasing amplitude and other's declining amplitude. Thirdly by the end the two negative peaks have become equal in amplitude.

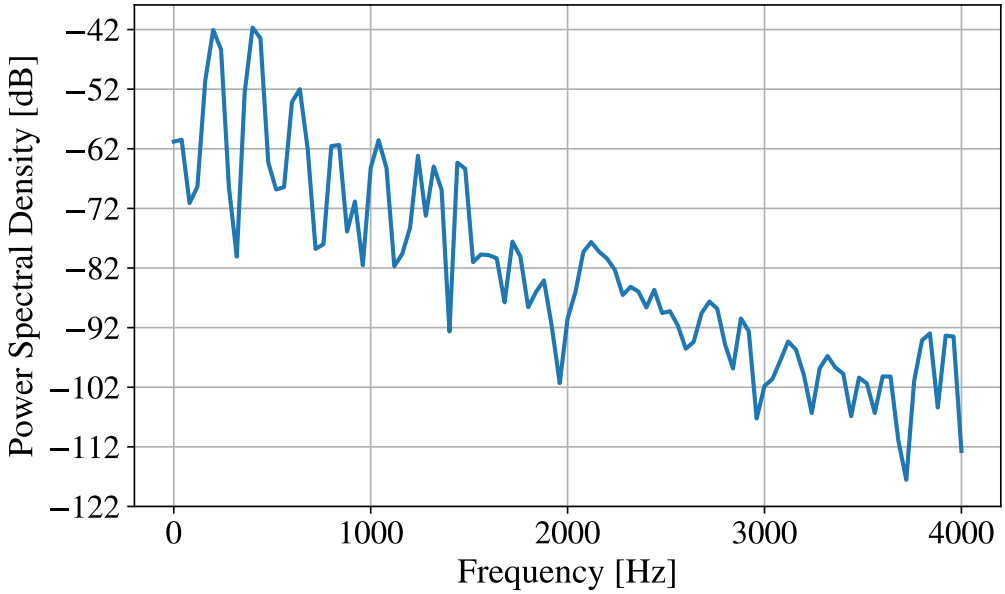


Figure 2.4: This is the power spectral density of the voiced sound ′/b/′ in figure 2.3. The first peak is around 210 Hz with the next at 415 Hz which corresponds to the fundamental frequency and the second harmonic. The peaks are not exact due to noise and only 25 milliseconds of data is being used.

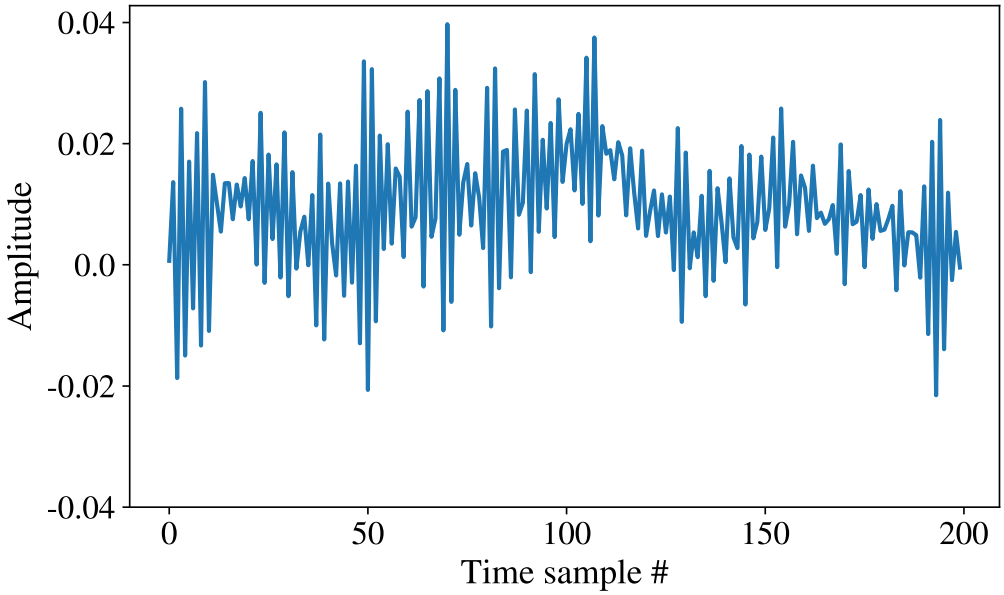


Figure 2.5: This is a 25 millisecond sample of the unvoiced sound ′/s/′ in the time domain using a 16-bit linear PCM and 8000 Hz sampling. The stochastic pattern is high frequency which can be seen in figure 2.6. The amplitude is significantly smaller than the voiced sound ′/b/′ in figure 2.3. There is a large offset relative to the amplitude of the sound segment.

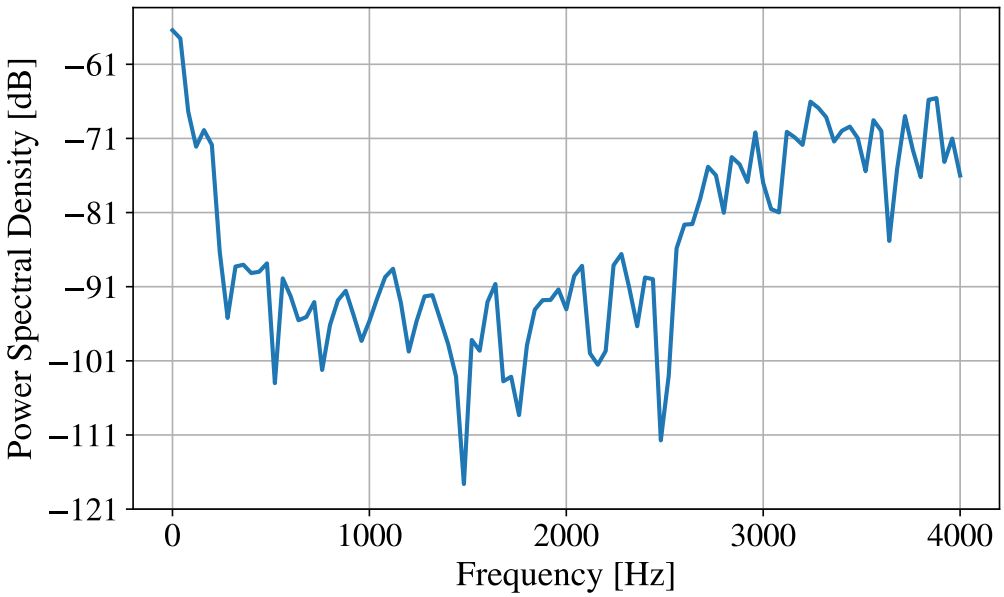


Figure 2.6: This is the power spectral density of the unvoiced sound ′/s/′ in figure 2.5. There are no peaks rather the energy is spread in the interval ranging from close to 2600 Hz to 4000 Hz. There is also significant energy at the lowest frequencies due to a relatively large offset in figure 2.5.

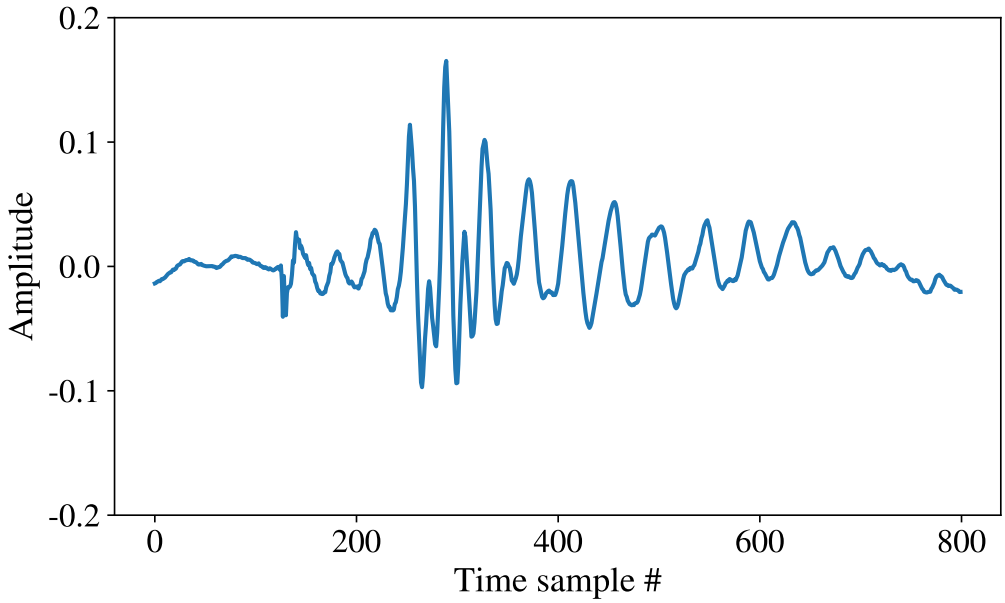


Figure 2.7: This is a 100 millisecond sample of the word 'the' in the time domain using a 16-bit linear PCM and 8000 Hz sampling. The plosive sound in the beginning has the largest amplitude and a quasi-period and low frequency pattern can be seen at the end of the word.

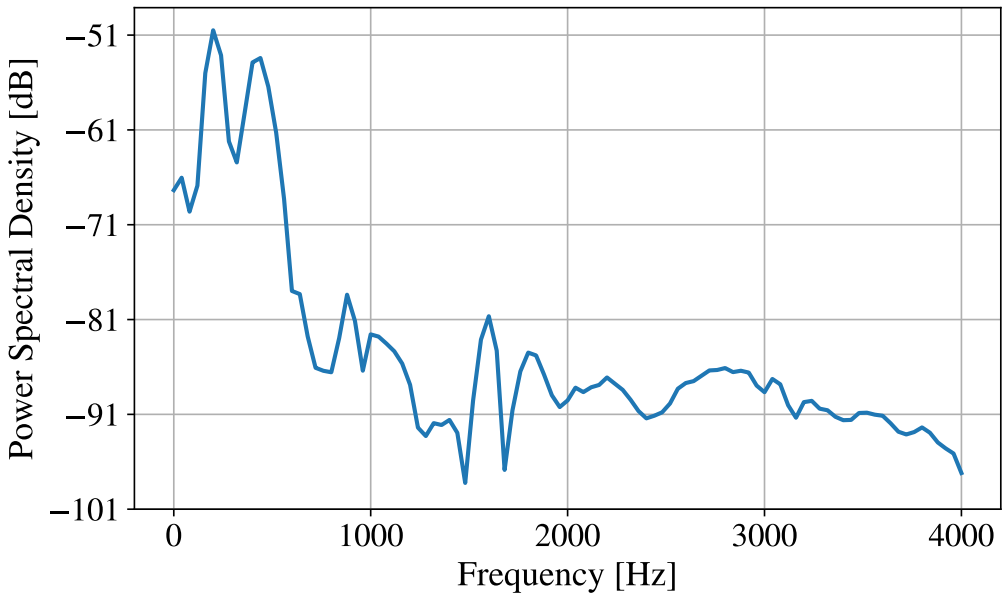


Figure 2.8: This is the power spectral density of the word 'the' in figure 2.7.

2.2.2 Speech Signal Sample Rate

Most of the data available for this project is narrow-band telephone speech signal that uses a sampling rate of 8000 Hz (*NIST 2017 Language Recognition Evaluation Plan* 2017b) and quantization with μ-law or A-law 8-bit encoding. Because of limited data sampled at a higher frequency the sampling frequency f_s of 8000 Hz is used in this project, corresponding to 0.125 milliseconds between samples. By the Nyqust-Shannon sampling theorem frequencies in the interval $[0, 4000)$ Hz can be represented. Human hearing is limited to 20000 Hz, but it is typically lower for adults. Parts of the speech signal is lost (Pisoni and Remez 2004), but a higher sample rate of 16000 Hz is only available for the newest data, and a choice is made to down sample everything to 8000 Hz.

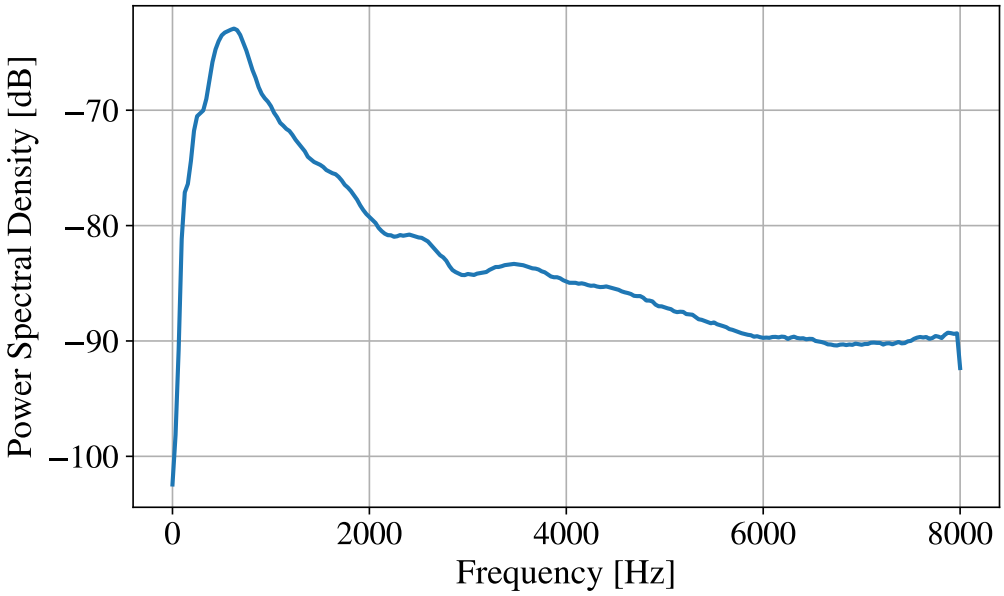


Figure 2.9: The average power spectral density of the TIMIT corpus. Using a 512 Hann window with no overlap. The average energy in the higher frequency bands decreases quickly, especially considering the logarithmic scale.

What is lost by down sampling? The average power spectral density of human speech sampled in 16000 Hz can be seen in figure 2.9. Only very little energy is lost on average by using the 8000 Hz sampling frequency. The energy in the 4000-8000 Hz frequency band is much lower than the 0-4000 Hz frequency band. Despite its low energy the 4000-8000 Hz frequency band might have variations from language to language containing useful information for language recognition. The speech signal is from the TIMIT corpus, specifically the training set that contains close to 4 hours of speech (Garofolo 1993). TIMIT consist of 16-bit 16000 Hz microphone speech signal, which is a higher quality than the 8-bit μ-law or A-law 8000 Hz narrow-band telephone speech signal. The narrow-band telephone speech signal has in general been band-pass filtered to the 300-3400 Hz frequency band, which help send more signals with a limited available analog transmission bandwidth

(*Recommendation G.191 STL-2009 Manual (11/09)* 2009). The power spectral density may be a biased estimate average human speech signal power spectral density. The TIMIT corpus have 438 male and 192 female American speakers, and women have a higher pitch than men (Archana and Malleswari 2015). Whether it differs from the gender distribution of LRE17 is unknown. In addition,, the corpus has a limited range of sentences 10 sentences being spoken per person. Two of the sentences are identical, five of them drawn from a group of 450 sentences, and the remaining three drawn from another group of 1890 sentences. There is a considerable overlap in the sentences compared to the LRE narrow-band telephone speech signals, where the sentences are chosen at random in conversation. That however does not guarantee that it is unbiased. To summarize the power spectral density may biased, but the indecation is most of the speech signal energy is preserved with the 8000 Hz sampling frequency. New language identification datasets with higher sample rates are required to conclude if the energy lost is important.

2.2.3 Speech Signal Quantization

The signal amplitude is quantified with a pulse code modulation (PCM) where each symbol corresponds to a quantization level. The simplest is linear PCM where the quantization levels are uniformly spread. In this case a 16-bit signed integer is used by assigning each of the integers in the interval $(-2^{15}, 2^{15}-1)$ which is $(-32768, 32767)$ as a quantization level. The recorded sound waves fluctuate around zero and waves with higher amplitudes reach higher negative and positive values. Care has to be taken to ensure the signal amplitude is contained within the quantization range or else the amplitude will be clipped to the smaller, but highest quantization level.

Other encodings exist such as the logarithmic PCM μ-law and A-law. The disadvantage of a linear quantization is that the quantization error and thereby the signal-to-noise-ratio will change with the amplitude of the speech signal. A lower amplitude corresponds to a lower signal-to-noise-ratio. The amplitude of the human speech signal fluctuates, this will lead to a fluctuating quality. In the late 1960's the two logarithmic quantization algorithms was developed making quantization noise more uniform and less dependent on the amplitude. The quantization levels are denser closer to zero. These encodings are designed to compress human speech recordings, for use in telephone applications, with a sample rate of 8000 Hz and 8-bits per sample, the bit-rate is 68 kbit/s (*Recommendation G.191 STL-2009 Manual (11/09)* 2009, Chapter 3). Most of the dataset used in training this model was narrow-band telephone signals which has been compressed using these encodings. The logarithmic PCM μ-law and A-law are based on analog compression characteristics. The compression characteristics for μ-law quantization is given by

$$c(x) = \frac{log(1+\mu|x|)}{1+log(\mu)}\text{sign}(x), \quad -1 \leq x \leq 1 \tag{2.1}$$

where μ is 255 because of the 8-bit choice and the range of x has been normalized

to $[-1, 1]$. The compression characteristics for A-law quantization is given by

$$c(x) = \begin{cases} \frac{A|x|}{1+log(A)}\text{sign}(x) & \text{if } |x| \leq \frac{1}{A} \\ \frac{1+log(A|x|)}{1+log(A)}\text{sign}(x) & \text{if } \frac{1}{A} \leq |x| \leq 1 \end{cases} \tag{2.2}$$

where $A = 87.56$ as chosen by ITU and the range of x has been normalized to $[-1, 1]$. The range of the compression characteristics is also in the interval $[-1, 1]$. The compression characteristics of μ-law is close to linear when the amplitude is small, but A-law is linear (*Recommendation G.191 STL-2009 Manual (11/09)* 2009, Chapter 3). A comparison of the two compression characteristics when the amplitude is small can be seen in figure 2.10.

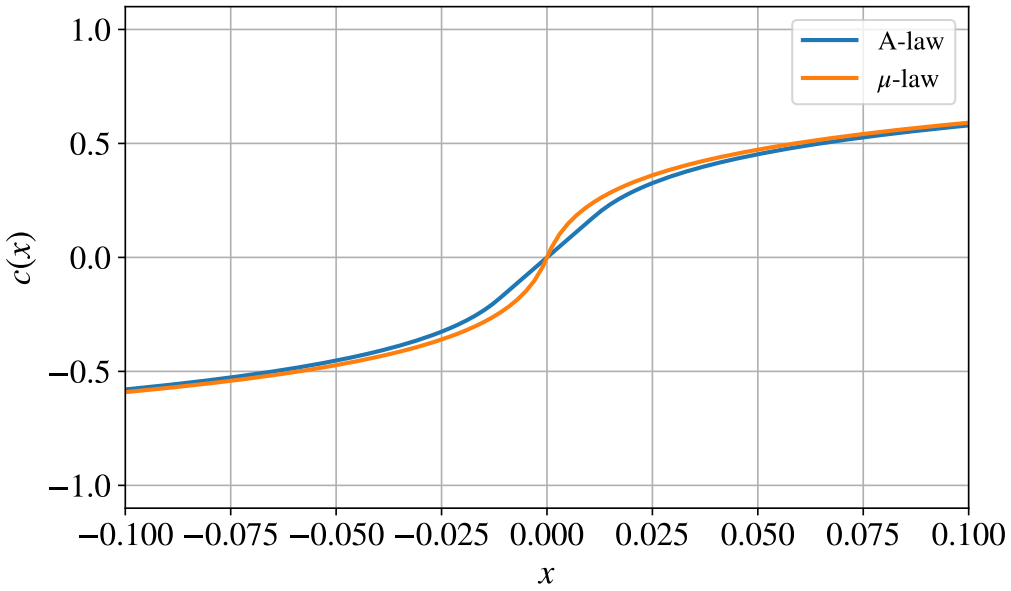


Figure 2.10: This is a comparison of the two logarithmic PCM μ-law and A-law for small amplitudes. $c(x)$ is the compression characteristics which is a function of the amplitude x, which has been normalized to the interval $[-1, 1]$. μ-law and A-law are similar for larger amplitudes. A-law is a piecewise function and for smaller amplitudes within $|x| < 0.011$ it is linear. The compression of x is related to the derivative of $c(x)$. A derivative of one implies uniform quantization levels. When the derivative is smaller than one expansion occurs and when it is larger than one compression of the quantization levels occurs.

A linear-piecewise approximation is used to convert the compression characteristics to digital 8-bit encodings. The fist bit is a sign, the next 3 bits is the identity of the linear-piecewise segment, and the remaining four is position in a given segment. For μ-law (2.1) that results in 15 linear-piecewise segments, because of symmetry the one segment on either side of zero can be joined as one. The 15 linear-piecewise approximation of the μ-law compression characteristics can be seen in figure 2.11. For A-law (2.2) two segments on either side of zero can be joined as one because of symmetry and the linear expression within $|x| \leq \frac{1}{A}$ in (2.2). The result is 13 linear-piecewise segments.

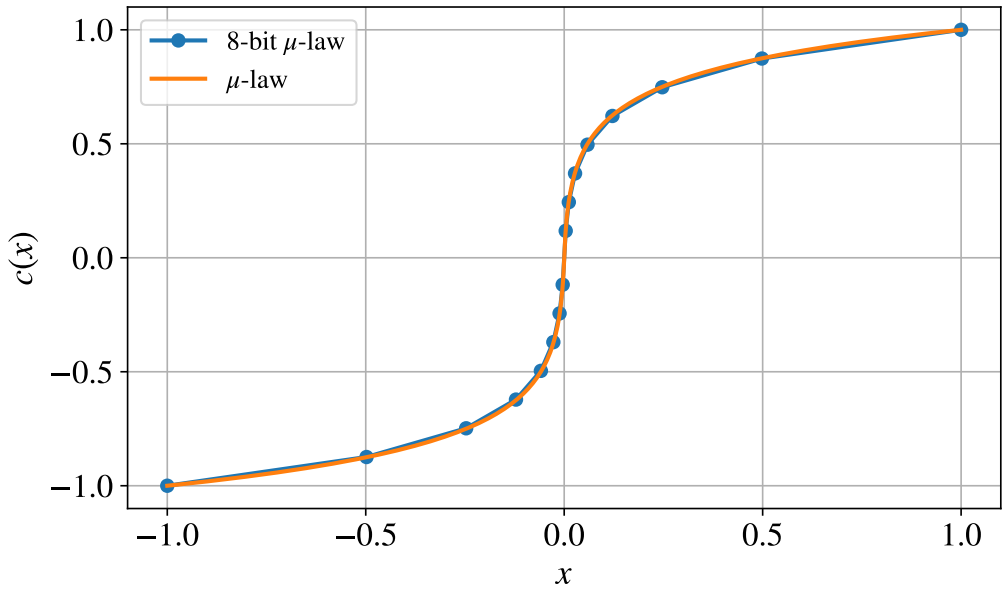


Figure 2.11: The 8-bit μ-law linear-piecewise approximation can be seen with the circles marking the connections. $c(x)$ is the compression characteristics which is a function of the amplitude x, which has been normalized to the interval $[-1, 1]$. There are 16 bins within in each segment, except the middle which has 32. Each bin correspond to a μ-law quantization level. The bin intervals are doubling in length with each interval away from zero. For a 16-bit linear PCM input x the outer bins each contain 1024 linear PCM quantization level.

To do the compression the tables of (*Recommendation G.711 (11/88)* 2009) contains the details of the 8-bit linear-piecewise approximation of μ-law and A-law. Decoding back to linear PCM can be done by table lookup or algorithmic conversion. The sounds signal segments of '/b/', '/s/', and '/t/' from the figures 2.3, 2.5, and 2.7 look visually unchanged if quantized with 8-bit μ-law quantization. As explained in figure 2.12 that would not have been the case for the sound '/s/' in figure 2.5. The lower amplitude of unvoiced sounds would be completely distorted using 8-bit linear PCM. At the other end of the scale the voiced '/b/' sound segment from 2.3 is scaled to the highest amplitude in figure 2.4, where it looks visually unchanged, a testament to the logarithmic PCM quantization methods.

Quantization to the 8-bit μ-law and A-law incur an error can be referred to as the quantization noise. There is also a quantization error in 16-bit linear PCM, but it is $2^8 = 256$ times less than 8-bit linear PCM. If the quantization noise is correlated with the signal, then harmonic distortions are introduced that corrupt the spectrum of the signal. Kaldi uses a process called dithering which adds noise to the signal such that the quantization noise appears uncorrelated with the signal. The harmonic distortions are reduced, but white noise is added to the spectrum instead. This is relevant when quantizing or encoding with fewer quantization levels.

float barrier

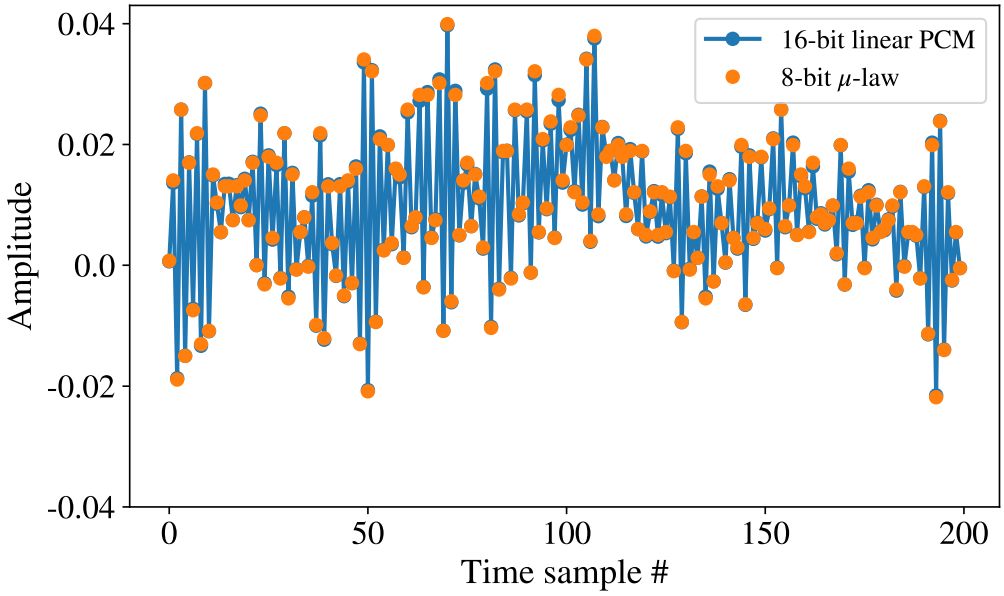


Figure 2.12: This is a comparison of 16-bit linear PCM versus 8-bit μ-law quantization encoding at low amplitudes. The speech segment used is from figure 2.5 and is a 25 milliseconds sample of the unvoiced sound '/s/' in the time domain using a 16-bit linear PCM and 8000 Hz sampling. The 8-bit μ-law samples in orange is on top of the 16-bit linear PCM samples, but 8-bit linear PCM would only have had $(2^8 \cdot 0.06)/2 \approx 8$ quantization levels, but 8-bit μ-law has 95 quantization levels available.

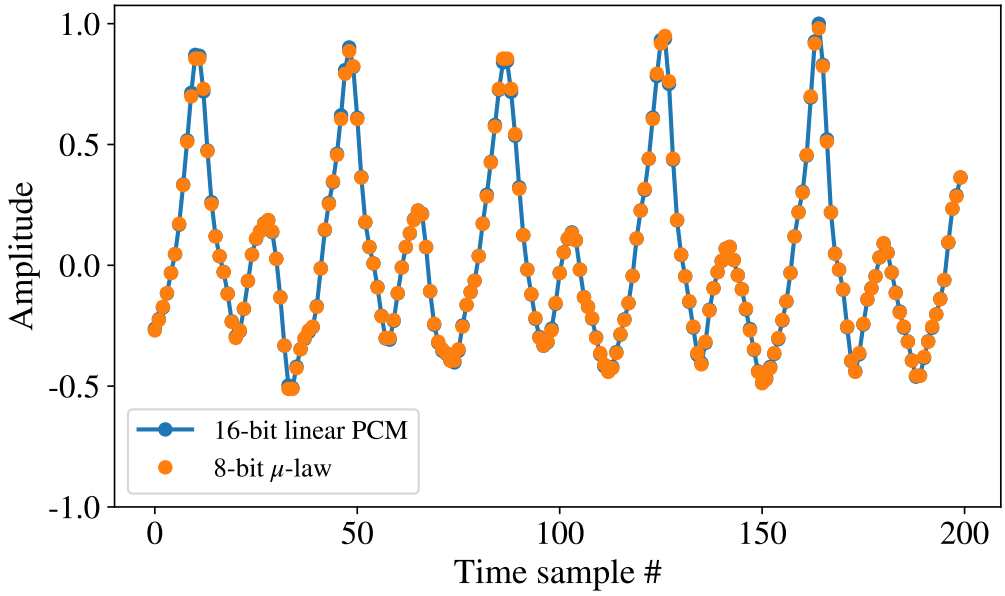


Figure 2.13: This is a comparison of 16-bit linear PCM versus 8-bit μ-law quantization encoding at high amplitudes. The speech segment used is from figure 2.5 and is a 25 milliseconds sample of the voiced sound '/b/' in the time domain using a 16-bit linear PCM and 8000 Hz sampling. The 8-bit μ-law samples in orange is on top of the 16-bit linear PCM samples, but 8-bit linear PCM would only have had 256 16-bit linear PCM values per quantization level at the edge, but 8-bit μ-law has 1024.

2.2.4 Preemphasis

By applying a digital high-pass filter before further processing emphasis is put on the higher frequencies, which have a lower energy in human speech signal. The filter also reduces the speech signal offset found in the '/s/' sound signal segment in figure 2.5. The filter is defined as

$$y_n = x_n - 0.97 \cdot x_{n-1} \tag{2.3}$$

where y_n is the output sample n and x_n is the input sample n. Its frequency response is shown in figure 2.14. The filter has one zero at 0.97 and is a finite impulse response filter of order 1. The filter has a gain of 0.03 at 0 Hz and a gain of a half at 650 Hz and a gain of 1 at 1350 Hz and gain of 1.97 at 4000 Hz. The effect of preemphasis on the power spectral density of a speech signal segment can be seen in figure 2.15.

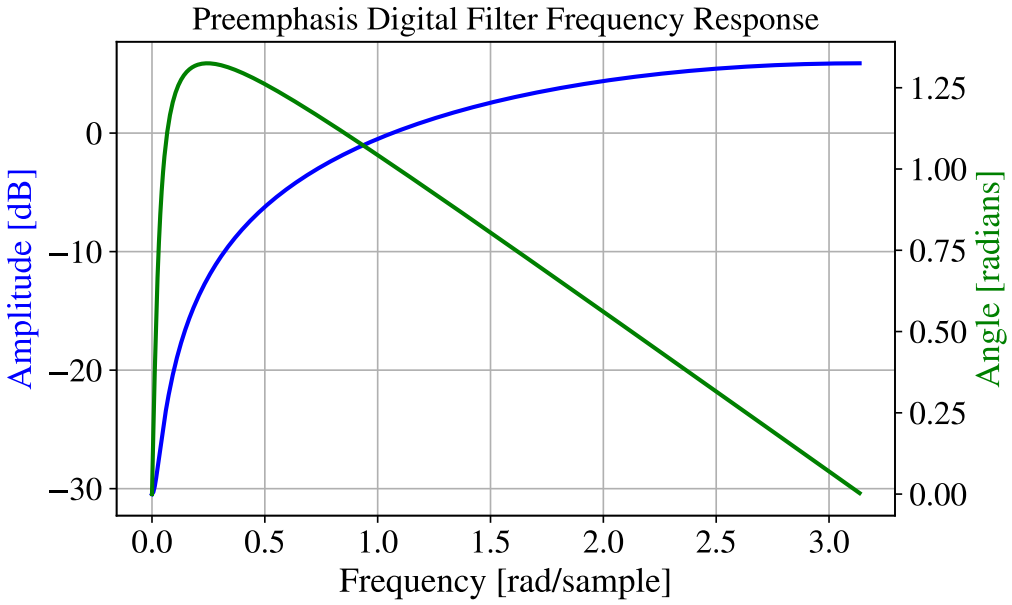


Figure 2.14: This is the frequency response of the preemphasis digital FIR filter. It is a high-pass filter meant to be used on the time domain speech signals.

Preemphasis is a computational cheap normalization method. It could be replaced with a full normalization, but it is still used today in the Kaldi toolbox (Povey et al. 2011). Normalization can be used to ensure all features have the same importance. Machine learning algorithms such as deep neural networks often use a full normalization of its input dataset. This could be rescaling or mean and variance removal. This is very expensive compared to applying the 1. order digital high-pass filter. Preemphasis would not have an effect on relative power between the low and high frequencies if such normalizations were used, if the normalization was used in an appropriate domain where the relative importance of the frequencies can be changed. For a simple example where this can not be done,

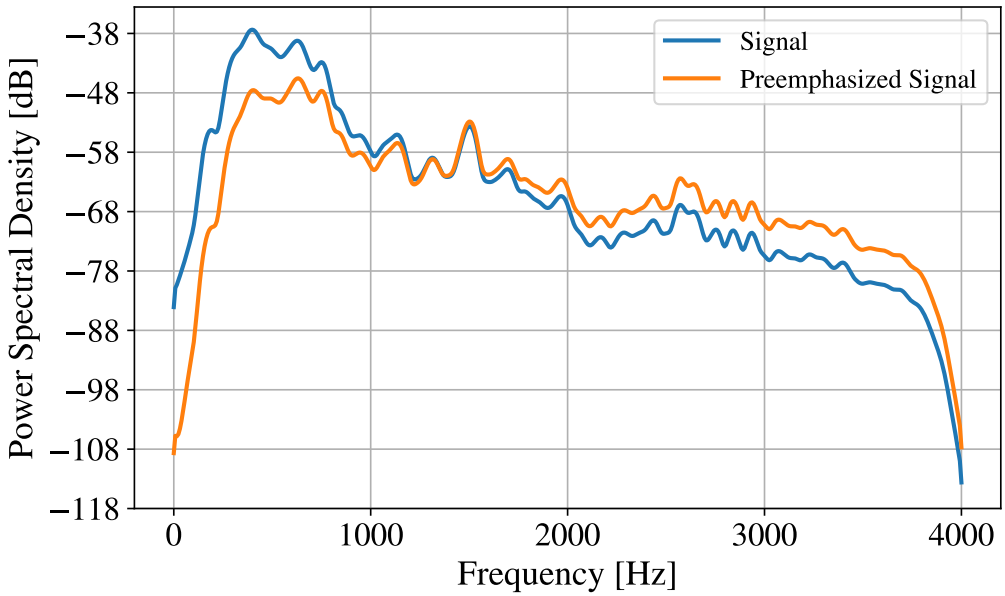


Figure 2.15: This is a comparison of the average power spectral density of the speech signal from 2.2 before and after applying the preemphasis digital FIR filter in equation (2.3). The preemphasized signal has almost lost energy at the lowest frequencies especially at 0 Hz. The higher frequencies have been amplified, but they are still weaker than the lower frequencies.

take total energy of the signal then the relative importance of energy from only a subset of the frequencies can not be increased.

Does all the languages behave in the same way as the small 5 second speech signal sample in figure 2.15? The average power spectral density of the TIMIT corpus in figure 2.9 showed that the average spectral power of the English language was lower for the higher frequencies, with more than a order of magnitude. To verify that this is also the case for all the other languages in the language groups the average power spectral density for all the languages is shown in figure 2.16. All the languages show the same declining trend. There is quite a bit of variety among the languages between 3400 Hz and 4000 Hz. Some languages seems to be dominated with narrow-band telephone data that have been low-pass filtered to the 300-3400 Hz frequency band. Data for some of the languages have been collected from multiple sources, that might explain some the differences in the 3400-4000 Hz frequency band (*NIST 2017 Language Recognition Evaluation Plan* 2017b).

see table for the language groups,

Why does the languages have low average power spectral density in the higher frequencies? The unvoiced '/s/' sound signal sample in figure 2.5 was shown to have most of its energy in the 2500-4000 Hz band. That segment has considerably lower energy than the voiced '/b/' sound signal sample in figure 2.3. Is the average power spectral density in the higher frequencies low because sounds there have lower energy, or is the average low because sounds in the higher frequencies are a

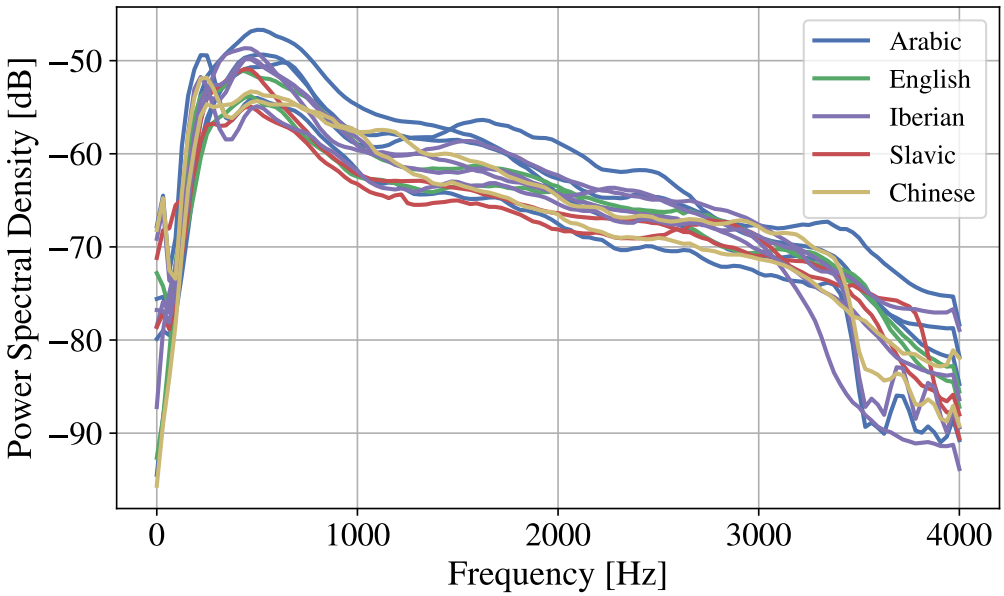


Figure 2.16: This is a comparison of the average power spectral density of the speech signal from in the LRE17 training set. The languages are color coded with the five language clusters of the LRE17 training set. The mean of the average power spectral densities falls from about -53 dB to -73 dB in the interval 500 Hz to 3400 Hz. That is an order of magnitude per 1450 Hz. Each language uses up to 12 hours speech signals with a few exceptions, due to less data being available. For each language hours of speech signals has been segmented into 32 millisecond segments with no overlap. The average power spectral density is calculated using the resulting 127 DFT bins from each segment.

rare occurrence? Figure 2.17 shows that the higher frequencies have lower energy and that their highest energy follows the same declining trend as all the languages in figure2.16. The figure uses the American English, but the same trend is assumed to be true for the other languages. Sounds in the higher frequencies might also be less commonly occurring, but the same trend of 20 dB decline in energy matches the observations of 2.16. In summation preemphasis increases the relative energy of the lower energy high frequency sounds in the speech signals of all the languages.

float barrier

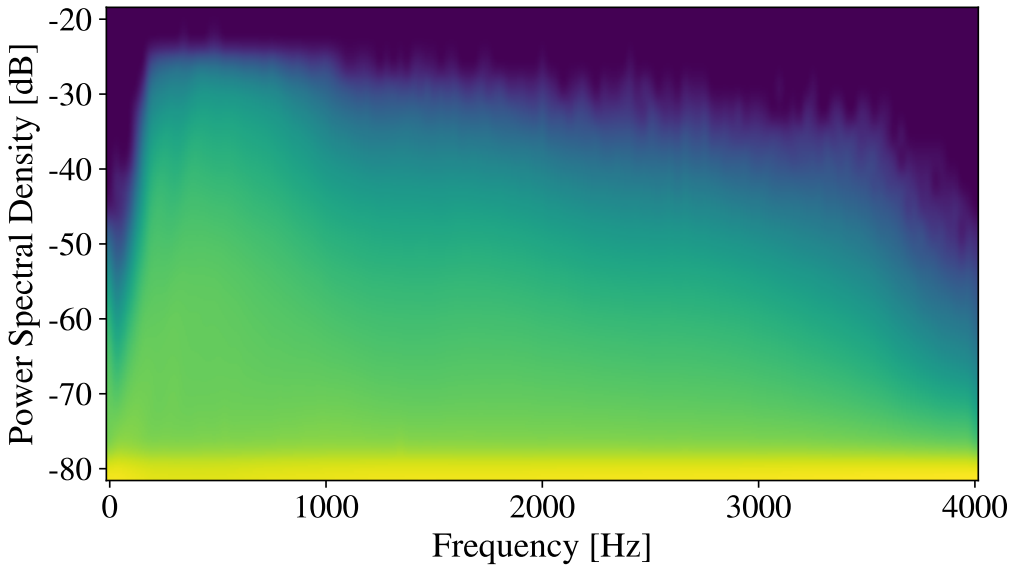


Figure 2.17: This is the distribution of power in the power spectral density of the American-English in the LRE17 training set. The rate of occurrence values are plotted in a log scale, because of the many empty segments caused by silence and sounds being sparse in the frequency domain. The 12 hours of the speech signals has been segmented into 32 millisecond segments with no overlap. The power spectral density for each segment is calculated in dB. A 20 bin histogram has been calculated for each of the 127 DFT frequency bins, and the values have been smoothed using quadratic polynomial interpolation. There is a declining trend of the energy with higher frequency.

2.2.5 Time-Frequency Domain

Speech can also be represented in the time-frequency domain using the discrete short-time Fourier transform (STFT). This has already been done implicitly during the power spectral density calculations before the averaging across the frames. The time domain representation of speech signals is a correlated and dense representation of speech, but the time-frequency representation is sparse. This time-frequency domain is called the power spectrogram, or spectrogram for short, when the magnitude is squared of the STFT values. Each time step is referred to as a frame and it corresponds to a column of the spectrogram containing a power spectrum. The spectrogram is plotted in figure 2.18. With the spectrogram the sounds can be differentiated by their short-time frequency content across time.

The STFT consists of first windowing sound segments into frames, computing the power spectrum for each frame, and then stacking the power spectrums as columns into a time-frequency matrix. The purpose of the windowing is to make stationary speech signal frames. Human speech signals vary by phoneme, but there is also a transition between the phonemes. This means that the human speech signal is non stationary and that the power spectrum changes over time. What would happen if the power spectrum was to be computed on a non stationary signal? Assuming that it consists of multiple sounds that by themselves are

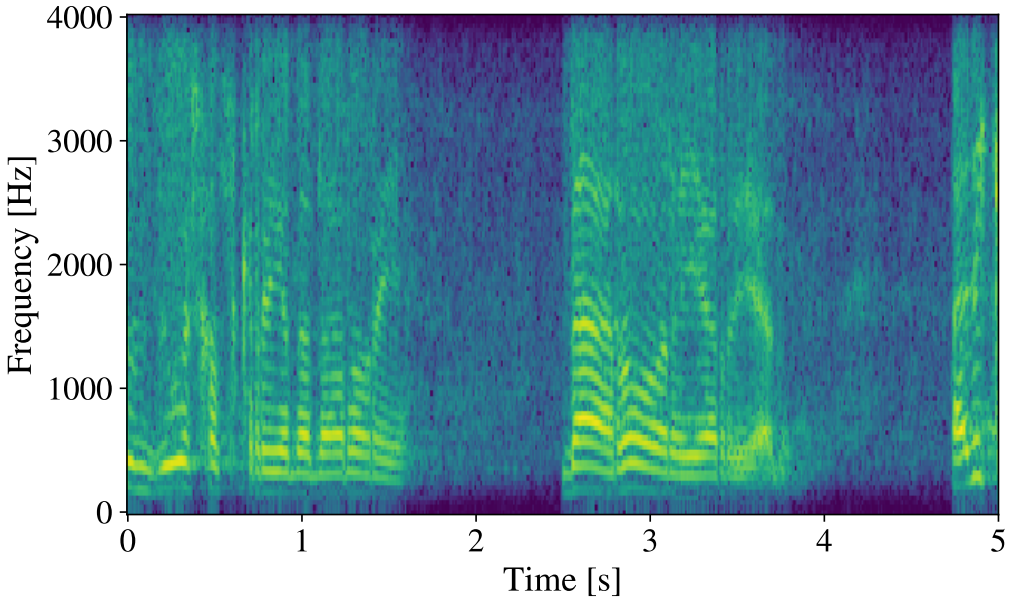


Figure 2.18: This figure illustrates the spectrogram of a 5 second speech segment which is in the time-frequency domain. It is the same speech segment as seen in the time domain in figure 2.2. Time is still on the x-axis, but the y-axis now has frequency. There are 129 frequency bins per frame and the color shows the log power of a frequency bin at that given time frame. A pattern can be seen that changes or gradually evolves with each phoneme in the words.

stationary, it would be similar to computing a power spectrum of several added stationary signals. Ignoring the transitions between sounds, there is still the matter of nonaligned or opposite phases that might cancel out energy from the power spectrum. The analogy tells us that parts of the human speech signal would be averaged and detailed information would be lost. Human speech can be assumed to be stationary for short 30-40 millisecond segments (Oppenheim and Schafer 2014) at the time. In Kaldi 25 milliseconds is used as the frame length (Povey et al. 2011). That corresponds to 200 samples with a sample rate of 8000 Hz.

Each frame is multiplied element wise with a window to remove discontinuities. The discontinuities come from the periodicity assumptions that the discrete Fourier transform (DFT) will use later. A window is in general highest at the center and goes towards zero at the edges, where the discontinuities have been introduced by segmenting (Oppenheim and Schafer 2014). This means that most of the signal is lost at the edges of the window. To avoid losing information more of the signal can be preserved by overlapping the frames. The Kaldi uses a 15 millisecond overlap corresponding to a 10 millisecond shift of each frame (Povey et al. 2011). With a sample rate of 8000 Hz the frames are shifted 80 samples. This is illustrated in figure 2.19. As an optional step the dc offset, which is mean amplitude shift from zero, is removed now. The samples s_m from the signal vector $\boldsymbol{s}_n$ that belonging to

the ith frame is written as

$$s_{i,n} = s_{vi+n}, \quad m = 0, 1, \ldots, N-1$$

where n is the sample number within each frame, $N = 200$ is the number samples per frame, and $v = 80$ is the frame shift in samples. The frame matrix $\boldsymbol{s}_i, m$ needs to have all the frames be the same length the signal is zero padded which is to say extended with zeros such that signal length M

$$M = N + v(I-1), \quad I \in \mathbb{Z}_{>0}$$

where I is the number of frames, with $i = 0, 1, \ldots, I-1$. It can found by

$$I = \max\left(1, \left\lceil \frac{M-N+v}{v} \right\rceil\right).$$

The time domain human speech signal has now been divided into discrete time step frames, with i denoting the time step.

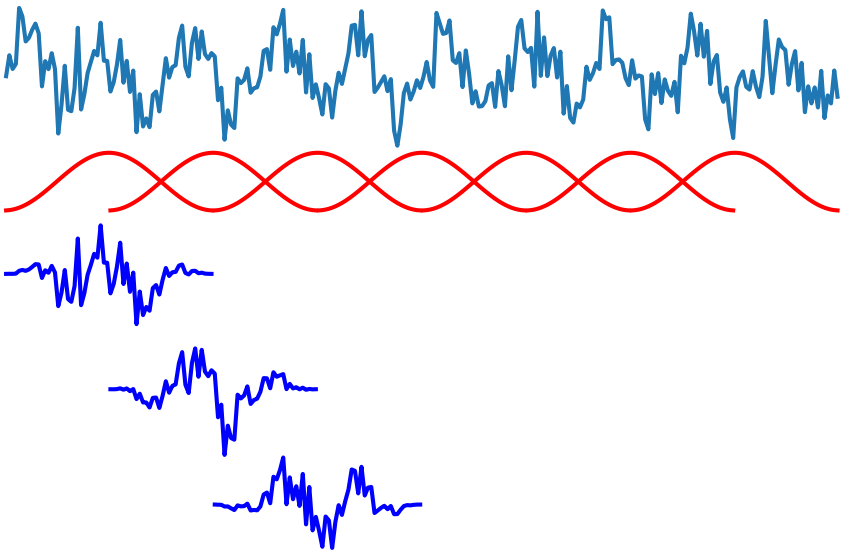


Figure 2.19: This figure illustrates windowing by plotting several windows and windowed segments called frames under the speech signal, with a shared time x-axis. Below the speech signal are several red windows overlapping by half. The next three signals are the frames, which is the result of multiplying each of the first 3 windows in turn with the speech signal. This figure is taken from the author's previously graded work.

differentite power spectrum desity as an estimate, and the power spectrum as the other, also named the periodegram

Kaldi uses the window referred to as Povey (Povey et al. 2011) which is

$$w_n = \left(0.5 - 0.5 * \cos\left(\frac{n2\pi}{N}\right)\right)^{0.85}, \quad n = 0, 1, \ldots, N-1$$

where $N = 200$ is the window length and equal to the frame length. The window is multiplied element wise with each frame. The next step is calculate the power spectral density by applying the discrete Fourier transform defined as

Definition 2.1 (Discrete Fourier Transform)
The discrete Fourier transform (DFT) of the signal vector $\boldsymbol{x}_n$, of length N, is defined as

$$\text{DFT}(\boldsymbol{x}_n) = \boldsymbol{X}_k = \sum_{n=0}^{N-1} x_n e^{-j\frac{2\pi}{N}kn}, \quad k = 0, 1, \dots N-1,$$

where $j^2 = -1$ and $\boldsymbol{X}_k$ is a vector the DFT coefficients (Oppenheim and Schafer 2014).

When the DFT is applied separately to each windowed frame the result is the short-time Fourier transform. Then the time-frequency domain is formed with the frames being a function of time and the DFT coefficients being a function of frequency. The DFT coefficients are complex numbers that can be split into a magnitude and a phase. The phase is discarded and only the magnitude that has information about the energy in the time-frequency is kept. For every frame i the one sided power spectral density is calculated as

$$\boldsymbol{P}_{i,k} = \frac{2}{N}|\text{DFT}(\boldsymbol{s}_{i,n} \circ \boldsymbol{w}_n)|^2, \quad i = 0, 1, \dots, I-1$$

where $\circ$ is the element wise multiplication operator and only first half of the symmetric power spectrum is kept (Kay 2006). The result is the Spectrogram $\boldsymbol{P}_{i,k}$, a $(I \times N/2 + 1)$ matrix which can be seen in figure 2.18.

2.2.6 Mel Filter Bank

The mel filter bank smooths the frequency envelope of each frame and compresses resolution of the high frequencies. The assumption is that the high frequencies of human speech signals only contain low resolution information. This could be justified by the fact that the resolution of the human ear is less sensitive at high frequencies (O'shaughnessy 1987). That a humans ability to differentiate or notice two tones with separate frequency is worse at higher frequencies. The mel scale is a function of frequency as defined by (Povey et al. 2011) is

$$\text{Mel}(f) = 1125 \log\left(1 + \frac{f}{700}\right) \tag{2.4}$$

where f is the frequency in Hz and the inverse is

$$\text{Mel}^{-1}(u) = 700\left(\exp\left(\frac{u}{1125}\right) - 1\right)$$

where u is the mel. There is more than one mel scale because it is an approximation of experimental data. The mel scale $\text{Mel}(f)$ in equation (2.4) is plotted in figure 2.20. The mel filter bank is constructed by triangle filters with the corners being uniformly separated in the mel scale. When converted back to Hz they will have a nonlinear separation distance. Kaldi uses 23 triangle filter (Povey et al. 2011), which is close to a 4.4 times reduction of the frequency resolution. The triangle filter uses the center points of the adjacent triangles filters as corners, which is to say they overlap by half. The outer edges of the triangle filters are chosen to be 20 Hz and 3700 Hz. The center points including the outer edges are

$$l_t = \{\text{Mel}(20) + t\Delta | t = 0, 1, T+2\}, \quad \Delta = \frac{\text{Mel}(3700) - \text{Mel}(20)}{T+1}$$
$$c_t = \{\text{Mel}^{-1}(c_t) | t = 0, 1, T+2\}$$

where the points l_t is in mel and uniformly separated. c_t is in Hz. The center points including the outer edges can also be seen in figure 2.20. The triangle filters magnitude frequency response of the mel filter bank is defined as

$$H_t(f) = \begin{cases} 0 & \text{if } f \leq c_{t-1} \\ \frac{f - c_{t-1}}{c_t - c_{t-1}} & \text{if } c_{t-1} \leq f \leq c_{t-1} \\ \frac{c_{t+1} - f}{c_{t+1} - c_t} & \text{if } c_t \leq f \leq c_{t+1} \\ 0 & \text{if } f \geq c_{t+1} \end{cases}, \quad t = 1, 2, \ldots, T \tag{2.5}$$

note that their magnitude frequency response sum to one between them, such that the energy is preserved.

These triangle filters are defined in continuous frequency and needs to be sampled to match the N frequency bins of the spectrogram.

$$\left\{ f_s \frac{k}{N} \middle| k = 0, 1, \frac{N}{2} \right\}$$

where N is even, when odd the endpoint of k will $(N-1)/2$. The center points can be rounded to the nearest frequency bin to have those bins have a magnitude frequency response of one, and the rest have rational numbers. The magnitude frequency response of the filter can be seen in figure 2.21. The sampled mel filters bank is

$$\boldsymbol{H}_{k,t} = H_t\left(f_s \frac{k}{N}\right), \quad k = 1, 2, \ldots, \frac{N}{2}, \quad t = 1, 2, \ldots, T$$

and it is a $(N/2 + 1 \times T)$ matrix with each row consisting of magnitude and the frequency response of a triangle filter. The filter bank features can be calculated by matrix multiplication

$$\boldsymbol{E}_{i,t} = \boldsymbol{P}_{i,k} \boldsymbol{H}_{k,t}$$

where $\boldsymbol{E}_{i,t}$ is a $(I \times T)$ matrix composed of the rows of frames containing the corresponding filter bank features. The sparseness of the triangle filter bank from

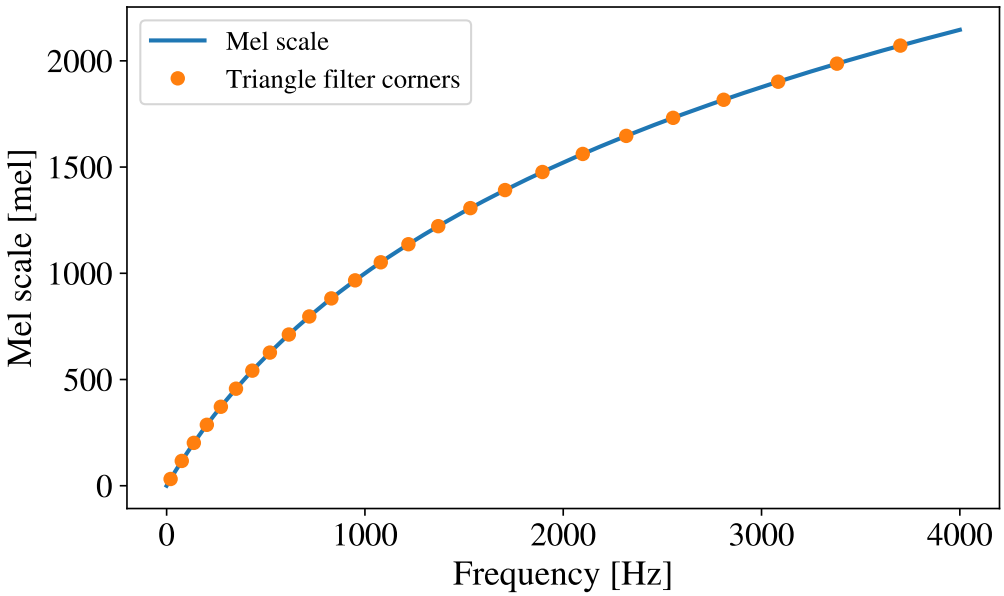


Figure 2.20: This is the mel scale as defined in equation (2.4). The orange circles represent the edges of the magnitude frequency response of the triangle mel filter bank. The triangle filters are each of the three consecutive points. The triangle filters are uniformly placed in the mel scale, but not in Hz.

equation (2.5) have not been utilized to save computation by matrix multiplication. The filter bank features seen in figure 2.22 correspond to the spectrogram in figure 2.18.

latex newcommand move caption up on all figure and clip some white space

2.2.7 Discrete Cosine Transform

The last step of the MFCC, the discrete cosine transform (DCT) that further reduces the dimensionality which can help due to the curse of dimensionality. The curse of dimensionality has two sides. The first is that more features require more data to fit properly. The second is that the complexity in big O notation of many machine learning algorithms is worse than linear. Growing computational power is reducing this restriction. The DCT also produces decorrelated or whitened features from the more correlated mel power spectrum. The differences can be seen in figures illustrating the mel filter bank features 2.22 and the MFCC features 2.23. The DCT is defined

Definition 2.2 (Discrete cosine Transform)
The discrete cosine transform (DCT) of the real signal vector $\boldsymbol{x}_n$, of length N, is defined as

$$\mathrm{DCT}(\boldsymbol{x}_n) = \boldsymbol{C}_k = \sum_{n=0}^{N-1} x_n \cos\left(\frac{\pi}{N}\left(n + \frac{1}{2}k\right)\right) \quad k = 0, 1, \ldots, N-1$$

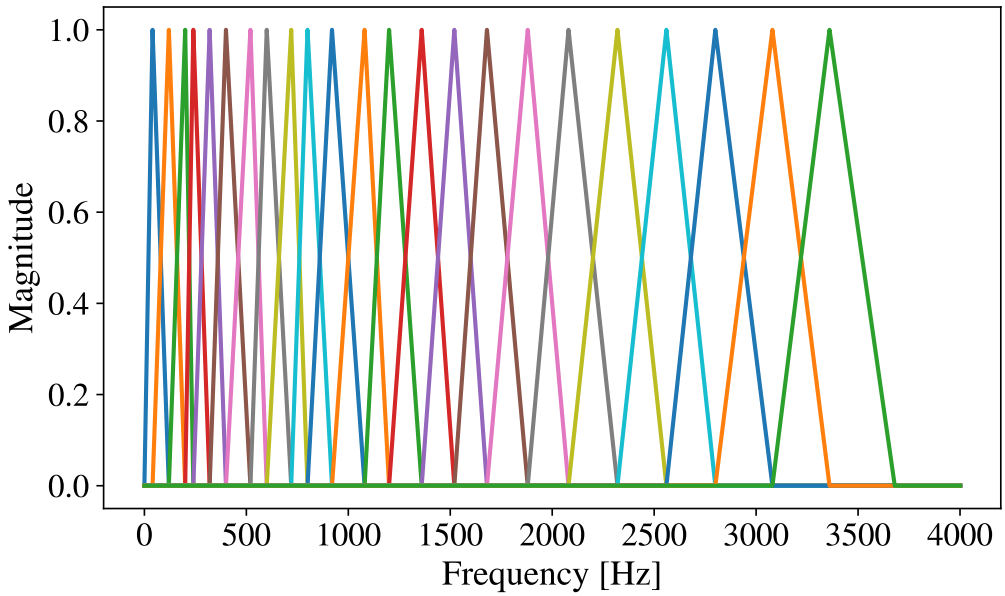


Figure 2.21: This is the mel filters used for the mel filterbank calculation. They are sparse and only smooth the power spectrum locally. The center points have been rounded to the nearest spectrogram frequency bin.

where $\boldsymbol{C}_k$ is a vector the DCT coefficients.

The DCT defined here is known as DCT-II. The DFT assumes the signal to be periodic, but the DCT-II assumes the boundary to be even around $n = -\frac{1}{2}$ and $n = N - \frac{1}{2}$. The assumption holds true for the mel power spectrum, but not the power spectrum as it is even around the zeroth coefficient corresponding to 0 Hz.

The DCT is applied to log of every row of $\boldsymbol{E}_{i,t}$ to get the DCT coefficient for every frame. The dimensionality is reduced by keeping the first q coefficient which in this case is 20. The first q coefficients are kept and the remaining $T - q$ coefficients corresponding to fast changes in the mel power spectrum are discarded, as to further use the smoothness assumption of the power spectrum and only represent its general shape. The MFCC features can be seen in 2.23. There is no longer a clear visible pattern as in mel filter bank features in figure 2.22.

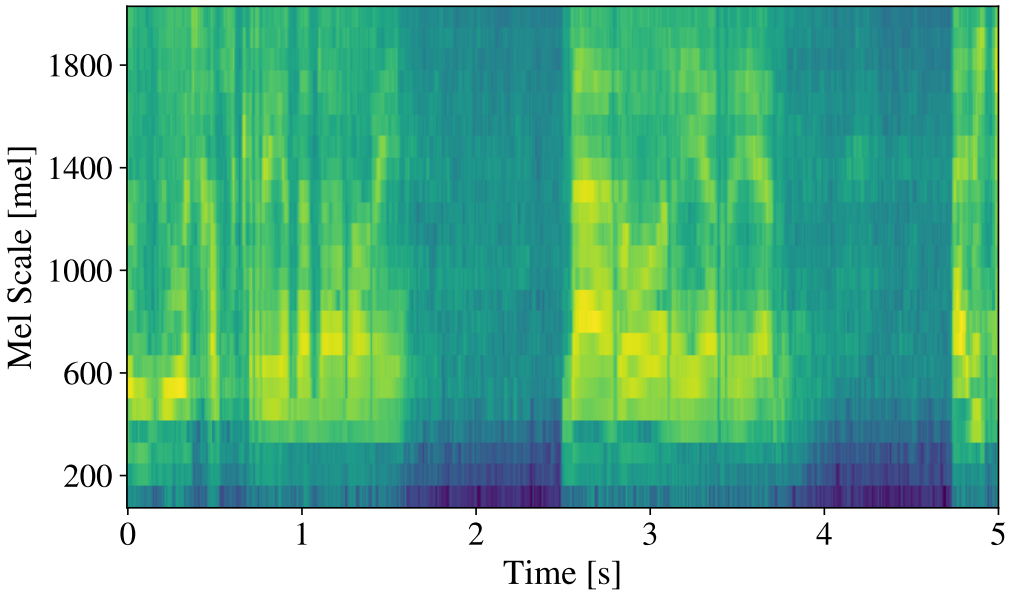


Figure 2.22: This figure illustrates the filter bank features of a 5 second speech segment. It is the same speech segment as seen in the time domain in figure 2.2. Time is still on the x-axis, but the y-axis now has the frequency in mel. There are 23 filter bank coefficients per frame and they are plotted in a log scale. A pattern can be seen that changes or gradually evolves with each phoneme in the words, similar to the spectrogram in figure 2.18.

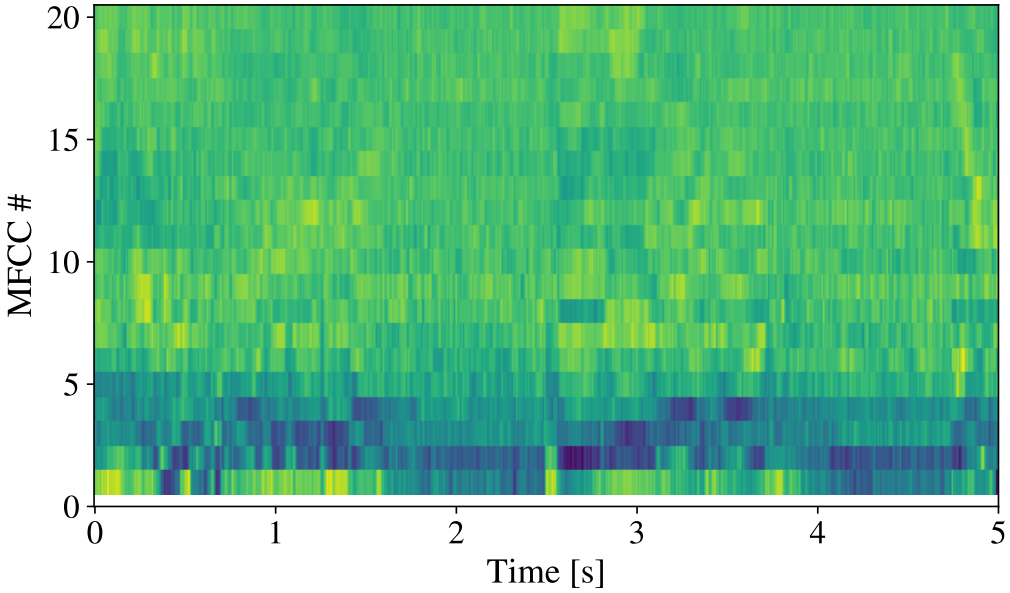


Figure 2.23: This figure illustrates the MFCC features of a 5 second speech segment. It is the same speech segment as seen in the time domain in figure 2.2. Time is still on the x-axis, but the y-axis now has MFCC. There are 20 MFCC per frame. The first coefficient has been omitted because of it large scale.

2.3 Bottleneck Features

The purpose of the bottleneck features block is to use the modeling abilities of deep neural networks with a large time context to discover and exploit structures in data to make information dense features. This is done by forcing it to move information though a bottleneck layer. The way to succeed at this is by encoding as much information as possible into this layer and decoding it on the other side.

Phonetic discriminant bottleneck features (BNF) is obtained from the MFCCs. This means that the model is trained to classify phonemes or rather senone acoustic units, which is phonemes and the many possible transitions from one phoneme to another. After the bottleneck layer the network has to decode all the information it can from it to classify the senone acoustic units. The classification task is not the important part. It is just an simple label that serves as a need to discover and encode structures in the data to predict the label. The idea is similar to the auto encoder concept, where the labels are the input and a bottleneck layer forces the discovery of structures in the data that can be used to encode the bottleneck layer. To calculate the bottleneck features give the MFCC sequence as input and extract the activation of the bottleneck layer yielding the sequence of BNFs.

The bottleneck network was trained on 1800 hours of Fisher English using Kaldi (Povey et al. 2011). The network consisted of 7 hidden layers, the 6th layer was an 80 dimensional linear bottleneck layer; the rest were Time delay neural network (TDNN) (Waibel et al. 1990) layers with p-norm activations with input/output dimension equal to 3500/350. The output layer was a softmax that classifies 5577 senone acoustic units. Short-term mean and variance normalization was applied with a 3-second sliding window and silent frames were removed.

TDNN

2.4 i-Vectors

remove we The i-vector paradigm (Dehak et al. 2011) transforms the sequence of BNFs into a fixed-dimensional embedding. Before this step the sequences would vary depending on their length. As an example the video audio all have the original length of their corresponding videos. The fixed-dimensional embedding is useful as it allows the use of standard classification algorithms.

Each speech segment is modeled by a Gaussian mixture model (GMM) whose super-vector mean $\mathbf{M}$ is assumed to be

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

where $\mathbf{m}$ is the GMM-UBM mean super-vector, $\mathbf{T}$ is a low-rank matrix and $\mathbf{w}$ is a standard normal distributed vector. $\mathbf{M}$ defines the total variability space, i.e. the directions in which the UBM can move to adapt it to a specific segment. The GMM-UBM represents the speaker-independent distribution of feature vectors. The *maximum a posteriori* (MAP) point estimate of $\mathbf{w}$ is the i-vector embedding. The GMM-UBM uses 2048 clusters with full covariance and the i-vector embed-

ding has a length of 600 (Richardson et al. 2018). The i-vector extractor is trained in Kaldi (Povey et al. 2011)

2.5 Gaussian Back-End (GBE) with Domain Adaptation

A linear Gaussian classifier was used to compute the language log-likelihood scores from the i-vectors. This back-end models each class with a Gaussian where the within-class covariance matrix is shared across languages. The weight of each language was equalized in the covariance estimation. There are 14 languages in LRE17.

For domain adaptation, the *a priori* back-end means and covariances were computed on out-domain data and applied *Maximum a posteriori* (MAP) adaptation using in-domain data. The adaptation equations for the Gaussian classifier are

$$\boldsymbol{\mu}_l = \alpha_l \boldsymbol{\mu}_{\mathrm{ML}_l} + (1-\alpha_l)\boldsymbol{\mu}_{0_l} \qquad l = 1, \ldots, L \tag{2.7}$$

$$\begin{aligned}\mathbf{S}_{\mathbf{W}} = \frac{1}{L}\sum_{l=1}^{L} \big[&\beta_l \mathbf{S}_{\mathrm{ML}_l} + (1-\beta_l)\mathbf{S}_0 \\ &+\beta_l(1-\alpha_l)\left(\boldsymbol{\mu}_{\mathrm{ML}_l} - \boldsymbol{\mu}_{0_l}\right)\left(\boldsymbol{\mu}_{\mathrm{ML}_l} - \boldsymbol{\mu}_{0_l}\right)^{\mathrm{T}}\big]\end{aligned} \tag{2.8}$$

where

$$\alpha_l = \frac{N_l}{N_l + r_{\boldsymbol{\mu}}} \qquad \beta_l = \frac{N_l}{N_l + r_{\mathbf{W}}}\ ; \tag{2.9}$$

L is the number of languages, N_l is the number of samples of language l; $\boldsymbol{\mu}_{0_l}$ and $\mathbf{S}_0$ are the prior means and covariance; $\boldsymbol{\mu}_{\mathrm{ML}_l}$ and $\mathbf{S}_{\mathrm{ML}_l}$ are the maximum likelihood means and covariances for language l computed on the in-domain data; and $r_{\boldsymbol{\mu}}$ and $r_{\mathbf{W}}$ are the relevance factors.

2.6 Calibration

Finally, a linear calibration function was applied to convert the Gaussian back-end scores into well-calibrated log-likelihoods. The calibration function had a language dependent bias and a common scaling parameter, and was trained using multi-class logistic regression.

Chapter 3
Speech Enhancement System

3.1 Speech Enhancement system

3.1.1 Speech enhancement system evaluation

To verify if the SE model itself works, it should be evaluated with a listening test to fully evaluate the performance. However, to quickly and cheaply evaluate development work, a number of objective algorithms are used instead. These algorithms are designed to emulate human evaluation of SE, with a higher score being better. The first is perceptual evaluation of speech quality (PESQ) which is meant to emulate human evaluation of the pleasantness of listening to the speech audio (Rix et al. 2001; ITU-T 2005). The PESQ score is defined in the interval $[-.5, 4.5]$. Another is the short-time objective intelligibility measure (STOI) (Taal et al. 2011) and the extended STOI (eSTOI) (Jensen and Taal 2016) meant to emulate human word comprehension, i.e. a human word error rate if you will. They are defined in the interval $[0, 1]$. Compared with the above measures, signal-to-distortion ratio (SDR) it aims to evaluate the audio source separation quality, but it is still used as a speech enhancement measure by regarding enhanced data and subtracted noise data as sources (Vincent, Gribonval, and Févotte 2006), which is defined in the interval $(-\infty, \infty)$. The enhancement algorithms in this project are evaluated with these measures by comparing their enhanced signals to the original uncorrupted signals. The need for uncorrupted signals restricts this evaluation form to simulated data.

3.1.2 Speech enhancement dataset

This section describes our speech enhancement dataset, which is generated for the purpose of speech enhancement experiments on the LRE17 task. The corruption of a speech signal can be seen as two types: additive and convolutional. Additive noise is typically independent of background noise, whereas convolutional noise can come from reverberation in rooms, and will be correlated with the speech signal. In this study only additive noise is considered, where the adopted signal model for the noisy speech signal y as

$$y(t) = s(t) + n(t) \tag{3.1}$$

where s is the speech signal and n is the noise signal.

In the dataset, noisy speech signals are created for each SNR level of {-3, 0, 3, 6, 9, 12, 15} dB equally. Simple voice activation detection is used to account for silence regions in speech signals, when calculating the energy. The training and validation datasets have no overlap and are split into 90 and 10 percents, respectively. The speech signals are taken from the LRE17 training set consisting of 2069 hours of telephone conversations. They are all sampled at 8 kHz with a mix of precision encodings. The noise signals come from the audio signals in the LRE17 development video domain. Most of these audios except for the talk shows contain noisy speech segments. Examples of background noise are babble, television, clapping, laughing, kitchen work and wind. The dataset also includes signals with reverberation which are left as is. Speech segments in these signals have been manually marked as speech intervals. A noise signal is a concatenation of all non-speech intervals in a noisy speech signal. The concatenation is performed with 128 samples of overlap and using a Hanning window of length of 256 samples. Noise intervals less than 125 milliseconds of length are discarded. This results in 6.6 hours of noise signals, which are expected to be closer to the noise sources in the target domain. Note that these noise signals potentially contain background speech since some recordings are annotated with segments of dominant speakers, and the aforementioned approach unintentionally includes speech segments of non-dominant speakers as noises. The noise signals are repeated to create 2069 hours of speech and noisy speech signal pairs, which are then cut into 5 second long segments.

The input feature for our BLSTM speech enhancement system is now explained. First, the noisy speech signal in the time domain is transformed using short time Fourier transform (STFT) into a time-frequency domain spectrogram. It uses a modified Hanning window w of length of 256 samples and an overlap/step of 128 samples.

$$w[k] = \frac{1}{2} + \frac{1}{2}\cos\left(2\pi\frac{\left(k - \frac{K-1}{2}\right)}{K}\right),\ k = 0, 1, \ldots, K-1 \tag{3.2}$$

After STFT, the 100-bin log Mel filterbank coefficients are extracted. Finally, the filterbank coefficients are normalized using the global mean and variance computed over the training samples. With these input features, the BLSTM model outputs the mask for each time-frequency bin, which is then multiplied by the original noisy speech magnitude spectrogram to get the enhanced magnitude spectrogram as an approximation of the uncorrupted speech. The time domain signal of the enhanced speech can be synthesized by using the inverse STFT, where the phase is taken from the original noisy speech spectrogram.

3.1.3 Model and training

The BLSTM-based model architecture is adopted for speech enhancement. BLSTM recurrent neural networks offer an elegant way to incorporate context information, instead of explicitly choosing the context based on feed-forward neural networks. The baseline BLSTM has 2 layers with 384 hidden units with an additional fully connected layer to transform a concatenation of the bi-directional output of 768

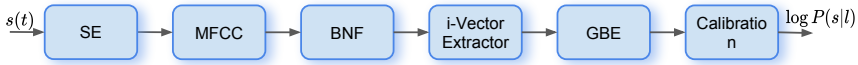


Figure 3.1: Proposed i-Vector language recognition system with single-channel enhancement.

units to 129 frequency bins for each time step. A sigmoid activation function is applied to constrain the mask to the interval from 0 to 1. By following the previous work of (Erdogan et al. 2015), the magnitude time-frequency approximation is used instead of a mask approximation for the objective function. First, consider the following distance function $D(\cdot)$:

$$D(\hat{a} \circ |Y| - a \circ |Y|) \tag{3.3}$$

where a is the ideal mask, $\circ$ is element-wise multiplication, $\hat{a}$ is the approximated mask obtained by BLSTM, and $|Y|$ is the magnitude time-frequency representation of the noisy speech. For the sake of simplicity, the time-frequency index is omitted in the formulation. Several masks have been proposed and an overview can be found in (Erdogan et al. 2015). The SE system uses the ideal amplitude mask a_{iam}

$$a_{\text{iam}} = \frac{|S|}{|Y|} \tag{3.4}$$

where $|S|$ is the magnitude time-frequency representation of the uncorrupted speech. The justification is that the language recognition system uses the magnitude only, and does not consider the phase. Equation (3.3) reduces to

$$D(\hat{a} \circ |Y| - a_{\text{iam}} \circ |Y|) = D(\hat{a} \circ |Y| - |S|). \tag{3.5}$$

With this representation, the mean squared error (MSE) based objective function is represented as:

$$\underset{\theta \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{M} \sum_{m=0}^{M-1} (\hat{a} \circ |Y| - |S|)^2 \tag{3.6}$$

with M being the number of total samples in a minibatch, n being the number of BLSTM parameters, and θ is the BLSTM parameter space. Adam is used as a stochastic minimizer. The model is implemented in the PyTorch framework.

3.2 Experimental setup

3.2.1 NIST LRE17 dataset

The approach is evaluated on the NIST language recognition evaluation 2017 (LRE17) task (*NIST 2017 Language Recognition Evaluation Plan* 2017a). The LRE17 task consists of closed set language identification between 14 languages from 5 language clusters (Arabic, English, Slavic, Iberian and Chinese).

The focus was on the fixed condition where the organizers constrained the datasets allowed for system development. NIST provided a training set (TRN17)

consisting of narrow-band telephony speech built from previous NIST evaluations (around 2000h). Switchboard and Fisher English telephony corpora was also allowed for training. Additionally, NIST provided a development set (DEV17) containing around 60 hours of speech from a domain similar to the evaluation set. Both, development and evaluation sets contain audios from two sources: narrow-band telephony and broadcast radio (MLS14); and wide-band video (VAST). MLS14 audio files consisted of segments of 3, 10 and 30 seconds while VAST audio files contained the full duration of the original source video file.

Language recognition systems were requested to provide a vector of calibrated log-likelihoods, one for each target language. Performance was measured using a detection cost function which is a weighted average of miss and false alarm rates.

$$C(\gamma) = \frac{1}{L}\sum_{i=0}^{L}\left[P_{\text{Miss}}(i,\gamma) + \frac{\gamma}{L-1}\sum_{j\neq i}P_{\text{FA}}(i,j,\gamma)\right] \tag{3.7}$$

where $\gamma = (1 - P_{\mathcal{T}})/P_{\mathcal{T}}$, $P_{\mathcal{T}}$ is the target language prior, and L the number of languages. $P_{\text{Miss}}(i,\gamma)$ is the miss rate for language i and $P_{\text{FA}}(i,j,\gamma)$ is the probability of detecting language i in an audio containing language j. Miss and false alarms are computed by applying detection thresholds $\log(\gamma)$ to the language log-likelihood ratios (derived from the calibrated log-likelihoods). The primary metric averages (3.7) for two operating points, $P_{\mathcal{T}} = 0.5$ and $P_{\mathcal{T}} = 0.1$. Also, the counts of each corpus (MLS14 and VAST) are equalized when computing the cost function so both have the same weight in the metric.

3.2.2 Experiments

The baseline is the language recognition system described in Section 2. The considered systems are with Gaussian back-end non-adapted to the LRE17 development set; adapted to the full development set (condition independent); and adapted to the specific domain (condition dependent), i.e., different adapted model for MLS14 and VAST. The condition independent and dependent score calibration is also considered. The development and evaluation data is processed with the OM-LSA and BLSTM SE methods. Thus, speech enhancement was included in the back-end adaptation and calibration steps. The training and validation loss is shown in figure 3.2 in Hyperparameter optimization was done for the BLSTM model but there as no change in the training or validation loss, when doubling the neuron, or adding an extra layer, or changing the batch size, or changing the learning rate. A sweep was done the language recognition from the epochs 25 to 150 in figure 3.3, where each epoch corresponds all the noise being used once which is 6 hours.

3.3 Results

3.3.1 Speech quality measures

Table 3.1 shows the SE performance with four performance measures (PESQ, STOI, eSTOI, and SDR), as introduced in Section 3.1.1. The performance of OM-LSA

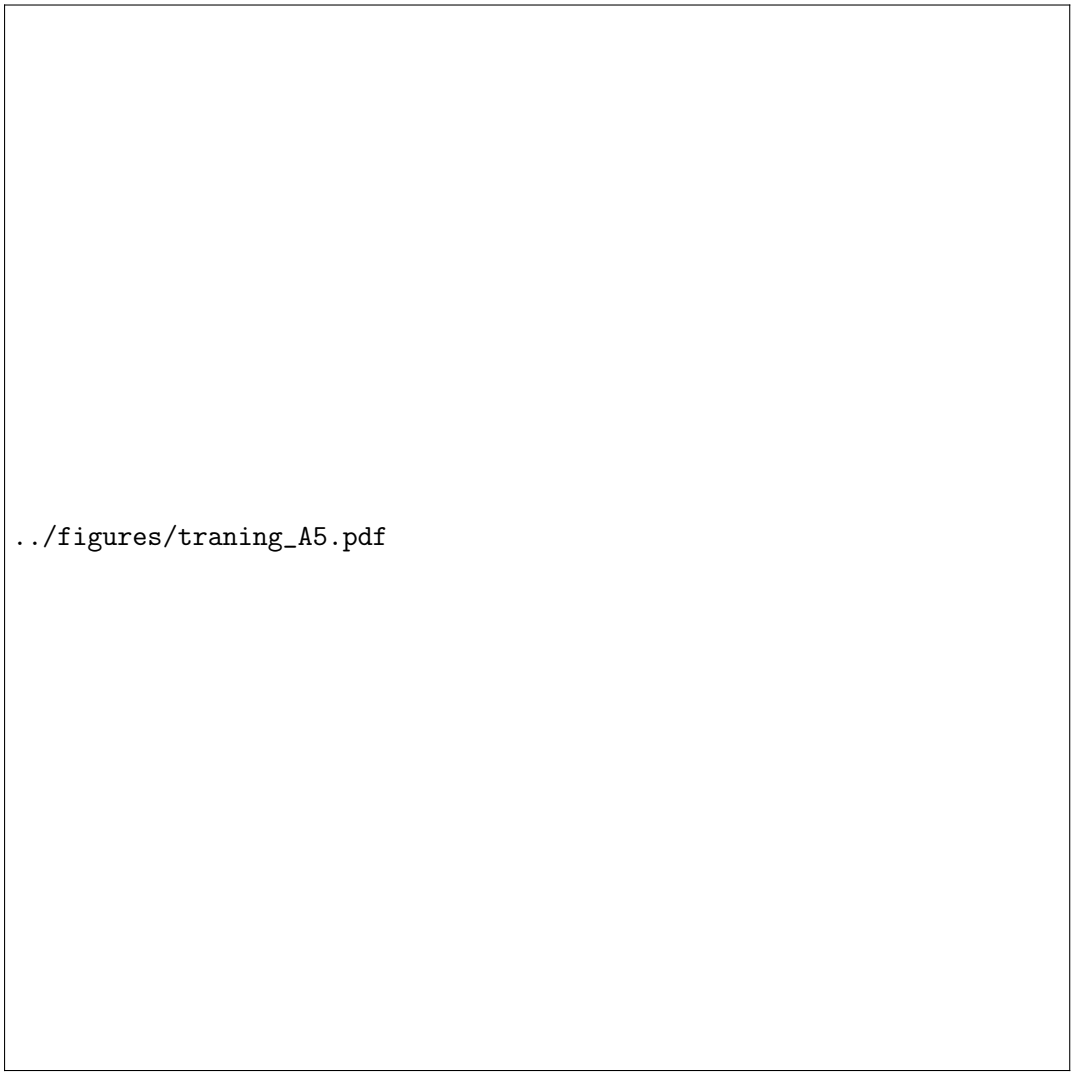


Figure 3.2: This is a plot of the training and validation loss of the BLSTM model.

was slightly degraded on the PESQ, STOI, eSTOI scores, but improved on the SDR score. This is because OM-LSA tends to remove noise components overly, which would affect the speech quality and intelligibility, especially for the high SNR setting. On the other hand, the BLSTM SE system outperformed OM-LSA for all measures consistently in both high and low SNR settings.

3.3.2 Language recognition

Table 3.2 presents language recognition in terms of the detection cost as defined in Section 3.2.1. The OM-LSA method improved the performance from the baseline in most of the cases. Meanwhile, the proposed BLSTM improved the performance in all the adaptation conditions, outperforming OM-LSA. For the MLS14 case, the BLSTM performance was degraded in some cases, but not significantly. For the VAST (noisy video) case, the improvement was very significant in all conditions.

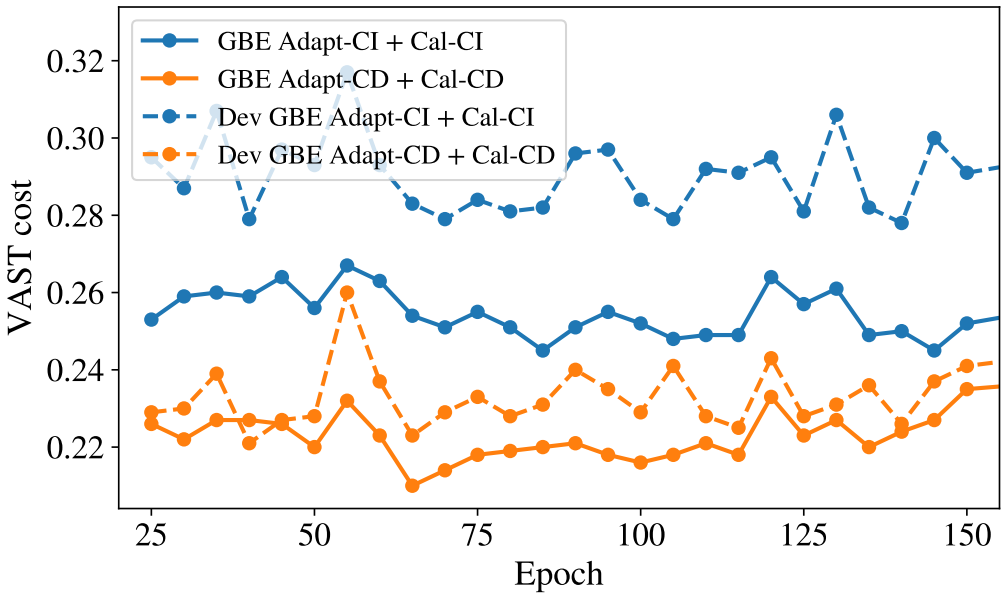


Figure 3.3: This illustrates a sweep of language recognition evaluations.

The best language recognizer, including condition dependent back-end and calibration, achieved 11.3% relative improvement when using our BLSTM SE. In average of the MLS14 and VAST cases, the relative improvement of BLSTM SE was around 6.3%, which is still significant.

Another thing worth mentioning is that, with applying SE, the gap between condition-dependent and condition-independent back-end systems was reduced. This property is quite useful in a real application, since it can be avoided to use a complicated condition-dependent system, which requires to have multiple domain-dependent models with a precise domain detector.

Table 3.1: Result for the speech quality (PESQ), speech intelligibility (STOI, eSTOI), and audio source separation (SDR) for the simulated validation set. The values should be compared relative to the reference values. Higher is better for all speech enhancement measures.

System	PESQ	STOI	eSTOI	SDR
All SNRs:				
Reference	2.456	0.733	0.565	4.395
OM-LSA	2.379	0.708	0.546	6.502
BLSTM	**2.815**	**0.793**	**0.634**	**12.333**
15 dB SNR:				
Reference	3.042	0.875	0.761	13.507
OM-LSA	2.895	0.844	0.730	13.249
BLSTM	**3.305**	**0.895**	**0.801**	**18.670**
-3 dB SNR:				
Reference	1.895	0.568	0.362	-4.626
OM-LSA	1.809	0.541	0.346	-1.722
BLSTM	**2.291**	**0.665**	**0.440**	**5.517**

Table 3.2: Results for the addition of a preprocessing speech enhancement step, for different language recognition systems. Consider systems with three types of back-end non-adapted to the development data, condition independent adapted (CI) and condition dependent adapted (CD); and two calibrations, condition independent and dependent. The values are from equation (3.7), where lower is better and the MLS14 and VAST display the result for the telephone and video audio respectively. The baseline is without SE

System	Baseline	OM-LSA	BLSTM
Cost average:			
GBE Non-adapt + Cal-CI	0.306	0.289	**0.269**
GBE Non-adapt + Cal-CD	0.292	0.277	**0.265**
GBE Adapt-CI + Cal-CI	0.234	0.238	**0.207**
GBE Adapt-CI + Cal-CD	0.221	0.227	**0.199**
GBE Adapt-CD + Cal-CI	0.219	0.235	**0.209**
GBE Adapt-CD + Cal-CD	0.206	0.218	**0.193**
MLS14:			
GBE Non-adapt + Cal-CI	0.198	0.218	**0.193**
GBE Non-adapt + Cal-CD	0.193	0.213	**0.192**
GBE Adapt-CI + Cal-CI	**0.165**	0.185	**0.165**
GBE Adapt-CI + Cal-CD	**0.162**	0.183	0.164
GBE Adapt-CD + Cal-CI	**0.168**	0.188	0.169
GBE Adapt-CD + Cal-CD	**0.164**	0.185	0.166
VAST:			
GBE Non-adapt + Cal-CI	0.414	0.360	**0.346**
GBE Non-adapt + Cal-CD	0.391	0.340	**0.337**
GBE Adapt-CI + Cal-CI	0.304	0.291	**0.249**
GBE Adapt-CI + Cal-CD	0.280	0.270	**0.235**
GBE Adapt-CD + Cal-CI	0.270	0.282	**0.249**
GBE Adapt-CD + Cal-CD	0.248	0.252	**0.220**

Chapter 4
Conclusion

We proposed a BLSTM speech enhancement technique to improve language recognition in a noisy signal condition. The BLSTM is trained to estimate a time-frequency mask indicating the quality of each frequency bin. Using this mask, we obtained an enhanced version of the signal spectrogram, and recover the time domain waveform. We evaluated the quality of the enhanced signals in the recent NIST 2017 language recognition evaluation, where there is a condition with noisy audio from Internet videos. We compared results using the proposed method and baseline OM-LSA; also adapting the language recognition system to the target domain and non-adapting. In the noisy condition, we obtained performance gains around 16% for the case without adaptation and around 11% for the case where we performed condition dependent adaptation of the recognizer. Performance in clean conditions was not degraded. Also, speech enhancement contributed to reduce the gap between condition dependent and independent recognizers, which could greatly simplify the systems.

As future work, adding external and more realistic noise databases like CHiME-4 (Vincent et al. 2017), and Musan (Snyder, Chen, and Povey 2015), it would be possible to investigate the importance of in domain versus out of domain noise data. And examine the relationship between the noise database size and performance. Are there diminishing returns with addition of more noise data? And is a large out of domain noise source beneficial in addition to the in domain noise source?

There are several straight forward improvements to the project such as adding reverberational noise, to handle the few noisy speech signals with considerable reverberations in the LRE17 challenge. Performing the speech enhancement in wide-band speech, and then down sampling to 8000 Hz could improve the speech enhancement on the 16000 Hz video audio data. Having the language recognition system operate in 16000 Hz would probably be beneficial but the language data is not available to train it. There might also be newer and better speech enhancement models and approaches to consider, such as mixture networks, generative adversarial networks, and convolution networks (Chazan, Goldberger, and Gannot 2017; Pascual, Bonafonte, and Serra 2017; Zhao et al. 2018). Adding a data augmentation baseline using the same noise sources to compare with the speech enhancement preprocessing step, would give valuable comparison of the two approaches.

With a realistic noise database the in domain noise extraction method could be examined, by simulating noisy speech and then extracting noise signals from it. Thus by comparing a system trained using the original noise source with one using the extracted noise source, it could reveal the effect of having a noise source contaminated with speech and the concatenation effect. To examine the concatenation effect alone the pure noise corresponding to the extracted segment from the noisy speech could be used to make the noise source instead.

The idea of utilizing in domain noisy speech in training speech enhancement could be explored further. The current approach is to simulate speech and noisy speech pairs to use supervised learning. l speech and noisy speech pairs could be used instead of simulation, but it requires a costlier special setup. The simulation requires a noise source, that can be in domain or out of domain. The in domain noise source would be preferential, but it is not always available. Sometimes and in this case noisy speech is available. Instead of extracting noise from noisy speech, a domain translation approach could be used. The method cycle generative adversarial network or CycleGAN for short is introduced in the paper "Unpaired image-to-image translation using cycle-consistent adversarial networks" (Zhu et al. 2017). It can be used to learn domain transformation using unpaired data from two domains. In our case the domains are the noisy speech data and speech domain data. There would be no need to rely on a concatenated noise source that only extracted 6 hours of noise from 45 hours of noisy speech data. Instead the model could learn to remove noise present in all the 45 hours of noisy speech data.

An end to end model possibly using multi-task learning could simplify the training of the speech enhancement system, and would provide training and validation curves directly related to the language recognition task. Instead of only relying on the MSE of the magnitude spectrogram approximation. The end to end model could be used with a bottleneck layer, so the features could be integrated into an i-vector for any other system in general. The multi-task learning would hopefully adapt the speech enhancement to the language recognition task.

Finally as language recognition is similar to speaker recognition the positive result could possibly be duplicated with the same approach.

add overfit eksperiment to conclusion

Bibliography

Archana, G. S. and M. Malleswari (2015). "Gender identification and performance analysis of speech signals". In: *2015 Global Conference on Communication Technologies (GCCT)*, pp. 483–489. DOI: 10.1109/GCCT.2015.7342709.

Bahari, Mohamad Hasan et al. (2014). "Non-Negative Factor Analysis of Gaussian Mixture Model Weight Adaptation for Language and Dialect Recognition". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 22.7, pp. 1117–1129. ISSN: 2329-9290. DOI: 10.1109/TASLP.2014.2319159.

Chazan, Shlomo E, Jacob Goldberger, and Sharon Gannot (2017). "Speech Enhancement using a Deep Mixture of Experts". In: *arXiv preprint arXiv:1703.09302*.

Cohen, I. (2003). "Noise spectrum estimation in adverse environments: improved minima controlled recursive averaging". In: *IEEE Transactions on Speech and Audio Processing* 11.5, pp. 466–475. ISSN: 1063-6676. DOI: 10.1109/TSA.2003.811544.

Cohen, Israel and Baruch Berdugo (2001). "Speech enhancement for non-stationary noise environments". In: *Signal Processing* 81.11, pp. 2403 –2418. ISSN: 0165-1684. DOI: https://doi.org/10.1016/S0165-1684(01)00128-1.

Dehak, Najim et al. (2011). "Front-End Factor Analysis For Speaker Verification". In: *IEEE Transactions on Audio, Speech and Language Processing* 19.4, pp. 788 –798. DOI: 10.1109/TASL.2010.2064307.

Deller Jr., J. R., J. H. L. Hansen, and J. G. Proakis (2000). *Discrete-Time Processing of Speech Signals*. IEEE Press.

Erdogan, H. et al. (2015). "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks". In: *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 708–712. DOI: 10.1109/ICASSP.2015.7178061.

Garofolo, J. et al. (1993). *TIMIT Acoustic-Phonetic Continuous Speech Corpus LDC93S1.* Web Download. Philadelphia: Linguistic Data Consortium.

ITU-T (2005). *PESQ, P.862.2*. URL: https://www.itu.int/rec/T-REC-P.862-200511-I!Amd2/en (visited on 03/23/2018).

Jensen, J. and C. H. Taal (2016). "An Algorithm for Predicting the Intelligibility of Speech Masked by Modulated Noise Maskers". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 24.11, pp. 2009–2022. ISSN: 2329-9290. DOI: 10.1109/TASLP.2016.2585878.

Kay, Steven (2006). *Intuitive probability and random processes using MATLAB®*. Springer Science & Business Media.

Kolbæk, Morten, Zheng-Hua Tan, and Jesper Jensen (2017). "Speech Intelligibility Potential of General and Specialized Deep Neural Network Based Speech Enhancement Systems". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 25.1, pp. 153–167.

Lu, Xugang et al. (2013). "Speech enhancement based on deep denoising autoencoder." In: *Interspeech*, pp. 436–440.

Michelsanti, Daniel and Zheng-Hua Tan (2017). "Conditional generative adversarial networks for speech enhancement and noise-robust speaker verification". In: *Proc. Interspeech*.

NIST 2017 Language Recognition Evaluation Plan (2017a). Tech. rep. NIST. URL: https://www.nist.gov/itl/iad/mig/nist-2017-language-recognition-evaluation.

NIST 2017 Language Recognition Evaluation Plan (2017b). Tech. rep. NIST. URL: https://www.nist.gov/sites/default/files/documents/2017/06/01/lre17_eval_plan-2017-05-31_v2.pdf.

Oppenheim, A. V. and R. W. Schafer (2014). *Discrete-Time Signal Processing*. Third. Pearson Education Limited.

O'shaughnessy, Douglas (1987). *Speech communication: human and machine*. Universities press.

Pascual, Santiago, Antonio Bonafonte, and Joan Serra (2017). "SEGAN: Speech enhancement generative adversarial network". In: *arXiv preprint arXiv:1703.09452*.

Pisoni, D. B. and R. E. Remez, eds. (2004). *The Handbook of Speech Perception*. Blackwell Reference Online. 10 April 2017 http://www.blackwellreference.com/subscriber/book.html?id=g9780631229278_9780631229278. Blackwell Publishing.

Povey, D. et al. (2011). "The Kaldi speech recognition toolkit". In: *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pp. 1–4. arXiv: arXiv:1011.1669v3.

Recommendation G.191 STL-2009 Manual (11/09) (2009). Tech. rep. ITU-T. URL: http://www.itu.int/rec/T-REC-G.191-200911-I/en.

Recommendation G.711 (11/88) (2009). Tech. rep. ITU-T. URL: http://www.itu.int/rec/T-REC-G.711-198811-I/en.

Richardson, Fred et al. (2018). "The MIT Lincoln Laboratory / JHU / EPITA-LSE LRE17 System". In: *submitted to Odyssey 2018*. Les Sables d'Olonne, France.

Rix, A. W. et al. (2001). "Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs". In: *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (ICASSP)*. Vol. 2, pp. 749–752. DOI: 10.1109/ICASSP.2001.941023.

Snyder, David, Guoguo Chen, and Daniel Povey (2015). "Musan: A music, speech, and noise corpus". In: *arXiv preprint arXiv:1510.08484*.

Taal, C. H. et al. (2011). "An Algorithm for Intelligibility Prediction of Time-Frequency Weighted Noisy Speech". In: *IEEE Transactions on Audio, Speech, and Language Processing* 19.7, pp. 2125–2136. ISSN: 1558-7916. DOI: 10.1109/TASL.2011.2114881.

Vincent, Emmanuel, Rémi Gribonval, and Cédric Févotte (2006). "Performance measurement in blind audio source separation". In: *IEEE Transactions on Audio, Speech, and Language Processing* 14.4, pp. 1462–1469.

Vincent, Emmanuel et al. (2017). "An analysis of environment, microphone and data simulation mismatches in robust speech recognition". In: *Computer Speech & Language* 46, pp. 535–557.

Waibel, Alexander et al. (1990). "Phoneme recognition using time-delay neural networks". In: pp. 393–404.

Wang, DeLiang et al. (2009). "Speech intelligibility in background noise with ideal binary time-frequency masking". In: *The Journal of the Acoustical Society of America* 125.4, pp. 2336–2347.

Wang, Yuxuan, Arun Narayanan, and DeLiang Wang (2014). "On training targets for supervised speech separation". In: *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)* 22.12, pp. 1849–1858.

Weninger, Felix et al. (2015). "Speech enhancement with LSTM recurrent neural networks and its application to noise-robust ASR". In: *International Conference on Latent Variable Analysis and Signal Separation*. Springer, pp. 91–99.

Xu, Y. et al. (2015). "A Regression Approach to Speech Enhancement Based on Deep Neural Networks". In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 23.1, pp. 7–19. ISSN: 2329-9290. DOI: 10.1109/TASLP.2014.2364452.

Zhao, Han et al. (2018). "Convolutional-Recurrent Neural Networks for Speech Enhancement". In: *arXiv preprint arXiv:1805.00579*.

Zhu, Jun-Yan et al. (2017). "Unpaired image-to-image translation using cycle-consistent adversarial networks". In: *arXiv preprint arXiv:1703.10593*.

Appendix A
Interspeech 2018 Paper

This appendix includes a paper version of the results that have been accepted to the Interspeech 2018 conference.

Effectiveness of Single-Channel BLSTM Enhancement for Language Identification

Peter Sibbern Frederiksen[1], *Jesús Villalba*[2], *Shinji Watanabe*[2], *Zheng-Hua Tan*[1], *Najim Dehak*[2]

[1]Department of Electronic Systems, Aalborg University, Denmark
[2]Center for Language and Speech Processing, Johns Hopkins University, Baltimore, MD

psf@ieee.org, jvillal7@jhu.edu, shinjiw@jhu.edu, zt@es.aau.dk, ndehak3@jhu.edu

Abstract

This paper proposes to apply deep neural network (DNN)-based single-channel speech enhancement (SE) to language identification. The 2017 language recognition evaluation (LRE17) introduced noisy audios from videos, in addition to the telephone conversation from past challenges. Because of that, adapting models from telephone speech to noisy speech from the video domain was required to obtain optimum performance. However, such adaptation requires knowledge of the audio domain and availability of in-domain data. Instead of adaptation, we propose to use a speech enhancement step to clean up the noisy audio as preprocessing for language identification. We used a bi-directional long short-term memory (BLSTM) neural network, which given log-Mel noisy features predicts a spectral mask indicating how clean each time-frequency bin is. The noisy spectrogram is multiplied by this predicted mask to obtain the enhanced magnitude spectrogram, and it is transformed back into the time domain by using the unaltered noisy speech phase. The experiments show significant improvement to language identification of noisy speech, for systems with and without domain adaptation, while preserving the identification performance in the telephone audio domain. In the best adapted state-of-the-art bottleneck i-vector system the relative improvement is 11.3% for noisy speech.

Index Terms: speech enhancement, BLSTM, language recognition, NIST LRE17.

1. Introduction

Language recognition refers to the process of automatically detecting the language spoken in a speech utterance. Its applications range across customized speech recognition, multilanguage translation, service customization and forensics [1]. The focus of research in the field has been on developing recognition methods to improve the performance of general systems, while little attention has been given to improving the noise-robustness of language recognition systems.

NIST Language recognition evaluations (LRE) has played an instrumental role in driving language recognition research over the years and LRE constantly increases the challenge level of its evaluations. The most recent LRE 2017 evaluation [2] presents a new scenario with a significant mismatch between training and evaluation data. The training dataset consists of a large amount of narrow-band telephone speech, which is in line with previous evaluations. However, the evaluation dataset consists of a combination of narrow-band telephone data and wide-band data from Internet videos. Furthermore, the LRE 2017 organizers provide a limited amount of in-domain development data for model adaptation and calibration purposes. While telephone speech contains low levels of noise and reverberation, we observed that the video data are severely degraded by babble noise, music and reverberation.

Single-channel speech enhancement (SE) can be used as preprocessing to mitigate the aforementioned degradation and reduce the mismatch between training and evaluation data. SE has been widely used as preprocessing for speech applications, such as automatic speech recognition (ASR) [3], speaker verification [4], mobile communications and hearing aids [5]. In this study we investigate the effectiveness of utilizing single-channel SE to improve the noise-robustness of a language recognition system.

It has been experimentally shown that applying ideal binary mask in the time-frequency domain is able to improve speech intelligibility of noisy speech signals for both normal hearing and hearing impaired listeners with various noise types [6]. Various ideal ratio masks have become preferable over ideal binary mask in recent studies [5, 7, 8]. In [9, 10] a DNN is trained to predict clean speech from noisy speech without the use of a mask by casting it as a regression problem. A long short-term memory (LSTM) network has shown to outperform feed-forward DNN methods, when used as preprocessing for noise robust ASR [3], and the bidirectional extension of LSTM (BLSTM) achieves further improvement [8]. This paper follows the success of the BLSTM SE method, and applies it to a language recognition system. The BLSTM SE is processed in the time-frequency domain, but only deals with the magnitude while the phase component remains corrupted, similar to the other DNN-based SE methods. The method internally predicts a mask from BLSTM, and the predicted mask is multiplied by the noisy speech magnitude, which yields the enhanced magnitude. The network is trained with the mean square error criterion between the clean and enhanced magnitudes. In BLSTM SE (and other DNN-based enhancement), only additive noise is considered, where the noise source is extracted from in-domain data with limited size in our setup. The effectiveness of BLSTM SE on the language identification is evaluated by a state-of-the-art bottleneck i-vector LRE system, where BLSTM SE is used as preprocessing of the LRE system [11].

To validate the effectiveness of BLSTM SE methods, we also compare our SE with the optimally-modified log-spectral amplitude (OM-LSA) speech estimator with the improved minima controlled recursive averaging (IMCRA) noise estimator [12], [13]. OM-LSA is a well-known signal processing method that does not require data-driven training and adaption stages.

2. Speech Enhancement system

2.1. Speech enhancement system evaluation

To verify if the SE model itself works, it should be evaluated with listening test to fully evaluate the performance. However,

to quickly and cheaply evaluate development work, a number of objective algorithms are used instead. These algorithms are designed to emulate human evaluation of SE, with a higher score being better. The first is perceptual evaluation of speech quality (PESQ) which is meant to emulate human evaluation of the pleasantness of listening to the speech audio [14, 15]. The PESQ score is defined in the interval $[-.5, 4.5]$. Another is the short-time objective intelligibility measure (STOI) [16] and the extended STOI (eSTOI) [17] meant to emulate human word comprehension, i.e. a human word error rate if you will. They are defined in the interval $[0, 1]$. Compared with the above measures, signal-to-distortion ratio (SDR) aims to evaluate the audio source separation quality, but it is still used as a speech enhancement measure by regarding enhanced data and subtracted noise data as sources [18], which is defined in the interval $(-\infty, \infty)$. The enhancement algorithms in this paper are evaluated with these measures by comparing their enhanced signals to the original uncorrupted signals. The need for uncorrupted signals restricts this evaluation form to simulated data.

2.2. Speech enhancement dataset

This section describes our speech enhancement dataset, which is generated for the purpose of speech enhancement experiments on the LRE17 task. The corruption of a speech signal can be seen as two types: additive and convolutional. Additive noise is typically independent of background noise, whereas convolutional noise can come from reverberation in rooms, and will be correlated with the speech signal. In this study we only consider additive noise, where we adopt the signal model for the noisy speech signal y as

$$y(t) = s(t) + n(t) \tag{1}$$

where s is the speech signal and n is the noise signal.

In the dataset, noisy speech signals are created for each SNR level of {-3, 0, 3, 6, 9, 12, 15} dB equally. Simple voice activation detection is used to account for silence regions in speech signals, when calculating the energy. The training and validation datasets have no overlap and are split into 90 and 10 percents, respectively. The speech signals are taken from the LRE17 training set consisting of 2069 hours of telephone conversations. They are all sampled with 8 kHz with a mix of precision encodings. The noise signals come from the audio signals in the LRE17 development video domain. Most of these audios except for the talk shows contain noisy speech segments. Examples of background noise are babble, television, clapping, laughing, kitchen work and wind. The dataset also includes signals with reverberation which are left as is. Speech segments in these signals have been manually marked as speech intervals. A noise signal is a concatenation of all non-speech intervals in a noisy speech signal. The concatenation is performed with 128 samples of overlap and using a Hanning window of length 256 samples. Noise intervals less than 125 milliseconds are discarded. This results in 6.6 hours of noise signals, which are expected to be closer to the noise sources in the target domain. Note that these noise signals potentially contain background speech since some recordings are annotated with segments of dominant speakers, and the aforementioned approach unintentionally includes speech segments of non-dominant speakers as noises. The noise signals are repeated to create 2069 hours of speech and noisy speech signal pairs, which are then cut into 5 seconds long segments.

Now we describe the input feature for our BLSTM speech enhancement system. First, the noisy speech signal in the time domain is transformed using short time Fourier transform (STFT) into a time-frequency domain spectrogram. We use a modified Hanning window w of length 256 samples and an overlap/step of 128 samples.

$$w[k] = \frac{1}{2} + \frac{1}{2}\cos\left(2\pi\frac{\left(k - \frac{K-1}{2}\right)}{K}\right),\ k = 0, 1, \ldots, K-1 \tag{2}$$

After STFT, we extract the 100-bin log Mel filterbank coefficients. Finally, the filterbank coefficients are normalized using the global mean and variance computed over the training samples. With these input features, the BLSTM model outputs the mask for each time-frequency bin, which is then multiplied by the original noisy speech magnitude spectrogram to get the enhanced magnitude spectrogram as an approximation of the uncorrupted speech. The time domain signal of the enhanced speech can be synthesized by using the inverse STFT, where the phase is taken from the original noisy speech spectrogram.

2.3. Model and training

We adopt BLSTM-based model architecture as speech enhancement. BLSTM recurrent neural networks offer an elegant way to incorporate context information, instead of explicitly choosing the context based on feed-forward neural networks. The baseline BLSTM has 2 layers with 384 hidden units with an additional fully connected layer to transform concatenation of the bi-directional output of 768 units to 129 frequency bins for each time step. A sigmoid activation function is applied to constrain the mask to the interval from 0 to 1. By following the previous work of [8], we consider magnitude time-frequency approximation instead of a mask approximation for the objective function. First, we consider the following distance function $D(\cdot)$:

$$D(\hat{a} \circ |Y| - a \circ |Y|) \tag{3}$$

where a is the ideal mask, $\circ$ is element-wise multiplication, $\hat{a}$ is the approximated mask obtained by BLSTM, and $|Y|$ is the magnitude time-frequency representation of the noisy speech. For the sake of simplicity, we omit the time-frequency index in the formulation. Several masks have been proposed and an overview can be found in [8]. The SE system uses the ideal amplitude mask a_{iam}

$$a_{\text{iam}} = \frac{|S|}{|Y|} \tag{4}$$

where $|S|$ is the magnitude time-frequency representation of the uncorrupted speech. Equation (3) reduces to

$$D(\hat{a} \circ |Y| - a_{\text{iam}} \circ |Y|) = D(\hat{a} \circ |Y| - |S|). \tag{5}$$

With this representation, the mean squared error (MSE) based objective function is represented as:

$$\underset{\theta \in \mathbb{R}^n}{\text{minimize}} \quad \frac{1}{M}\sum_{m=0}^{M-1}\left(\hat{a} \circ |Y| - |S|\right)^2 \tag{6}$$

with M being the number of total samples in a minibatch, n being the number of BLSTM parameters, and θ is the BLSTM parameter space. Adam is used as a stochastic minimizer. The model is implemented in the PyTorch framework.

3. Language recognition system

Figure 1 shows the pipeline of a state-of-the-art i-vector language recognition system with an additional speech enhancement step. Following, we explain each of the steps.

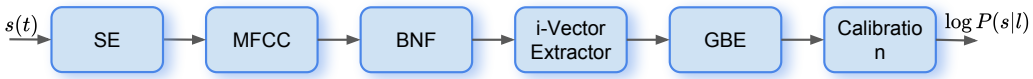


Figure 1: *Proposed i-Vector language recognition system with single-channel enhancement.*

3.1. Feature extraction

We computed 20 dimensional Mel-frequency cesptral coefficients (MFCCs) from the noisy/enhanced speech signal. From MFCCs, we obtained phonetic discriminant bottleneck features (BNF). The bottleneck network was trained on 1800 hours of Fisher English using Kaldi NNet2 [19]. The network consisted of 7 hidden layers, the 6th layer was an 80 dimensional linear bottleneck layer; the rest were TDNN layers with p-norm activations with input/output dimension equal to 3500/350. The output layer was a softmax that classifies 5577 senone acoustic units. Short-term mean and variance normalization was applied with 3 second sliding window and silence frames were removed.

3.2. i-Vectors

The i-vector paradigm [20] transforms the sequence of BNFs into a fixed-dimensional embedding. Each speech segment is modeled by a Gaussian mixture model (GMM) whose super-vector mean $\mathbf{M}$ is assumed to be

$$\mathbf{M}_s = \mathbf{m} + \mathbf{T}\mathbf{w}_s \tag{7}$$

where $\mathbf{m}$ is the GMM-UBM mean super-vector, $\mathbf{T}$ is a low-rank matrix and $\mathbf{w}$ is a standard normal distributed vector. $\mathbf{M}$ defines the total variability space, i.e. the directions in which we can move the UBM to adapt it to a specific segment. The GMM-UBM represents the speaker-independent distribution of feature vectors. The *maximum a posteriori* (MAP) point estimate of $\mathbf{w}$ is the i-vector embedding.

3.3. Gaussian back-end (GBE) with domain adaptation

We used a linear Gaussian classifier to compute the language log-likelihood scores from the i-vectors. This back-end models each class with a Gaussian where the within-class covariance matrix is shared across languages. We equalized the weight of each language in the covariance estimation.

For domain adaptation, we computed the *a priori* back-end means and covariances on out-domain data and applied *Maximum a posteriori* (MAP) adaptation using in-domain data. The adaptation equations for the Gaussian classifier are

$$\boldsymbol{\mu}_l = \alpha_l \boldsymbol{\mu}_{\mathrm{ML}_l} + (1-\alpha_l)\boldsymbol{\mu}_{0_l} \qquad l = 1, \ldots, L \tag{8}$$

$$\begin{aligned}\mathbf{S}_{\mathbf{W}} = \frac{1}{L}\sum_{l=1}^{L} \big[&\beta_l \mathbf{S}_{\mathrm{ML}_l} + (1-\beta_l)\mathbf{S}_0 \\ &+\beta_l(1-\alpha_l)\left(\boldsymbol{\mu}_{\mathrm{ML}_l} - \boldsymbol{\mu}_{0_l}\right)\left(\boldsymbol{\mu}_{\mathrm{ML}_l} - \boldsymbol{\mu}_{0_l}\right)^{\mathrm{T}}\big]\end{aligned} \tag{9}$$

where

$$\alpha_l = \frac{N_l}{N_l + r_{\boldsymbol{\mu}}} \qquad \beta_l = \frac{N_l}{N_l + r_{\mathbf{W}}}\ ; \tag{10}$$

L is the number of languages, N_l is the number of samples of language l; $\boldsymbol{\mu}_{0_l}$ and $\mathbf{S}_0$ are the prior means and covariance; $\boldsymbol{\mu}_{\mathrm{ML}_l}$ and $\mathbf{S}_{\mathrm{ML}_l}$ are the maximum likelihood means and covariances for language l computed on the in-domain data; and $r_{\boldsymbol{\mu}}$ and $r_{\mathbf{W}}$ are the relevance factors.

3.4. Calibration

Finally, we applied a linear calibration function to convert the Gaussian back-end scores into well-calibrated log-likelihoods. The calibration function had a language dependent bias and a common scaling parameter, and was trained using multi-class logistic regression.

4. Experimental setup

4.1. NIST LRE17 dataset

We evaluated our approach on the NIST language recognition evaluation 2017 (LRE17) task [2]. The LRE17 task consists of closed set language identification between 14 languages from 5 language clusters (Arabic, English, Slavic, Iberian and Chinese).

We focused on the fixed condition where the organizers constrained the datasets allowed for system development. NIST provided a training set (TRN17) consisting of narrow-band telephony speech built from previous NIST evaluations (around 2000h). Switchboard and Fisher English telephony corpora were also allowed for training. Additionally, NIST provided a development set (DEV17) containing around 60 hours of speech from a domain similar to the evaluation set. Both, development and evaluation sets contain audios from two sources: narrow-band telephony and broadcast radio (MLS14); and wide-band video (VAST). MLS14 audio files consisted of segments of 3, 10 and 30 seconds while VAST audio files contained the full duration of the original source video file.

Language recognition systems were requested to provide a vector of calibrated log-likelihoods, one for each target language. Performance was measured using a detection cost function which is a weighted average of miss and false alarm rates.

$$C(\gamma) = \frac{1}{L}\sum_{i=0}^{L}\left[P_{\mathrm{Miss}}(i,\gamma) + \frac{\gamma}{L-1}\sum_{j\neq i} P_{\mathrm{FA}}(i,j,\gamma)\right] \tag{11}$$

where $\gamma = (1-P_{\mathcal{T}})/P_{\mathcal{T}}$, $P_{\mathcal{T}}$ is the target language prior, and L the number of languages. $P_{\mathrm{Miss}}(i,\gamma)$ is the miss rate for language i and $P_{\mathrm{FA}}(i,j,\gamma)$ is the probability of detecting language i in an audio containing language j. Miss and false alarms are computed by applying detection thresholds $\log(\gamma)$ to the language log-likelihood ratios (derived from the calibrated log-likelihoods). The primary metric averages (11) for two operating points, $P_{\mathcal{T}} = 0.5$ and $P_{\mathcal{T}} = 0.1$. Also, the counts of each corpus (MLS14 and VAST) are equalized when computing the cost function so both have the same weight in the metric.

4.2. Experiments

The baseline is the language recognition system described in Section 3. We considered systems with Gaussian back-end non-adapted to the LRE17 development set; adapted to the full development set (condition independent); and adapted to the specific domain (condition dependent), i.e., different adapted

Table 1: *Result for the speech quality (PESQ), speech intelligibility (STOI, eSTOI), and audio source separation (SDR) for the simulated validation set. The values should be compared relative to the reference values. Higher is better for all speech enhancement measures.*

System	PESQ	STOI	eSTOI	SDR
All SNRs:				
Reference	2.456	0.733	0.565	4.395
OM-LSA	2.379	0.708	0.546	6.502
BLSTM	**2.815**	**0.793**	**0.634**	**12.333**
15 dB SNR:				
Reference	3.042	0.875	0.761	13.507
OM-LSA	2.895	0.844	0.730	13.249
BLSTM	**3.305**	**0.895**	**0.801**	**18.670**
-3 dB SNR:				
Reference	1.895	0.568	0.362	−4.626
OM-LSA	1.809	0.541	0.346	−1.722
BLSTM	**2.291**	**0.665**	**0.440**	**5.517**

model for MLS14 and VAST. We also considered condition independent and dependent score calibration. We processed the development and evaluation data with the OM-LSA and BLSTM SE methods. Thus, speech enhancement was included in the back-end adaptation and calibration steps.

5. Results

5.1. Speech quality measures

Table 1 shows the SE performance with four performance measures (PESQ, STOI, eSTOI, and SDR), as introduced in Section 2.1. The performance of OM-LSA was slightly degraded on the PESQ, STOI, eSTOI scores, but improved on the SDR score. This is because OM-LSA tends to remove noise components overly, which would affect the speech quality and intelligibility, especially for the high SNR setting. On the other hand, the BLSTM SE system outperformed OM-LSA for all measures consistently in both high and low SNR settings.

5.2. Language recognition

Table 2 presents language recognition in terms of the detection cost as defined in Section 4.1. The OM-LSA method improved the performance from the baseline in most of the cases. Meanwhile, the proposed BLSTM improved the performance in all the adaptation conditions, outperforming OM-LSA. For the MLS14 case, the BLSTM performance was degraded in some cases, but not significantly. For the VAST (noisy video) case, the improvement was very significant in all conditions. The best language recognizer, including condition dependent back-end and calibration, achieved 11.3% relative improvement when using our BLSTM SE. In average of the MLS14 and VAST cases, the relative improvement of BLSTM SE was around 6.3%, which is still significant.

Another thing worth mentioning is that, with applying SE, the gap between condition-dependent and condition-independent back-end systems was reduced. This property is quite useful in a real application, since we can avoid to use a complicated condition-dependent system, which requires to have multiple domain-dependent models with a precise domain detector.

Table 2: *Results for the addition of a preprocessing speech enhancement step, for different language recognition systems. We consider systems with three types of back-end non-adapted to the development data, condition independent adapted (CI) and condition dependent adapted (CD); and two calibrations, condition independent and dependent. The values are from equation (11), where lower is better and the MLS14 and VAST display the result for the telephone and video audio respectively.*

System	Baseline	OM-LSA	BLSTM
Cost average:			
GBE Non-adapt + Cal-CI	0.306	0.289	**0.269**
GBE Non-adapt + Cal-CD	0.292	0.277	**0.265**
GBE Adapt-CI + Cal-CI	0.234	0.238	**0.207**
GBE Adapt-CI + Cal-CD	0.221	0.227	**0.199**
GBE Adapt-CD + Cal-CI	0.219	0.235	**0.209**
GBE Adapt-CD + Cal-CD	0.206	0.218	**0.193**
MLS14:			
GBE Non-adapt + Cal-CI	0.198	0.218	**0.193**
GBE Non-adapt + Cal-CD	0.193	0.213	**0.192**
GBE Adapt-CI + Cal-CI	**0.165**	0.185	**0.165**
GBE Adapt-CI + Cal-CD	**0.162**	0.183	0.164
GBE Adapt-CD + Cal-CI	**0.168**	0.188	0.169
GBE Adapt-CD + Cal-CD	**0.164**	0.185	0.166
VAST:			
GBE Non-adapt + Cal-CI	0.414	0.360	**0.346**
GBE Non-adapt + Cal-CD	0.391	0.340	**0.337**
GBE Adapt-CI + Cal-CI	0.304	0.291	**0.249**
GBE Adapt-CI + Cal-CD	0.280	0.270	**0.235**
GBE Adapt-CD + Cal-CI	0.270	0.282	**0.249**
GBE Adapt-CD + Cal-CD	0.248	0.252	**0.220**

6. Conclusions

We proposed a BLSTM speech enhancement technique to improve language recognition in a noisy signal condition. The BLSTM is trained to estimate a time-frequency mask indicating the quality of each frequency bin. Using this mask, we obtain an enhanced version of the signal spectrogram, and recover the time domain waveform. We evaluated the quality of the enhanced signals in the recent NIST 2017 language recognition evaluation, where there is a condition with noisy audio from Internet videos. We compared results using the proposed method and baseline OM-LSA; also adapting the language recognition system to the target domain and non-adapting. In the noisy condition, we obtained performance gains around 16% for the case without adaptation and around 11% for the case where we performed condition dependent adaptation of the recognizer. Performance in clean conditions was not degraded. Also, speech enhancement contributed to reduce the gap between condition dependent and independent recognizers, which could greatly simplify the systems.

As future work, we plan to use more realistic noise databases like CHiME-4 [21], and Musan [22]. Additionally, reverberation could be simulated as well to reduce the noise mismatch further. Also, we want to perform speech enhancement in wide-band speech, instead of downsampling to 8 kHz, which should improve the language recognition performance on videos.

7. References

[1] M. H. Bahari, N. Dehak, H. Van hamme, L. Burget, A. M. Ali, and J. R. Glass, "Non-Negative Factor Analysis of Gaussian Mixture Model Weight Adaptation for Language and Dialect Recognition," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 7, pp. 1117–1129, jul 2014.

[2] "NIST 2017 Language Recognition Evaluation Plan," NIST, Tech. Rep., 2017.

[3] F. Weninger, H. Erdogan, S. Watanabe, E. Vincent, J. Le Roux, J. R. Hershey, and B. Schuller, "Speech enhancement with lstm recurrent neural networks and its application to noise-robust asr," in *International Conference on Latent Variable Analysis and Signal Separation*. Springer, 2015, pp. 91–99.

[4] D. Michelsanti and Z.-H. Tan, "Conditional generative adversarial networks for speech enhancement and noise-robust speaker verification," *Proc. Interspeech*, 2017.

[5] M. Kolbæk, Z.-H. Tan, and J. Jensen, "Speech intelligibility potential of general and specialized deep neural network based speech enhancement systems," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 25, no. 1, pp. 153–167, 2017.

[6] D. Wang, U. Kjems, M. S. Pedersen, J. B. Boldt, and T. Lunner, "Speech intelligibility in background noise with ideal binary time-frequency masking," *The Journal of the Acoustical Society of America*, vol. 125, no. 4, pp. 2336–2347, 2009.

[7] Y. Wang, A. Narayanan, and D. Wang, "On training targets for supervised speech separation," *IEEE/ACM Transactions on Audio, Speech and Language Processing (TASLP)*, vol. 22, no. 12, pp. 1849–1858, 2014.

[8] H. Erdogan, J. R. Hershey, S. Watanabe, and J. L. Roux, "Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks," in *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, April 2015, pp. 708–712.

[9] X. Lu, Y. Tsao, S. Matsuda, and C. Hori, "Speech enhancement based on deep denoising autoencoder." in *Interspeech*, 2013, pp. 436–440.

[10] Y. Xu, J. Du, L. R. Dai, and C. H. Lee, "A regression approach to speech enhancement based on deep neural networks," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 1, pp. 7–19, Jan 2015.

[11] F. Richardson, P. A. Torres-carrasquillo, J. Borgstrom, D. Sturim, Y. Gwon, J. Villalba, N. Chen, J. Trmal, and N. Dehak, "The MIT Lincoln Laboratory / JHU / EPITA-LSE LRE17 System," in *submitted to Odyssey 2018*, Les Sables d'Olonne, France, jun 2018.

[12] I. Cohen and B. Berdugo, "Speech enhancement for non-stationary noise environments," *Signal Processing*, vol. 81, no. 11, pp. 2403 – 2418, 2001.

[13] I. Cohen, "Noise spectrum estimation in adverse environments: improved minima controlled recursive averaging," *IEEE Transactions on Speech and Audio Processing*, vol. 11, no. 5, pp. 466–475, Sept 2003.

[14] A. W. Rix, J. G. Beerends, M. P. Hollier, and A. P. Hekstra, "Perceptual evaluation of speech quality (pesq)-a new method for speech quality assessment of telephone networks and codecs," in *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings (ICASSP)*, vol. 2, 2001, pp. 749–752.

[15] ITU-T. (2005) Pesq, p.862.2. [Online]. Available: https://www.itu.int/rec/T-REC-P.862-200511-I!Amd2/en

[16] C. H. Taal, R. C. Hendriks, R. Heusdens, and J. Jensen, "An algorithm for intelligibility prediction of time-frequency weighted noisy speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 7, pp. 2125–2136, Sept 2011.

[17] J. Jensen and C. H. Taal, "An algorithm for predicting the intelligibility of speech masked by modulated noise maskers," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 24, no. 11, pp. 2009–2022, Nov 2016.

[18] E. Vincent, R. Gribonval, and C. Févotte, "Performance measurement in blind audio source separation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 4, pp. 1462–1469, 2006.

[19] 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 *Proceedings of the IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, dec 2011, pp. 1–4.

[20] N. Dehak, P. Kenny, R. Dehak, P. Dumouchel, and P. Ouellet, "Front-End Factor Analysis For Speaker Verification," *IEEE Transactions on Audio, Speech and Language Processing*, vol. 19, no. 4, pp. 788 – 798, may 2011.

[21] E. Vincent, S. Watanabe, A. A. Nugraha, J. Barker, and R. Marxer, "An analysis of environment, microphone and data simulation mismatches in robust speech recognition," *Computer Speech & Language*, vol. 46, pp. 535–557, 2017.

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