Deep Learning Approaches for Automatic Sung Speech Recognition

Adapting Spoken Technologies to Sung Speech

Gerardo Roa Dabike

Supervisor: Professor Jon Barker

Department of Computer Science
University of Sheffield

This thesis is submitted for the degree of
Doctor of Philosophy

Speech and Hearing Research Group May 2022

I want to dedicate this thesis to my wife Monserrat, and children Tomas and Catalina, who have been by my side throughout this journey.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements. The length of this thesis including footnotes, appendices and references is approximately 52000 words. This thesis contains 57 figures and 37 tables.

Gerardo Roa Dabike
May 2022

Acknowledgements

First and foremost, I would like to express my utmost gratitude to my supervisor, Professor Jon Barker, for his mentorship, guidance, feedback and support throughout this great 4-year journey. Secondly, I would like to thank Professor Guy Brown and Dr Mark Hepple for all their insightful comments in our PhD panel meetings. The biggest thanks go to my wife who helped me with the illustrations in Chapter 2 and gave me her unconditional support and encouragement. Last but not least, I thank my children, who were next to me, especially on the hard times.

Abstract

Automatic sung speech recognition is a challenging problem that remains largely unsolved. Challenges are due to both the intrinsic poor intelligibility of sung speech and the difficulty of separating the vocals from the musical accompaniment. In recent years, deep neural network techniques have revolutionised spoken speech recognition systems through advances in both acoustic modelling and audio source separation.

This thesis evaluates whether these new techniques can be adapted to work for sung speech recognition. For this, it first presents an analysis of the differences between spoken and sung speech. Then motivated by this analysis, the thesis makes four major contributions.

First, the thesis addresses the lack of large, standardised sung speech datasets suitable for evaluating sung speech recognition. The opportunity for building a suitable dataset has recently arisen with the release of Smule's DAMP-MVP dataset, a large unaccompanied karaoke performance dataset. However, constructing a well-balanced and easy-to-use evaluation dataset from this weakly-labelled and weakly-annotated data presents many challenges. This thesis presents solutions to these challenges.

Second, the thesis reconsiders the problem of sung speech acoustic modelling. New musically-motivated features are considered to capture the importance of the vocal source information. Features considered include pitch, voicing degree, voice quality, and beat-based features. It is shown that pitch and voicing degree features are useful for improving recognition performances.

Third, accompanied sung speech recognition poses a challenging source separation problem. This thesis investigates the use of modern time-domain source separation networks. Also, it investigates whether 'speaker embedding' ideas can be employed for music source separation by considering the use of 'instrument' embeddings.

Finally, a complete system that combines the deep neural network based source separation and speech recognition components are jointly evaluated, dealing with the mismatch between the distorted sung speech originated from the separation network and the 'clean' sung speech used for acoustic modelling.

Table of contents

List of figures

List of tables

Nomenclature

Acronyms / Abbreviations

AC	acoustic model
aLDA	acoustic latent Dirichlet allocation
ASR	automatic speech recognition
CMLLR	constrained MLLR
DCT	discrete cosine transform
DFT	discrete Fourier transform
DNN	deep neural network
FPE	fine pitch error
FRC	functional residual capacity
GMM	Gaussian mixture model
GPE	gross pitch error
GP	glimpse proportion
HMM	hidden Markov model
HNR	harmonic to noise ratio
IPA	International Phonetic Alphabet
LM	language model
LM-MMI	lattice-free maximum mutual information

LPC linear prediction coefficients

LSTM long short-term memory

MAE mean absolute error

MAP maximum a posteriori

MFCC Mel-frequency cepstral coefficients

MLLR maximum likelihood linear regression

MLP multilayer perceptron

MSE mean square error

NCCF normalised cross-correlation function

PER phone error rate

PESQ perceptual evaluation of speech quality

PLP perceptual linear prediction

POV probability of voicing

PP perplexity

P_S subglottic pressure

RELU rectified linear unit function

RMS root mean square

RNN recurrent neural network

RV residual volume

SAR signal-to-artifact ratio

SDR signal-to-distortion ration

SIR signal-to-interference ratio

SI-SDR scale-invariant SDR

SNR signal-to-noise ratio

SPL sound pressure level

SPS syllables per second

STOI short-time objective intelligibility

SVM support vector machines

TCN temporal convolutional network

TDDN-F factorised TDNN

TDNN time delay neural network

TRAP temporal patterns

TVL total lung volume

tVSA triangular VSA

VAD voice activity detection

VC vital capacity

VQ voice quality

VSA vowel space area

WER word error rate

WFST weighted finite-state transducer

WLPC warped linear predictive coefficients

Chapter 1

Introduction

1.1 Motivation

Solo-singer sung speech recognition (also referred to as "lyrics transcription"), the task of automatically transcribing the lyrics of a song, is a very challenging problem that has remained largely unsolved. When singing (sung speech), the intelligibility of the speech is often of secondary importance compared to the overall artistic impression. This may mean that speech sounds (i.e., phonemes) become distorted or reduced in ways that make them intrinsically less intelligible. The recognition problem then becomes even more challenging when the singing is accompanied by an instrumental background. In fact, even humans can find it difficult to understand the lyrics of some popular songs correctly. Mishearing lyrics are common enough to justify the creation of websites to report common mishearing[1], and comedians like Peter Kay have sketches based on mishearings[2]. And yet, human sung speech recognition remains far more robust than the current state-of-the-art automatic systems.

There are many scenarios where a robust sung speech recognition system might be useful. Applications include, but are not limited to, music retrieval systems, content labelling, automatic lyrics alignment and hearing impaired lyrics reading:

- Sung speech recognition is important for lyric-based music retrieval. For example, it can be used to form a transcription of the tracks that are being searched. Alternatively, in a voice-based query system, it can be used to turn the sung query into text. Similar retrieval applications have existed in the spoken speech domain for many years (i.e., 'spoken document retrieval') and interestingly were shown to be usable even when recognition word error rates are high (Abberley et al., 1998).

[1] https://www.kissthisguy.com/

[2] https://www.youtube.com/watch?v=7my5baoCVv8

- Automatic sung speech recognition (ASR) systems could automatically detect 'explicit content' in lyrics (Vaglio et al., 2020). Online music services like Deezer[3] and YouTube Music[4] may constantly need to classify new content in their catalogue as containing explicit content so they can alert parents about lyrics potentially unsuitable for children.

- Sing-along options are increasingly demanded in music services like Apple Music and Amazon Music. This has been made more evident with the recent introduction of lyrics by Spotify[5] in November 2021. However, the current sing-along option from some services is not provided for all the songs, and songs containing lyrics may present the lyrics phrases at erroneous timings. ASR technologies could automatically transcribe the lyrics and aid the correct alignment of the lyrics presented to the user.

- Hearing-impaired people with cochlear implants tend to report lower singing understanding than people with normal hearing (McDermott, 2004), which reduces their music enjoyment (Fuller et al., 2013). Cochlear implant users may benefit from automatically generated lyrics when listening to music, helping them understand the song message and, in the process, improving their speech perception (Fuller et al., 2018).

Beyond purely music application-orientated motivations, the challenge of sung speech recognition is interesting in its own right. The fact that sung speech is intrinsically less intelligible makes sung speech recognition a powerful test of the performance of speech acoustic modelling. Examining the gap between human and machine performance, and trying to close it, is a possible route to building better models of speech acoustics. This will have importance beyond sung speech. For example, better acoustic modelling is needed for processing other atypical speech sources (e.g., handling dysarthric speech, stuttered speech and other speech pathologies), for dealing with other hypo-articulated speech instances (casual speech, mumbling, etc), and for building models that can be robustly adapted in data-scarce scenarios (e.g., building recognisers for low resourced languages).

With the deep learning revolution, spoken ASR system have experienced significant improvements. In fact, every state-of-the-art ASR system for spoken tasks uses deep learning technologies. This is the case, for example, for the LibriSpeech audiobooks (Panayotov et al., 2015), for which the lowest recognition error rate of 1.8% was achieved by employing a very sophisticated deep neural network (DNN) architecture (Hsu et al., 2021). However, the advances in deep learning have not been applied to sung speech recognition. Recent research

[3]https://deezer.io/detecting-explicit-content-in-songs-274967de7fd1

[4]https://support.google.com/youtube/answer/174084

[5]https://newsroom.spotify.com/2021-11-18/you-can-now-find-the-lyrics-to-your-favorite-songs- in-spotify-heres-how/

has only employed simple multi-layer perceptron networks with one hidden layer reporting high phoneme recognition errors of 80% (Kruspe, Anna Marie, 2018).

The lack of research in sung speech recognition is partly due to sung speech applications being perceived as of less immediate importance. This has led to little funding being available for research in the area compared with tasks such as telephone speech, meeting transcription and broadcast media transcription. This, in turn, has meant that the necessary sung speech resources have not become available (compared to the resources available for spoken tasks like the Switchboard corpus (LDC97S62), the AMI meeting corpus (McCowan et al., 2005) and the MGB Challenge (Bell et al., 2015)). Previous research on sung speech recognition has resorted to using large amounts of spoken data for acoustic modelling and small sets of private singing data for adaptation (Kawai et al., 2017; Kruspe, 2016b; Mesaros and Virtanen, 2010a; Tsai et al., 2018), reporting poor performances.

In this thesis, we will deal with the challenge of the accompanied sung speech recognition task by dividing the problem into sub-tasks that separately deal with audio source separation, unaccompanied sung speech recognition and the integration of both. This is done by employing spoken technologies and adapting them to singing acoustic conditions, such as DNN source separation architecture and traditional hybrid DNN/ hidden Markov model (HMM) ASR systems.

1.2 Research Questions and Goals

This thesis addresses the challenge of recognising the lyrics from a single channel musical signal. The approach to the problem will involve analysing technologies designed for spoken speech and adapting them by exploiting sung speech properties. For this, the problem is divided into three more constrained tasks. The first part focuses on the challenge of unaccompanied solo-singer sung speech recognition. This stage investigates acoustic modelling using unaccompanied sung speech data and state-of-the-art DNN-HMM acoustic models for spoken speech trained on standard spectral features. The second stage addresses the source separation problem. The objective is to separate the singing segment from the musical accompaniment preserving the necessary information needed for the ASR task. The third and final stage evaluates the ASR performance from a combined system using the previous stages' source separation and ASR models. A survey of the existing sung speech datasets that can be adapted for the task is carried out for each stage.

In order to design an effective sung speech recognition system, a number of research questions will have to be addressed at each stage of the work. These are outlined below.

RQ.1 Sung and spoken speech share several characteristics, in that they both originate from the same speech production mechanisms, and they can be semantically similar. However, they have several acoustic differences resulting in lower sung speech intelligibility. **What are the differences in the speech production mechanisms between sung and spoken speech, and how are these differences reflected in the acoustic signal?**

RQ.2 Sung speech is typically found as part of a musical composition where all components are designed to work together in a complementary way. This means that the masking effect of the instrumentation is quite different to background noise masking in the well-studied speech plus *noise* situation. **What is the masking effect of the musical accompaniment, and how does this masking impact the intelligibility of the sung speech signal?**

RQ.3 Modern spoken speech recognition and separation systems need large corpora that broadly and deeply sample a range of acoustic conditions to train robust and generalisable models. This kind of readily available data is scarce in the case of sung speech scenarios. **What datasets exist that can be shaped for modelling sung speech recognition and vocal source separation?**

RQ.4 Modern speech recognition systems achieve incredibly good results on spoken speech in isolation (i.e., without background noise) because they are highly adapted to the nature of the spoken speech signal. **How can we best re-adapt spoken speech ASR systems to meet the demands of unaccompanied sung speech recognition? In particular, how much benefit can be gained by re-considering the features on which the acoustic models are trained to capture the musical properties of the signal better?**

RQ.5 Source separation techniques have been shown to be useful as the 'front-ends' to noise-robust spoken speech recognition. These techniques often rely on the speech and the noise being independent sources in these applications. However, the source independence assumption is no longer valid in musical mixtures. Therefore, **how can spoken source separation approaches be adapted to exploit the musical signal's constraints better? In particular, can recent speaker embedding approaches be used to characterise and filter out instrument-specific musical accompaniment?**

RQ.6 In **RQ.4** we deal with the unaccompanied sung speech recognition problem, and in **RQ.5** we deal with the vocal separation problem that can serve as a front-end for the sung speech recognition systems. However, speech separation algorithms that remove the accompaniment are imperfect, i.e., they will leave residual noise

and distort the speech signal leading to errors in the subsequent speech recognition stage. **How severely does the separation distortion impact the speech recognition performance, and to what extent can the recognition systems be adapted to accommodate distortion?**

1.3 Contributions

Review study of the sung and spoken speech differences

The study of the differences between sung and spoken speech production presented in Chapter 2 is a comprehensive study that can inform both music and speech research. There are several applications where music research can benefit from spoken technologies, such as language identification, text-to-audio alignment and speech recognition. However, these applications may not be easily applied to music due to the differences between speech styles. This study provides some insights that may help reduce the distance between these two research fields so one field can benefit from the other.

DSing dataset: Sung speech corpus for lyrics recognition

The thesis addresses the lack of large, standardised solo-singer sung speech datasets suitable for evaluating unaccompanied sung speech recognition. The opportunity for building such a dataset has recently arisen with the release of the Digital Archive of Mobile Performances - Smule Multilingual Vocal Performance 300x30x2 dataset (DAMP-MVP) (Smule, Inc., 2018), a large unaccompanied karaoke performances dataset. DAMP-MVP provides singing recording and the prompt timings of when a lyrics sentence is presented to the user in the Smule karaoke mobile application. However, constructing a well-balanced and easy-to-use evaluation dataset from this weakly-labelled and weakly-annotated dataset presents a number of challenges. Chapter 4 presents solutions to the problems of voice activity detection and segmentation, text normalisation, alignment of prompts with singer activation, and design of training, development and test sets with disjoint singer and song identities. The results of this work have led to the DSing corpus, a corpus containing three increasingly larger training sets of 15, 44 and 149 hours of unaccompanied singing. DSing also includes two ground truth test sets for validation and evaluation. This dataset has been made publicly available for sung speech recognition research (Roa Dabike and Barker, 2019). At the time of writing, this dataset has already been proved beneficial for other researchers (Demirel et al., 2020a, 2021a,b; Watanabe and Goto, 2020; Zhang et al., 2021).

The effect of sources-based features in sung speech recognition

Chapter 4 first presents a spoken speech state-of-the-art hybrid ASR architecture trained using the DSing corpus (Roa Dabike and Barker, 2019). The system is composed of a factorised time delay neural network (TDNN-F) acoustic model trained using traditional features, i.e., 40 Mel-frequency cepstral coefficients (MFCC) and 100 i-vectors (Dehak et al., 2011). The language model consist of a 4-gram model trained on 1.7 million lines of lyrics text. The system was used to evaluate the value of different musical properties of the sung speech signal (Roa Dabike and Barker, 2021). It was found that source-based features, such as pitch, voicing degree and vocal quality measurements, were more helpful for normalising the acoustic model, especially for high pitch singing.

Preparation work for using DAMP-VSEP vocal separation dataset

Since 2018, music separation research has been based on the use of the MUSDB18 dataset (Rafii et al., 2017). However, music separation models have reached a point where the sole use of the MUSDB18 dataset is insufficient and more extensive corpora are needed. Researchers have been resorting to augmenting the training data using private datasets (Défossez et al., 2019). This started to change with the recent release of the Digital Archive of Mobile Performances - Vocal Separation dataset (DAMP-VSEP) (Smule, Inc., 2019), which opened the opportunity of constructing a common large-scale music separation dataset. However, the DAMP-VSEP dataset has been released in a raw state that is not directly suitable for source separation development and evaluation. It possesses several challenges that make it difficult to use including, misalignment between background and vocal sources containing bleeding from the background. Chapter 5 presents work to deal with these challenges, resulting in the construction of a standardised dataset for vocal separation. The resultant dataset contains three increasingly larger training sets of 14, 24 and 66 hours of accompanied singing. At the time of writing, the resulting dataset is due to be released on GitHub[6].

Instrumental accompaniment embedding for vocal source separation

Chapter 5 investigates the challenges of separating the singing segment from the background accompaniment. A model is first constructed using the conventional convolutional time-domain audio source separation network (Conv-TasNet) architecture trained on a novel dataset based on the DAMP-VSEP corpus (Smule, Inc., 2019). Then, motivated by speakers embedding ideas, this model is extended by introducing an instrumental embedding module to inform the separator network with the song's musical structure extracted from the background

[6]https://github.com/groadabike/VSEP-dataset

accompaniment. Results show that the use of instrumental embeddings is helpful to increase the separation performance for both the vocal and background sources. However, the benefits depend on how the embeddings are computed when applying the embedding in practical applications. In this thesis, the embeddings are computed by employing a cascade system of two steps that first estimates a background, and then computes embeddings from the estimated background.

Adapting clean sung speech acoustic models to distorted sung speech

Chapter 6 investigates the challenge of recognising accompanied sung speech by employing an audio source separation front-end to separate the singing before presenting it to an ASR back-end. Six different evaluation datasets are used to evaluate the performances under different musical scenarios. First, using an ASR system trained on unaccompanied sung speech, recognition performances are computed on the mixture with and without activating the separation module. This evaluation shows the extent of the benefit of estimating before recognising the lyrics. However, this system operates on mismatched singing conditions, i.e., the separated singing contains several distortions that affect the accuracy of the acoustic model's phoneme classification. This mismatch in singing conditions is addressed by adapting the acoustic model to a separated singing condition using the 'weight transfer' learning technique by leveraging two different adaptation sets, improving recognition.

1.4 Structure of the thesis

The thesis is composed of seven chapters. Figure 1.1 shows an overview of how these chapters are related. The thesis starts in Chapter 2 by studying the differences between spoken and sung speech that make the latter less intelligible than the former. Then, the relevant literature in audio source separation and sung speech recognition is reviewed and presented in Chapter 3. These two initial chapters inform the work for unaccompanied sung speech recognition and music source separation challenges, presented in Chapters 4 and 5. Note that Chapters 4 and 5 deal with separated challenges and the work done in one can be considered independently of the other. However, the outcomes from these two chapters are connected in Chapter 6 to investigate the challenges of accompanied sung speech recognition. The content of this thesis is accompanied by audio samples on the website https://thesis.gerardoroadabike.com.

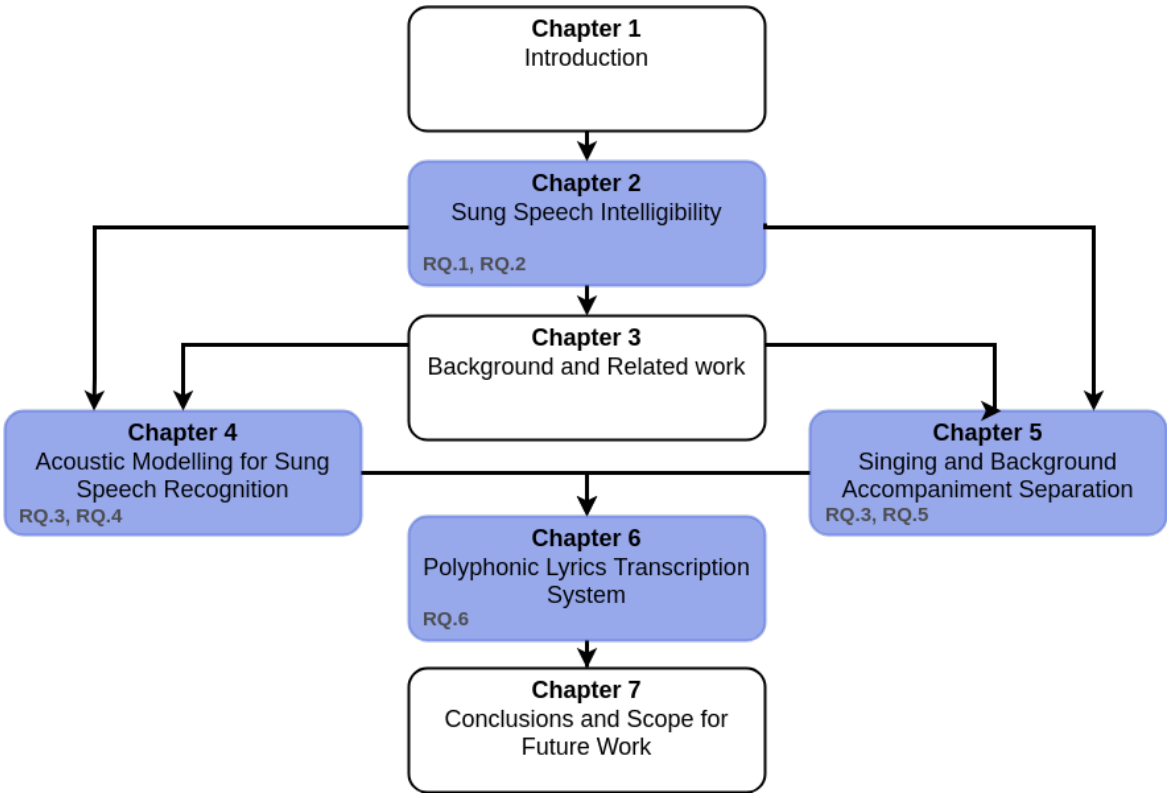


Figure 1.1 Organisation of the thesis. The blue boxes indicate investigation chapters. The research questions addressed by each chapter are indicated.

1.4.1 Chapters

Chapter 2 contextualises the work for the following chapters. It focuses on analysing the characteristics of the sung and spoken speech production, identifying how both speech styles differ and how these differences affect the signal properties (**RQ.1**). It also analyses the impact of the musical accompaniment on the intelligibility of the sung speech and how this compares with the more investigated noisy spoken speech problem (**RQ.2**). This chapter addresses these question by providing a comprehensive analysis of the similarities and differences between the different sung and spoken speech scenarios, highlighting some of the common shortcomings faced by different sung speech tasks and what singing properties can be exploited to alleviate these problems.

Chapter 3 reviews the relevant literature on sung speech processing, sung speech recognition and musical audio source separation. Together with Chapter 2, this chapter provides the foundations for the rest of the thesis.

Chapter 4 is dedicated to investigating the unaccompanied sung speech recognition problem. First, it presents the work of making a novel weakly-labelled (i.e., the data contains

start times for each expected lyric line that may deviate from the recordings) sung speech corpus suitable for recognition training (**RQ.3**). Then, adopting a spoken state-of-the-art ASR system, this chapter offers the construction of a sung speech recognition baseline. The chapter ends with an investigation of the effect on ASR performances of musically-motivated features from the singing voice (**RQ.4**).

Chapter 5 investigates the problem of separating the singing from the background accompaniment (**RQ.5**). It first presents the construction of a solo-ensemble singing separation corpus based on novel unprocessed data for vocal source separation. Next, it presents the construction of a baseline separation system by employing and adapting a well-known time-domain source separation model for speakers separation. Then, it investigates ways to improve the separation performances by utilising a composite loss function and employing embeddings ideas to inform the models with the background accompaniment characteristics.

Chapter 6 investigates the problem of recognising the sung speech from sung accompanied scenarios by evaluating the previous two chapters' outcomes in a combined system (**RQ.6**). This chapter deals with the mismatch between separated versus clean sung speech and attempts to identify how to reduce the impact of this mismatch on the recognition system. Also, it identifies the advantages and disadvantages of using a combined system versus a jointly optimised one. This chapter ends with a final evaluation using different datasets representing different difficulties on sung speech conditions.

Finally, **Chapter 7** provides a review of the main research findings from Chapters 4, 5 and 6, along with scope for future work.

1.4.2 Appendices

The thesis includes three appendices. Appendix A reports the individual performance of each recognition experiment performed in Chapter 4. Appendix B describes the computation of syllables per second score (SPS) and how it is applied to the DSing sung speech dataset. SPS score is used in Chapter 4. Appendix C describes a data selection algorithm based on 'acoustic latent Dirichlet allocation' (aLDA). The algorithm is then used in Chapter 5.

1.5 List of Publications

The following publications arise from the work presented in Chapter 4:

- Roa Dabike, G. and Barker, J. (2019). "Automatic lyric transcription from karaoke vocal tracks: Resources and a baseline system," in *Proceedings of Interspeech 2019*, pages 549–583.

- Roa Dabike, G. and Barker, J. (2021). "The Use of Voice Source Features for Sung Speech Recognition," in *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2021)*, pages 6513–6517.

The next publication refers to the participation in the MIREX 2020: Lyrics transcription task challenge, which corresponds to the preliminary results in Chapter 6:

- Roa Dabike, G. and Barker, J. (2020). "The Sheffield University System for the MIREX 2020:Lyrics Transcription Task," in *Proceedings of Music Information Retrieval Evaluation eXchange (MIREX 2020)*.

Currently, there are plans to publish the outcome of the rest of the thesis, for example:

- Chapter 2 can be published in a paper containing the key difference in production between sung and spoken speech and how these differences translate into the acoustic signal.
- Chapter 5 presents work on using background embedding to enhance source separation models. This chapter can be published after further investigation of how to better incorporate the embedding into the system.

Chapter 2

Analysis of the Differences Between Sung and Spoken Speech

2.1 Introduction

This thesis aims to develop an automatic sung speech recognition approach by adapting spoken speech technologies to sung speech. In order to do that, it is necessary first to understand the differences between sung and spoken speech styles and how these differences affect the intelligibility of the message. This chapter focuses on understanding the lyrics recognition from a human point of view to identify the challenges of the task and, therefore, how to inform the design of sung speech recognition systems.

Speech intelligibility can be defined as *"the extent to which a listener understands a speaker's or singer's message"* (Fine and Ginsborg, 2014; Munro and Derwing, 1995). Rossi et al. (2020) reported that the extraction of the speech meaning seems to operate similarly for spoken and sung speech, i.e., the same brain areas activate to process the sung and spoken speech. This means that under similar conditions, for a given listener, differences in the sung and spoken speech intelligibility result from differences in the speech production mechanisms and not from the listener.

In a spoken language, the intelligibility of words is crucial for communication. However, the perception of the spoken words can be complex, and the cues that aid in understanding the message may vary between speech styles. For example, *read speech* style is that which occurs when the spoken words are being read aloud by the speaker. Read speech may represent a form of speech where the phonemes are produced without any physical or external disruption, following the correct grammatical structures. *Clear speech* is a hyper-articulated speech style that is consciously adopted when the talker believes they are not being understood.

Loud speech results from increasing the vocal effort while speaking (e.g., the raising of the vocal effort as an unconscious response to noisy environments; this is known as the Lombard effect (Lane and Tranel, 1971)). *Spontaneous speech*, as the opposite of the read and clear speech, is unplanned with several disfluencies such as pauses, fillers (e.g., *uh* or *um*), word fragments, word repetitions and self-corrections. *Sung speech* is a style of speech resulting from using the sound production mechanisms simultaneously to speak and as an instrument. This chapter will focus on comparing the differences between the read speech (from now on referred to as spoken speech) and sung speech styles.

Fine and Ginsborg (2014) reported that trained and amateur singers (including singing teachers) believe that the understanding of the sung message while hearing a song is very important. They grouped the factors that affect the sung speech intelligibility into four categories: *performer*, *music- and word-setting*, *listener* and *environmental*.

The performer category is perhaps the most important category that affects the sung speech intelligibility. This category refers to factors like the singer's articulation and diction. Sung speech has several characteristics that makes it harder to understand than spoken speech. For instance:

- Sung vowels are shifted in frequency compared to spoken vowels (Sundberg, 1977b).
- Male singers may produce an increment of energy in frequency around 3,000 Hz called singer's formant (Bartholomew, 1934).
- High pitch singers tend to modify the vowel's articulation to increment the energy of the phonation frequency (Morozov, 1965).

The music- and word-setting category group factors that affect the intelligibility that are related to the song genre and composition style. For example, *Jazz* music is the most intelligible genre with an average of correct recognition of approximately 95%, and *Classical* music is the least intelligible with an average of less than 50% (Condit-Schultz and Huron, 2015).

The listener and environmental categories group factors that affect the correct understanding of the sung message unrelated to the singer's performance or the song's characteristics. The former refers to factors related to the music consumer (e.g., low hearing ability or lack of attention while listening to a song), and the latter to the physical location and environmental conditions where the singer performs (e.g., singing in a close music studio or performing on an open space like a stadium). The factors from these categories will not be represented in the data that will be used in further chapters. Therefore, these categories will not be discussed further but are mentioned for completeness.

This chapter is organised as follows. Section 2.2 introduces the challenges studied in this chapter by presenting various samples of commonly misheard lyrics from popular commercial songs. Next, the following two sections, Section 2.3 and 2.4, study different factors related to the performer that affect the intelligibility. Section 2.3 describes the sound production mechanisms and reviews how they operate and vary when producing sung and spoken speech. Section 2.4 presents a novel data-driven analysis of the fundamental differences between sung and spoken speech using the NUS-48E (Duan et al., 2013) sung and spoken speech parallel corpus. Then, Section 2.5 analyses factors from the music- and word-setting category that affects the sung speech intelligibility by conducting a novel analysis of the effect of the background accompaniment. Section 2.6 summarises the key findings of this chapter.

2.2 Common Lyrics Mishearing

This section presents several samples of real lyrics mishearing to illustrate how different factors may reduce the sung speech intelligibility. The samples were sourced from the *KissThisGuy*[1] website, an archive website where people can report their misheard lyrics from popular songs.

2.2.1 Mishearing Samples

The selected samples correspond to lyric mishearings from commercial recordings, representing mishearing produced by a combination of factors belonging to the *performer* and *music- and word-setting* categories described above. The musical genre of the samples may be classified as Pop or Rock. They were selected from mishearing reported by several people, taking care that they correspond to genuine mishearing and not to a simple humorous change of words. A humorous misheard lyric would be an intentional change of words to make a joke, and they do not represent what the user understood from the song. For instance, one user reported hearing “Annie, are you voting?” instead of "Annie, are you OK?" from *Michael Jackson*’s *Smooth Criminal*. However, they explained that the misheard lyric was the result of trying to make their daughter laugh.[2].

Table 2.1 presents nine samples of honest mishearing. The table includes the name of the artist, the sex of the singer, the song name, and the real (R) and the misheard lyrics (M). In italic bold are indicated the correct and misheard words for each song.

[1] http://www.kissthisguy.com

[2] https://www.kissthisguy.com/Annie–are-you-voting–misheard-25457.htm

Table 2.1 List of misheard lyric samples from popular songs sourced from the KissThisGuy misheard lyrics archive website. **R**: Real lyrics, **M**: Misheard lyrics

Song	Sex	Artist	Song	Type	Lyrics
1	F	Adele	Someone Like You	R M	... sometimes it ***lasts*** in love ... ... sometimes it ***laughs*** in love ...
2	F	Stevie Nicks	Edge Of Seventeen	R M	... like a ***white winged dove*** ... ... like a ***wide window*** ...
3	M	Survivor	Eye Of The Tiger	R M	... ***stalks his prey*** in the n... ... ***stocks his bread*** in the n...
4	M	R.E.M	Losing My Religion	R M	... that's me in the ***spotlight*** ... ... that's me in the ***spa, like*** ...
5	M	Toto	Africa	R M	... I bless the ***rains*** down ... ... I bless the ***grains*** down ...
6	M	Genesis	Invisible Touch	R M	... invisible ***touch, yeah*** ... ... invisible ***top shelf*** ...
7	M	David Bowie	Cracked Actor	R M	... show me ***you're real*** ... ... show me ***your rear*** ...
8	M	Bruce Springsteen	Brilliant Disguise	R M	... just a ***brilliant disguise*** ... ... just a ***brick in disguise*** ...
9	M	Walk The Moon	Shut Up and Dance	R M	... Shut up ***and dance*** with me... ... Shut up ***that duck*** with me...

Table 2.2 shows the American phonetic transcription (based on the open Carnegie Mellon University Pronouncing Dictionary[3] (CMU Dictionary)) of the misheard lyrics by using the International Phonetic Alphabet (IPA) symbols. The phonetic transcription corresponds only to the misheard words and not to the whole sentence.

Table 2.3 groups the samples presented in Table 2.1 by the possible sources of mishearing, i.e., *Performer* or *music- and word-setting*. The *Performer* category was divided into two subcategories; *Articulation - Pronunciation* and *Pitch*. The former groups mishearings originated by a poorly articulated pronunciation, phonemes omission or by stops in the incorrect part of the word. The latter identifies the songs where high pitches may have caused a reduction of intelligibility. The music- and word-setting category identifies songs where a specific musical sound overlaps with sung speech (*frequency overlap*) or songs where the

[3]http://www.speech.cs.cmu.edu/cgi-bin/cmudict

Table 2.2 Phonetic transcription of the misheard lyrics samples from Table 2.1, using the IPA symbols. **R**: Real lyrics, **M**: Misheard lyrics

Song	Type	Lyrics	Phonetic Transcript
1	R	lasts	læst
	M	laughs	læfs
2	R	white winged dove	waɪt wɪŋd dʌv
	M	wide window	waɪd ˈwɪndoʊ
3	R	stalks his prey	stɔks hiz preɪ
	M	stocks his bread	stɒks hɪz brɛd
4	R	spotlight	ˈspɒtlaɪt
	M	spa, like	spɑ, laɪk
5	R	rains	reɪnz
	M	grains	greɪnz
6	R	touch, yeah	tʌtʃ, jeə
	M	top shelf	tɒp ʃɛlf
7	R	you're real	jʊr rɪəl
	M	your rear	jʊər rɪr
8	R	a brilliant	ə ˈbrɪljənt
	M	a brick in	ə brɪk ɪn
9	R	and dance	ænd dæns
	M	that duck	ðæt dʌð

background was particularly louder in the misheard area, compared to the singing (*balance*). Each sample can be classified into one or more categories. For example, in sample 2, the singer raises her pitch in a specific syllable and at the same time, and perhaps as an effect of the pitch raising, the articulation of that syllable is degraded. This table illustrates how several factors can simultaneously interact, affecting the intelligibility of the lyrics.

2.3 Sung Speech Production

The speech production mechanisms are composed of three systems; *breathing*, *voicing* and *filtering*. Each system relates to specific organs involved in speech production. Figure 2.1 shows a schematic diagram of the different systems, indicating some of the most important organs involved in each of them. The speech production starts with breathing by generating an airstream using the human *respiratory system*. Then, in the voicing, this airstream from

Table 2.3 Categorisation of possible factors that affect the intelligibility of the lyrics mishearing presented in Table 2.1.

Song	Category			
	Performer		Music- and Word-Setting	
	Articulation Pronunciation	Pitch	Frequency overlap	Volume balance
1		✓	✓	
2	✓	✓	✓	
3				✓
4	✓			
5			✓	
6	✓			
7	✓		✓	
8	✓			
9		✓		✓

the breathing makes the *vocal folds* vibrate, generating a quasi-periodic signal. Last, in the filtering, this signal is filtered by the interaction of several movable parts from the *vocal tract*. The sequence interaction of these three stages produces the speech sounds we use for talking and singing.

This section reviews the operation of the speech production mechanisms, highlighting the production differences when talking and singing as described in the literature.

2.3.1 Breathing

The production of speech sounds starts with breathing. As shown in Figure 2.1, the main organs involved in breathing are the *lungs*, the *rib cage*, the *trachea*, the *diaphragm* and the *abdominal muscles*. The lungs are spongy structures suspended in the rib cage. They have small cavities connected to tubes called bronchi, which are then connected to the trachea. The trachea is a tube that extends from the bronchi to the vocal folds. The diaphragm is a sheet of muscle located below the lungs and inserted into the lower contour of the rib cage, taking the shape of a dome when it is relaxed. The abdominal muscles are a flat sheet of muscles attached to the rib cage and the higher part of the pelvis.

The dynamics of the respiratory process is an interaction between the diaphragm (primary muscle for inspiration) and the abdominal muscles (primary muscles for expiration). During inspiration, the diaphragm muscle contracts, flattening its shape to be like a plate. This action increases the rib cage volume and expands the lungs' volume, causing air to rush into the lungs. The diaphragm flattening also pushes the abdominal content (i.e., the digestive organs)

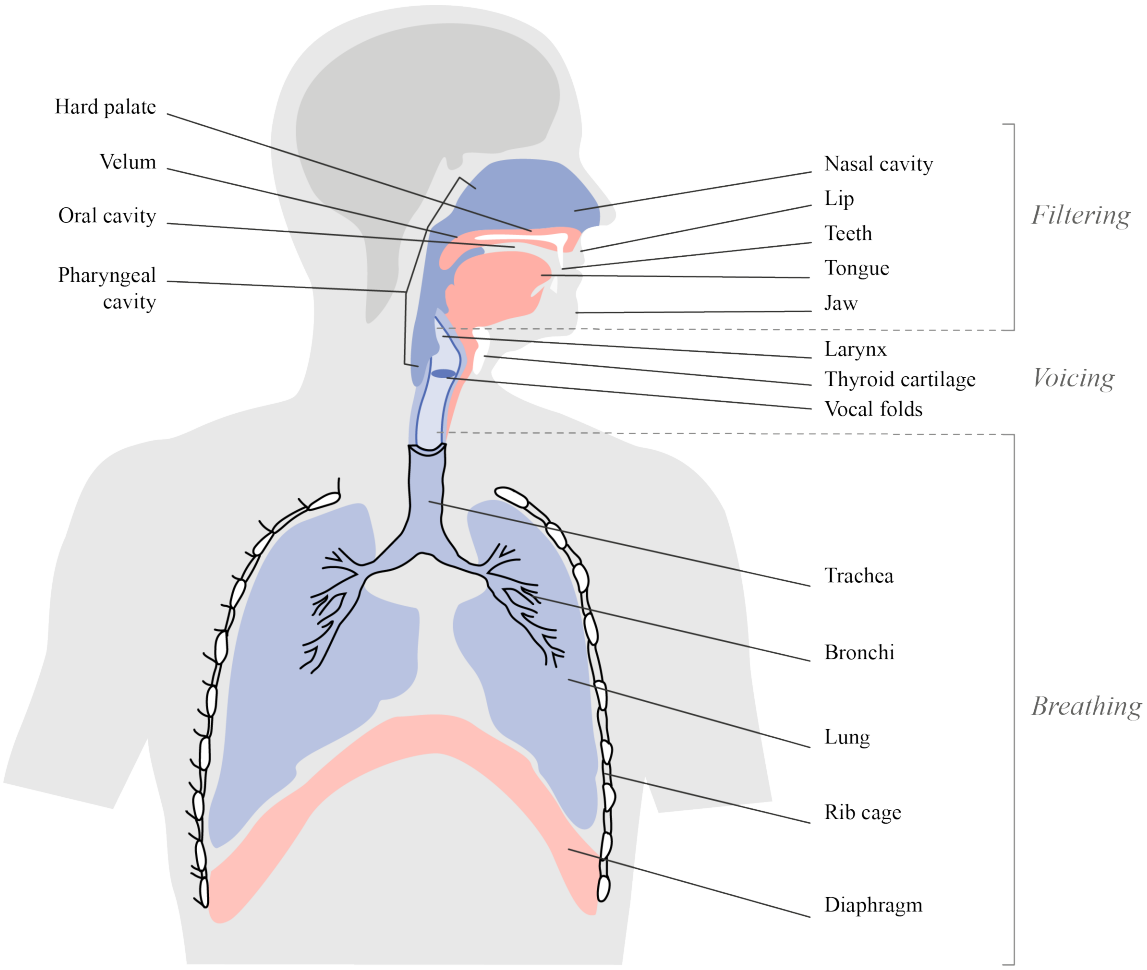


Figure 2.1 Schematic diagram of the human speech production mechanisms.

downward, pressing the abdominal muscles outward. During expiration, the abdominal muscles contract, pressing the abdominal content back up again. This pushes the diaphragm up again to its original shape, compressing the lungs and forcing the air contained in the lungs to flow through the trachea to reach the vocal folds. Figure 2.2 shows a sagittal view of the rib cage, diaphragm and abdominal muscles.

The maximum volume of air that the lungs can contain after inhaling is called *total lung volume* (TLV). After exhalation, a *residual volume* (RV) of air is retained in the lungs. Walsdorff et al. (2015) estimated that the TLV and RV for female adults average 5.6±1.0 and 1.8±0.7 litres, respectively, and 7.6±1.2 and 2.2±0.8 for male adults. The difference between the TLV and RV corresponds to the *vital capacity* (VC), which is the total amount of air one can use to produce speech sounds. Baldwin et al. (1948) investigated the relation of the VC with several physical characteristics. For this, he studied the VC from 92 healthy, non-smoker subjects (52 males and 40 females). He found that the VC correlates to the body

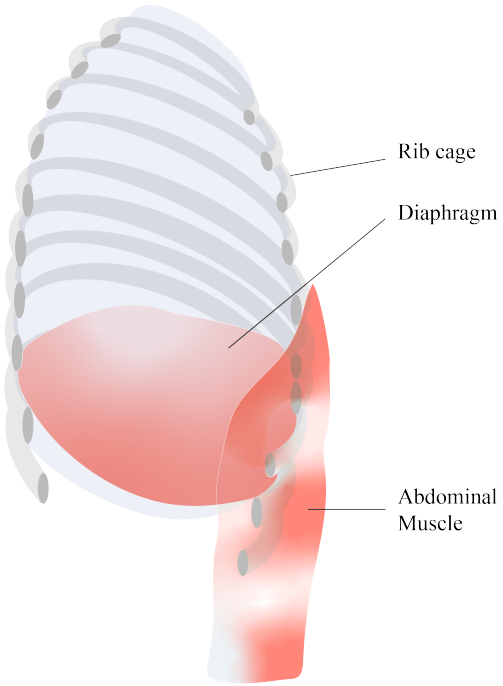


Figure 2.2 Schematic diagram of a sagittal view of the rib cage, diaphragm and abdominal muscles.

height positively, negatively to age and has no correlation with body weight. They estimated the adults' VC as a function of gender, age and height,

$$VC = \begin{cases} [2.763 - (0.112 \times A\ )] \times L & Males \\ [2.178 - (0.101 \times A\ )] \times L & Females \end{cases} \tag{2.1}$$

where L indicates the body height in centimetres and A the age in years. In a passive exhalation, the volume of air remaining in the lungs is called *functional residual capacity* (FRC). This volume is slightly less than half of the VC.

Bouhuys et al. (1966) reported that in spoken speech, it is common not to use more than 70% of the VC and that it is usual to take breaths when the volume goes just below the FRC, with breaths being taken every four or five seconds. In contrast, in singing, it is not unusual to use the totality of the VC to cover one lyrics' sentence per breathing interval, resulting in extended breathing periods of ten seconds or more. The extended breathing periods in singing are a result of conscious action, with the frequency depending on several variables like the music genre, the style of the song and the artistic performance. Gould (1977) found that trained singers increase their VC by reducing their RV, having an average of 20% more

VC than non-singers. Moreover, he found that, perhaps due to the vocal training, trained singers can use their VC more efficiently than untrained singers.

The efficient use of the vocal capacity allows singers to extend sounds to meet the needs of the music performance, i.e., adjust the speech sounds to encompass the required melody. Sharma et al. (2021) reported that the duration of sung words are, on average, 2.5 times longer than when the same words are spoken, and sung sentences are twice as long as the spoken versions. The duration of sung sentences seems not to be related to singing training, i.e., a given sung sentence will be extended similarly by trained and untrained singers (Brown et al., 2000).

As described above, on each exhalation, the air contained in the lungs flows up through the trachea, reaching the vocal folds. This produces an air pressure below the glottis (described in Section 2.3.2) called the *subglottic pressure* (P_S), measured in centimetres of water (cmH_2O). The level of the P_S will depend on the degree of muscle contraction and glottal resistance. For example, the P_S for spoken speech is around six cmH_2O and can increase to up to 15 cmH_2O in loud speech (Sundberg, 1987). In contrast, the P_S for sung speech can vary from 2 to 50 cmH_2O (Proctor, 1980).

Loudness is the subjective perception of the sound pressure and it is positively correlated to P_S. It is usually measured in terms of the *sound pressure level* (SPL), in decibels (dB). The SPL of a given sound pressure P_1 is measured as $20\log_{10}(P_1/P_0)$ with P_0 equals 20 μPa (micro Pascals). For every time the P_S doubles, the SPL increases on average by 9-10 dB. This seems to be true for both cases, spoken (Holmberg et al., 1988) and sung (Bouhuys et al., 1968) speech styles.

2.3.2 Voicing

The main organs that constitute the voicing system are the *vocal folds* and the *larynx*. The vocal folds (also called *vocal cords*) are muscles shaped as folds that originate in the posterior part of the *thyroid cartilage* (marked by the "Adam's apple") and insert in the *arytenoid cartilages*. The vocal fold lengths between 9 and 13 mm in adult females and between 15 and 20 mm in males. The arytenoid cartilages are two cartilages shaped like three-sided pyramids lying in the posterior part of the larynx. With rotating movements, the arytenoid cartilages control the *adduction* (opening) and *abduction* (closing) of the vocal folds by separating the posterior end of the vocal folds. The slit between the vocal folds is called the *glottis*. The larynx is a narrow and short tube (about 2 cm long) inserted in the pharynx in the cross-way between the respiratory and food passages. It is conformed primarily of cartilage, with the thyroid, epiglottis and arytenoid cartilages being the most important for understanding the voicing system. The larynx extends from the glottis to the lower part to the epiglottis, a

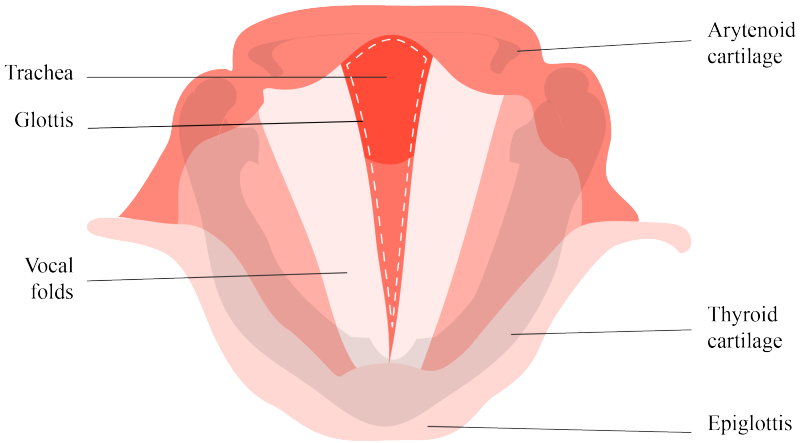


Figure 2.3 Schematic diagram of a transverse view of the vocal folds. This view is looking down the throat with the bottom towards the front of the neck.

cartilage attached to the thyroid cartilage shaped like a curled leaf. When swallowing, the epiglottis covers the passage to the larynx and the vocal cords close, preventing food from entering the breathing passages.

The subglottal pressure forces the vocal folds apart. The air passing through the glottis generates a Bernoulli force that closes the folds. This causes the subglottic pressure to rise again, repeating the cycle. In general, the P_S decreases at inspiration and increases at expiration (Ladefoged, 1961). The cycle of the opening and closing of the vocal folds produces a quasi-periodic vibration, called the *fundamental frequency* (F_0), measured in Hertz (Hz) (vibrations per second). To be heard, F_0 must be in the range of $20\,Hz$ to $20000\,Hz$. The perception of F_0 is commonly referred to as the *pitch* of a sound, where a high pitch sound corresponds to a high frequency and a low pitch to a low frequency. The pitch of a pure tone (a sine wave at a single frequency) can be considered unambiguous, i.e., with very little training, most people can discriminate minimal differences in frequency.

The sounds generated by the vocal folds corresponds to harmonic complex tones. That is, the sounds are composed of the F_0 (the lowest tone in the series) and overtones (partials). Together the fundamental frequency plus the overtones are called harmonic series, where the Nth element of the series is N times F_0. The pitch estimation of a complex tone usually corresponds to the F_0. Most harmonic complex tones tend to have more energy at low harmonics, with the amplitude of the partials decreasing uniformly at $12\,dB$ per octave (one octave corresponds to a doubling of frequency, e.g., one octave above $440\,Hz$ equals $880\,Hz$) (Sundberg, 1977b).

The length of the vocal folds determines the range of the fundamental frequencies; larger vocal folds result in lower frequency ranges. This becomes evident when comparing genders.

Table 2.4 Pitch ranges and vocal folds length by gender and by sung and spoken speech.

Gender	**Vocal Folds**	**Spoken**	**Sung**	
	(mm)	Range (Hz)	Vocal Category	F_0 Range (Hz)
Male	15 - 20	65 - 260	Bass Baritone Tenor	82.41 (E2) - 261.63 (C4) 98.00 (G2) - 329.63 (E4) 123.47 (B2) - 392.00 (G4)
Female	9 - 13	100 - 525	Contralto Mezzo-soprano Soprano	174.61 (F3) - 587.33 (D5) 220.00 (A3) - 698.46 (F5) 267.63 (C4) - 880.00 (A5)

Table 2.4 shows the pitch ranges and vocal folds lengths per gender and type of speech. Singers can be classified into one of the different vocal categories that divide singers by their singing frequencies. However, there is no universal definition of such ranges, but for the sake of completeness, Table 2.4 presents the vocal category and ranges from Randel (1986). Adult males' vocal folds lengths between 15 and 20 millimetres (mm), with spoken F_0 typically ranging between 65 and 260 Hz; and adult females' vocal folds lengths between 9 and 13 mm, with spoken F_0 typically ranging between 100 and 525 Hz.

In contrast, when singing, the range of the fundamental frequency that a singer can achieve increases. Singers of the same gender possess a different frequency range where they are more comfortable singing. For example, male singers could be classified into one of the three groups presented in Table 2.4, i.e., Bass (F_0 ranging between 82.41 and 261.63 Hz), Baritone (F_0 ranging between 98 and 329.63 Hz) or Tenor (F_0 ranging between 123.47 and 392 Hz); and female singers into Contralto (F_0 ranging between 174.61 and 587.33 Hz), Mezzo-soprano (F_0 ranging between 220 and 698.46 Hz) or Soprano (F_0 ranging between 267.63 and 880 Hz). Lycke and Siupsinskiene (2016) assessed the relationship between female voice parameters and the level of singing training. They reported that the maximum singing frequency a singer can achieve depends on the level of singing training, e.g., while trained female singers can achieve an average maximum frequency of $1068.8 \pm 233.2\,Hz$, untrained female singers achieve a lower maximum frequency of $720.1 \pm 218.4\,Hz$. Moreover, trained female singers showed a larger pitch range (distance between lowest to highest singing tone) of 35.3 semitones (12 semitones correspond to one octave) than untrained female singers (25.8 semitones), and a lower average minimum frequency ($144.1 \pm 22.7\,Hz$ versus $158.0 \pm 44.5\,Hz$).

Variations in $F0$ frequency could be made by adjusting the tension of the vocal folds. When the vocal folds are relaxed, they will get thicker and shorter, producing a lower pitch. In contrast, if the vocal folds are tensed, they will get thin and long, producing a higher

pitch. Additionally, the fundamental frequency is also affected by the subglottic pressure. Lieberman et al. (1969) reported that for every time the P_S increases by one cmH_2O, the fundamental frequency rises by 3 to 18 Hz, and that greater variations in F_0 seem to occur at a low vocal effort and at high average fundamental frequencies.

In Western music styles, good quality singing is usually characterised by the presence of a sinusoidal frequency modulation called *vibrato* (Bartholomew, 1934). Vibrato is characterised by *rate* (number of undulation per second) and *extent* (how far modulations vary relative to mean pitch). Reported vibrato rates are usually between 5 to 7 Hz and extents of about ± 1 semitone (Bartholomew, 1934; Hakes et al., 1988). Despite the frequency variation, vowel intelligibility may not be significantly influenced by vibrato, i.e., similar vowel perception scores are obtained from vibrato and vibrato-free phonation (Sundberg, 1975). Additionally to vibrato, singers may resort to another kind of vocal ornamentation such as *trill* and *trillo*. The former is a modulation similar to vibrato but with a greater extent of several semitones. *Trillo* is a vocal ornamentation where a single pitch is repeated, starting with a slow individual pulse separated by silent intervals and gradually increasing in rate.

Since the vocal folds are an organic structure, their oscillations are not exactly periodic, producing variations in period and amplitude, known as jitter and shimmer, respectively. Jitter is defined as the frequency variation from cycle to cycle. It is measured by detecting the timings of the fundamental periods, returning the absolute peaks, i.e., the beginning of the glottal pulse. Once the glottal pulses are obtained, jitter can be measured in terms of the average absolute difference between two consecutive periods:

$$jitter = \frac{1}{N-1} \sum_{i=1}^{N-1} |T_i - T_{i-1}| \tag{2.2}$$

where T_i is the extracted fundamental frequency period length, and N is the number of extracted fundamental frequency periods.

Shimmer relates to the amplitude variation of the sound wave. It is obtained in a similar way to jitter, but instead of considering the glottal periods, shimmer takes into account the maximum peak amplitude of the signal. Shimmer can be measured in terms of the average absolute difference between the amplitudes of two consecutive periods, divided by the average amplitude, expressed as:

$$shimmer = \frac{\frac{1}{N-1} \sum_{i=1}^{N-1} |A_i - A_{i+1}|}{\frac{1}{N} \sum_{i-1}^{N} A_i} \times 100 \tag{2.3}$$

where A_i is the extracted peak-to-peak amplitude, and N is the number of fundamental frequency periods.

Additionally to jitter and shimmer, speech quality can be assessed in terms of the *harmonic to noise ratio* (HNR). HNR measures the ratio between the periodic components of the voice speech (harmonic part) and the aperiodic components or glottal noise, i.e., a high HNR is associated with a sonorant and harmonic voice. It is obtained from the auto-correlation function of the voice signal and expressed in decibels (Boersma, 1993), defined as:

$$HNR = 10\log_{10}\frac{AC_v(T)}{AC_v(0) - AC_v(T)} \tag{2.4}$$

where $AC_v(0)$ is the auto-correlation coefficient of the origin and $AC_v(T)$ is the component of the auto-correlation corresponding to the fundamental period. The difference between $AC_v(0)$ and $AC_v(T)$ is assumed to be the noise energy.

Farrus et al. (2007) reported that jitter and shimmer carry speaker-specific characteristics and could increase the performance of speaker recognition systems. However, as reported by Slyh et al. (1999), these parameters are sensitive to the speech style, and different values for the same speaker are obtained in, for example, soft, loud, fast and slow speech. Similarly, several factors can affect the jitter, shimmer, and HNR values when singing:

1. Jitter and shimmer reduce after "warming-up" the vocal muscles before singing (Mezzedimi et al., 2018).

2. Different singing frequency modulations like *vibrato* and *trillo* present different levels of jitter, being in general lower for vibrato (Hakes et al., 1988).

3. Sung speech presents a lower degree of glottal noise than spoken speech, highlighting a discrepancy between sung and spoken speech styles (Lundy et al., 2000).

4. Jitter is inversely correlated to the fundamental frequency (Horii, 1979).

5. Jitter and shimmer are inversely correlated to the sound pressure level (Orlikoff and Kahane, 1991), but shimmer's correlation seems less robust than jitter's.

The speech *intonation* is the pitch variation over time. In the English language, the intonation conveys emotional cues, e.g., happiness or angriness. In tonal languages, like Mandarin, changes in the intonation change the word meaning. However, the role of the voice source in singing is fundamentally different from its role in speaking. Its first priority may be to follow the melody and stay in tune with the background accompaniment. This means that it has very different characteristics. First, the sung pitch range is much larger, and

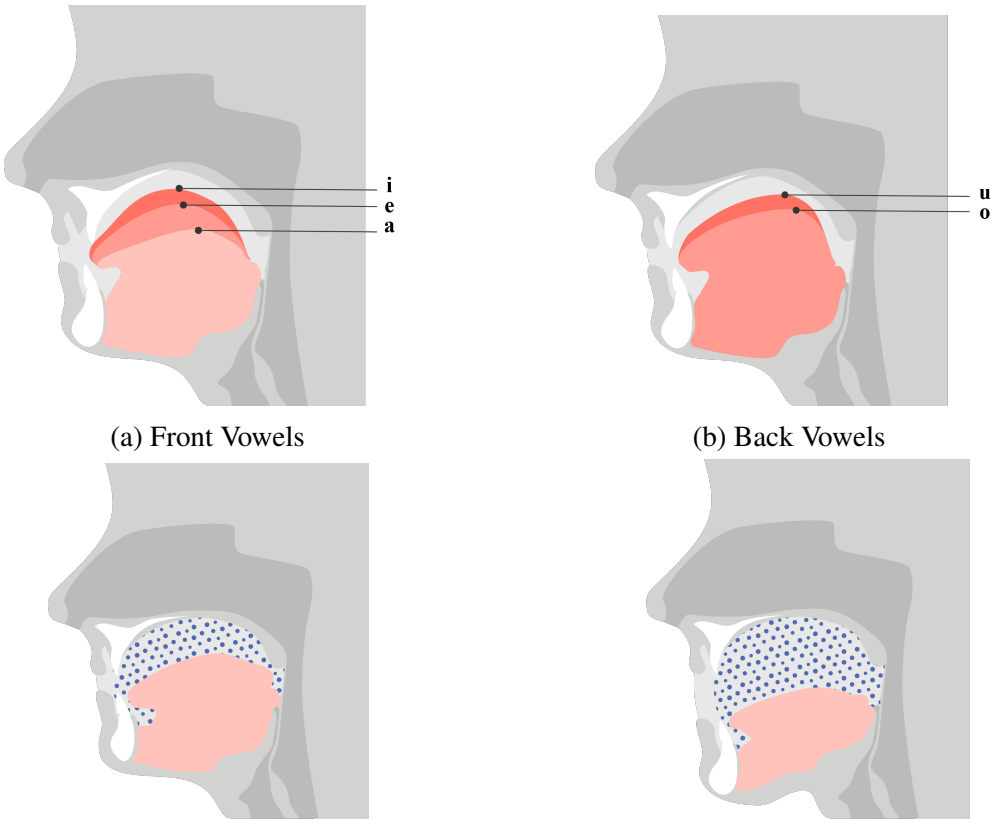
the average sung pitch is higher than in spoken speech (Merrill and Larrouy-Maestri, 2017). Second, the spoken pitch frequency varies freely, rising and falling within one syllable, with changes up to 12 semitones (Patel et al., 2008). In contrast, Vos and Troost (1989) reported that the pitch variation in melodies from Classical and Western music is expected to be more discrete with relatively infrequent changes greater than two semitones between two notes. However, as discussed before, a sung word length is longer than spoken ones, which means that one could expect that a sung syllable comprises several notes with a pitch variation more considerable than two semitones. The pitch variation in a sung and spoken word will be evaluated in Section 2.4 utilising a sung and spoken parallel corpus.

2.3.3 Filtering

The last part of the speech production mechanism is the filtering stage. This stage is composed of the *vocal tract*, a passage that includes the oral, nasal, and pharyngeal cavities through which air passes in speech production. The vocal tract starts above the vocal folds, in the epiglottis, and ends at the tip of the lips, comprising several movable agents (i.e., articulators) such as the lips, tongue, jaw, velum and the larynx (Figure 2.1). Lindblom and Sundberg (1971) characterised the articulators' movements as:

- The lips can be extended or rounded, changing the area of the mouth opening. They can vary in position with the help of the jaw.
- The jaw can move up or down and to the front and back. An increment of the jaw opening results in widening of the mouth and narrowing of the larynx.
- The tongue is a very dynamic articulator. It can constrict towards the velum, reaching the hard palate, constrict to the pharynx cavity, or extend to reach the teeth.
- The velum opens and closes the passage between the nasal and mouth cavity.
- The larynx can raise or lower its position giving less or more space to the pharynx.

Phonemes are the smallest sound units of speech, and they are divided into *vowels* and *consonants*. Vowels are produced with the articulators not coming very close together and without blocking the air passage, i.e., the tip of the tongue is always located behind the lower incisors, e.g., the English vowels [a], [ɑ], [e], [ɪ], [o], [ʊ]. Vowels can be described in terms of the shape and position of the tongue (Ladefoged and Johnson., 2015). When the highest point of the tongue is close to the front of the mouth, the vowels are called *front vowels* (e.g., the vowels [ɪ] and [a] from the words "hid" and "had", respectively). On the other hand,



(a) Front Vowels (b) Back Vowels

(c) Closed Vowels (d) Open Vowels

Figure 2.4 Schematic diagram of a sagittal view of the position and shape of the tongue in vowels production. (a) and (b) show the degree of the tongue height for the front vowels (a), and the back vowels (b). (c) Shows the tongue height and lip separation for closed vowels. (d) Shows the tongue height and lip separation for open vowels.

when the highest point of the tongue is located close to the back of the mouth, the vowels are called *back vowels* (e.g., the vowels [ʊ] and [ɑ] from the words “food” and “father”, respectively). Further, when the tongue is located close to the roof of the mouth, the vowels are called *closed vowel* (e.g., vowels [ɪ] and [ʊ]), and *open vowels* are vowels where the position of the tongue is far from the roof of the mouth (e.g., vowels [ɑ] and [a]). In close vowels, the lips are closer together, and they separate in open vowels. Figure 2.4 shows a sagittal view of the tongue position for front, back, open and closed vowels.

Consonants are sounds produced by partially or entirely stopping the air being breathed out through the mouth. Consonant production can be described by the *place of articulation* (position of the greatest vocal tract constriction) and the *manner of articulation* (the amount of airflow obstruction) (Ladefoged and Johnson., 2015). Table 2.5 summarises the most

Table 2.5 List of the most common places of articulation in the English spoken language, including some of the consonants and the key articulators for each place. The consonants included are only examples and do not form a complete list.

Place of Articulation	**Consonant**	**Key Articulators**
Bilabial	[p] [b] [m]	Upper and lower lips
Labiodental	[f] [v]	Lower lip and upper teeth
Dental	[θ] [ð]	Tip of tongue and upper teeth
Alveolar	[s] [d] [n]	Tip of tongue and alveolar ridge
Post-Alveolar	[ʃ] [ʒ]	Tip of tongue and behind alveolar ridge
Palatal	[j]	Front of tongue and hard palate
Velar	[k] [g]	Back of tongue and velum
Glottal	[h] [ʔ]	Vocal folds

common places of articulation of the English consonants, including some examples and indicating the key articulators involved in their production.

The manner of the articulation refers to the degree of constriction of the articulator, i.e., for each place of articulation, there may be different patterns of the constriction. There are three main manners of articulations; *plosive*, *fricative* and *approximant*. Plosives are sounds produced when the air is stopped from moving past the point where the articulators close, e.g., the sound [t] at the start of the word “tie”. Fricatives are sounds where the air passes by a narrow gap between articulators, becoming turbulent, e.g., the sound [z] at the start of the word “zoo”. Approximants sounds are produced when the articulators create a narrow gap wide enough not to produce turbulence, e.g., the sound [j] at the start of the word “yes”. Other manners of articulation are characterised by rapid changes or alternation in the degree of constriction, such as *trill* sounds, which corresponds to an intermittent alteration between a plosive and some degree of approximation, like the sound [r] at the start of the word “road”. *Affricative* sounds are fricative sounds preceded by a brief stop, such as the sound [tʃ] at the start of the word “chain”. Last, *nasal* sounds are produced when the velum opens the nasal cavities letting the airstream flow through the nasal cavities, like the sound [n] at the start of the word “no”.

Additionally, speech sounds are classified as to whether they originate from the vocal cords or not. When they originate by the vibration of the vocal folds, they are called *voiced sounds*. All vowels and several consonants belong to this group. The second group of sounds are called *unvoiced sounds*. They are produced with the vocal fold pulled apart, letting the air pass freely, so there is no vocal folds vibration. In most cases, for every place and manner, there exists a voiced and unvoiced pair of sounds. For example, the [b] and the [p] sounds

are voiced-unvoiced pair of sound, both produced by an abrupt stopping of the air using the lips, i.e., bilabial plosives.

As discussed, the position and shape of the articulators shape the vocal tract. Sundberg (1987) highlighted that the vocal tract's shape could be described in terms of a transfer function that characterises the sound transfer ability of the vocal tract, i.e., a function of the frequency of sound to be transferred. The frequencies that are most efficiently transferred (resonance) are called **formants** frequencies. The formants enhance the closest harmonics, i.e., harmonics closest in frequency to the centre of the formant have greater amplitude than others. According to Sundberg (1977b), the vocal tract has four or five most important formant frequencies for governing the perception of speech sounds.

Opening the nasal cavities introduces antiresonances that reduce the acoustic power of the signal. Nasal consonants, like [n] and [m], possess a very low first formant around 250 Hz and a large region above the first formant without energy. Although the articulators can affect all the formants, some of them affect one formant more than others. Lindblom and Sundberg (1971) made the following observations,

$F1$ is highly sensitive to changes in the jaw opening, rising when the jaw opening increases.

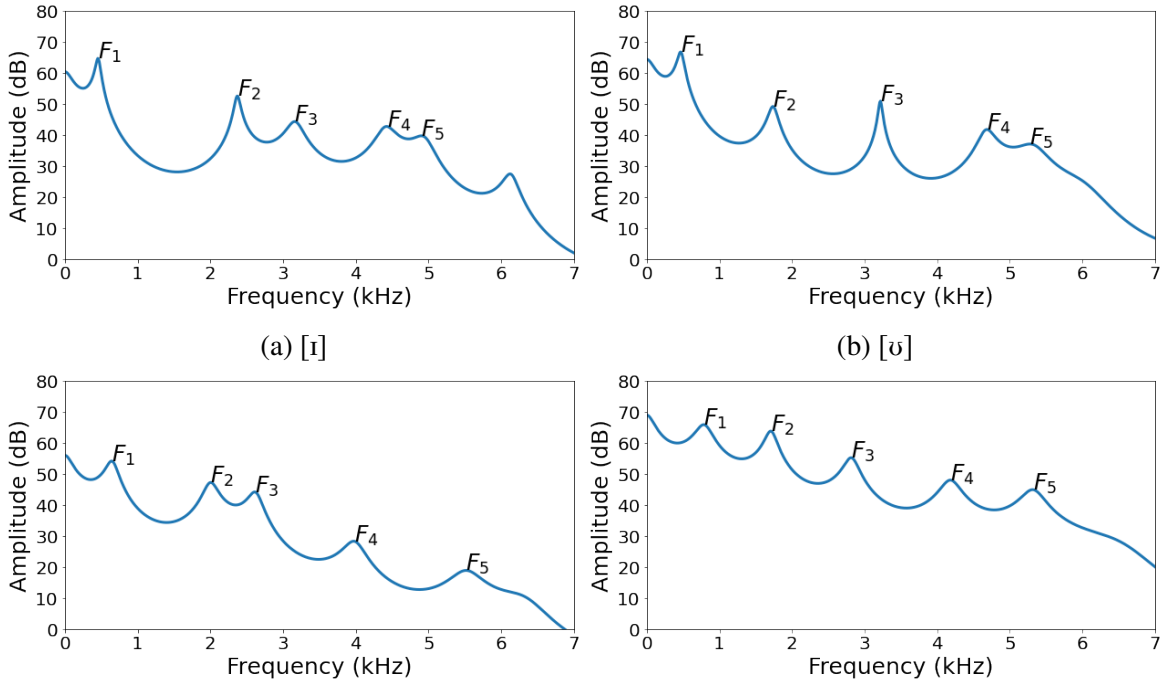
$F2$ is more sensitive to the tongue shape, reaching a maximum frequency when the tongue constricts at the anterior part of the vocal tract and minimising when the tongue constricts in the velar region of the vocal tract.

$F3$ is more sensitive to the position of the tip of the tongue behind the incisors, decreasing when the cavity between the teeth and the tongue increases.

$F4$ is more influenced by the length of the vocal tract and the larynx configuration.

The frequencies of the first two formants (F_1 and F_2) are the most sensitive to the articulators' position (Sundberg, 1977b). These two formants determine to a large extent the perceived speech sound, i.e., the perception of the phonemes are most correlated to the F_1 and F_2 frequencies. For a male adult, F_1 averages between approximately 250 and 700 Hz, while F_2 averages between 700 and 2500 Hz. Figure 2.5 shows the first five formant frequencies from four different vowels ([ɪ], [ʊ], [æ], and [a]) spoken by a female speaker. The vowels [ɪ] (2.5a) and [ʊ] (2.5b) from the top row have lower F_1 frequency than the vowels [æ] (2.5c) and [a] (2.5d) in the lower row. But, vowels [ɪ] and [æ] from the left column have higher F_2 frequency than vowels in the right column.

The formants construction operates in the same way for sung and spoken speech. However, singers possess characteristics that deviate their voice quality from spoken speech. For



(a) [ɪ] (b) [ʊ]

(c) [æ] (d) [a]

Figure 2.5 Formant frequencies of the vowels [a], [æ], [ɪ] and [ʊ] from a female speaker from the NUS-48E corpus (detailed in Section 2.4).

example, in sung speech, the [e] sound in "head" is shifted towards the sound [ɜ] in "heard". Sundberg (1977b) called this shift in the vowel quality a result of a "darker" voice (as observed when a person yawns while speaking). This darker voice is reflected as lower F_1 and F_2 frequencies and in an increment of energy in frequencies between 2500 and 3000 Hz, called the *singer's formant* (first described by Bartholomew (1934)). The singer's formant appears in male and low register female operatic singers. Its level, defined as the deviation from the expected level of the third formant, depends on several factors, including singer proficiency, vowel and fundamental frequency. The frequency average of the singer's formant depends on the vocal category (vocal categories described in Section 2.3.2), i.e., the average frequency is lower for basses and higher for tenors (Sundberg, 2001). The singers' formant is the acoustic result of lowering the larynx and widening the pharynx, which generates a resonance effect that increases the energy at higher frequencies without increasing the air pressure. Mainka et al. (2015) found that, on average, male singers lower their larynx by about eight *mm* on average compared to when they speak. Sundberg (1977a) suggested that the importance of developing the singers' formant lies in the need for operatic and concert singers to compete with the orchestra. An orchestra produces the highest energy at about 450 Hz, which is about the same frequency at which spoken speech produces maximum

Table 2.6 Formant frequency values from vowels spoken and sung at a high pitch by a Spanish soprano (as reported by (de Julián, 2016)).

Vowels	**Spoken Speech** (F_0: 309 Hz)		**Sung Speech** (F_0: 704 Hz)	
	F_1	F_2	F_1	F_2
[i]	353.52	2542.30	725.08	1890.63
[e]	608.42	2020.59	728.34	1834.14
[a]	841.96	1916.33	735.55	1739.70
[o]	736.30	1867.89	716.93	2000.22
[u]	434.26	1031.24	718.28	1965.83

average energy. With the help of the singers' formant, trained singers are much more easily heard in the presence of background music.

Unlike male operatic singers, high pitch female singers resort to different adjustments of their formants. At high pitch singing, the pitch frequency is higher than the first formant, reducing its energy and weakening the vowel sound clarity. For example, if a soprano sang the vowel [e] at a pitch of middle C, her first formant will be close to 270 Hz, and her pitch will be one octave higher at 523 Hz, seriously reducing the energy of the pitch. To compensate for this, soprano singers use the jaw opening to adjust their first formant frequency to match the pitch frequency, enhancing its energy. The process of adjusting F_1 to match F_0 is known as *formant tuning*. However, this adjustment shifts the vowels reducing the confidence in the vowel discrimination. de Julián (2016) reported that a Spanish soprano singing at a high pitch of 704 Hz tuned the formants of all her vowels, shifting and concentrating all of them to values closest to her spoken [o]. The values per vowel obtained by de Julián (2016) are presented in Table 2.6.

Borch and Sundberg (2002) investigated sung speech production differences between operatic and pop singers. They asked pop and operatic singers to sing the same sentences and found that pop singers do not produce the singer's formant. Sundberg (1977b) suggested that pop singers do not resort to the use of the singer's formant due to the vowel shifting; instead, they depend on electronic amplification. As was discussed before, operatic singers develop the singer's formant to compete with the orchestra, with a cost of reduced vowel intelligibility. Pop singers, on the other hand, may need to raise their vocal efforts while competing with the musical accompaniment, or rely on sound engineers to raise the level of their voices in frequencies around 3000-4000 Hz, which in live performances allows them to hear themselves.

People from common regional or social/cultural groups may share the same accent (a distinctive manner of pronunciation). Accents can vary to the degree that the phonetic transcription of a word can vary from speaker to speaker (e.g., the word "buck" can be transcribed as [b] [ʌ] [k] in some southern British English regions and as [b] [ʊ] [k] in some northern British English). They can also present differences in the acoustic parameters, such as the speech rate (phonemes or words per second), phoneme duration, pitch range and formant frequencies. Additionally, it is not uncommon between different regions to have a pronunciation that deviates from the "standard" phonetic transcription, presenting some substitution, insertion or deletion of phonemes. For example, Grabe et al. (2000) reported that the Cambridge accent has a larger word mean duration than the Newcastle, Belfast and Leeds accents, with the Newcastle accent presenting the shortest duration. Also, they reported that the Newcastle accent presented the lowest word duration variation between short and long vowel words. In singing, native English speakers tend to neutralise their accents (Gibson, 2010), and there is a tendency to move the accent towards American pronunciation (Konert-Panek, 2017), and non-native English speakers tend to neutralise their accent during singing (Hagen et al., 2011; Mageau, 2016).

2.4 Acoustic Analysis of Sung Speech

The previous section presented a review of the three systems, breathing, voicing and filtering, that make up the human sound production mechanisms. Specifically, it explained how these mechanisms vary when producing the different spoken and sung speech sounds. In this section, I use the *NUS-48E Sung and Spoken Lyrics Corpus* (Duan et al., 2013) to conduct a novel study of how the differences in the speech sound production translate into differences that can be observed in the acoustic signal, for example, differences in speech intensity, vowels lengthening, pitch range, formants frequencies and voice quality. The NUS-48E is a sung and spoken parallel corpus widely used in tasks such as singing voice generation (Blaauw et al., 2019; Chandna et al., 2019; Gao et al., 2020b), singing voice conversion (Deng et al., 2020; Gao et al., 2019; Liu et al., 2021; Nachmani and Wolf, 2019; Sisman and Li, 2020; Takahashi et al., 2021), speech-to-singing conversion (Parekh et al., 2020; Vijayan et al., 2019; Wu and Yang, 2020), and sung and spoken speech temporal alignment (Vijayan et al., 2018). To the best of my knowledge, this is the first time this dataset has been used for a comprehensive analysis of acoustic differences between sung and spoken speech.

2.4.1 Dataset Description

The *NUS-48E Sung and Spoken Lyrics Corpus* (NUS-48E) (Duan et al., 2013) is a parallel corpus containing 12 native or proficient English speakers representing various accents; six females and six males. Each speaker recorded a sung and a spoken version of four different English song lyrics from a pool of 20 unique songs, totalling 48 recordings per speech style, and 115 minutes of sung data and 54 min of spoken data. The recordings were performed in a sound-proof recording studio (STC 50+) using an Audio-Technica 4050 microphone with a pop filter. Analogue signals were digitised at 44100 Hz and using 16 bit samples using the software software Pro Tools[4] version 9.

During singing recordings, singers were fed with a downbeat through the headphones (i.e., the beat was not included in the recording) to set the tempo and guide the singing. Singers were not provided with any other accompaniment and were recorded as acappella. Singers were free to choose the key where they were most comfortable and make some rhythm and pitch changes.

The recordings are organised by speaker and separated by speech style (i.e., spoken vs sung). Each recording includes phone-level annotations, indicating the starting and ending point of each phoneme in seconds. For the phonetic annotation, the 39-phoneme set from the CMU dictionary[5] plus labels *SIL* (silence) and *SP* (aspiration) to mark long silences and words boundaries was adopted. The annotation process for sung recordings was performed utilising the Audacity[6] software. First, a human annotator selected from a pool of three labelled the phonemes as how they were uttered to capture the effect of singing. Then, labels from one annotator were verified by another to ensure consistency between annotators.

Due to time and resource limitations during the corpus construction, only the sung data were manually annotated. The spoken recordings were annotated by aligning labelled phonemes to the spoken lyrics utilising a Gaussian Mixture Model (GMM) - Hidden Markov Model (HMM) system trained on the Wall Street Journal (Paul and Baker, 1992) (WSJ0) corpus using the Hidden Markov Model Toolkit (Young, 1993) (HTK). The system had 2419 tied triphone states, and a GMM modelled each state with 16 components. The model was trained on 12 MFCC plus energy, first and second order derivatives features vector, totalling 39 features per frame. The audio signals were processed using a window of 25 milliseconds (*ms*) length and ten (*ms*) of shifting. Duan et al. (2013) evaluated the model using a closed vocabulary of 5000 words and a bigram language model, obtaining a 5.51% word error rate

[4] https://www.avid.com/pro-tools

[5] Web site: http://www.speech.cs.cmu.edu/cgi-bin/cmudict.

[6] Audacity® software is copyright ©1999-2021 Audacity Team. Web site: https://audacityteam.org/. It is free software distributed under the terms of the GNU General Public License. The name Audacity® is a registered trademark.

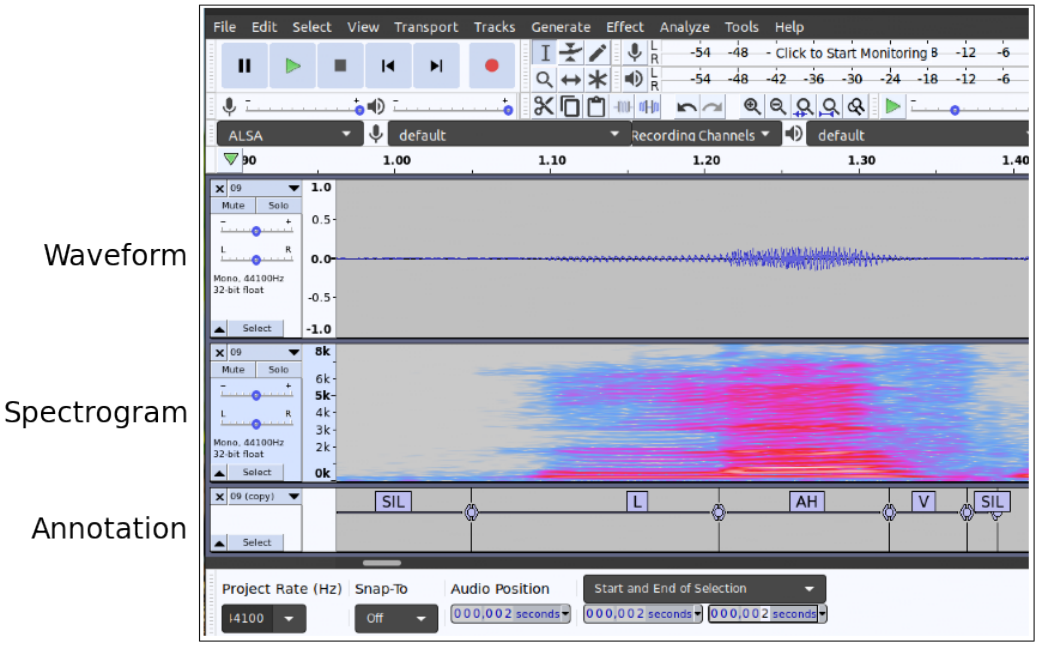


Figure 2.6 Example of one phoneme-level annotation of the word **LOVE** from the song *Love Me Tender* by *Elvis Presley* (1956), spoken by one female speaker from the dataset NUS-48E. This figure was constructed by employing the Audacity software.

for a benchmark. Figure 2.6 shows an example of the annotation from a spoken version of the word **LOVE** from the song *Love Me Tender* by *Elvis Presley* (1956).

Unlike sung song lyrics annotation, where six decimal places precision were used (i.e., the default annotation precision used by Audacity), the spoken annotations possess two decimal points of precision (i.e., as determined by the 10 *ms* frame shift used in the automatic alignment process). However, it is unlikely that humans can achieve phoneme boundaries annotation precision of 10 *ms* or lower, and phonemes boundaries annotations are sometimes arbitrary as sounds tend to blend into each other. The following subsections will utilise the annotations as they are provided.

2.4.2 Methodology

The various phone level analyses were performed using the ten single sound vowels in the NUS-48E corpus, corresponding to 8438 vowel samples per speech style, roughly equally distributed per gender. NUS-48E includes diphthong annotations, i.e., two-sound vowels annotated in a single composite phoneme, like [oʊ] from the word “boat” and [eɪ] from the word “bait”. Diphthong sounds were not included in the phoneme level analysis as they may

Table 2.7 List of the ten vowels from NUS-48E included in the analysis, including examples of words containing that vowel.

Vowel	Example
[ɑ]	b**a**lm
[æ]	b**a**t
[ʌ]	b**u**tt
[ɔ]	st**o**ry
[ɛ]	b**e**t

Vowel	Example
[ɛˑ]	b**i**rd
[ɪ]	b**i**t
[iː]	b**ea**t
[ʊ]	b**oo**k
[uː]	b**oo**t

affect the results due to the vowel transition within the diphthong (pitch, formants) and the duration of the combined sounds.

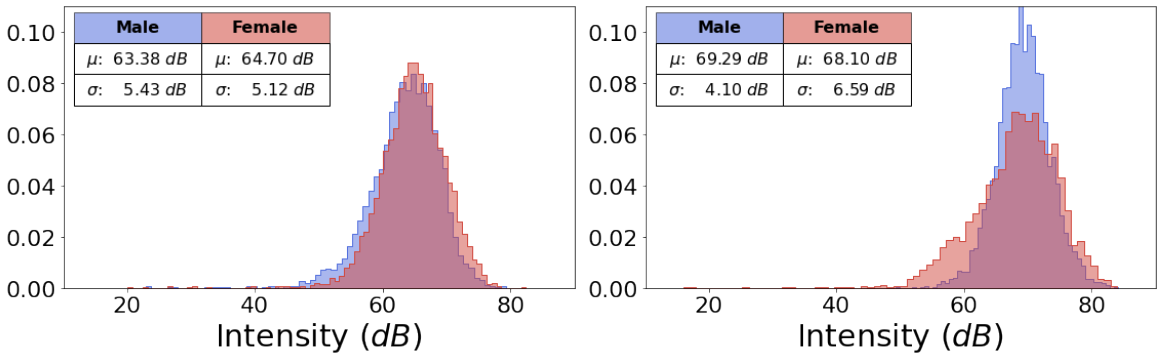
The NUS-48E corpus annotations are only provided at the phoneme level. Therefore, the corpus was preprocessed to construct word and sentence level annotations, i.e., annotations marking the start and end of words and sentences that are needed for word and sentence level analyses. The NUS-48E provides word boundaries in the form of silence or aspiration (some with no length, i.e., exact start and end timestamp). Therefore, the word-level transcription is obtained by simply grouping all consecutive phonemes between silence or aspirations.

Sentence-level annotations are constructed by using the *spoken* version of the lyrics. First, all consecutive phonemes are grouped until a silence or aspiration annotation longer than 300 *ms* is found, resulting in 1030 spoken sentences. Then, this segmentation is utilised to process both the spoken and sung versions of the lyrics, i.e., if the first sentences of one spoken lyrics encompass the 20 first phonemes, then the first sentence for the sung version will also be composed of the first 20 phonemes. This process ensures the comparison between a spoken and sung version of the same sentence.

The feature extraction was performed using Parselmouth (Jadoul et al., 2018), a Python interface for Praat (Boersma and Weenink, 2021). The details of how features were extracted are included in the relevant subsections.

2.4.3 Energy

Energy (loudness of speech) is one of the main distinguishing characteristics between spoken and sung speech styles. As stated in Section 2.3.1, the subglottic pressure is higher in sung speech than in spoken speech. Sharma et al. (2021) reported that the dynamic range of short-time energy in sung sentences is larger than in spoken ones. However, they did not include a gender comparison in their analysis.



(a) Spoken speech.

(b) Sung speech.

Figure 2.7 Histogram of the mean speech intensity distribution from eleven English vowels in the NUS-48E corpus per sung and spoken speech styles. (a) shows the spoken and (b) the sung distribution per male and female genders.

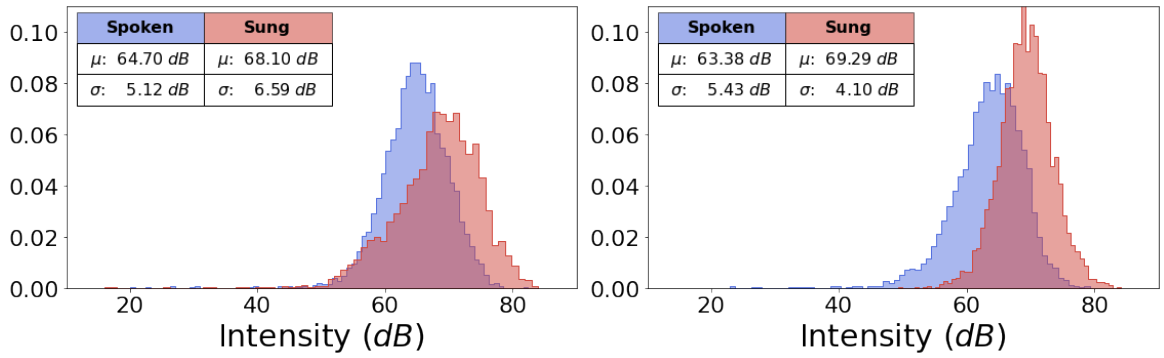
The energy difference between sung and spoken speech styles was analysed by plotting the distribution of the mean intensity from ten English vowel sounds from all recordings in NUS-48E. The mean speech intensity can be defined as the mean power (in Pascals square per second $\frac{Pa^2}{sec}$) between two points in time t_1 and t_2, expressed in decibels (dB). It is calculated as:

$$Intensity = 10\log_{10}\left(\frac{1}{t_2 - t_1}\int_{t_1}^{t_2} 10^{x(t)/10}dt\right) \tag{2.5}$$

and it should be close to the SPL. Due to the lack of information about the dB SPL utilised during the recordings, this analysis will focus on the relative difference in energy between female and male speakers for sung and spoken speech.

The per gender distribution of the mean speech intensity was estimated for the sung and spoken speech styles. Both genders present a similar mean (μ) and standard deviation (σ) for spoken speech, with the mean female intensity about 1.3 dB higher and the standard deviation 0.3 dB smaller than the male values. However, when singing (Figure 2.7b), the mean intensity for the males is 1 dB higher, and the standard deviation is more than 2 dB smaller than the females. This may suggest that, given the same song, male singers vary their loudness less than female singers.

Figure 2.8 shows the same data as Figure 2.7 but comparing the sung and spoken speech styles intensity distribution per gender. Figure 2.8a shows the sung and spoken distribution for females speakers and Figure 2.8b for male speakers. It can be seen from both figures that both female and male speakers increased their energy when singing compared to when speaking. Males' sung speech mean intensity raises 6 dB when singing and reduces their



(a) Female speakers.

(b) Male speakers.

Figure 2.8 Histogram of the mean speech intensity distribution from eleven English vowels in the NUS-48E corpus per male and female genders. (a) shows the female and (b) the male distribution per sung and spoken speech styles.

standard deviation by 1.3 *dB*. On the other hand, female speakers increase the mean by 3.7 *dB* and the standard deviation by 1.4 *dB*.

In summary, Figures 2.7 and 2.8, show that for spoken speech, there is little observable energy variation between genders, with both genders presenting a similar distribution. However, both genders increase their level when singing, with the male distribution having a lower standard deviation than the female.

2.4.4 Duration

As discussed in Section 2.3.1, during singing, the breathing periods are more extended than spoken ones, as is required to meet the needs of the musical performance. This breathing period extension translates into stretched sung sentences. This is illustrated in Figure 2.9, where a sung and spoken version of the sentence *"Edelweiss, Edelweiss"* is contrasted. In this figure, the red lines connect the starting time of each phoneme from the spoken and sung versions. The sung versions of the sentence possess large areas that correspond to a single vowel. In contrast, in the spoken version, the same vowels last a fraction of a second. For example, the duration of the vowels [eɪ] and [aɪ] is not greater than 500 *ms* when spoken, but both vowels are stretched to a large degree (more than one second) when singing.

Sharma et al. (2021) reported that word and sentence durations are larger for sung speech than for spoken speech. Their results agree with the observations from Figure 2.9, where a three-second spoken sentence can be stretched to seven seconds when singing. However, as shown in this figure, the degree of stretch depends on the class of the phoneme, i.e., consonants may not be stretched as much as vowels are.

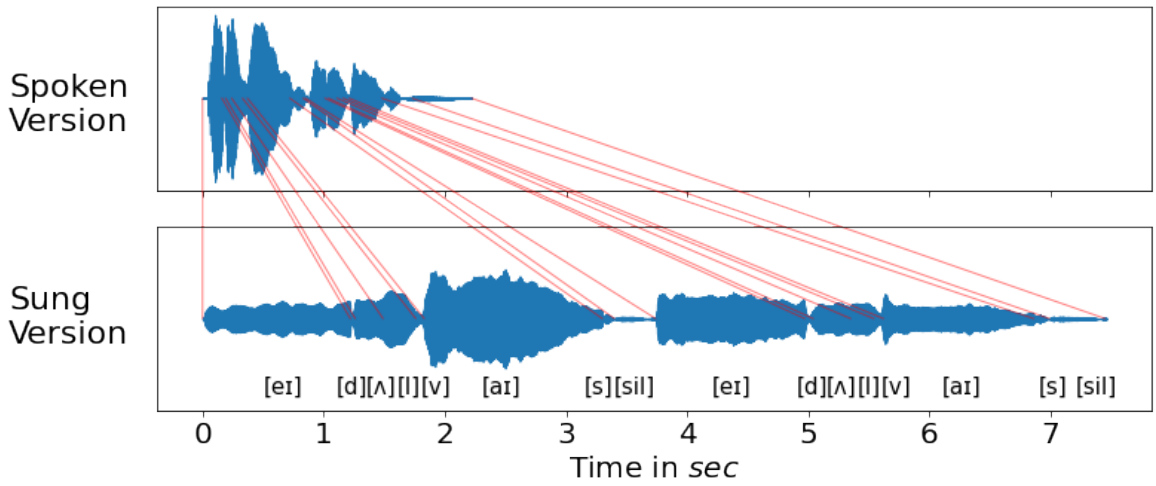


Figure 2.9 A sung and spoken female example of the sentence *"Edelweiss, Edelweiss"* from the song **Edelweiss** from the movie *The Sound of Music* utter by a female speaker. The red lines connect the starting time of each phoneme, utilising the annotation provided in NUS-48E. Additionally, the phoneme annotation is included below the sung utterance.

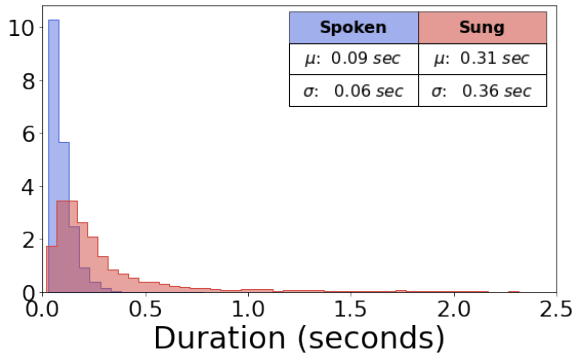
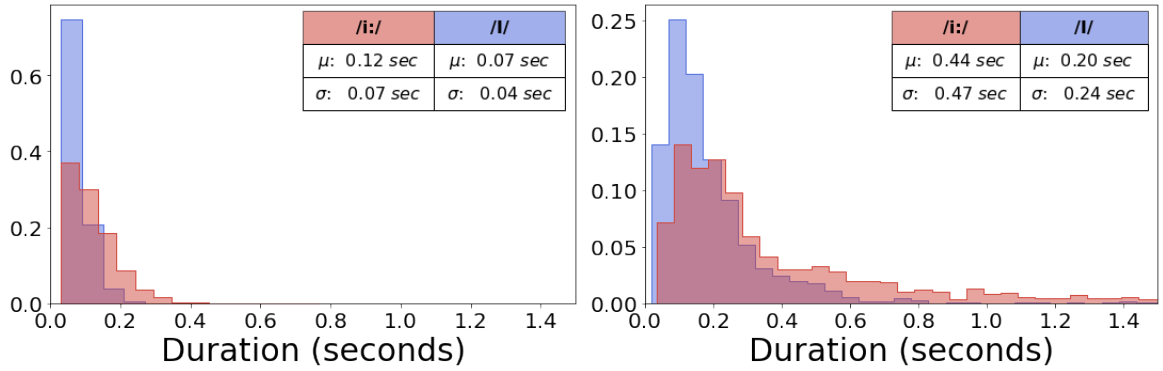


Figure 2.10 Sung and spoken normalised histogram of the duration distribution from the ten vowels present in the NUS-48E dataset.

Firstly, it was analysed how sung vowels' stretch compared with spoken vowels. Figure 2.10 shows the distribution of the sung and spoken duration from the single sound vowels contained in the NUS-48E corpus. The duration was obtained directly from the transcriptions by subtracting the end and start timestamp. The spoken vowel durations are relatively short, averaging 90 *ms* with a standard deviation of 60 *ms*. In contrast, sung vowels are often much longer, averaging 310 ms per vowel segment with a considerable standard deviation of 360 *ms* and a maximum duration over 1000 *ms*.

Vowel length is an important phonetic cue, aiding discrimination between short and long vowel pairs. For example, Figure 2.11 shows the duration distribution of the vowel pair [ɪ] vs [iː]. For spoken speech, most of the short [ɪ] vowels have duration of less than 100 *ms*,



(a) Spoken speech.

(b) Sung speech

Figure 2.11 Duration distribution of [ɪ] vs [iː] short and long vowel pairs per speech style.

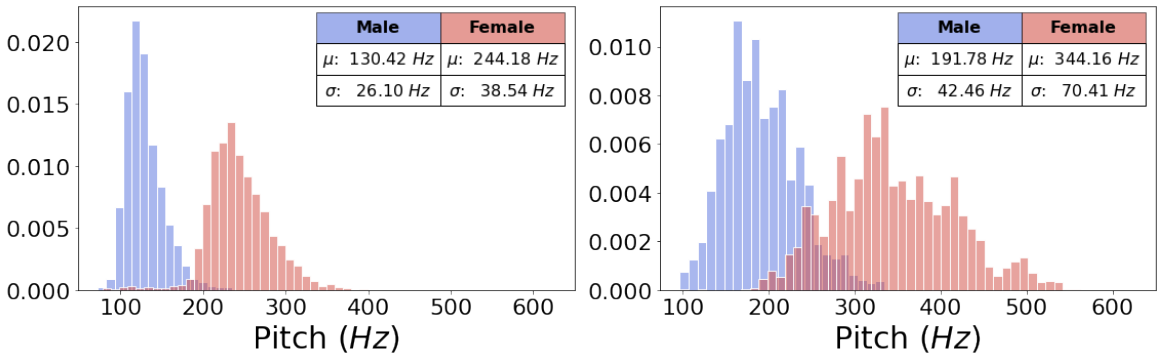
averaging 70 *ms*, while the duration of the longer [iː] is more spread with values up to 400 *ms*, averaging 120 *ms*. However, in sung speech, the vowels' duration specification is lost, with both vowels reaching values up to 800 *ms*, disrupting the correct identification.

In the automatic speech recognition (ASR) task, a vowel's duration can aid in discriminating between short and long vowels, such as "BIT" vs "BEAT". The vowel stretching during singing may invalidate the duration as a discriminator, leading to an increased number of insertion or substitution speech recognition errors.

2.4.5 Pitch

As has been discussed in Section 2.3.2, sung speech has a larger pitch range than spoken speech, with female speakers having a higher pitch range on average. Also, it was discussed that whereas the spoken pitch varies freely in a sentence, the sung speech is expected to have approximately discrete variations.

First, the characteristics of the pitch frequency ranges per gender in sung and spoken speech styles were compared. Figure 2.12 shows the pitch distribution, from ten single sounds vowels from NUS-48E, for speech style and gender. Spoken speech has virtually no overlap between pitch ranges by gender. In contrast, the extended fundamental frequency ranges in sung speech mean that the top end of the male range overlaps with the lower end of female values. The frequency distribution per gender shows different peaks at the position of specific musical notes, e.g., the male singer peaks match the frequencies to the notes E_3, $F_3^{\#}$ and $G_3^{\#}$. Sharma et al. (2021) reported that, in a single song, pitch frequencies tend to concentrate on a discrete number of specific notes, which agree with the results from Figure 2.12.



(a) Spoken speech.

(b) Sung speech

Figure 2.12 Pitch ranges per gender, separated by speech style.

The large sung pitch range implies that, virtually, a singer can have different voices when singing at different frequencies of their vocal range, i.e., singers change the way they articulate the phonemes when singing at different pitches. This can affect the role of different speaker normalisation techniques used in ASR systems (Chapter 4).

In addition to the pitch range, the pitch variation in a word was measured as the difference between the maximum and minimum pitch value during word production. The pitch variation was measured in terms of semitones as is more meaningful in musical terms. To measure the pitch variation in semitones, the maximum and minimum pitch in a word were obtained and then the interval in semitones was calculated as:

$$cents = 1200 \times \log_2 \left(\frac{max\ pitch}{min\ pitch} \right) \tag{2.6}$$

$$semitones = \frac{cents}{100} \tag{2.7}$$

where 1200 cents or 12 semitones correspond to one octave in the musical chromatic scale.

Figure 2.13 shows a box and whisker plot for the pitch variation in semitones in a word from the NUS-48E corpus. The pitch within spoken words varies 17 semitones on average with a standard deviation of 11 semitones. In contrast, for sung speech, the pitch varies on average by little more than one octave with a standard deviation of 7 semitones. The less variable interval in a sung word may be explained by the song performance, where the song structure guides the pitch. Patel et al. (2008) and Vos and Troost (1989) reported more conservative numbers per speech style, but the overall conclusion remains valid; the spoken speech pitch interval within a word is larger than in sung speech.

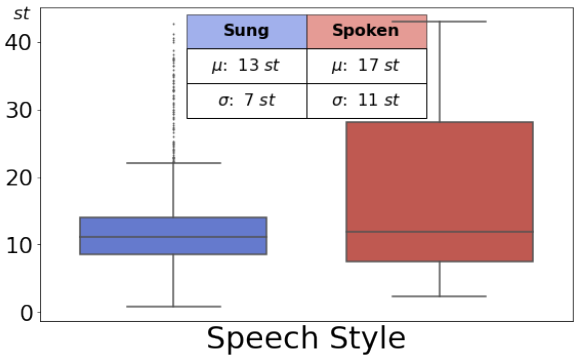


Figure 2.13 Within vowel pitch variation per speech style in semitones.

2.4.6 Formants

As discussed in Section 2.3.3, the formant frequencies of sung vowels are shifted in comparison with their spoken counterpart. The shifting may occur for different reasons, being these reasons different for male and female singers. For example, while male singers develop the singer's formant, high pitch female singers control the F_1 frequency to match the pitch.

The formant frequencies were extracted by first resampling the signal to a sampling frequency twice the frequency of the maximum frequency of the formant search range. Next, a pre-emphasis function is applied. Then, a Gaussian-like window is applied to each frame, and LPC coefficients are computed using Burg's algorithm (Vetterling et al., 1992). Formants were extracted at the middle of each vowel and plotted in F_1-F_2 'vowel space'.

Figure 2.14 shows the vowel space for the spoken frequencies for six vowels; [iː], [e], [æ] [ɑ], [ʌ], and [uː]. In this figure, $F2$ frequencies increase from right to left, and F_1 increases from top to bottom. As discussed before, F_1 values increase by increasing the jaw opening, and F_2 values increase depending on the position of the tongue (explained in Section 2.3.3). This figure shows that, for spoken speech, female speakers have, on average, higher formant frequencies than male speakers. The main reason for these differences is the different vocal tract lengths between genders; females have shorter vocal tracts than males.

When singing, the articulators are not in exactly the same location as when speaking. These differences originate from the musical requirements of the singing, impacting the frequencies of the formants. Figure 2.15a shows the females vowel space for sung and spoken speech. As was discussed before, females tend to open the jaw to control the first formant to match it with the pitch, which shifts the vowel space, as is shown in Figure 2.15a. The males' figure (Figure 2.15b) shows a movement of the vowel space in the opposite direction, with F_1 decreasing. Note that there is a reduction in F_2 in both figures when singing.

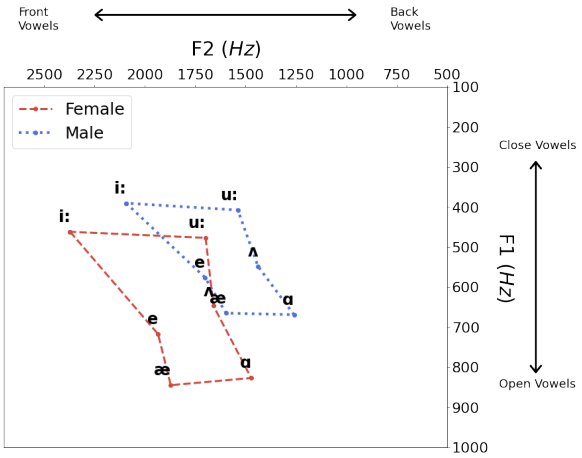
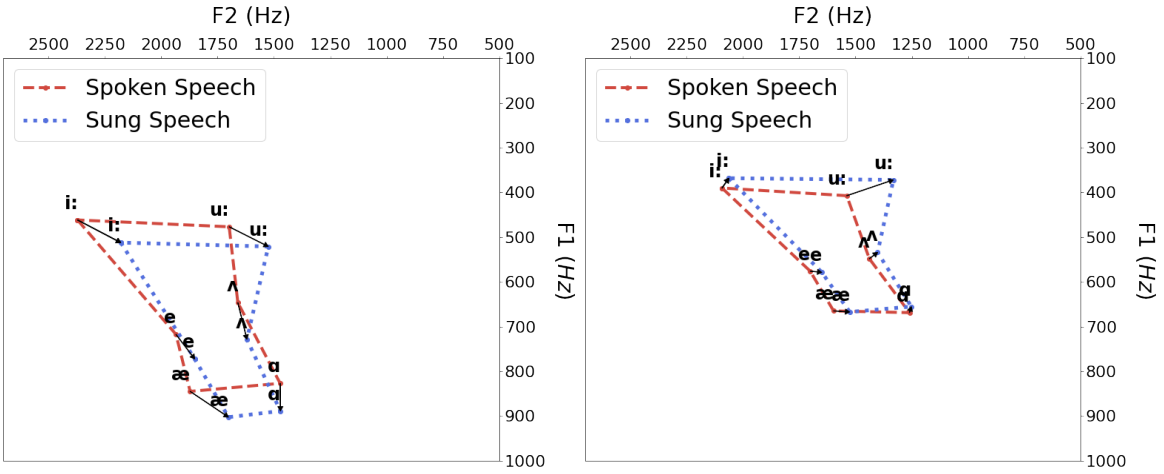


Figure 2.14 F_1 and F_2 vowel space for spoken speech. The figure compares the position of six vowels for female and male speakers.



(a) Female speakers.

(b) Male speakers.

Figure 2.15 Diagram illustrating the sung and spoken F_1 and F_2 average values per gender.

Due to the articulators' movements, the vowel space area (VSA) is modified during singing. VSA is a common metric used to evaluate deficit in speech production and speech intelligibility reduction (Mirarchi et al., 2017; Sandoval et al., 2013). VSA can be calculated by its triangular definition (triangular VSA or tVSA), which corresponds to the Euclidean distance between F_1 and F_2 coordinates for the vowels [iː], [uː] and [ɑ]. In this case, tVSA was utilised to measure how the VSA area varies between speech styles and was calculated as (Liu et al., 2005):

$$tVSA = \frac{ABS\left[F_{1i} \times (F_{2a} - F_{2u}) + F_{1a} \times (F_{2u} - F_{2i}) + F_{1u} \times (F_{2i} - F_{2a})\right]}{2} \tag{2.8}$$

Table 2.8 Triangular vowel space area per gender for sung and spoken speech.

Gender	**Speech Style**	**tVSA**	
		mean Hz^2	**std** Hz^2
Female	Spoken	5.97×10^5	1.50×10^5
	Sung	4.80×10^5	0.87×10^5
Male	Spoken	3.92×10^5	0.87×10^5
	Sung	4.19×10^5	0.53×10^5

where $ABS[\cdot]$ is the absolute value operator, F_{1i} correspond to F_1 for vowel [iː], F_{2a} correspond to F_2 for vowel [ɑ], and so on.

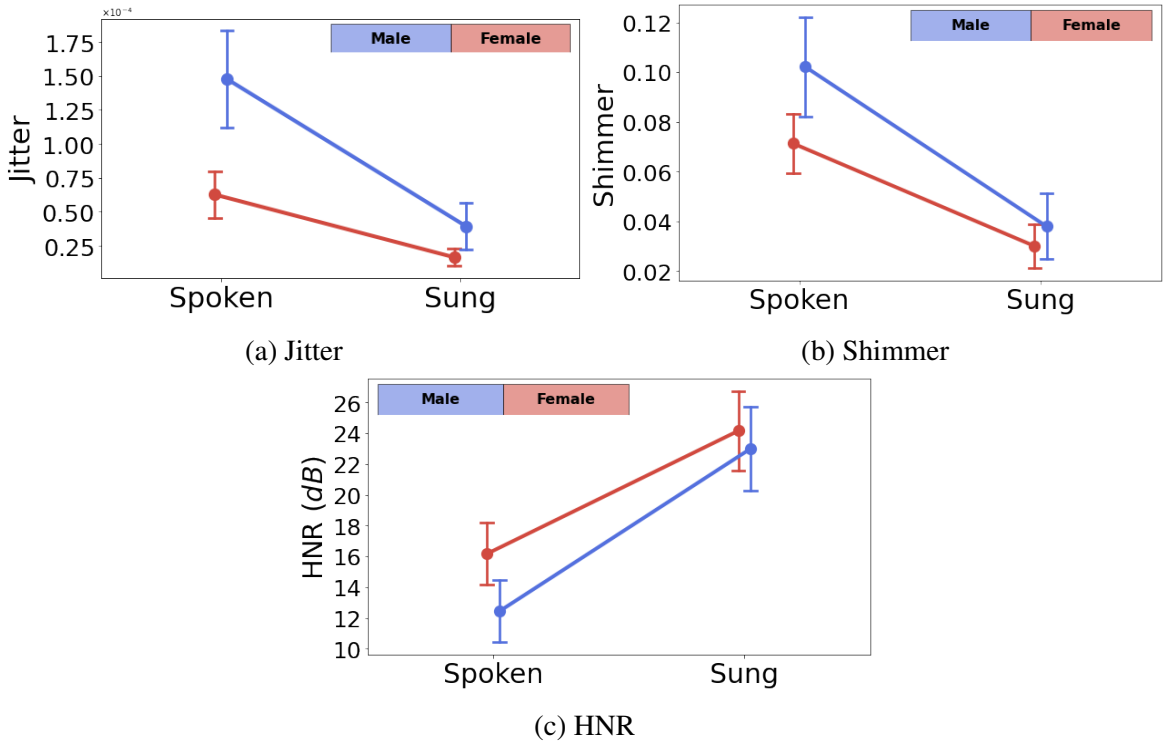
Table 2.8 summarises the mean and standard deviation of the tVSA per gender and speech style. On average, female speakers reduce the VSA when singing. These results agree with the high pitch effect discussed in Section 2.3.3. Females singers tend to increase their jaw opening to tune the formants during singing, moving the vowels closer to each other (de Julián, 2016). In contrast, male speakers increase their vowel area during singing. Both genders reduced the tVSA standard deviation, showing a reduced area variability during singing.

2.4.7 Voice Source Quality

As discussed in Section 2.3.2, parameters like *jitter*, *shimmer* and *harmonic to noise ratio* (HNR), are parameters that are strongly associated with speaker identity. Farrus et al. (2007) reported that jitter and shimmer could be used to improve speaker recognition systems. However, these parameters are very sensitive to factors such as speech styles, pitch frequency and loudness.

The jitter, shimmer and HNR parameters were calculated using the definitions described in Section 2.3.2, and the extraction was performed at the sentence level. Figure 2.16 presents three pair plots showing the jitter (Figure 2.16a), shimmer (Figure 2.16b) and HNR (Figure 2.16c) variation for female and male speakers and speech styles.

Spoken jitter parameter shows a clear distinction between genders, with male jitter being higher and more spread than that of females. When singing, both genders reduces their jitter significantly. Despite the fact that when singing the jitter for both genders decreases and gets closer, it seems that some gender specificity remains. Similar to jitter, the spoken shimmer is higher than the sung shimmer. However, when singing, the shimmer seems to lose the gender specificity. HNR, on the other hand, increases in value when singing by $10\,dB$ for males



(a) Jitter

(b) Shimmer

(c) HNR

Figure 2.16 Voice quality parameters between sung and spoken parameters.

and 8 dB for females speakers, reaching similar values for both genders. This increment is somewhat expected due to the reduction of glottal noise during singing (Lundy et al., 2000).

Jitter, shimmer and HNR parameter ranges showed a clear gender distinction in spoken speech and have been proved to be very useful for speaker recognition systems in spoken conditions (Farrus et al., 2007). However, due to the correlation of these parameters to speech styles and pitch frequency, the gender distinction virtually disappears; only the jitter maintains some degree of gender specificity and, shimmer and HNR become more gender independent. The weaker link to speaker identity may reduce the value of these parameters as speaker normalisation conditioning variables. This will be put on test in Chapter 4.

2.4.8 Beat

In continuous spoken speech, some syllables can be stressed by increasing the energy in that syllable, i.e., the stressed syllable would have greater relative respiratory energy than surrounding syllables. Stress variation is one of the so called 'suprasegmental' features superposed on the syllables. Other suprasegmental features include intonation and duration variation. 'Prosody' (Wennerstrom, 2001) refers to the use of the suprasegmental features to provide information beyond the sentence literal meaning, e.g., to provide emotional

information. Note that a detailed account of prosody is not necessary and is outside the scope of this thesis.

In the English language, the stress variation is commonly used to distinguish between nouns and verbs pairs where the first syllable is stressed for nouns, and the second for verbs, e.g., the words 'impact' and 'compact' are stressed in the first syllable when used as a noun and in the second syllable when they are used as a verb. However, when the words are part of a sentence, the syllables' stress changes depending on how the word is used in the sentence to keep regular or 'isochronous' intervals. For example, the word 'sixteen' is stressed in the first syllable but, in the sentence 'She's only sixteen', the stress of the word sixteen is moved to the second syllable because the word 'only' is stressed in the first syllable (example presented by Conlen (2016)). To maintain a constant interval between stresses, vowel duration can be extended or reduced so that the duration of stressed syllables is longer than the unstressed ones. This is known as "stress-timed" rhythm languages (Abercrombie, 1967; Pike, 1945). Stress-timed languages also include languages like Russian and Arabic. On the other hand, languages like Spanish, Italian and French are defined as "syllable-timed" rhythm languages (Abercrombie, 1967; Pike, 1945). In syllable-timed languages, the distance between stressed syllables varies depending on the number of the inter-stress syllables. In this case, vowels are less likely to be shortened and lengthened with all syllables having similar duration regardless of whether they are stressed or not, i.e., they have isochronous intervals between syllables.

Music has its own rules of rhythm that can operate quite differently from those that apply to spoken speech. In music, the rhythm defines a sequence of strong and weak pulses, called beats, which, together with the texture of the song, creates a guideline that musicians and singers must follow to perform the song (Arnold and Kramer, 2016). In a song composition, beats are grouped into bars and organised by the time signature (how many beats in one bar) and the tempo (how many beats in one minute). Perricone (2018) argue that a good song prosody should match relevant syllables with the strong beats and the whole lyrics phrase with a musical phrase, which makes the performance of the song more natural for the singer.

Bella et al. (2015) reported that non-professional singers with accurate pitch production tend to be precise and consistent in synchronising to a predictable series of beats. Figure 2.17 shows the spectrogram of a five-second fragment from one performance of the song *Edelweiss* from the NUS-48E sung and spoken corpus (Duan et al., 2013). The song has a time signature of three beats per bar. In this figure, each beat is identified with vertical red-dashed lines. It can be seen that the beginning of each word matches with the location of one beat, presumably a strong one, and the duration of three beats holds a whole word.

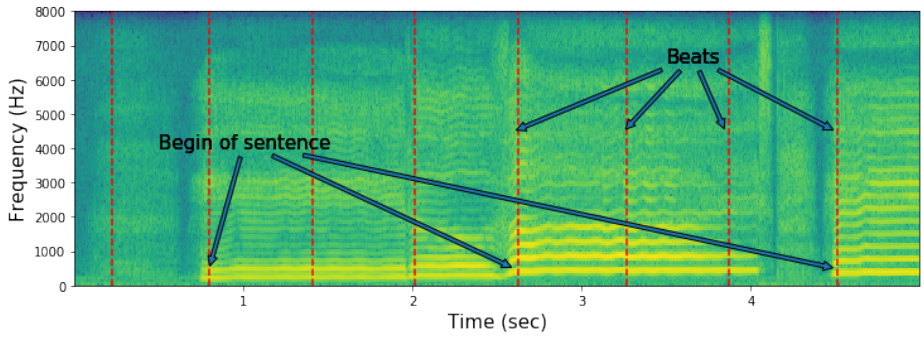


Figure 2.17 Spectrogram with the beat time and words boundaries from a five seconds excerpt from the song Edelweiss from the NUS-48E Sing and Spoken Lyrics corpus Duan et al. (2013).

Informing sung speech recognition systems with beat cues may help them learn information related to the song structure, the song's tempo and the word stress.

2.5 The Impact of Musical Accompaniment

Spoken speech communication often occurs in noisy environments where interfering sources accompany the target speech (e.g., noise or competing speech). Knowing the ratio between the average speech power to the power of the background noise – the signal-to-noise ratio (SNR) – is important because the background affects the ability to understand the message by masking the speech signal. SNR is often expressed in decibels (dB), and a ratio higher than $0\,dB$ indicates a higher speech level. In music, sung speech is typically found in the presence of instrumental background accompaniment. The degree of speech masking by the instrumentation may depend on the genre and the instruments that make up the background accompaniment.

In this section, I am conducting a novel analysis of the local impact of the background accompaniment on the sung speech intelligibility. This impact belongs to the music- and word-setting category that groups factors that affect the sung speech intelligibility related to the song structure and composition. The analysis is performed using the MedleyDB multitrack recordings corpus (Bittner et al., 2014), where the vocal segment of each song is provided in isolation from the corresponding background accompaniment.

2.5.1 Dataset Description

The MedleyDB corpus (Bittner et al., 2014) is an audio database of royalty-free multitrack recordings aimed to support research on MIR tasks, such as instrument recognition, audio

source separation and automatic mixing by providing annotations of melody F_0 and instruments activation along with the audio recordings. In its first release, MedleyDB contained 122 multi-genre full-length songs where the instruments are provided in an isolated manner in independent channels. Seventy-four new tracks were introduced in the second release (Bittner et al., 2016), totalling 196 songs, from which only 86 songs contain vocals.

MedleyDB contains songs from multiple sources: independent artist, NYU's Dolan Recording Studio[7], Weathervane Music[8] and Music Delta. During recording sessions, a set of microphones were used to record the different sources separately (these recordings are referred to as a *raw* track). All raw tracks from a single instrument were then grouped into stereo *stems*. Stems include all raw recordings from one instrument (e.g., raw tracks for *kick drum*, *snare drum* and *toms* are grouped as *drum set* stem), possible effects processing, gain control and panning.

MedleyDB provides metadata information for each song, including the artist's name, name of the song, musical genre, dictionary of raw and stem tracks with the corresponding instrument, and mixing coefficients per stems. The genre label is somewhat subjective, and songs classified as " Pop" or "Singer/Songwriter" may not be classified in this way in another dataset. For the genre label, there are nine possible values ("Singer/Songwriter", "Classical", "Rock", "World/Folk", "Fusion", "Jazz", "Pop", "Musical Theatre", "Rap").

Figure 2.18 shows an example of the hierarchy structure for recordings of a jazz quartet. In this figure, the double bass and trumpet stems are constructed from a single raw recording. However, the piano and drum set stems are constructed by the addition of several raw recordings. For each song, the mix, stems and raw audios are provided in *WAV* format, $44100\,Hz$ and 16bit.

2.5.2 Methodology

MedleyDB provides each instrument – including *male* and *female singers* (vocals) – in independent stems, making it ideal for acoustically analysing the balance and interaction between the background accompaniment and the sung speech across genders and musical genre. Genres Rock, Pop and Singer/Songwriter are relabelled as "Pop/Rock" to match with the labels used by Condit-Schultz and Huron (2015).

A *background accompaniment* track was constructed by adding all instrument stems, excluding female and male singers. This new grouping is illustrated in Figure 2.19. Figure 2.19a shows the original tracks provided composed of four stems: drum set, electric bass, electric guitar and male singer. Figure 2.19b shows the background accompaniment "analysis

[7] https://steinhardt.nyu.edu/programs/music-technology/facilities
[8] https://weathervanemusic.org/

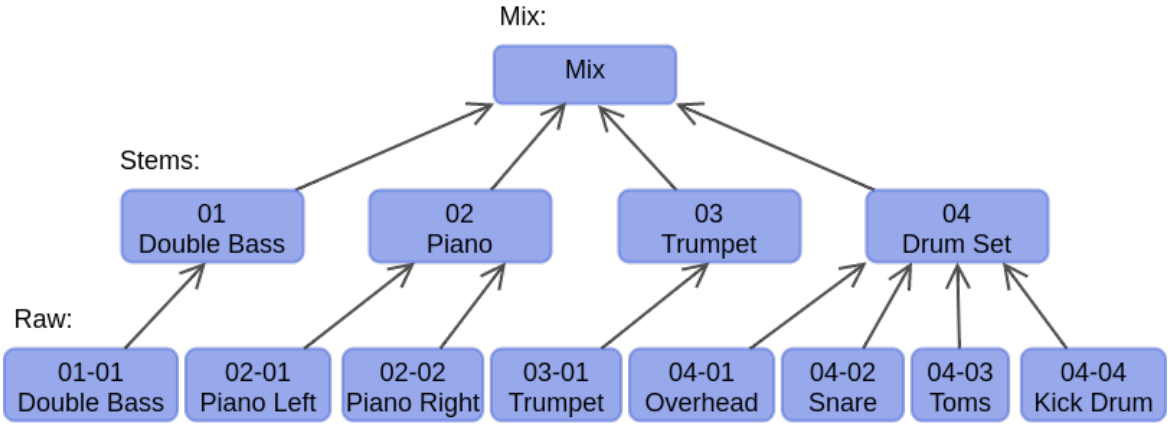
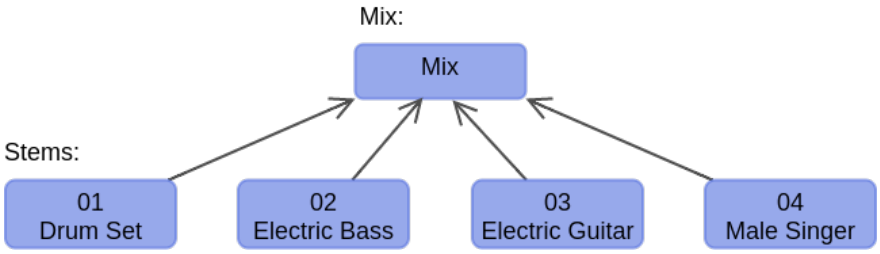
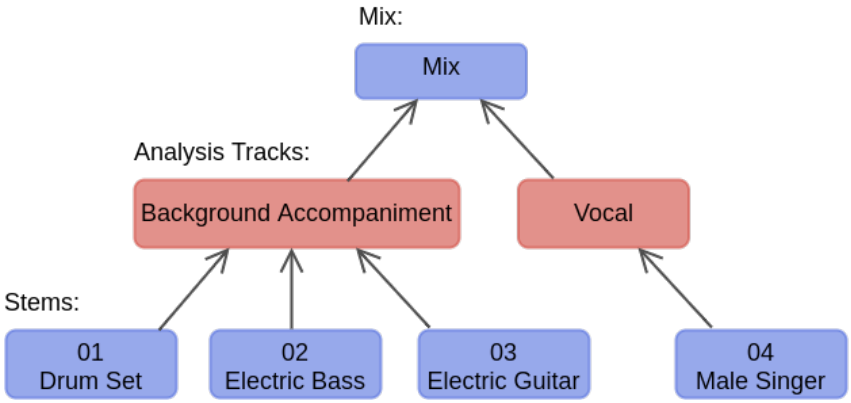


Figure 2.18 Diagram with the hierarchy of audio files for a jazz quartet (as presented by Bittner et al. (2014, 2016))



(a) Original stem hierarchy.



(b) Regrouping of stems hierarchy.

Figure 2.19 Diagram with the hierarchy of mixture audio files for a punk song from MedleyDB dataset (Bittner et al., 2014, 2016). (a) Shows the stem hierarchy as defined by MedleyDB, and (b) shows the hierarchy for the construction of the tracks for the background accompaniment impact analysis.

track" constructed by adding the instruments stems (i.e., drum set, electric bass, electric guitar) using the original mixing coefficients provided with the corpus' metadata.

The following analysis considers using only the vocal and background accompaniment analysis tracks.

2.5.3 Glimpse Analysis

The intelligibility of the message may be affected by the balance between the speech and background noise. Specifically, the background has an *energetic masking* effect on the speech, which occurs when the background noise energy is greater than the speech energy in some spectro-temporal region. Barker and Cooke (2007) investigated the correlation between spoken speech intelligibility and energetic masking using a "glimpsing" model to characterise the degree of masking (Cooke, 2006). This model exploits the fact that there will be regions where the speech energy is greater than the noise energy, even in unfavourable SNRs. In this model, a "glimpse" is defined as a spectro-temporal region in which the target signal is least affected by the background. The model measures the percentage of spectro-temporal glimpses over the total of spectro-temporal points.

The model begins by computing an auditory spectrogram representation of the audio data (*ratemap*) (Brown and Cooke, 1994) for the speech signal and the noise used to construct the mixture. The ratemap computation follows the procedure described by Cooke (2006) with the parameters utilised by Barker and Cooke (2007). A bank of 64 gammatone filters processes the audio signal. The filters are linearly spaced on an ERB-rate scale between frequencies ranging from 50 Hz and 8,000 Hz. The Hilbert envelope is computed and subjected to a leaky integrator with an 8 *ms* time constant within each channel. The output consists of 64 points sampled at 100 Hz.

Given the ratemap for the speech and noise signals, the glimpse proportion is computed by comparing the speech and noise spectra and finding the local points where the speech SNR is greater than the noise SNR by a threshold of T dB. The resulting spectro-temporal plane results in a series of connected regions (points are connected if they are adjacent in the same channel at consecutive time frames or adjacent in the same time frame in consecutive channels). As it is unlikely that listeners can detect very small glimpses areas, connected regions smaller than a threshold of N spectro-temporal points are discarded as unreliable. The parameters T and N were set to 3 dB and 5 points, respectively, equalling the values empirically found and used by (Barker and Cooke, 2007).

The glimpse proportion (GP) is computed as:

$$GP = 100 \times \frac{Area_{glimpsed}}{Area_{spectrogram}}\% \tag{2.9}$$

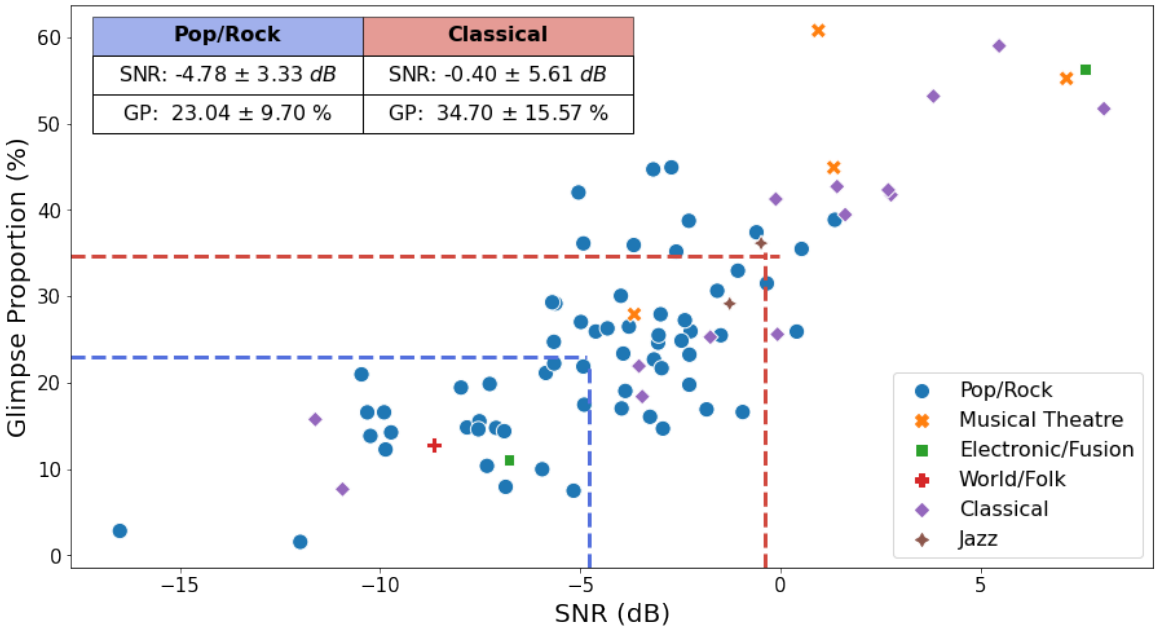


Figure 2.20 Glimpse proportion from the original mixtures from MedleyDB as a function of the original SNR of the sources.

where $Area_{glimpsed}$ is obtained by counting the number of spectro-temporal points from all glimpses, and $Area_{spectrogram}$ is the number of frequency channels multiplied by the number of frames. Glimpse proportion varies between 0% (the speech is completely masked) and 100% (the speech dominates across the entire spectro-temporal plane).

The same procedure and parameters from Barker and Cooke (2007) were utilised to calculate the glimpse proportion of musical recordings from MedleyDB under 'real' conditions, i.e., utilising the SNR used in MedleyDB. To ensure that the glimpse computation was calculated only in the speech areas, a voice activity detection algorithm was utilised to filter out the solo-background areas (voice activity detection described in Section 4.2.4), resulting in several singing segments per track. First, the glimpse proportion was computed for each segment. Then, the scores from the segments were averaged, weighted by the segment length, obtaining a glimpse proportion per song. (Equation 2.10).

$$\mu_{weighted} = \frac{\sum_{i=1}^{n} x_i W_i}{\sum_{i=1}^{n} W_i} \tag{2.10}$$

2.5.4 Results

Figure 2.20 shows the glimpse proportion as a function of the SNR of the songs for the 86 tracks in MedleyDB containing vocals. The SNR of the dataset ranges from -16.5 to 8.1 dB

with an average of 3.6 dB and a standard deviation of 4.4 dB. However, some genres show more beneficial SNR than others. For example, "Pop/Rock" songs' SNR ranges from -16.5 to 1.35 dB with an average of -4.8 dB and 3.33 dB standard deviation. However, "Classical" songs showed a higher SNR, ranging from -11.6 to 8.1 dB and averaging -0.4 dB with a 5.6 dB standard deviation. In spoken speech scenarios, speakers tend to increase their vocal effort to be heard in increasingly louder environments (Lane and Tranel, 1971). Pearsons et al. (1976) recorded several samples of spoken speech and environmental noises from different locations ('Schools', 'Homes', 'Hospitals', 'Department Stores', 'Trains' and 'Aeroplane'). This data provides speech levels and SNRs in different everyday scenarios. The main results of this work are summarised by Olsen (1998). In most scenarios, speech levels were between 55 and 66 dB SPL, maintaining an SNR between 5 and 15 dB. The variations of the speech level are explained by the Lombard effect (Lane and Tranel, 1971). Pearsons et al. (1976) reported that the speech level is increased by 0.6 dB for each 1 dB increment in noise level. The speech level was considerably higher in train and aeroplane conditions, reaching 73 and 77 dB SPL. However, the background noise level was even higher than the spoken level, resulting in lower SNRs of -1 and -2 dB for train and aeroplane, respectively.

As shown in Figure 2.20, the proportion of glimpses computed from the MedleyDB dataset ranges between 1.5 to 60.8%, with an average of 26.4% and a 13% standard deviation. As expected, "Pop/Rock" songs have a lower glimpse proportion than "Classical" songs due to the lower SNR of "Pop/Rock" music. "Pop/Rock" glimpse proportion averaged 23% with a standard deviation of 9.7%, and "Classical" song averaged 34.7% with 15.6% standard deviation. Barker and Cooke (2007) investigated the correlation between the glimpse proportion of artificially mixed spoken speech samples with speech-shaped noise and the word recognition from 20 native British English speakers. They reported that at -4 dB SNR, the intelligibility reaches 80-90%, with less than 10% of glimpse proportion. However, the correlation between glimpse proportion and intelligibility may not be as straightforward in music as it seems to be in spoken speech. Condit-Schultz and Huron (2015) reported intelligibility of 48% for "Classical" music and a higher 70% for "Pop/Rock".

2.6 Summary and Conclusion

This chapter reviewed the differences between sung and spoken speech styles production and analysed how these differences affect the message's understanding, addressing **RQ.1**. The chapter began with the presentation of several sung segments representative of common lyrics mishearing, illustrating some of the challenges of the task. Next, the chapter reviewed the operation of the sound production mechanisms, highlighting how the different mechanisms

(breathing, voicing and filtering) differ for singing and talking and how these differences affect the sung speech intelligibility. For example:

In the breathing system, both speech styles differ in using their vital capacity (VC) volume. Singers tend to use the total VC, stretching their sounds further than spoken speech (Bouhuys et al., 1966). This requires singers to control and even optimise their VC by reducing the RV (Gould, 1977). Additionally, both speech styles present different subglottal pressure, being the range of the sung speech pressure between 2 and 50 cmH_2O (Proctor, 1980) and between 5 and 15 cmH_2O Sundberg (1987) for spoken speech.

In the voicing system, the sung speech pitch range is larger than in spoken speech. Male speakers range between 65 and 260 Hz, but their range increases to between 82 and 392 Hz when singing–similarly, female range between 100 and 525 Hz when speaking and 171 and 880 Hz when singing.

In the filtering system, male operatic singers present a special formant frequency at around 3000 and 4000 Hz called singer's formant (Bartholomew, 1934), and when the sung pitch is higher than F_1, female singers resort to formant tuning technique to match F_1 with the pitch, increasing the energy of the formant. Both the singer's formant and formant tuning alter the way that phonemes are articulated from how they are articulated in spoken speech, reducing the intelligibility. Additionally, spoken accents can vary significantly between regions (Grabe et al., 2000), however, in sung speech, native English speakers tend to neutralise their accent (Gibson, 2010), with a tendency to American pronunciation (Konert-Panek, 2017), and non-native English speakers also have a tendency to neutralise their accent (Hagen et al., 2011; Mageau, 2016).

The rest of the chapter was dedicated to analysing the acoustic differences between sung and spoken speech signals. First, the goal was to understand how the sung and spoken speech signals differ to inform automatic speech recognition systems. For this, the NUS-48E sung and spoken speech parallel corpus (Duan et al., 2013) was utilised to compare the differences under "clean" conditions, i.e., sung and spoken lyrics recorded in isolation.

Last, using the MedleyDB dataset (Bittner et al., 2014, 2016), this chapter addressed **RQ.2** by utilising a glimpse proportion (GP) model to analyse the masking effect of the background musical over the sung speech signal. It was concluded that a GP model might be a good indicator of intelligibility within a genre but not between songs of different genres. The next chapter will review the current literature on automatic speech recognition and audio source separation in the context of music.

Chapter 3

Background and Related Work

The previous chapter discussed several differences between spoken and sung speech. These differences, including extended sung vowel duration, pitch range and pitch variation within a vowel, make the sung speech less intelligible than spoken speech.

This thesis is concerned with studying the differences between sung and spoken speech and applying these differences to adapt spoken speech technologies to sung speech. Unlike modern deep learning-based ASR systems, where the systems jointly learn the acoustic model and the language model, previous work have used classical approaches where the recognition system is subdivided into separately-trained acoustic and language models.

The present chapter starts by introducing the principles and terminology of traditional ASR systems based on the definitions and descriptions from Jurafsky (2009) and Vincent et al. (2018). This introduction will help to better comprehend how ASR systems operate and therefore how can best adapt spoken speech systems to sung speech. Additionally, future chapters will refer to these techniques when developing sung speech recognition systems. Then, the chapter presents a review of previous investigations of sung speech recognition and of music source separation for vocal separation.

The chapter is organised as follows. Section 3.1 presents a description of the different components comprising a traditional ASR system. Section 3.2 reviews previous investigations concerning the recognition of sung speech from an audio signal. Section 3.3 presents a review of previous work on music source separation designed to separate the singing segment from the background accompaniment. Last, Section 3.4 summarises the key content of the chapter.

3.1 The Basis of Automatic Speech Recognition

Speech recognition refers to the task of finding the most probable word sequence, $\mathbf{W}$, from all possible word sequences in the language $\mathscr{L}$ given an acoustic signal. The signal can be

represented as a length-N sequence, $\mathbf{O} = [\mathbf{o}_1, \mathbf{o}_2, ..., \mathbf{o}_{N-1}]$, of speech feature vectors $\mathbf{o}_n$. Note that representing the signal as a series of feature vectors is a typical first step. However, some end-to-end systems may not need this step as they can operate directly with the signal. The recognition problem can be formulated according to Bayes' decision theory as:

$$\hat{\mathbf{W}} = \underset{\mathbf{W} \in \mathscr{L}}{\arg\max}\, P(\mathbf{W} \mid \mathbf{O}) \tag{3.1}$$

where $\hat{W}$ is the predicted word sequence.

However, directly computing $P(\mathbf{W} \mid \mathbf{O})$ is a very difficult task to solve as it considers all possible word sequences. End-to-end systems are designed to solve this problem directly. However, they require a large amount of training data, which is scarce and hard to collect. Instead, in a conventional system, using Bayes' rule, this problem can be re-expressed as:

$$\hat{\mathbf{W}} = \underset{\mathbf{W} \in \mathscr{L}}{\arg\max}\, \frac{P(\mathbf{O} \mid \mathbf{W})P(\mathbf{W})}{P(\mathbf{O})} \tag{3.2}$$

where $P(\mathbf{W})$ is the prior probability of the word sequence that can be estimated by a 'language model' (LM) and $P(\mathbf{O} \mid \mathbf{W})$ is the likelihood of the observation vector given a phoneme sequence estimated using an 'acoustic model' (AM). $P(\mathbf{O})$ is the probability of the acoustic observation. Note that $P(\mathbf{O})$ can be omitted as it is independent of the word sequence, i.e., it is a constant scalar across all word sequences and so does not effect the outcome when applying the argmax function. Therefore, the problem can be reformulated as:

$$\hat{\mathbf{W}} = \underset{\mathbf{W} \in \mathscr{L}}{\arg\max}\, \overbrace{P(\mathbf{O} \mid \mathbf{W})}^{\text{Acoustic Model}} \underbrace{P(\mathbf{W})}_{\text{Language Model}} \tag{3.3}$$

Figure 3.1 shows a schematic diagram of a high-level overview of a general automatic speech recognition (ASR) system that solves Equation 3.3. Typically, these systems start with a *front-end* stage that converts a speech signal into feature vectors. The acoustic model computes the likelihood of the observed feature vectors given some linguistics units, i.e., words or phones. The language model expresses the probability of a given sequence of words in the language $\mathscr{L}$. Finally, the decoding stage takes the acoustic model, the language model, and the sequence of observations, and uses a search algorithm to estimate the most probable sequence of words.

ASR systems are typically designed to be speaker-independent, i.e., they are trained using speech recordings from several speakers (or singers in the case of sung speech recognition) to recognise the speech regardless of who is speaking (or singing). However, the speech characteristics of the training data may be different from those in the testing data, creating

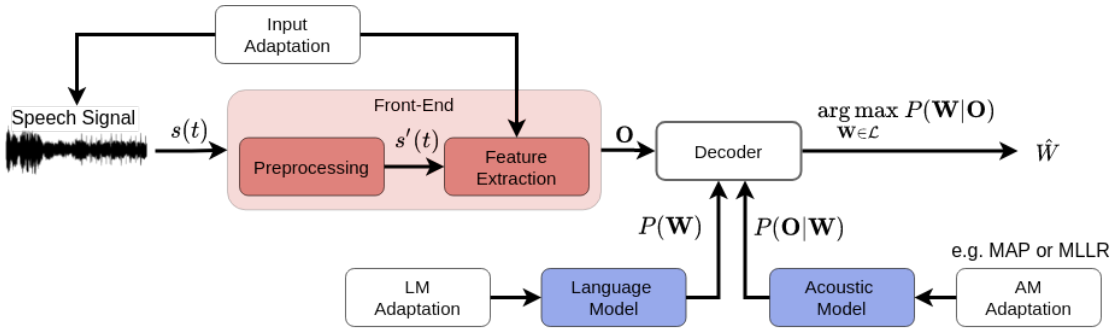


Figure 3.1 Schematic diagram of a high-level overview of a general Automatic Speech Recognition system.

a mismatch between both sets. This variability between the training and testing sets can be reduced by employing an *adaptation* process. Different adaptations can be applied to the various stages in a traditional ASR system. For example, speed perturbations can be applied to the speech signals before presenting them to the front-end stage to train models under different speech rates. Acoustic model adaptations can be employed to adapt the acoustic model to certain conditions, such as speaker adaptation that can adjust the model's parameters to recognise the speech of a target speaker better. Last, language models can be adapted by conditioning them to specific grammatical structures.

3.1.1 Front-End

The front-end stage takes a length-T speech signal, $s(t)$, as input and returns a length-N sequence of observation vectors, $\mathbf{O}$. This stage may begin with any necessary steps to normalise the signal for compatibility. The normalisation steps aim to reduce uninformative sources of variability between different speech signals. For example, different signals might have a different number of channels (e.g., mono, stereo, or more channels), different sample rate (e.g., 8000 Hz, 16000 Hz, or 32000 Hz), different energy (e.g., variation of the mean volume of the speakers), or they may come from different age or gender speakers (e.g., spectral variations due to differences in vocal tract length). In this project, the normalisation process consists of transforming the audio signal into a single channel by averaging all existing channels (typically from stereo to mono channels) and down-sampling the signal to 16000 Hz. Next, if the audio signal is composed of a mixture of speech and background noise, audio source separation or speech enhancement techniques can be applied to isolate the speech before extracting the features. Then, given the preprocessed signal, the observation sequence is generated by applying a transformation function $f(\cdot)$. Speech signals can be

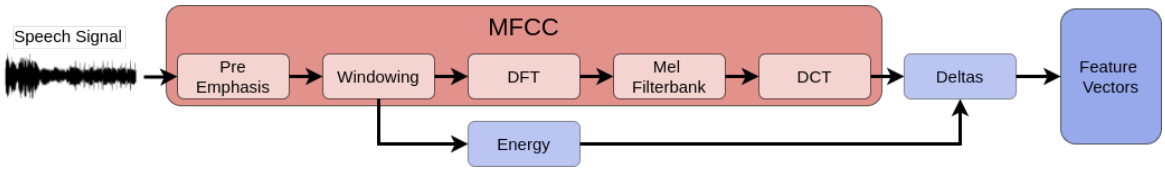


Figure 3.2 Schematic diagram of the Mel-frequency cepstral coefficients features extractions.

assumed to be quasi-stationary signals, i.e., the properties of the speech can be approximated as being constant over a short period. Under this assumption, many speech features represent the acoustic information in small windows of the signal (typically, spoken ASR systems use windows of 25 *ms*). By far, the most common speech features in ASR systems are the Mel-frequency cepstral coefficients (MFCC). Note that speech recognition systems are not restricted to the use of MFCC and other spectral representation options include linear prediction coefficients (LPC) (Atal and Schroeder, 1967), perceptual linear prediction (PLP) (Hermansky, 1990) and warped linear predictive coefficients (WLPC) (Strube, 1980). All these speech features try to estimate the shape of the smoothed spectrum to infer the configuration of the vocal tract.

Figure 3.2 shows a typical feature vector used in traditional ASR systems. Such vectors are typically composed of the concatenation of MFCCs and energy features, plus the first and second temporal derivative of both. The description of how these features are computed is presented below.

Mel-frequency cepstral coefficients

The extraction of MFCC features is divided into several steps, including windowing the signal, applying the DFT transform, warping the frequencies into the Mel scale and applying the DCT transform.

1. **Pre-Emphasis:** Speech sounds, like vowels, have more energy at the lower frequencies. The pre-emphasis step aims to enhance the amount of energy in the higher frequencies. The objective is to make the information from the higher formants more available to the acoustic model. The pre-emphasis is done by using a first-order high-pass filter:

$$s'(t) = s(t) - \alpha s(t-1) \tag{3.4}$$

where $s(\cdot)$ is the signal in the time domain, $0.9 \leq \alpha \leq 1$, and $s'(\cdot)$ is the pre-emphasised signal.

2. **Windowing:** Speech signals are non-stationary signals; that is, their statistical properties vary over time. However, we can assume stationarity in very small windows of speech. The MFCCs' extraction is typically computed on 25 *ms* length frames obtained every 10 *ms*. This process abruptly cuts off the signal at its boundaries, creating discontinuities that produce problems when performing Fourier analysis. To alleviate this, a window function that shrinks the values towards zero at the frame borders is applied for each frame, avoiding the discontinuities.

   The signal is extracted by multiplying the signal at time n, $s[n]$ by the value of the windows at time n, $w[n]$:

   $$y[n] = w[n]s[n] \tag{3.5}$$

   One of the most common window functions is the Hamming window. Assuming a window with length L frames, the Hamming window is defined as:

   $$w[n] = \begin{cases} 5.54 - 0.46\cos\left(\frac{2\pi n}{L}\right) & , \ 0 \leq n-1 \leq L-1 \\ 0 & , \ otherwise \end{cases} \tag{3.6}$$

3. **Discrete Fourier transform:** The discrete Fourier transform (DFT) function transforms the time domain signal to the frequency domain components, extracting the spectral information from the windowed signal, i.e., the amount of energy that the signal contains at different frequency bands. The input of the DFT is a windowed signal $x[n]....x[m]$. DFT is computed as:

   $$X[k] = \sum_{n=0}^{N-1} x[n] exp\left(\frac{-j2\pi kn}{N}\right) \quad , \ 0 \leq k \leq N-1 \tag{3.7}$$

   where $X[k]$ is a complex number representing the magnitude and phase at each frequency band and N is the number of points to compute the DFT. The 'spectrum' can be visualised by plotting the magnitude against the frequency.

4. **Mel-filterbank:** Human hearing is most sensitive to frequencies lower than 1000 Hz. Modelling the human hearing properties for features extraction improves speech recognition performances. The intuition is to create a bank of bandpass filters that

collect energy from each frequency band. For MFCC features extraction, this is performed by, first, warping the frequencies output by the DFT to the Mel units (Stevens et al., 1937), defined as:

$$Mel(f) = 1127ln\left(1+\frac{f}{700}\right) \tag{3.8}$$

Then, triangular filters are used to create triangular Mel-filterbanks. This filterbank is typically conformed of ten linearly spaced filters below $1000\,Hz$ and logarithmically spaced filters above $1000\,Hz$. The Mel-filters are constructed by computing the absolute square DFT value to obtain the DFT spectrum. Then, triangular band-pass filters are applied to obtain the Mel-scale power spectrum.

$$s(m) = \sum_{k=0}^{N-1}\left[|X_t(k)|^2 H_m(k)\right] \tag{3.9}$$

where

$$H_m(k) = \begin{cases} 0 & , k < f(m-1) \\ \dfrac{k-f(m-1)}{f(m)-f(m-1)} & , f(m-1) \leq k \leq f(m) \\ \dfrac{f(m+1)-k}{f(m+1)-f(m)} & , f(m) \leq k \leq f(m+1) \\ 0 & , k > f(m+1) \end{cases} \tag{3.10}$$

where M is the number of filters $1 \leq m \leq M$, k is the DFT bin number $1 \leq k \leq N$, $f(\cdot)$ is the list of $M+2$ Mel-spaced frequencies.

5. **Log scaling:** The log is computed on the mel spectrum values to compress the dynamic range, which is very large in the power domain. The log makes the features less sensitive to power variations like the proximity of the speaker's mouth to the microphone. Additionally, the use of logarithm simplifies the operations as it transforms the multiplication of very small numbers into simple addition. Also, the log scale means that the source and the filter that have been multiplied in the energy domain are now additive and can be separated by filtering, i.e. keeping the low quefrency[1] MFCC

[1] Quefrency is defined as the inverse of Fourier transform, measured in seconds

coefficients and discarding the high quefrencies (this process is also know as liftering) (Bogert et al., 1963).

6. **Discrete cosine transform:** Applying the discrete cosine transform (DCT) log Mel-filterbank vector results in a set of cepstral coefficients (the spectrum of the log spectrum), defined as:

$$c(n) = \sum_{m=0}^{M-1} log_{10}(s(m)) \cos\left(\frac{\pi n(m-0.5)}{M}\right) \quad , \; n = 0,1,2,...,C-1 \tag{3.11}$$

where C is the number of MFCCs, $c(n)$ is the cepstral coefficient n, M is the number of filters $1 \leq m \leq M$.

This process decorrelates the log Mel spectrum features, separating the source and filter influence. The low quefrency coefficients will be predominantly describing the filter and the high quefrency ones the 'less important' source, which is why only the first 13 are typically kept. The decorrelation is also an important step for Gaussian mixture model (GMM) based acoustic models with diagonal covariance. Note that the zeroth coefficient, $c(0)$, is often excluded as it represents the average log-energy of the signal, which only carries little speaker-specific information.

Energy

Despite that the zeroth MFCC's coefficient may be excluded, the energy of the frames correlates with the phone identity. For example, vowels and fricative sounds have more energy than stop sounds. Therefore, engineered energy features are typically included. The frame energy can be computed as the sum over time of the power of the samples in the frame. For a signal, $x[t]$, in a window from time sample t_1 to time sample t_2, energy can be defined as:

$$Energy = \sum_{t=t_1}^{t_2} x^2[t] \tag{3.12}$$

Deltas

MFCC features are static information computed from each frame. However, speech is not constant and has fluctuations from frame to frame, which is a very useful phone identity cue. 'Deltas' (velocity) are the first derivative of the MFCC features providing the rates of variation of the coefficients. 'Delta-deltas' (acceleration) are computed as the first derivative of the delta features. Deltas can be computed using a linear regression over several frames:

$$d(t) = \frac{\sum_{n=1}^{N} n\,(c(t+n) - c(t-n))}{2\sum_{n=1}^{N} n^2} \tag{3.13}$$

where $d(t)$ is the delta coefficient from the frame t and N is the number of consecutive frames used for computation.

3.1.2 Acoustic Modelling

The acoustic model, $P(\mathbf{O} \mid \mathbf{W})$, estimates the match between the acoustic observations, $\mathbf{O}$, and the word sequence, $\mathbf{W}$. This expression is very complicated to solve as it has to deal with all possible pronunciations of the word sequence. However, since speech possesses a temporal structure and can be encoded as a sequence of observations, hidden Markov models (HMM) become the natural framework for modelling. Acoustic modellings for traditional ASR systems are typically based on HMM with Gaussian mixture models (GMM) or deep neural networks (DNN) for emission probability function; these are then known as GMM-HMM or DNN-HMM, respectively.

The HMMs are statistical models of Markov processes, i.e., a sequence of states where the probability of each state only depends on the previous state. An HMM models a sequence of observations assuming an underlying set of unobservable states (hidden). In HMM-based acoustic modelling, transitions between states are constrained from left-to-right, i.e., the states can only transition to themselves (remain in the state) or to the successive states.

Hidden states can be used to model different words. However, we are not interested in words but rather in sequences of words, which are HMM models made of the concatenation of words' HMM where the language model provides the transition probabilities.

HMMs for ASR are defined by the following components:

- $\mathbf{O} = [o_1, o_2, ..., o_T]$, a sequence of $\mathbf{T}$ observations, where each observation is the spectrum and energy information at a certain point in time.
- $\mathbf{Q} = [q_1, q_2, ..., q_N]$, a set of N hidden states, corresponding to phones or subphones.
- $\mathbf{A} = [a_{11}, a_{12}, ..., a_{n1}, ..., a_{nn}]$, a transition probability matrix where each a_{ij} represents the probability of each subphone taking a self-loop of going to the next subphone.
- $\mathbf{B} = b_i(o_t)$, a sequence of emission probabilities, each expressing the probability of an observation o_t being generated from the subphone state i. GMM or DNN models can be used for emission probability functions.
- $\mathbf{q_0}, \mathbf{q_{end}}$, special start and end states that are not associated with observations.

Note that when HMM phone models are trained without context (i.e., context-independent models), they are called *monophones*. Models can be trained taking phonetic neighbours into account (i.e., context-dependent models). When the models take one previous or the following phone as context, they are called *biphone*. However, the most common models used are *triphones*, i.e., models with two phones of context, the previous and the following.

Emission Probability

In traditional HMM-based acoustic models, GMMs are typically employed to express the probability of an observation o_t being generated by the state i, i.e., the emission probability. These models are known as GMM-HMM models.

The Gaussian or normal distribution define a density function over continuous applications. Two parameters define the Gaussian distributions: the mean and variance in the case of a single variable and the mean vector (μ) and covariance matrix (Σ) for multidimensional variables (which is the case for audio features). The density function for a multivariate Gaussian distribution is defined as:

$$p(x) = \frac{1}{(2\pi)^{D/2}|\Sigma|^{1/2}} exp\Big\{ -\frac{1}{2}(x-\mu)^T\Sigma^{-1}(x-\mu)\Big\} \tag{3.14}$$

where x and μ are a $1 \times D$ dimensional observation vector and Σ is a $D \times D$ covariance matrix. A single Gaussian distribution is usually unable to properly model real data as they only have a very small number of parameters. More complex probability distributions can be modelled by extending the Gaussian distribution by a linear superposition of a number of Gaussian distributions, called Gaussian mixture models (GMM).

GMMs are unsupervised probabilistic models that assume that a mixture of Gaussian distributions can generate all the data points. That is, given a set of observations $\mathbf{O} = [o_1, o_2, ..., o_T]$, their probability distributions are modelled by a weighted sum of K components, where:

$$p(o_n) = \sum_{k=1}^{K} \pi_k \mathcal{N}(o_n \mid \mu_k, \Sigma_k) \tag{3.15}$$

where $\mathcal{N}(o_n \mid \mu_k, \Sigma_k)$ is the k Gaussian distribution from the K total components with mean vector μ_k, covariance matrix Σ_k and mixing coefficient π_k.

Alternatively to GMM, DNN models can be employed for emission probability computation. These kinds of models are known as hybrid DNN-HMM models. In this case,

a network can be trained as a phone classifier where each output neuron corresponds to a different phone. Typically, these networks employ a softmax activation function to the output layer and a cross-entropy loss function as a way to assign a probability to each phonetic label at a given frame. Note that N previous and following frames can be easily incorporated to provide the frame context.

The output of these networks can be interpreted as $P(phone|acoustic\ frame)$, i.e., the posterior probability of a phone given the acoustic frame. However, in an HMM we need the emission probabilities $p(x|q)$. Let q be the phone class and x the acoustic frame. Therefore, using Bayes rule:

$$P(q|x) = \frac{p(x|q)p(q)}{p(x)} \tag{3.16}$$

then, we can get *scaled likelihoods* by dividing the network output $P(q|x)$ by the relative frequency of the class q in the dataset,

$$\frac{p(x|q)}{p(x)} = \frac{P(q|x)}{p(q)} \tag{3.17}$$

Note that we can use $\frac{p(x|q)}{p(x)}$ instead of $p(x|q)$ because the scaling term $p(x)$ is independent of the class q.

3.1.3 Language Modelling

A language model (LM) is a statistical model concerned with learning the probability distribution, $P(\mathbf{W})$, over the k words sequence, $\mathbf{W} = (w_1, w_2, ..., w_k)$, for a given language $\mathscr{L}$. So, the probability of a k-words sentence is computed by:

$$P(\mathbf{W}) = P(w_1, w_2, ..., w_k) = \prod_{k=1}^{K+1} P(w_k \mid w_1, w_2, ..., w_{k-1}) \tag{3.18}$$

One way to approximate $P(\mathbf{W})$ is by employing an 'N-gram' language model. N-gram models assume that the probability of the next word can be estimated given knowledge of the N-1 preceding words. The model can then compute the probability of sequences with arbitrary lengths. So, the probability of a word w_k is approximated by:

$$P(w_k \mid w_0, w_1, ..., w_{k-1}) \approx P(w_k|w_{k-N}, ..., w_{k-1}) \tag{3.19}$$

Models where the probability of any word is assumed to depend on only the preceding two or three words are called *bigrams* (2-gram) and *trigrams* (3-gram), respectively.

Using maximum likelihood estimation (MLE), it is possible to estimate the probability of an N-gram model. This can be computed as:

$$P(w_n \mid w_{n-N+1}, ..., w_{n-1}) = \frac{C(w_{n-N+1}, ..., w_{n-1}, w_n)}{C(w_{n-N+1}, ..., w_{n-1})} \tag{3.20}$$

where $C(\cdot)$ is the count operation.

Other language models employ recurrent neural networks to predict the next word in a sequence given the previous context. However, these kinds of language models will not be covered here as they are beyond the scope of the thesis. For more information about these kinds of language models and how they operate, refer to Kamath et al. (2019).

Vocabulary

In some scenarios, the whole set of possible of words that can occur is known. This is referred to as closed vocabulary, and the test set can only contain words from this lexicon. However, in large vocabulary scenarios, some words may have no or rare occurrences in the training set. These unseen words are known as 'out of vocabulary' words. To deal with these scenarios, first, a predefined set of words are selected. Next, all words not included are replaced by '<UNK>' pseudo-word. Then, the probability of <UNK> is computed as any other word.

Perplexity

Language models can be evaluated in terms of their 'perplexity' (PP), which is the capacity of the LM to predict a sample. The perplexity of an LM on a test set can be defined as the inverse probability of the test set, normalised by the number of words,

$$PP = \sqrt[N]{\frac{1}{P(w_1, w_2, ..., w(N))}} \tag{3.21}$$

Expanding the probability of W using chain rule,

$$PP = \sqrt[N]{\prod_{i=1}^{N} \frac{1}{P(w_i \mid w_1, w_2, ..., w_{i-1})}} \tag{3.22}$$

3.1.4 Decoding

The decoding process in ASR consists of combining the acoustic and language models' outputs into a lattice[2] of hypotheses and searching for the most likely one. The final hypothesis is selected by using the 'Viterbi' algorithm. However, evaluating all possible word sequences is costly and can be very slow, especially for a large vocabulary task. Therefore, the search space is typically reduced by pruning the low probability hypothesis using beam search.

Modern systems often employ a weighted finite-state transducer (WFST) to compose a context-dependent HMM network in the acoustic model with the lexicon and language model (Mohri et al., 2002). In the Kaldi toolkit (Povey et al., 2011), the decoding process is performed using a WFST with beam-pruning, so states with a score less than a best-score minus the beam are pruned.

Note that the necessary pruning reduces the accuracy of the decoding. Errors can be introduced when the most probable hypothesis is erroneously pruned. This is particularly true for poor acoustic models where there may be many competing hypotheses requiring a very wide beam.

3.1.5 Acoustic Models Adaptations

Speaker dependent ASR systems can achieve a factor of two or three lower average word error rate (WER) than speaker independent systems if they are trained using the same amount of data (Huang and Lee, 1993; Huang, 1991; Paul, 1989). However, collecting enough data from a specific speaker to train such a system can be very costly. However, one can train HMM-GMM-based acoustic models on an extensive collection of speech samples from several speakers and adapt it to a target speaker using a small amount of data, obtaining speaker dependent-like performances.

These same techniques can be employed for singing ASR systems. For example, one could adapt a singer independent system to improve the lyrics transcription from a target singer. Additionally, f only a few samples of singing data are available, speaker adaptations techniques can be employed to adapt a spoken speech AM to singing conditions.

Two popular speaker adaptations are the 'maximum likelihood linear regression' (MLLR) and 'maximum a posteriori' (MAP). These techniques aim to update the AM's parameters, so the mismatch between the trained model and the target speaker is reduced. For a detailed account of these techniques, refer to the review presented by Antipolis and Woodland (2001).

[2]Lattice is a representation of the alternative word sequences that are likely transcriptions of a particular utterance.

Maximum Likelihood Linear Regression

Maximum Likelihood Linear Regression (MLLR) computes a series of transformations for the mean and variance of the GMM-HMM models, so the HMM is more likely to generate the adaptation data. When employing MLLR, new mean, $\hat{\mu}$, and covariance matrices, $\hat{\Sigma}$, are computed:

$$\hat{\mu} = W\xi \tag{3.23}$$

$$\hat{\Sigma} = B\Sigma B^T \tag{3.24}$$

where $\xi = [1\ \mu]^T$, with μ being the mean vector, and W can be decomposed into $W = [b\ A]$ where b is the bias and A is the transformation matrix.

There are two variations of MLLR adaptation: *mean only MLLR* and *constrained MLLR*. Mean only MLLR only transform the mean vector, keeping the covariances unchanged (3.23). Constrained MLLR applies the same transformation matrix to both the mean and covariance matrix, where,

$$\hat{\Sigma} = A\Sigma A \tag{3.25}$$

Maximum a Posteriori

Maximum a posteriori (MAP) adaptation involves using prior knowledge of the model parameters before seeing any adaptation sample. In a GMM-HMM model, a particular Gaussian mean, with prior μ_0, the estimate is:

$$\hat{\mu} = \frac{\tau\mu_0 + \sum_{t=1}^{T} \gamma(t) o_t}{\tau + \sum_{t=1}^{T} \gamma(t)} \tag{3.26}$$

where τ, $2 \leq \tau \leq 20$, is a hyperparameter that gives the bias between the maximum likelihood estimate of the mean and the prior mean, o_t is the adaptation vector at time t, and $\gamma(t)$ is the Gaussian probability at time t.

One of the advantages of MAP adaptation is that as the amount of adaptation data increases, the MAP estimate converges to the maximum likelihood estimate. However, unlike MLLR adaptations, the MAP adaptation is local, i.e., only the parameters belonging to the Gaussian of the observed states will be adapted. Therefore, some Gaussians may not be adjusted, especially in large vocabulary systems that can contain thousands of Gaussians.

3.1.6 Evaluation

The standard evaluation for speech recognition systems is the word error rate (WER). WER estimates how much the hypothesised word sequence differs from the reference one. It is computed by the sum of the deletions, D, substitutions, S and insertions, I, normalised by the length of the reference sequence, N (Equation 3.27). The first step in computing WER is to align the hypothesised and reference sequences so the Levenshtein distance (Levenshtein et al., 1966) is minimised. WER can be greater than 100 when there are more errors than words in the sequence, i.e., as can happen when there are one or more insertions.

$$WER(\%) = \frac{D+S+I}{N} \times 100 \tag{3.27}$$

Note that when the reference transcription is at the phone level, this score is called the phone error rate (PER).

3.2 Unaccompanied Sung Speech Recognition

In the first part of the chapter, the theoretical aspects of ASR systems were covered. We will now review related studies focused on sung speech recognition. Chapter 2 discussed several similarities and differences between sung and spoken speech. While sung and spoken speech are produced using the same speech production mechanisms, they have different acoustic characteristics, like syllable duration and pitch ranges, that makes the former less intelligible than the latter.

One of the requirements for successful acoustic modelling is to have sufficient data to model the statistical representation of the feature vector sequence given a phoneme sequence. Spoken speech recognition is a well-established challenge with many speech corpora with a long history containing spoken speech recordings together with their corresponding aligned transcriptions that allow comparable results between different studies. For example, the TIMIT corpus (Garofolo et al., 1983) is composed of over 5 hours of speech sentences phonetically transcribed. The Wall Street Journal (WSJ) corpus (Paul and Baker, 1992) is composed of 80 hours of recordings of speakers reading newspaper text paragraphs. The TED-LIUM corpus (Rousseau et al., 2012) is composed of 118 hours of transcribed speech from TED[3] talks. More recently, the LibriSpeech audiobooks dataset (Panayotov et al., 2015) contains 1000 hours of recordings derived from audiobooks that are part of the LibriVox[4] project. However, acoustic models trained using spoken data perform poorly when used to

[3]https://www.ted.com/
[4]https://librivox.org/

recognise sung speech due to the acoustic differences between the sung and spoken speech. To the best of my knowledge, no transcribed unaccompanied sung speech corpus for sung speech recognition is currently available.

Due to the lack of readily available unaccompanied sung speech data, sung speech recognition research has been focused on information retrieval applications. A speech recognition system can be employed in the front-end to recognise a keyword or a small sung sentence. Then, the recognised words are used to retrieve a list of songs or lyrics text containing those words from a musical collection. One can achieve good retrieval performances by using some fuzzy matching, even with high WER. This was, in fact, one of the earliest successful applications for spoken speech ASR, i.e. the NIST spoken document retrieval tasks (Garofolo et al., 2000).

Authors have been dealing with the lack of data for acoustic modelling in different ways. For example, some authors used a large amount of spoken speech data (Hosoya et al., 2005; Kawai et al., 2016, 2017; Kruspe, 2014, 2015a,b; McVicar et al., 2014; Mesaros and Virtanen, 2010a,b; Wang et al., 2003) and resorted to adaptation techniques to adapt the spoken acoustic model to singing. Other authors have used small manually annotated sung speech data (Hansen, 2012) or, an extensive collection of sung speech data that uses a spoken acoustic model to automatically generate the alignments to transcriptions of lyrics retrieved from the Internet (Kruspe, 2016a,b). Last, some authors used distorted sung speech data by employing some preprocessing steps to estimate the speech segment from an accompanied sung speech signal (Gruhne et al., 2007a,b; Szepannek et al., 2010; Tsai et al., 2018).

Language modelling for ASR systems tends to be trained on a large text corpus matching the grammatical structure of the speech signal. For example, spoken ASR systems based on the LibriSpeech dataset used language models trained on a large collection of text excerpted from 14500 public domain books. Note that in the LibriSpeech case, the text used for language modelling does not overlap with the text appearing in the acoustic test and development sets. In the same way, sung speech recognition systems can train language models using text lyrics not present in the acoustic data (Kawai et al., 2016, 2017; Mesaros and Virtanen, 2010a,b; Tsai et al., 2018). However, in music information retrieval, some authors constrained their systems to retrieval applications where the input song exists in the target collection. This constrains the language modelling to learn from texts contained in the test data (Hosoya et al., 2005; McVicar et al., 2014; Wang et al., 2003). A more detailed account of the development of sung speech acoustic and language modelling is provided in the sections that follow.

3.2.1 Sung Acoustic Modelling

Based on Spoken Data

One of the earliest works on sung speech recognition was presented by Wang et al. (2003). They trained triphone HMM acoustic models using MFCC features, 33 hours of Taiwanese and Mandarin spoken speech data and 925 sung phrases in the same languages for evaluation. Despite the mismatch of the acoustic conditions between the training and evaluation sets, they achieved 7% of WER. However, this performance was only possible due to the use of a very constrained language model restricted to lines in the test data.

More flexible systems dealt with the mismatch between the spoken training data and the sung testing data by utilising a small set of singing data for adapting the acoustic models to sung speech conditions (Hosoya et al., 2005; Kawai et al., 2016; Kruspe, 2015a; McVicar et al., 2014; Mesaros and Virtanen, 2010a,b). In this line, Hosoya et al. (2005) trained monophone HMM acoustic models using read speech data and adapted the models to the sung speech by using the MLLR speaker adaptation technique (Gales and Woodland, 1996) and 127 song chorus recordings performed by six male singers as adaptation data. Their testing data was composed of several 5-word length utterances generated from 128 Japanese children's songs sung by five male university students. The system achieved 23% WER using a language model consisting of a finite state automaton with constrained transitions based on Japanese grammatical rules.

Similar to Hosoya et al., Mesaros and Virtanen (2010a,b) trained monophone HMM acoustic models using 13 MFCCs plus deltas and double deltas and the CMU Arctic spoken dataset[5]. However, for adaptation, they employed a two-step procedure using the constrained MLLR (CMLLR) adaptation technique (Mesaros and Virtanen, 2009). In the first step, models are adapted based on eight regression classes of phonetic similarities, and in the second step, further adaptations are applied using classes determined by acoustic similarities. Testing and adaptation were performed using a 5-fold cross-validation setup from forty-nine sung fragments (19 male and 30 female) of 12 pop songs, ranging from 20 to 30 seconds, achieving 88% WER. To evaluate the model on polyphonic music, Mesaros and Virtanen first employed a source separation technique (Mesaros and Virtanen, 2009) to estimate the singing. In this case, 100 segments of separated singing were used, and the acoustic model was adapted using all forty-nine sung fragments, resulting in 95% WER. In both cases, the language model consisted on a trigram model trained using 4470 song lyrics retrieved from http://www.azlyrics.com website.

[5] http://www.festvox.org/cmu_arctic/

McVicar et al. (2014) employed the same two-step adaptation proposed by Mesaros and Virtanen. They trained triphone HMM acoustic models using the Wall Street Journal spoken corpus (Paul and Baker, 1992) and ten songs for adaptation. In this case, testing was performed using the chorus sections from 18 songs, obtaining 69% WER using a bigram language model trained on lyrics from the test set.

Kawai et al. (2016) adopted a more sophisticated adaptation approach. They first trained a neural network to transform read-speech MFCCs to sung speech MFCCs. The network consisted of two hidden layers with 24 sigmoid units each, 12-unit (12 MFCCs) input layer, and 12-unit output layer. The transformation network was trained using a private parallel corpus composed of 15 sung and spoken lyric recordings from seven singers. Acoustic modelling was performed using 12 transformed MFCCs plus log energy, deltas and double deltas and 122 hours of spontaneous spoken speech. Then, acoustic models were speaker and singing adapted using the maximum a posteriori (MAP) adaptation technique. They used seven songs for speaker adaptation, 40 songs for singing adaptation, and seven for testing. Note that the adaptations and testing sets were obtained by separating the singing from polyphonic music using third-party software. The language model consisted of a 3-gram trained on 230000 lyric texts using the 20000 most frequent words. The best performances obtained reached 41% WER.

As discussed in Chapter 2, the vowels distance (i.e., distance between the vowels when plotted in F1-F2 space) is highly affected by the singing pitch, reducing as pitch increases. In a later work, Kawai et al. (2017) evaluated the use of pitch-based features to increase the vowel phone class distance. They explored the use of three different pitch-based features: V_0, which corresponds to a flag indicating if the pitch is detectable (1) or not (0); Log F0; V_4, which corresponds to a one-hot vector indicating pitch range (undetectable F_0, $55Hz \leq F_0 < 173Hz$, $174Hz \leq F_0 < 260Hz$, $261Hz \leq F_0$). The acoustic model consisted of a five-layer DNN network trained using 122 hours of spontaneous speech data and 90 minutes of sung speech data. The training process was divided into two steps. In the first step, the model was trained using a 39-dimensional feature vector (12 MFCCs, log energy, deltas and double deltas) plus dummy zero-valued pitch features. In the second step, the model was fine-tuned by including the actual pitch features for the sung data but keeping the dummy pitch features for the spoken data. Using the same language model than their previous work, the system achieved a performance of 39% WER. Note that they only evaluated the effect of male singers pitch range, which are much smaller than female singers range, making the current definition of the V_4 features less likely to be useful for female singing recognition.

In a later work, Kruspe (2015b) employed a different adaptation approach. Instead of transforming the feature vector or adapting the acoustic models, they implemented a

preprocessing step to transform the spoken data to sung-like data. Three transformations were evaluated: time stretching, pitch shifting and vibrato. Time stretching randomly extends the duration of the vowels between 5 to 100 times the original duration – parameters motivated by an analysis between the spoken and sung sounds duration in their previous work (Kruspe, 2014). Pitch shifting randomly shifts the pitch between 60% to 120% of the previously estimated original pitch. Vibrato was applied only to previously extended vowels by estimating the original pitch and shifting it using a sine curve with amplitude 0.2 and frequency $6\,Hz$. Triphone HMM acoustic models were trained using the TIMIT dataset (Garofolo et al., 1983) transformed with one or more of the three transformations. The evaluation set consisted of the 12 songs with phone-level annotation presented by Hansen (2012) (work described below), obtaining best performance of 107% PER when using pitch shifting and time stretching modifications.

Note that the previous unaccompanied sung speech ASR systems based on spoken data tend to use less than 50 recordings of singing data for adaptation. Such small datasets is insufficient for successfully adapting spoken acoustic models to the various singing characteristics.

Based on Clean Sung Data

Hansen (2012) presented an MLP acoustic model approach trained on a small singing corpus of 19 word-level manually annotated songs, from which only 12 are also manually annotated at phone-level. In this work, only the 12 phone-level songs were used. The data was split into 16000 frames for training, 80000 frames for validation and 6000 frames for testing. They trained two MLP networks, one using MFCC features, and another using temporal pattern features (TRAP) (Hermansky and Sharma, 1998). The result of these two networks are then combined to boost the phoneme classification. The motivation for using the combined system was to take advantage of the temporal information of the TRAP features that may help to improve the recognition of dynamic sounds like fricative and plosive phonemes. In this work, the performances were evaluated in terms of the recall score, i.e., the capacity of the AM to correctly classify the phone from a frame out of all the correct predictions. The combined system achieved a recall of 48% in comparison with 44% for MFCCs and 42% for TRAPs.

In 2015, Smule[6] released the DAMP Vocal performance (multiple performance) dataset (Smule, Inc., 2015), a large karaoke sung speech dataset, comprising 34,620 performances covering 302 different songs. However, this corpus has a significant imbalance between the number of recordings per song and it does not include lyric alignments. In a later work, Kruspe (2016a,b) exploited this dataset by automatically aligning the singing audio with

[6] http://smule.com

the transcriptions. Using a monophone HMM acoustic model trained on TIMIT, words and phonemes from the lyrics recovered from the Smule website were aligned. Using the aligned corpus, several training sets with different number of frames per phoneme were constructed. For acoustic modelling, a DNN-HMM model with three hidden layers (1024, 850 and 1024 units) was used. The inputs were 13 MFCCs plus deltas and double deltas (39 dimension) and the output layer corresponds to 37 monophones. For evaluation, 300 performances per gender (one for each song) were used. Phone recognition resulted in 80% PER.

Based on Distorted Sung Data

Some authors explored the utilisation of singing data estimated from accompanied sung speech recordings. In these cases, the singing was estimated using some preprocessing steps to reduce the masking effect of the background accompaniment.

Gruhne et al. (2007a,b) presented a study that evaluated several audio features and various machine learning algorithms to classify singing into phoneme classes. For this analysis, they annotated 37 polyphonic songs at the phone-level (21 males and 16 females). This data was then split into 51% for training and 49% for testing. In this work, Gruhne et al. implemented a harmonic extraction algorithm before extracting the features (Fujihara et al., 2005). The algorithm starts with fundamental frequency estimation using a multi-resolution fast Fourier transform (Dressler, 2013). Next, based on the estimated fundamental frequency, harmonic partials are extracted from the spectrogram. Then, the signal is re-synthesised using a sinusoidal re-synthesis using only the estimated fundamental frequency and harmonics. The re-synthesised signal resulted in sung speech signals containing a large amount of distortion (residual from the accompaniment), which affected the recognition of unvoiced formants. Therefore, Gruhne et al. restricted their work to categorise singing into 15 voiced phoneme classes.

Four different features were extracted from the re-synthesised signal; 13 MFCCs, 9 PLP coefficients, 8 LPCs and 8 WLPCs. For phoneme classification, GMM, multilayer perceptron (MLP) and support vector machines (SVM) classifiers were trained. Best performance was obtained when training an SVM classifier obtaining 57.7% of correct classified instances. Szepannek et al. (2010) extended Gruhne et al.'s work by exploring the use of a large set of perceptually-motivated features. However, no significant improvement was obtained.

In more recent work, Tsai et al. (2018) utilised 130 English sung songs sourced from YouTube. Singing segments were provided by the video owners, i.e., there are no details of the recording conditions or recording equipment used. Using an acoustic model trained using the LibriSpeech data, 20 songs with more than 95% WER were removed. The remaining 110 songs were segmented into lyric phrases resulting in 640 segments for training and

97 for testing. Tsai et al. employed the Kaldi toolkit (Povey et al., 2011) to construct their ASR system. Acoustic modelling consisted of a time delay neural network (TDNN) (Waibel et al., 1989)-long short-term memory (LSTM) (Hochreiter and Schmidhuber, 1997) DNN model, trained using 40 dimension MFCCs and 50 dimension i-vectors (Dehak et al., 2011). Speed perturbation was employed for data augmentation to deal with the changes in singing speech rates. Using a trigram language model trained using 40 million sentences from the LibriSpeech text corpus plus 574,000 lyric texts sourced from the Internet, they obtained 73.9% WER. This work may represent the first system using a modern and more sophisticated acoustic model. However, the size of the training data used (less than five hours of data) remains insufficient for robust sung speech acoustic modelling compared to the more extensive datasets used for spoken speech acoustic modelling mentioned above.

3.2.2 Results

For completeness, the results of the studies reviewed in this section are presented in Table 3.1, for sung phone recognition results, and Table 3.2, for sung word recognition results. Note that the results reported in both tables correspond to experiments using different private test sets, which means that care has to be taken when comparing results. Note that different works reported performances using different metrics, especially in the results of studies recognising sung phonemes, presented in Table 3.1.

Despite the fact that the results are not directly comparable, they clearly illustrate the complexity of the task. For example, from the sung word recognition results in Table 3.2, the systems using a closed LM reported higher performances then open LM, which is expected because the test sung utterance are known by the LM. However, for results reported using 'Lyrics' or 'Prose + Lyrics' LM the error did not drop from 40% WER in the best reported results. This shows a very large gap compared with spoken tasks, e.g., for LibriSpeech audiobooks (Panayotov et al., 2015) lowest WERs are 1.8% (Hsu et al., 2021), for the TED talks corpus TED-LIUM (Rousseau et al., 2012) the lowest WER is 5.6% (Zhou et al., 2020), and for the WSJ corpus (Paul and Baker, 1992) WERs have reached 2.9% (Hadian et al., 2018).

Table 3.1 Results of phone-level sung speech recognition systems. In Train data and Test data, the term 'Enhanced' refers to using some method to enhance the sung segment from a polyphonic song, 'Spoken' refers to the use of spoken data, and 'Clean' refers to the use of unaccompanied sung speech. Note that authors used different evaluation metrics in their phone-level experiments.

Study	Gruhne et al. (2007a)	Mesaros and Virtanen (2010b)		Hansen (2012)	Kruspe (2015b)	Kruspe (2016a)
Train data	Enhanced	Spoken	Spoken	Clean	Spoken	Clean
Adaptation	X	✓	✓	X	✓	X
Test Data	Enhanced	Clean	Enhanced	Clean	Clean	Clean
Evaluation Metric	Correct Classified Instance	PER	PER	Recall	PER	PER
Results	58%	65%	81%	48%	107%	80%

Table 3.2 Results of word-level sung speech recognition systems. In Train data and Test data, 'Enhanced' refers to using some method to enhance the sung segment from a polyphonic song, 'Spoken' refers to the use of spoken data, and 'Clean' refers to the use of unaccompanied sung speech. In Language Model, 'Closed' refers to the use of the lyric texts from the test set, 'Lyrics' refers to the use of a collection of lyric texts independent from the test set, and 'Prose + Lyrics' refer to the use of a large corpus of spoken text augmented with lyrics text. All results are presented in terms of the word error rate. Note that the results reported are not directly comparable as they used different training and evaluation datasets.

Study	Wang et al. (2003)	Hosoya et al. (2005)	Mesaros and Virtanen (2010b)		Kawai et al. (2016)	Kawai et al. (2017)	McVicar et al. (2014)	Tsai et al. (2018)
Train data	Spoken	Spoken	Spoken	Spoken	Spoken + Clean	Spoken + Clean	Spoken	Spoken
Adaptation	X	✓	✓	✓	✓	X	✓	X
Test Data	Clean	Clean	Clean	Enhanced	Enhanced	Enhanced	Clean	Enhanced
Language Model	Closed	Closed	Lyrics	Lyrics	Lyrics	Lyrics	Closed	Prose + Lyrics
Results	7%	23%	88%	95%	41%	39%	69%	74%

3.3 Audio Source Separation

As discussed above, the sung speech recognition problem becomes more difficult in accompanied scenarios where the instrumental background accompaniment masks or partially masks the singing. In many speech recognition systems for spoken speech in noise (e.g., scenarios presented in the CHiME 3 (Barker et al., 2017) and CHiME 4 (Vincent et al., 2017) challenges), the speech segment is separated from the noise prior to feature extraction. Similarly to spoken scenarios, sung speech recognition systems may benefit from a vocal separation front-end.

Music source separation presents different challenges from those presented in conventional noise-robust spoken speech recognition. In many spoken scenarios, sources are uncorrelated in their behaviour and have little time and frequency overlap, e.g., speech recordings in a crowded street (Barker et al., 2017). However, in music, sources are strongly correlated in onset (the sources follow the same rhythm structure with synchronous strong and weak beats) and frequency (instruments playing at octaves produce several overlapping harmonics).

Most robust spoken speech ASR applications deal with sounds with "acoustically mixed" signals, i.e. sounds that have occurred in the same acoustic space and recorded over the same microphone channel. For music, this may or may not be the case. i.e. professional music mixed in a recording studio where each instrument has been independently recorded and probably digitally manipulated, invalidating the assumption that sources share qualities, like the same reverberant space. Moreover, in some cases, the mixture may include 'non-physical' sources, e.g., music generated by a synthesiser.

Music tends to be structured with different redundancies observable in the spectrogram. For example, percussion events tend to be precisely located in time, while harmonics are located in frequency. FitzGerald (2010) exploited this quality to construct a percussion/harmonics separation system. While this kind of redundancy pattern offers little benefit for vocal separation, the highly period structure of the instrumental background accompaniment, unlike singing, which tends to be much less periodic, can be exploited to construct a time-frequency mask to separate the background from the mixture. This is the case of FitzGerald (2012) and Rafii and Pardo (2011, 2012), that proposed an unsupervised technique that estimates the rhythm patterns of the song to predict a background's time-frequency mask. These kinds of techniques represented the state-of-the-art before the introduction of deep learning in music source separation.

Modern DNN approaches for music source separation started to arise with the 2018 Signal Separation Evaluation Campaigned (SiSEC) (Stöter et al., 2018) release. This campaign comes with the release of the MUSDB18 datasets (Rafii et al., 2017). MUSDB18

comprises ten hours of labelled music with separate tracks per instrument, representing the first large scale dataset for standardised music source separation research. Since then, many sophisticated DNN approaches have been proposed (Défossez et al., 2019; Jansson et al., 2017; Luo and Mesgarani, 2019; Samuel et al., 2020; Stoller et al., 2018b; Stöter et al., 2019). Details of these approaches will be discussed below.

This section starts by presenting some of the most common performance measurements for audio source separation evaluation used in this thesis. Next, it presents details of music source separation systems based on the music redundancies. Then, it presents some of the state-of-the-art deep learning vocal separation architectures.

3.3.1 Evaluating Audio Source Separation

Audio source separation evaluation techniques are usually evaluated using objective measurements, i.e., measurements that rate the separation performance by contrasting a separated source with its ground truth source. Perhaps, the most commonly-used objective evaluation measurements are the three energy ratios proposed by Vincent et al. (2006): 'signal-to-distortion ratio' (**SDR**), 'signal-to-interference ratio' (**SIR**) and 'signal-to-artifacts ratio' (**SAR**).

An estimated source $\hat{s_j}$ is assumed to be composed of four components:

$$\hat{s_j} = s_{target} + e_{interf} + e_{noise} + e_{artif} \tag{3.28}$$

where s_{target} is the true source, e_{interf} is the interference error, e_{noise} is the noise error and e_{artif} is the artifact error. Note that the computation of these terms is based on a decomposition of $\hat{s_j}$ by orthogonal projections. Some amount of distortion is allowed in the estimation accounted by a time-invariant 512-tap filter.

Given the computation of these terms, the measurements are defined as:

- Signal-to-Distortion Ratio (SDR): considered a measure of the overall performance of the separation and is typically used to compare different systems.

$$SDR = 10log_{10}\left(\frac{||S_{target}||^2}{||e_{interf} + e_{noise} + e_{artif}||^2}\right) \tag{3.29}$$

- Signal-to-Interference Ratio (SIR): interpreted as the amount of other sources that can be heard in the estimated source.

$$SIR = 10log_{10}\left(\frac{||S_{target}||^2}{||e_{interf}||^2}\right) \tag{3.30}$$

- Signal-to-Artifacts Ratio (SAR): represents the amount of unwanted artifacts in the estimated source.

$$SAR = 10log_{10}\left(\frac{||S_{target} + e_{interf} + e_{noise}||^2}{||e_{artif}||^2}\right) \tag{3.31}$$

In general, SDR is considered a measure of the overall performance of the separation and is typically used to compare different systems. However, Roux et al. (2019) showed that SDR might not produce fairly comparable results because its filtering process can mask large distortions. Roux et al. (2019) proposed a scale-invariant version of SDR (SI-SDR) that alleviates over-optimistic SDR scores by removing SDR's dependency on the amplitude scaling of the signal. This is done by computing a scaling factor to match the reference and estimated sources such that a mismatch in scale causes no residual error. SI-SDR is defined as:

$$\text{SI-SDR} = 10log_{10}\left(\frac{||\alpha s||^2}{||\alpha s - \hat{s}||^2}\right), \quad \text{for } \alpha = \underset{\alpha}{\text{argmin}}\,|\alpha s - \hat{s}|^2 \tag{3.32}$$

where s is the target source, $\hat{s}$ is the estimated source and α is the scaling factor.

In addition to the *distortion* measurements described above, the quality of the estimated speech sources can be measured in terms of their intelligibility. This is done by using perceptual measurements, which correlates to subjective quality evaluations. The two most common measurements are the 'perceptual evaluation of speech quality' (PESQ) (Di Persia et al., 2008) and the 'short-time objective intelligibility' (STOI) (Taal et al., 2010).

PESQ is defined in the standard ITU P.862 as the mean for evaluating speech quality transmitted by a narrowband telephony system. It has been shown that PESQ has good properties to evaluate speech enhancement systems (i.e., systems where the target is only the estimate of the speech content from a mixture) in terms of subjective perceptual quality (Hu and Loizou, 2006).

STOI is an intelligibility measure highly correlated with the intelligibility of a speech signal with additive noise. STOI is an intrusive measurement (i.e., a function of the target and estimated speech) computed as the average of the correlation across frames and frequency bins.

3.3.2 Exploiting the Periodicity in Music

Several approaches for music source separation have been proposed based on the periodic structure of the background accompaniment. The detection of the rhythm pattern can be exploited to predict a time-frequency mask of the background from the mixture. One

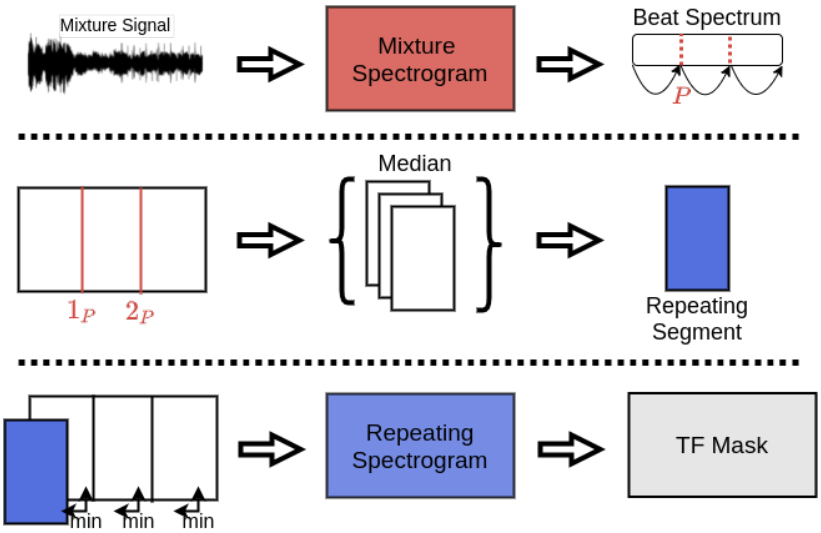


Figure 3.3 REPET: Process to build the repetition pattern. Stage 1: Analyse the spectrogram and identify the repetition period. Stage 2: The mixture is split into patches of the same length and take the median of them. Stage 3: The repetition pattern is used to construct a time-frequency mask.

of the most popular systems that exploit music periodicity is REPET (REpeating Pattern Extraction Technique) presented by Rafii and Pardo (2011). REPET is an algorithm that separates the background accompaniment from a mixture signal. It exploits the fact that the accompaniment of a song often has a spectrogram that is periodic in contrast to the vocal that is not usually very repetitive. The process to construct the background repetition is illustrated in Figure 3.3. The algorithm starts by identifying the period on which the power spectrum of the accompaniment is repeated. This is done by detecting the beat spectrum. Then, the power spectrum of the accompaniment is estimated as an averaging of the different repetitions. The estimated vocal segment corresponds to the residual between the mixture and the estimated background. REPET can work on a large variety of music signals. However, it assumes a strict repetition accompaniment, which is not realistic in a whole song length where different repetition patterns can be found, e.g., the verse and the chorus. Liutkus et al. (2012) presented an extension of REPET called 'adaptive REPET'. Adaptive REPET allows time-varying repetition pattern periods, which enable the separation of full-length songs with verse/chorus/verse structure.

Rafii and Pardo (2012) presented another extension to REPET, 'REPET-SIM', that instead of looking at periodic repetitions, REPET-SIM adaptively looks for similar parts at non-fixed periods. It starts by computing a self-similarity matrix that indicates what frames from the spectrogram are close to another. Then, the power spectrum of the accompaniment is estimated for each frame as the median of all frames in the song identified as similar.

Inspired by the repetition patterns exploited by REPET, FitzGerald (2012) proposes an approach that detects similar frames by using distant metrics. This method assumes that the local periodicity is not necessary and the repetition may occur further in a song.

Systems based on the periodicity of the music presented above can result in fast source separation. However, they rely on the correct identification of the periodicity of the accompaniment. Moreover, songs from genres like 'Pop/Rock' may have different sections across the song with different repetition patterns, like the 'introduction', 'bridge' and 'coda', limiting the effectiveness of the techniques.

3.3.3 Deep Music Source Separation

Currently, deep learning technologies are reporting state-of-the-art performances in audio source separation. These systems can be divided into systems that learn a mask to apply to the mixture spectrogram and systems that estimate the sources by operating directly on the waveform. Below is a brief review of the leading deep learning architectures for music source separation.

Mask-based Systems

In general, mask-based source separation systems take the spectrogram of the mixture as input, and they learn a time-frequency mask of the target source, e.g., the vocal. Then, the learned mask is applied to the mixture magnitude spectrogram to estimate the magnitude spectrum of the target source. The reconstruction of the estimated signal is made by using the masked magnitude spectrum in conjunction with the mixture phase spectrum, and transforming back to the time domain. Various deep learning architectures have been investigated for estimating the mask:

Open-UnMix (Stöter et al., 2019)

Open-UnMix is based on a recurrent neural network (RNN) architecture with bidirectional LSTM (BLSTM) layers proposed by Uhlich et al. (2017). For the music source separation problem, Uhlich et al. (2017) showed that the BLSTM layers better take into account the context information than the use of supervectors with context information in feed forward networks. Additionally, compared with traditional RNN, BLSTM have the advantage of not suffering from the vanishing or exploding gradients problem.

The Open-UnMix architecture is conformed of a fully connected layer, followed by three BLSTM layers and two fully connected layers. In 2019, Open-UnMix reported state-of-the-art performances despite being designed to favour simplicity over

performance. Open-UnMix served as a baseline for the SigSeg challenge (Stöter et al., 2018).

U-Net (Jansson et al., 2017)

U-nets are fully convolutional encoder-decoder architectures. The U-Net system applies six strided 2-D convolutional layers with a 5x5 kernel size and strides of 2 that progressively encode the input into smaller representations. Each encoder layer is followed by a batch normalisation (Ioffe and Szegedy, 2015) and a rectified linear unit function (RELU) activation function. A dropout of 50% is applied to the first three layers. The smallest representation (the result of the sixth encoder layer) is scaled back using five transpose convolutional layers with the same kernel and stride sizes as a paired encoder. The decoder layers also include batch normalisation and RELU activation function. The consecutive encoder and decoder steps are more or less symmetric, leading to a u-shaped architecture, hence the name. The final layer has a sigmoid activation function that makes a mask for the target source. The loss is computed between the spectrogram of the estimated source obtained by multiplying the mixture with the mask and the target source's spectrogram.

The U-Net architecture has several extensions, such as learning several sources simultaneously (Kadandale et al., 2020); or jointly learning to separate the vocal segment and estimate the singing F_0 (Jansson et al., 2019) or the singer activation (Stoller et al., 2018a). Some variants follow the same architecture as the original U-Net, but others use different activation functions or different numbers of convolutional layers.

Waveform-based Systems

Waveform-based systems can be regarded as end-to-end systems where the input and outputs are all waveforms, i.e., they operate directly on the time domain rather than on a time-frequency representation such as a spectrogram. They take the mixture signal as input and estimate the target signal directly. In these cases, the model takes several decisions including how it will model the phase and the temporal and frequency resolution of the features. However, successfully learning the masks and the signal representations require more data than mask-based systems. Note that there are still design choices in terms of kernel lengths and number of channels.

Conv-TasNet (Luo and Mesgarani, 2019)

Conv-TasNet is a modification of the TasNet (Luo and Mesgarani, 2017) speech separation system. TasNet is a time-domain audio source separation system consisting

of three modules, namely *encoder*, *separator* and *decoder*. The system starts by encoding the signal using a 1-D convolution layer to transform small segments of audio into filterbank-like representations (Pariente et al., 2020). Then, a separator network, consisting of a deep LSTM network, learns the masks. Finally, using transpose convolution, the decoder reconstructs the estimated sources obtained by masking the encoder output with the learned masks, transforming them back to the time domain.

Unlike TasNet, the Conv-TasNet architecture is a fully convolutional architecture. It replaces the LSTM separator network with a fully convolutional module motivated by the temporal convolutional network (TCN) (Lea et al., 2016). The TCN is composed of a series of 1-D dilated convolutional blocks. Each layer is a 1-D convolutional block with an exponentially increasing dilation factor. In Conv-TasNet, M convolutional blocks with dilation factors 1, 2, 4, .., $2^M - 1$ are repeated R times. The result from the R stacked blocks is passed to a point-wise convolution ($1 \times 1 - conv$ block) to estimate a mask function for each source (C) at each time step.

Both Conv-Tasnet and Tasnet use SI-SNR loss between the target and estimated waveforms. Only Conv-TasNet has been adapted to musical separation (Défossez et al., 2019).

Meta-TasNet (Samuel et al., 2020)

Meta-TasNet is an extension of Conv-TasNet that implements several modifications for adapting for music separation. First, the encoder and decoder are extended to a series of 1-D convolutional/deconvolutional layers with different kernel sizes to capture a wider range of information. Second, it implements a multi-stage architecture that processes the audio samples at $8\,kHz$, $16\,kHz$ and $32\,kHz$. The representation of the target source from a lower sample rate is concatenated to the representation of the mixture from the next higher sample rate. Third, the loss function is extended to a weighted sum of four terms: the SI-SDR loss, a 'dissimilarity' loss that minimises the similarity between instruments within a sample, a 'similarity' loss to maximise the similarity of instruments in different samples, and 'reconstruction' loss to increase the SI-SDR between the mixture and the signal processed without masking. Lastly, the main contribution of Meta-TasNet is the incorporation of a 'generator' network that predicts the parameters of the masking network for a target instrument. The generator learns an embedding for instruments.

Wave-U-Net (Stoller et al., 2018b)

Wave-U-Net is a multi-channel 1-D adaptation of the U-Net architecture that operates directly on the time-domain allowing large temporal context. It replaces the 2-D

Table 3.3 Average vocal SDR performance computed across all evaluation samples from MUSDB18 dataset (Rafii et al., 2017). REPET and REPET-SIM scores were computed using the 'NUSSL' Python module implementation (Manilow et al., 2018). U-Net scores corresponds to results reported as 'dedicated U-Net' by Kadandale et al. (2020).

	Model	**Vocal SI-SDR (*dB*)**
Repetition-Based	REPET (Rafii and Pardo, 2011)	-2.35
	REPET-SIM (Rafii and Pardo, 2012)	-2.65
Mask-Based	U-Net (Jansson et al., 2017)	5.09
	Open-UnMix (Stöter et al., 2019)	6.32
Waveform-based	Conv-TasNet (Luo and Mesgarani, 2019)	6.43
	Meta-TasNet (Samuel et al., 2020)	6.40
	Wave-U-Net (Stoller et al., 2018b)	3.25
	Demucs (Défossez et al., 2019)	6.84

convolutional/deconvolutional layers with 1-D convolutional/deconvolutional layers. The error between the target and the estimated source is computed by using the mean square error (MSE) loss function.

Demucs (Défossez et al., 2019)
Demucs is similar to the Wave-U-Net architecture with six convolutional layers and paired deconvolutional layers. However, Demucs has two BLSTM layers at the centre. The error is computed using the mean absolute error (MAE) loss function.

Mask-based system have the disadvantage that they estimate the spectrogram of the sources but not the phase and using the phase from the mixture can limit their performance. Additionally, the spectrogram representation performs well on stationary signals. However, music is a continuous signal. These two problems are solved by waveform-based systems where the phase and the representation are learnt by the model.

3.3.4 Performance

In the previous section, several music source separation architectures were introduced. This section will present the performance of these models in terms of the SDR score obtained using the MUSDB18 (Rafii et al., 2017) dataset. MUSDB18 is currently the most popular dataset used for music separation since its introduction for the 2018 SigSeg challenge (Stöter et al., 2018).

Table 3.3 presents the performance of the models described above utilising the MUSDB18 dataset. Repetition-based results correspond to the average score across all MUSDB18 evaluation samples obtained using the REPET implementation from the 'NUSSL' Python (Manilow et al., 2018) module. Mask- and waveform-based DNN models results correspond to scores reported by the authors.

The poorest performances were obtained by REPET and REPET-SIM algorithms despite that they exploit the intrinsic repetition patterns in music. These algorithms rely on several free parameters that constrain how far the algorithm will search for the repetitions. However, parameters that work for one song may not work for another. Additionally, there are many songs, especially in the 'Jazz' musical genre, where few redundancies can be found, completely escaping the algorithm assumptions.

Mask-based DNN systems reported higher performance than repetition based-systems. However, best performances are obtained from time-domain systems that estimate the sources directly from the waveform. One reason that may explain the better performance of these systems is that the models learn the audio representation, presumably capturing better information than the supervectors used in mask-based systems can.

Note that the performances reported from some of the wave-based systems was improved by augmenting the data from the MUSDB18 training set with private sets. For example, the Demucs performances increases from 6.81 dB to 7.29 dB and Conv-TasNet from 6.43 dB to 6.74 dB when augmenting the MUSDB18 training set with 150 songs. The increased performance obtained when training models using extra training data is an indication that the MUSDB18 dataset may not be large enough for training robust audio source separation models.

3.4 Summary

This chapter introduced the speech recognition problem in the context of sung speech recognition. Traditional speech recognition systems are conformed of several modules, namely the front-end, acoustic model, language model and decoder, that work together to predict the more likely sequence of words given a sequence of observation vectors. The front-end module takes a length-T speech signal and returns a length-N sequence of observation vectors. This step may include several signal normalisation steps, like resampling and changing the number of channels. In the case of a speech in noise signal, the front-end may consist of an audio source separation process to separate the speech from the noise before computing the observation vectors. Acoustic modelling deals with estimating the match between the observation vectors with the sequence of words. This is done by employing

hidden Markov models (HMM) using Gaussian mixture models (GMM) or deep neural networks (DNN) for emission probabilities. Language models (LM) learn the probability distribution over a word sequence. Typically, this is done using n-gram models that compute a word's conditional probability given the $N-1$ previous words. The decoder dynamically uses the acoustic model and language model to search for the word sequence with the higher score.

Robust acoustic models need a large amount of annotated data to model the statistical representation of the feature vector sequence given a phoneme sequence. However, there is a lack of a large annotated dataset for sung speech recognition research. Authors have been proposing different ways to overcome the lack of data by implementing different techniques. Some authors trained the acoustic model (AM) using a large amount of spoken speech data and a small set of sung speech data to adapt the models to singing (Hosoya et al., 2005; Kawai et al., 2016, 2017; Kruspe, 2014, 2015a,b; McVicar et al., 2014; Mesaros and Virtanen, 2010a,b; Wang et al., 2003). Other authors limited their ASR systems to phoneme recognition by using a small amount of manually transcribed sung data (Hansen, 2012) or by using a spoken AM to force aligned a large amount of sung data with lyrics retrieved from the Internet (Kruspe, 2016a,b). Last, some authors have resorted to techniques to enhance the harmonics of the sung speech from polyphonic recordings (Gruhne et al., 2007a,b; Szepannek et al., 2010; Tsai et al., 2018). The comparison of previous work is made difficult because they all used different training and evaluation sets. This brings us to **RQ.3** and **RQ.4** that will be dealt with in Chapter 4.

Music audio source separation possesses different challenges than spoken scenarios. In spoken separation, the sources are uncorrelated with little harmonics overlap. However, in music, sources are highly correlated in onset, and instruments may have several harmonics overlapping when playing at fifths or octaves. On the other hand, music is highly structured and contains several redundancies that source separation algorithms can exploit. The repetition structure of the music has been exploited to estimate a mask to separate the background from the mixture. In these models, the singing is estimated by a difference between the mixture spectrogram and the spectrogram of the estimated background. Estimated signals are generated by utilising the phase from the mixture. Since the introduction of the MUSDB18 dataset (Rafii et al., 2017), several novel deep learning models have been designed to separate the 'vocal', 'bass', 'drums' and 'others' sources from a mixture. MUSDB18 enabled a fair comparison between the different architectures. Several DNN models have been proposed that learn from the time-frequency representation (Jansson et al., 2017; Stöter et al., 2019) or directly from the waveform (Luo and Mesgarani, 2019; Samuel et al., 2020; Stoller et al., 2018b) using this dataset. However, MUSDB18 contains only 100 songs for training data,

which may not be sufficient for training highly generalised systems. This leads us to **RQ.5** which will be investigated in Chapter 5.

As was mentioned above, some ASR systems may integrate an audio source separation step as part of the front-end stage (Tsai et al., 2018). While Chapter 4 will investigate an ASR system for unaccompanied singing, Chapter 6 will investigates the effect of integrating a music source separation model with the unaccompanied sung speech ASR system to recognise the lyrics from a polyphonic song, which bring us to **RQ.6**. First, it will present an analysis of recognising directly from the separated singing. Then, it will employ acoustic model adaptations techniques to adapt the AM to distorted singing conditions.

Chapter 4

Acoustic Modelling for Sung Speech Recognition

4.1 Introduction

This thesis tackles the challenge of automatically recognising the lyrics from a recording of an accompanied singing performance. We have seen how the problem can be divided into three more constrained tasks: unaccompanied singing recognition, audio source separation and the integration of both. This chapter addresses the first of these challenges - the task of unaccompanied signing recognition. Note that there are two major parts to consider in an automatic speech recognition system: the *language model* and the *acoustic model*. Language modelling is a research field by itself and a very complex task for lyrics. The chapter will focus on unaccompanied sung speech acoustic modelling.

Chapter 2 presented a discussion of several differences in the production of sung and spoken speech that lead to differences in the signal domain. These differences, such as vowel duration, pitch range, pitch variation within a vowel, vowel space area and energy, act to make sung speech less intelligible than spoken speech. These effects originate partly because, in singing, the speech intelligibility becomes secondary to the needs of the artistic performance, making it more challenging to recognise and resulting in spoken speech acoustic modelling performing poorly on sung speech data without acoustic modelling adaptation (Chapter 3).

Sung speech recognition research is made even more challenging by the lack of readily available data. English acappella singing data suitable for training and evaluating acoustic models are scarce, small, and usually not openly available. Previous research has used these smaller datasets for sung speech adaptation or evaluation purposes, while needing to use larger spoken speech datasets for training. For example, Mesaros and Virtanen (2010b)

utilised 49 segments of unaccompanied singing ranging between 20 to 30 seconds for spoken to sung speech adaptation and evaluation. The sets were generated by using a 5-fold setup, with one-fifth of the data for the evaluation. Hansen (2012) utilised 12 manually annotated songs for sung speech phoneme classification. Mauch et al. (2012) used 20 full-duration English spoken songs for assessment of an ASR model trained on Japanese sung speech data (Goto et al., 2002).

Recently, novel singing datasets have been made available for different sung speech research tasks. However, these datasets have not been designed directly for ASR purposes and care needs to be taken when using them. For example, the DALI dataset (Meseguer-Brocal et al., 2018) contains over 5000 full-duration songs of semi-automatically aligned commercial music. However, DALI is only composed of accompanied singing, making it difficult to use for sung speech acoustic modelling evaluation. In contrast, Smule[1] released several large karaoke sung speech datasets. The first release, the DAMP Vocal performance (multiple performance) dataset (Smule, Inc., 2015), comprises 34,620 performances covering 302 different songs. However, there is a big imbalance in the number of performances per song, and the data is made hard to use by a lack of timing information to align lyrics to the performance. The second DAMP Vocal performance (balanced) database (Smule, Inc., 2017) extends the previous release containing 24,874 performances from 5429 singers but covers just 14 song arrangements. Although this dataset is acoustically rich, with only 14 songs, it does not have enough linguistic variability to train robust speech models. It may be better suited to studying singer variability than speech recognition. The third release, the Digital Archive of Mobile Performances - Smule Multilingual Vocal Performance 300x30x2 dataset (DAMP-MVP) (Smule, Inc., 2018), is a multilingual dataset that provides 18,676 performances from 13,154 singers covering 5690 songs with an equal number of performances per gender separated by country of the singers. Further, it also provides the lyric prompts presented to the singer and the prompt timings; thus, it is much easier to align the song lyrics to the signals. The number of singers and songs in DAMP-MVP makes it a good candidate for constructing a sung speech dataset for ASR. It offers the necessary variability for learning acoustic patterns. However, the data still requires several further preprocessing steps before effectively being used for ASR training. This leads us to **RQ.3**, can the DAMP-MVP corpus be shaped to construct a sung speech recognition dataset? This chapter presents work on constructing the DSing dataset, a novel dataset based on the DAMP-MVP corpus constructed for English sung speech recognition.

Having built a large sung speech dataset, we can then investigate new acoustic modelling approaches to exploit the musical characteristics of the sung speech derived from the

[1] http://www.smule.com

differences between the sung and spoken speech or musically-motivated features (**RQ.4**). Chapter 3 explained how traditional state-of-the-art ASR systems designed for spoken speech typically use MFCC acoustic features to capture the phonetic characteristics of the signal and i-vectors (Dehak et al., 2011) to capture the speakers' characteristics (Han et al., 2017; Panayotov et al., 2015). This kind of representation has also been used in sung speech ASR systems, obtaining promising results, but with error rates still far above those seen in spoken speech tasks (Kruspe, 2016b; Mesaros and Virtanen, 2010a; Roa Dabike and Barker, 2019; Tsai et al., 2018). However, F_0 has an important influence on the formant frequencies (Chapter 2). For example, female singers constantly adjust F_1 to approximate their frequency to F_0 when singing in their higher pitch range. The much higher sung speech pitch range increases the phonetic boundaries in the MFCC space and makes it harder to characterise the speaker by features like i-vectors due to the higher within speaker variability. It is hypothesised that pitch and other musically-motivated features could be informative when supplied in conjunction with MFCCs as long as the classifier can model the correlations well. Note that some novel end-to-end deep learning ASR systems attempt to learn all the ASR modules simultaneously directly from the raw signal, i.e., without the need for engineered features (Chorowski and Jaitly, 2017; Graves et al., 2006). However, one of the goals of this chapter is to evaluate the effect of using musical knowledge directly.

This chapter is organised as follows. Section 4.2 presents the design and construction of the 'DSing' dataset, a novel dataset based on the DAMP-MVP corpus constructed for English sung speech recognition. Section 4.3 evaluates how well a state-of-the-art spoken speech recognition architecture performs when utilising the DSing data for training acoustic models and lyrics text for training a language model. This system serves as a baseline for the sung speech recognition stage of the thesis. Section 4.4 presents different musically-motivated features of the singing evaluated in this thesis, describing how these features are extracted from the signal and presented to the acoustic model to improve recognition performance. Section 4.5 presents the experimental methodology employed to analyse the effect of the musically-motivated features described in the previous section. The results and analysis are provided in Section 4.6. Finally, Section 4.7 summarises the chapter.

4.2 Construction of the corpus

This section presents the construction of the *DSing* dataset, a large dataset designed for unaccompanied sung speech ASR based on the Digital Archive of Mobile Performances - Smule Multilingual Vocal Performance 300x30x2 dataset (DAMP-MVP). DAMP-MVP is the third corpus originated from karaoke performances and released by the Smule collaborative

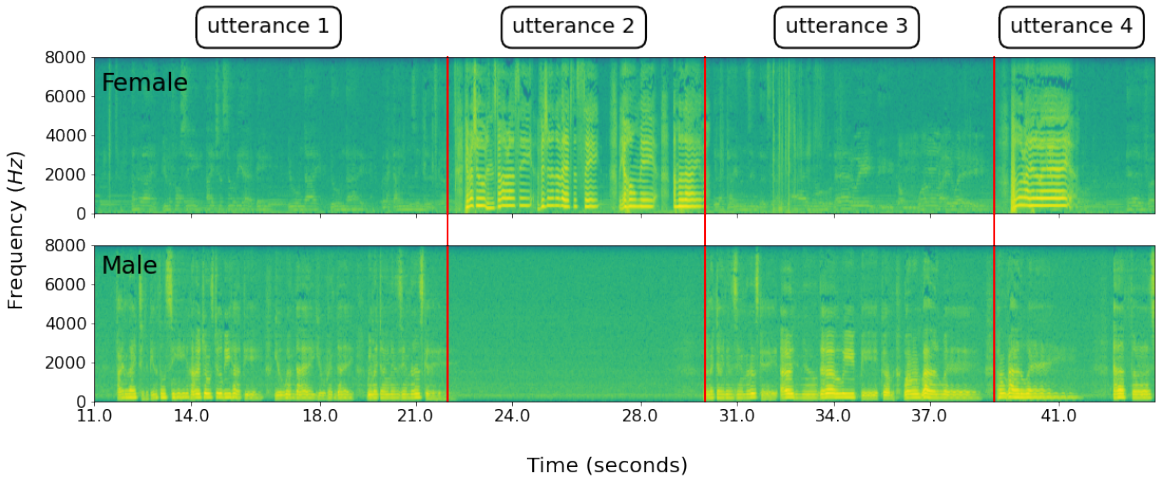


Figure 4.1 Spectrograms of a 33-seconds (seconds 11 to 44) excerpt from a performance of the song *Diamonds* by *Rihanna*. The song was performed in duet ensemble by a female (top) and male (bottom) singer. The song was performed by female (top) and male (bottom) singers in a duet ensemble. Each singer sang an utterance in turns, except for *utterance 4*, which was sung simultaneously.

karaoke mobile application. Smule is a mobile karaoke social network application that allows users to sing popular songs using their mobile phones. In Smule, a version of a song is called an "arrangement". An arrangement describes different components of the song, including the characteristics of the instrumentation of the background accompaniment, the lyrics and timings, which means that there might be multiple arrangements for the same song. For example, two arrangements of one song could share the same lyrics and timings but differ in the instruments composing the background accompaniment, e.g., one arrangement using the original background accompaniment of the song and the other using an acoustic guitar or piano recording. On Smule, songs can be performed as a solo or duet. In a solo performance, one singer performs the whole song. However, in a duet, two singers perform different parts of a song. Figure 4.1 shows the spectrograms from a 33-seconds audio excerpt of two singers singing the song *Diamonds* by *Rihanna* in duet mode. In this sample, a male and female singer take turns performing different parts of the song. The male singer sings solo in the first and third utterances, the female in the second utterance and both singers sing the fourth utterance simultaneously.

4.2.1 Description of the DAMP-MVP dataset

The DAMP-MVP dataset is a collection of 18,676 singing performances from 13,154 singers covering 5716 arrangements. The performances are equally distributed by gender, organised

by the country (30 in total) where the performance was recorded, i.e., the country label refers to the singer location during the recording and not necessarily to their nationality. It can be assumed that the country is a good indicator of nationality, therefore, of the accent. Further, DAMP-MVP provides the lyric prompts presented to the singer, along with the prompt timings making it easier to align the song lyrics to the signals.

Smule collected and processed the DAMP-MVP data during the second half of 2017 and released it in early 2018. Samples selected correspond to the two most popular singers (male and female), from the 300 most popular arrangements, from 30 countries. The popularity of the arrangements was determined by counting the number of performances within a country. The performance popularity was determined by counting the numbers of listens and votes cast by users of the Smule application. Smule assumed that the up-voted performances represent well sung and good sound quality recordings.

When using the Smule application, users sing along to a karaoke accompaniment track playing on their device. Users will typically use their headphones so that their voice is captured in isolation from the accompaniment. This means the data can be used to study sung speech recognition in isolation from musical source separation challenges. Research can instead focus on the challenges of sung speech recognition itself. This assumption was tested by previewing 100 randomly selected recording samples. It was found that for 88% of these recordings, users were wearing headphones, while for the remaining 12%, this was not as evident by the presence of the background accompaniment. Of the accompaniment-free data, about 15% had appreciable levels of environmental noise (i.e., performers were using the application in a noisy location). No data was discarded at this stage. However, noisy samples will be filtered out by employing a cleanup stage during the acoustic modelling (refer to Section 4.3.2).

As discussed in Chapter 2, singers with different levels of proficiency present different breathing periods, pitch range, minimum and maximum pitch, and format control. Due to DAMP-MVP is a corpus originated from karaoke singing, containing singers with different levels of singing proficiency, the DAMP-MVP corpus may not contain sufficient recordings portraying specific singing characteristics like very high pitch singing. However, the karaoke characteristic of the data also means that most of the songs represent popular songs that most singers (proficient and amateurs) feel more comfortable singing, i.e., songs that are familiar to the singer with expected pitch within the singers' voice range. Additionally, due to the numerous songs and singers contained in the corpus, it can be assumed that acoustic properties like pitch variation within a word, phoneme extension and phonemes at different pitches are well depicted in the data, allowing the learning of the acoustic patterns. The

process of adapting the DAMP-MVP corpus to construct the DSing ASR dataset was divided into four steps:

1. Normalise the lyrics prompt to an utterance-level structure.
2. Select English language songs.
3. Align the lyrics prompt with the corresponding performing times.
4. Split the data into train, development and evaluation sets.

4.2.2 Lyrics Prompt Normalisation

The DAMP-MVP dataset provides the text prompts shown to performers along with the time they were displayed. For most songs, these are in a convenient utterance-level format, i.e., one prompt and timestamp per sentence to be sung (typically a single line from the song lyrics). However, some of the text prompts are provided in small speech units corresponding to monosyllabic words or to parts of multi-syllabic words. For the syllabic level prompts, the timestamps mark the start of the corresponding syllable. Note that there are no labels indicating if the speech unit corresponds to a word or part of a word. Figure 4.2 shows six syllabic-level prompts from one arrangement of the song *Play That Song* by *Train*. In this arrangement, the sentence "play that song" is provided utilising one prompt per each word (i.e., prompts with timestamps 4.8, 5.4, and 6.0 seconds in Figure 4.2a). But, multi-syllabic words are split into sub-words comprising one or two syllables, like the word "favorite" that is provided into two separated prompts, "favo" timestamp 29.6 and "rite" timestamp 30.1 seconds (Figure 4.2b).

The Smule application has an option on the website that allows users to submit their arrangement of different songs. The submission process starts by requesting the submission of the backtrack audio and the lyrics. Then, the process request users to mark the times when each lyrics sentence starts (e.g., an acoustic arrangement of the song "Chandelier" by "Sia"[2]). However, some songs are generated internally by Smule (e.g., "Bad Habits" by "Ed Sheeran"[3]) or submitted by the original artist (e.g., "Play that Song" by "Train"[4]). These songs include annotations to the vocal melody that the singer should produce when performing the song. The melody would typically vary in a sentence, having a different pitch per syllable. In the DAMP-MVP, the user-generated arrangements are provided as

[2]https://www.smule.com/song/sia-chandelier-karaoke-lyrics/474572_111629/arrangement
[3]https://www.smule.com/song/ed-sheeran-bad-habits-karaoke-lyrics/15464100_15464100/arrangement
[4]https://www.smule.com/song/train-play-that-song-karaoke-lyrics/3769302_3769302/arrangement

```
{
    "t": 4.848488,
    "l": "Play"
},
{
    "t": 5.454548999999999,
    "l": "that"
},
{
    "t": 6.06061,
    "l": "song,"
},
```


(a) Monosyllabic words.


```
{
    "t": 29.696988999999995,
    "l": "favo"
},
{
    "t": 30.101029666666662,
    "l": "rite"
},
{
    "t": 30.303049999999995,
    "l": "re"
},
```

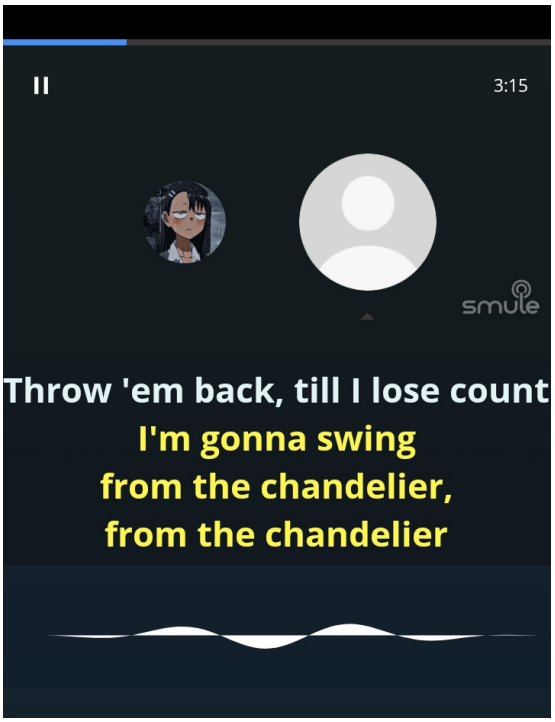

(b) Sub-words.

Figure 4.2 Six prompt lyrics from two parts of the same arrangement of the song *Play That Song* by *Train*. This arrangement provides the prompts at sub-word units level. In some cases, (a) the prompts contain monosyllabic words for the sentence "play that song" and, in others, (b) the prompts contains one or more syllables of the multi-syllabic words of the sentence "favorite record".

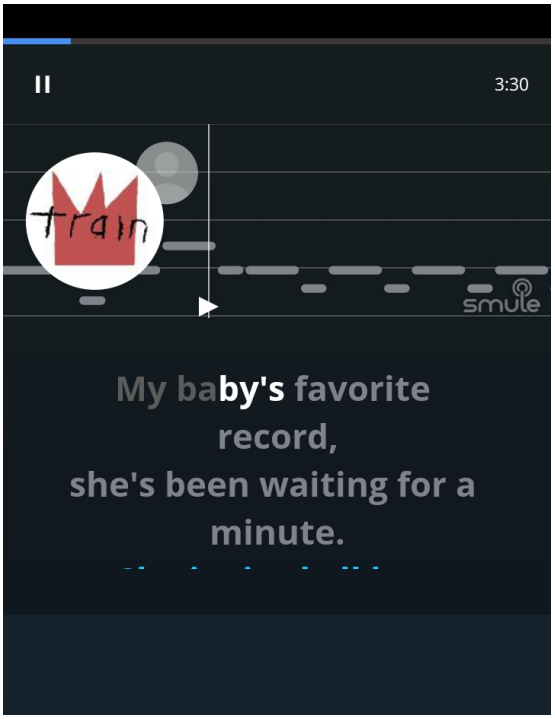
sentence-level prompts, and the arrangements generated by Smule or the original artist are provided as syllable-level. Note that the melody annotation is not included in the corpus.

Figure 4.3 presents two screenshots from the Smule application showing how the arrangements are presented to the singer. Figure 4.3a shows how a user-generated song is presented to the user. In this figure, the lyrics are presented by sentences, highlighting the current sentence in white colour and the following sentences in yellow. Figure 4.3b shows a song generated by the original artist. The user is presented with the expected melody and the pitch currently produced at the top of the screen. When a Smule user is singing, the expected melody (represented by the horizontal grey lines) moves from right to left. The vertical white line is a static line that intersects with the grey lines to indicate which syllable is sung at that moment. The white triangle on the white line changes its position along the line to show the singer's pitch. When this arrow matches the horizontal grey lines, it indicates that the singer is producing the expected pitch. The same screen distribution for an artist-generated song is presented for Smule-generated ones.

Chapter 3 mentioned that acoustic modelling aims to learn the relationship between speech audio signals and phonetic units by utilising datasets containing speech audio recording and corresponding text transcription. To meet that structure, syllable-level lyrics from DAMP-MVP have to be transformed to sentence-level. This can be done by matching the text prompts to the lines of the song lyrics recovered from the Smule website. However, to minimise the number of queries to the Smule website, the process first identifies the lyrics provided at syllable-level by counting the words and sub-words in each prompt. Some syllable-level lyrics may include prompts containing several words, e.g., one syllable-level



(a) Screenshot of the song "Chandelier" by "Sia", a song providing sentence-level lyrics annotation .



(b) Screenshot of the song "Play that Song" by "Train", a song providing syllable-level lyrics annotation.

Figure 4.3 Screenshots from the Smule application showing how sentence and syllable-level lyrics are presented to the user. In (a), sentence-level lyrics are presented by highlighting the whole current sentence, without any indication of target or produced pitch. In (b), syllable-level lyrics are presented by highlighting the current syllable in the lower half of the screen and indicating the expected versus the produced pitch for that syllable in the upper half.

arrangement of the song "Rockabye" by "Anne-Marie"[5] has a prompt comprising two syllables (i.e., the text "chose to" at timestamp 94.7 seconds). Therefore, arrangements with more word/sub-word structured prompts than prompts containing several words are considered syllable-level and sentence-level otherwise.

With the syllable-level arrangements identified, the process recovered their lyrics from the Smule website. This is made possible by a unique song arrangement label. Then, the words/sub-words prompts were matched with the recovered lyrics, setting the start time for the reconstructed sentence equal to the timestamp of the prompt text corresponding to the beginning of the sentence.

[5]https://www.smule.com/song/clean-bandit-ft-anne-marie-rockabye-karaoke-lyrics /3771372_3771372/arrangement

4.2.3 Selecting English Language Songs

DAMP-MVP corpus does not provide information related to the language of the lyrics. Considering this work investigates acoustic modelling for English sung speech, it is crucial to know the language of the lyrics. Therefore, non-English songs were filtered out by detecting the lyrics' language using the Compact Language Detector 2[6] (CLD2) library wrapped by the *Polyglot*[7] *Python* module (a natural language module that supports several multilingual applications). CLD2 probabilistically detects up to 165 languages from plain text or HTML. A language is identified either by Unicode script (e.g., Greek or Thai), or using a Naïve Bayesian classifier operating with 4-letter n-grams ("quadgrams") or single letter ("unigrams") for CJK languages. The training corpus was manually constructed based on chosen web pages for each language and augmented by an automated scraping of over 100 million additional web pages. For each text processed, CLD2 reports the three most likely languages with a confidence score. Note that CLD2 is not designed to perform well for a short text.

Lyrics from DAMP-MVP (now all transformed in a convenient sentence-level structure) are classified as English or non-English using the CLD2 package (wrapped by Polyglot). As CLD2 works on large text, the language was detected for the song lyrics, i.e., sentences were joined in a single text. Songs were considered English sung language when the confidence score of being English was greater than 85% and non-English otherwise. The confidence score threshold was selected to exclude multi-languages songs (e.g., several "Reggeaton" songs are made by a mixture of Spanish and English sentences). An inspection of 50 randomly selected songs from different countries showed this process to be robust in selecting English songs. The process resulted in 818 English arrangements and 5547 performances.

4.2.4 Audio Realignment

The realignment step aimed to match utterances with their corresponding transcription, using the prompts data and the unsegmented audio performance as input.

There are four main challenges to the alignment process. First, there is often a mismatch between the prompted lyrics and the words actually sung by the performer. This occurs because singers could omit, change or insert entire phrases, either by mistake or on purpose. Second, Smule songs are typically performed in duet mode; therefore, the recording from one performer is expected to contain only portions of the whole song, as illustrated in Figure 4.1. This means that, per each singer, there are prompt lyrics without a singing counterpart.

[6] https://github.com/CLD2Owners/cld2
[7] https://polyglot.readthedocs.io/en/latest/

Third, there can be considerable differences between the prompt timings and the onsets of the corresponding utterances. Generally, prompts appear early to allow the singer time to prepare, but the lead time is not always the same. Further, singers may start utterances considerably late if they are not familiar with the song. Finally, there is not a one-to-one correspondence between utterance-level prompts and sung utterances. A continuously sung utterance may span more than one prompt, i.e., there is no natural pause at the end of every line of a song. This is especially true for experienced singers who know the song lyrics and do not need to pause to read or prepare.

The realignment process in this thesis only attempts to match lyrics lines with the singing counterpart. However, there is no guarantee that the singing perfectly matches the transcription. Therefore, a simple temporal match that is not concerned with the singing and transcription content will suffice. Note that several lyrics-to-audio alignments approaches in the literature deal with the challenges of the task. However, these approaches assume that the singing segments match the lyrics text. To read more about lyrics-to-audio alignment and its application, refer to Fujihara and Goto (2012).

The alignment process attempted to deal with the above challenges using a rule-based algorithm. The algorithm matches a sequence of utterance-level prompts with a sequence of non-silence signal segments extracted from the recordings. Utterance-level prompts are recovered using the process described above. An end-time is associated with each prompt taken as the start time of the following prompt. The non-silence segments are extracted from the signal using a simple energy-based activity detector implemented in the *Pydub*[8] *Python* module. The algorithm uses a 20 *ms* window and a 1 *ms* frame step and classifies frames as either silence or non-silence according to whether the window root mean square (RMS) energy is lower or higher than -25 *dB* below the maximum signal amplitude. Silences segments shorter than 20 *ms* (e.g., silence within a word) are converted to non-silence. Then all non-silence segments are located (i.e., sequence of non-silence frames bounded by silence). The start and end time of each segment is noted.

The alignment algorithm uses the start and end times to pair utterance prompts to corresponding signal segments. However, it is necessary to join two or more non-silence segments to match with a single prompt in some cases. This occurs when an utterance has been split by the existence of a small silence (e.g., due to aspiration). In other cases, it is necessary to join several prompt texts to match a single utterance, i.e., when the performer sings more than one line without an intervening pause. To achieve this, the algorithm proceeds as follows:

[8] https://github.com/jiaaro/pydub

1. Prompts that do not intersect with any existing non-silence segment are discarded (the singer failed to sing the lyric).
2. Non-silence segments that do not intersect with any existing prompts are discarded (i.e., typically extraneous noise such as coughing).
3. Wherever more than one non-silence segment intersects with the same prompt, the segments are joined.
4. Wherever more than one prompt intersects with the same non-silence segment, the prompts are joined.
5. If every segment does not intersect with only one prompt, return to step 3.
6. Non-silence segments are now paired to their intersecting prompts.

After running the algorithm, a sample of 100 segments was examined to evaluate the quality of the alignment. It was found that 60% of the segments were correctly aligned to the prompt, i.e., with correct timings and with prompts that provided the correct transcription. A further 32% were only partially correct, i.e., the segment was correctly assigned to prompt lyrics, but the performer added, substituted or omitted some words. 8% of the segment alignments had totally failed. Typically, in these cases, prompts were aligned to segments containing only background noise introduced by a failure of the earlier voice activity detection stage. Due to these imperfections, the baseline alignment processing is only used for generating the training data.

The testing data was constructed by manually correcting the alignment timings and re-transcribing the lyrics when singers altered the expected lyrics. Sentences that were not possible to correctly realign or where the sung words were not clear for obtaining a correct transcription were discarded. This ensures that accurate recognition performances can be measured.

4.2.5 Defining the Train And Tests Sets

Having processed the DAMP-MVP dataset and generated several utterances matched with their corresponding transcriptions, this section describes how these utterances are distributed into training and testing data.

First, taking advantage of the singer country information provided by DAMP-MVP, the utterances were split into three datasets, *DSing1*, *DSing3*, and *DSing30*, that progressively introduce performances from a broader set of countries. DSing1 is constructed using the

Table 4.1 Description of the DSing dataset, detailing the DSing1, DSing3 and DSing30 training sets and the hand-corrected development and evaluation test sets.

Set	Singers	Arrangements	Performances	Utterances	Hours
DSing1	352	219	434	8794	15.1
DSing3	1050	398	1343	25,526	44.7
DSing30	3205	821	4320	81,075	149.1
Development	40	40	66	482	0.7
Evaluation	43	42	70	479	0.8

subset of recordings from singers registered as users in Great Britain. DSing3 is constructed from the subset of recordings from singers registered in one of the three native English speaking countries, namely, Great Britain, the USA and Australia. Finally, the largest data set, DSing30, is constructed using singers from all 30 countries available in the DAMP-MVP dataset. Note that only the English songs are being used in all cases, i.e., DSing30 will contain many recordings sung in English by non-native English speakers.

The data from DSing1 was further split into train, development and evaluation sets, including 80%, 10% and 10% of the data, respectively. Care has been taken to ensure that the sets are disjoint with respect to both singers and arrangements, i.e., no singer or arrangement seen in one set is seen in any other set. The many-to-many association between singers and arrangements made this complicated, and some data has to be lost to meet this constraint. This was done by first counting the number of arrangements per singer that intersect with the different sets. Then, singers were assigned to the set where the number of arrangements was greater, discarding the other arrangements. Any arrangement from the development and evaluation sets were excluded from the DSing3 and DSing30, so these sets can be used for training.

Six hundred utterances (about one hour) were randomly selected from the development and evaluation set and manually corrected to ensure their quality and balance. The utterances were realigned, correcting the start and end time. Also, the prompt transcriptions were fixed by replacing the lyric annotation with the actual sung words. Utterances were discarded when background accompaniment was found in the recording (i.e., the user was not using earphones). Care was taken to keep roughly 20 utterances per speaker. This process resulted in 482 utterances covering 40 speakers (27 female and 13 males) for the development set and 480 utterances covering 43 speakers (30 females and 13 males) for the evaluation set. The final size of the DSing training and test sets is summarised in Table 4.1.

4.3 Development of the Baseline ASR System

The previous section described the procedure to construct the DSing dataset for sung speech ASR training based on the DAMP-MVP corpus. This section reports the work on evaluating the performance of a state-of-the-art ASR system designed for spoken speech and trained on the DSing data utilising the Kaldi toolkit (Povey et al., 2011), i.e., using conventional acoustic modelling ideas taken from spoken speech ASR. Language models are trained using an in-domain lyrics corpus and acoustic model using the DSing1, DSing3 and DSing30 sung speech train datasets.

The models evaluated in this section will serve as a reference for the rest of the chapter.

4.3.1 Language Model

The language model was constructed based on lyrics obtained from LyricWiki[9] (website was closed on September 21, 2020), a free wiki website that stored lyrics of over 2 million songs. Care was taken to select a list of lyrics that match the music style and artist name found in the DAMP-MVP dataset and to include popular artists that are more likely to be found in karaoke performances. The corpus was constructed by, first, including the lyrics of the songs by the artists featured in the DSing3 training set (LMSmule). Then, the LMSmule was augmented with all the available lyrics by all the artists from the Billboard list "The Hot 100" for the 31st December of the years 2015 to 2018 (LMSmule+).

Care has been taken to ensure that the LM test data does not overlap with the training data unfairly. To avoid the inclusion of lyrics from songs in the DSing test sets, arrangements that share more than half of their sentences with one of the test set songs were discarded. This filtering is based on content rather than song title because song titles in the Smule data and LyricWiki are not always easily comparable. For example, arrangements in Smule can have a suffix describing some characteristic of the arrangement (e.g., *bohemian rhapsody short version*) that will not match the official song title. Additionally, care was taken when selecting songs to avoid songs being included multiple times, as can happen when songs appear on multiple albums covered by different artists. This prevented unfairly increasing the probability of sequences of words, e.g., the sentence *"A mosquito, my libido"* from "Smells Like Teen Spirit" by "Nirvana" is an improbable sentence. Still, the song has at least ten different cover versions.

The resulting LMSmule and LMSmule+ corpora were then 'normalised', i.e., text were converted from their written form into their verbalised form. This stage included converting numbers into text; excluding non-lyrics content, like 'chorus' or 'verse' labels; replacing

[9] https://web.archive.org/web/20200830142257/https://lyrics.fandom.com/wiki/LyricWiki

all non-ASCII characters to Unicode strings using the *normalisation form compatibility decomposition* (NFKD) algorithm (Whistler, 2021). Some lyrics contain atypical spellings with over-repeated letters to indicate that the singer should extend the sound. For example, in an arrangement of "Love Can Move Mountains" by "Celine Dion", the word "LOVE" is spelt as "LOOOOOOVE" to indicate an extended vowel sound. These letter repetitions can be detected by counting the number of repeated letters in the word and removing the 'extra' repetitions, i.e., in the English language, vowels can be found as single or double in some words but never with more repetitions. Then, words can be corrected by matching them with a dictionary, e.g., The words "LOOVE" does not exist in a dictionary but, the word "LOVE" does.

The resulting language model corpus contains the lyrics of 44,287 songs from 456 artists, where 125 artists are featured in the DSing3 training set. In total, there are 1,747,731 lines of text from lyrics consisting of 11.5 million tokens and 91,654 unique words. A lexicon of 26K words was defined by selecting the most frequent words. Three approaches were evaluated to select the most frequent words in the corpus. However, selecting the N more frequent words in the corpus showed similar performance to more complex approaches (see Section 4.3.1). This lexicon was found to encompass 92% of the DSing1 training dataset vocabulary and 97% of the development dataset. Pronunciations were obtained from the CMU pronunciation dictionary[10], which covered 80% of the words. For the remaining words, pronunciations were automatically generated using the Phonetisaurus G2P toolkit (Novak et al., 2015). The 20% of words not included in the CMU dictionary mainly correspond to offensive words, names, stylised spellings, and foreign words.

A 3-gram and 4-gram MaxEnt language model were built using the SRLIM toolkit (Stolcke, 2002). These models were assessed in terms of their perplexity, which measures how well the model predicts a sample. A low perplexity is better. For comparison, the perplexities of the out-of-domain LibriSpeech (Panayotov et al., 2015) language model was used. LibriSpeech is a well-labelled set but sourced from spoken speech audio-books data. The perplexities were evaluated on the DSing development set, and the results are presented in Table 4.2. The out of domain LibriSpeech LM proves to be the worst model with the highest perplexity of 206 for 3-gram and 193 for 4-gram models. This was not unexpected as the grammatical structure of lyrics tends to be closer to that of poetry than to that of prose. A significant improvement was obtained when using the LMSmule LM, obtaining perplexities of 103 and 100 for 3-gram and 4-gram, respectively. This improvement showed the benefits of using an in-domain corpus. A further perplexity decrease was achieved using the extended

[10]http://www.speech.cs.cmu.edu/cgi-bin/cmudict

Table 4.2 Language Models Perplexity

Model	**3-gram**	**4-gram**
LibriSpeech	206	193
LMSmule	103	100
LMSmule+	73	60

language model LMSmule+, reducing from 103 to 73 for 3-gram and from 100 to 60 for 4-gram. The model selected for the baseline was the LMSmule+.

Selecting the Vocabulary

The vocabulary was selected by evaluating three approaches:

1. tf: A *term frequency* approach that counts the number of times a term (word) appears in the corpus divided by the total of terms in the corpus, defined as:

$$tf = \frac{\sum w_i}{Total\ Words} \tag{4.1}$$

   where $\sum w_i$ is the number of times that the word w_i appears in the corpus and $Total\ Words$ is the total number of words in the corpus.

2. tf_{song}: extends tf by weighting the word frequency by the song frequency computed as the number of songs in the corpus containing the term divided by the total number of songs. This approach reduces the probability of choosing common words that are too song- or genre-specific, e.g., slang words in "rap" music.

$$tf_{song} = tf \times \frac{\sum Song_{w_i}}{Total\ Songs} \tag{4.2}$$

   where tf_{song} is the frequency of the word w_i weighted by the song frequency, $\sum Song_{w_i}$ is the number of songs that contain the word w_i, and $Total\ Songs$ is the total number of songs in the corpus.

3. $tf_{song/artist}$: extends tf further by weighting by the number of artists in the corpus that use the term. This approach attempts to increase the probability of choosing common words that several artists use.

$$tf_{song/artist} = tf_{song} \times \frac{\sum Artist_{w_i}}{Total\ Artist} \tag{4.3}$$

Table 4.3 Coverage percentage per term frequency approach and vocabulary size. For each term frequency approach, the effective number of words selected is included. Values in bold show the final vocabulary size selected.

Words Target	tf		tf_{song}		$tf_{song/artist}$	
	Words	Coverage	Words	Coverage	Words	Coverage
5000	5032	92.44	5000	92.72	5000	93.15
10000	10,039	96.01	10,000	96.15	10,001	96.15
15000	15,206	97.86	15,071	98.00	15,004	98.15
20000	20,832	98.57	20,230	98.72	20,015	98.72
25000	**26,645**	**98.86**	25,339	99.00	25,035	99.14
30000	34,093	99.29	30,300	99.29	32,059	99.29

where $tf_{song/artist}$ is the frequency of the word w_i weighted by the song frequency and artist frequency, $\sum Artist_{w_i}$ is the number of artists in the corpus that use the word w_i, and $Total\ Artist$ is the total number of distinct artist in the corpus.

The three approaches were evaluated in terms of the *coverage percentage* corresponding to the percentage of words that intersect with the development set. The coverage percentage using vocabularies of sizes 5000, 10000, 15000, 20000, 25000 and 30000 were evaluated for the three approaches. For all the vocabulary sizes, the word lists were constructed incrementally. The process starts by including all the words with the highest frequency. Then, if the number of the currently selected words is lower than the target size, all the words with the next descending frequency are included. This process is repeated until reaching the target vocabulary size. Note that no hard cut-off procedure was implemented, i.e., the final size of the vocabulary can be higher than the target number of words.

Table 4.3 shows the number of selected words and coverage percentage for each term frequency approach and the different vocabulary sizes. Note that when selecting a vocabulary size of 5000 words for the tf_{song} approach, the resulting size is 5032 words. The extra 32 words are the result of the increasing number of words that share the same frequency. The same behaviour was obtained with the different vocabulary sizes and approaches. The $tf_{song/artist}$ approach resulted in the best word coverage percentage with, for example, 99.14% of coverage when selecting 25000 words. However, similar performance is obtained using the simplest tf with a reduction of 0.28% coverage but with a lower computation cost. Therefore, the final vocabulary was selected utilising the simplest tf approach with an effective size of 26K that covers 98.86% from the words in the development set.

4.3.2 Acoustic Model

The initial experiments aimed to construct a well-performing baseline. The acoustic model was constructed using the state-of-the-art factorised Time Delay Neural Networks (TDNN-F) (Povey et al., 2018) for spoken speech. The system is based-on the LibriSpeech (Panayotov et al., 2015) system from Kaldi's recipes.

The acoustic features used are 13 Mel frequency cepstral coefficients (MFCCs) plus delta, delta-delta and energy, with 25 milliseconds (*ms*) frame length and 15 *ms* of overlapping. Initial alignments are performed with a triphone speaker adapted Gaussian mixture model (GMM) and feature-space maximum likelihood linear regression (fMLLR). The triphone model is used to clean the training data using the standard Kaldi cleanup process[11]. This process is set to remove bad utterances from the training data (e.g., filtering utterances with an incorrect transcription). This process removed about 10% of the training utterances, which mainly corresponds to erroneous singing segments resulting from the realignment step during the DSing dataset construction. Using the resulting 'clean' training data, a factorised time-delay neural network (TDNN-F) (Povey et al., 2018) acoustic model is then trained using 40 MFCCs with two frames context plus 100-dimensional i-Vectors, and a lattice-free maximum mutual information (LF-MMI) loss function (Povey et al., 2016).

4.4 Acoustic Modelling Using Musically-Motivated Cues

The previous section evaluated the performance of a spoken speech ASR system trained on singing data. For acoustic modelling, three increasingly large training sets were defined utilising English sung performances of unaccompanied karaoke singing based on data released by the Smule application. Language models were trained on in-domain lyrics data sourced from the LyricWiki website. The performances obtained from these models are utilised as a reference for the current section.

Chapter 2 discussed several differences in the sound production mechanisms when singing and talking and how these differences make sung speech less intelligible than spoken speech. For example, in singing, the pitch has a larger range, a higher mean and a lower variation within a vowel than in spoken speech, and sung speech tends to have less glottal noise and less variation of period and amplitude in the signal. Singing at different pitches results in shifting the formant frequencies and reducing the vowel space. These differences may reduce the effectiveness of using MFCC acoustic features and i-vectors speaker-specific representations. It is known that phoneme classification is independent of the pitch, i.e., let p

[11] https://github.com/kaldi-asr/kaldi/blob/master/egs/wsj/s5/steps/cleanup/clean_and_segment_data.sh

be the pitch observation, f some MFCC features, and q the phoneme then

$$p(q|p) = p(q) \tag{4.4}$$

However, this does not mean that the probability of a phoneme having observed the MFCC is also independent of the pitch,

$$p(q|f,p) \neq p(q|f) \tag{4.5}$$

F_0 is highly correlated to some aspects of the spectral vector (Singer and Sagayama, 1992), and vowel perception changes from lower to higher vowels as F_0 increases (Hirahara and Kato, 1992). This correlation between the pitch and the spectrum envelope affects the boundaries of the MFCCs space. Consequently, due to the higher pitch range of sung speech (Section 2.4.5 on page 37), the variability of the MFCCs space for sung speech will be higher than for spoken speech, reducing the ability of speaker-dependent features like i-vectors to characterise the speaker correctly.

Farrus et al. (2007) reported that voice source quality parameters like jitter and shimmer carried speaker-specific characteristics that can help spoken speech ASR systems to improve recognition performance. However, Section 2.4.7 on page 41 shows that the link between voice quality features and speaker characteristics is reduced in sung speech due to their correlation to pitch frequency and speech styles. Nonetheless, some information, especially in the jitter parameter, is retained.

Chapter 2 discussed that in good song compositions, strong beats should match relevant syllables (Perricone, 2018). The beat cues hold information related to the song structure (Section 2.4.8 on page 42), tempo and word stress that may be helpful to improve the phoneme discrimination by normalising the syllable stress and tempo. Note that beat tracking is the first step for constructing beat-synchronous features, which try to find similar segments. These features are widely used in music information retrieval, for example, to find the correlation between two audio signal segments (Ellis et al., 2008).

This section hypothesises that musically motivated features, such as pitch, voice quality features and beat, may be informative when supplied in conjunction with MFCCs as long as one has sufficient data and a classifier (like a DNN) that can model their correlation well. The structure of the section is as follows. First, it presents the description of the musically-motivated features investigated, namely pitch, voicing degree, voice source quality features (i.e., jitter, shimmer and HNR) and musical beat. Then, it presents an evaluation of the use of speech rate using syllables-per-second measurements (Appendix B).

4.4.1 Pitch and Voicing Estimation

For pitch feature extraction, the Kaldi pitch tracker (Ghahremani et al., 2014) was utilised. The Kaldi pitch features were specifically designed for training spoken acoustic models together with MFCCs. It is based on the RAPT algorithm (Talkin and Kleijn, 1995), a time-domain F_0 estimator that uses normalised cross-correlation function (NCCF) to generate period candidates and dynamic programming to estimate the pitch and voicing state for each frame. The RAPT algorithm has several parameters that can be set depending of the scenario, including $F0_{min}$ and $F0_{max}$ for the minimum and maximum frequency search in Hertz, and proceeds as follow:

1. Provides two versions of the sample speech data, one at the original sample rate and one at a reduced sample rate.

2. Using a two-pass procedure, computes the NCCF.

   (a) Using the low sample rate signal, computes the NCCF for all lags in the F_0 range of interest. Then, it records all local maxima values, i.e., values closest to 1, excepting for zero lag which is 1 by definition.

   (b) Using the original sample rate signal, computes the NCCF for seven lags in the vicinity of the peaks estimated before.

3. Each NCCF peak estimated for the original sample rate signal generates a F_0 candidate.

4. Using dynamic programming, it determine the most likely estimated pitch and binary voicing classification (voiced or unvoiced) for each frame.

Unlike RAPT, the Kaldi pitch tracker does not make a hard voicing decision. Instead, it treats all frames as voiced sounds with an associated pitch and a probability of voicing (POV). Kaldi pitch works under the assumption that all frames correspond to voiced sounds, allowing a Viterbi search to interpolate pitch estimates across unvoiced frames. This is made possible by not limiting the search to the local maxima of the NCCF but allowing it to take any value. Additionally, before the pitch estimation, Kaldi pitch globally normalises the energy of the signal and then sub-samples the signal by using a low-pass filter to improve the accuracy and optimise the algorithm. The algorithm outputs pitch estimate and the NCCF on each frame. However, the Kaldi pitch provides four features resulting from a post-processing of the pitch estimate and NCCF:

1. A ‘log pitch’ value of the estimated pitch.

2. A Gaussian-distributed 'POV' feature computed as:

$$POV = 2((1.0001 - NCCF)^{0.15} - 1) \tag{4.6}$$

3. A 'normalised log pitch' by using a short-time mean subtraction. On each time t, a weighted average pitch is subtracted, computed over a window of 151 frames, centred at t and weighted by a probability of voicing computed as:

$$p = \frac{1}{1 + exp(-l)} \tag{4.7}$$

where l is an approximation of the log-likelihood ratio $log(p(voiced)/p(unvoiced))$ computed as:

$$l = -5.2 + 5.4\exp(7.5(a-1)) + 4.8a - 2\exp(-10a) + 4.2\exp(20(a-1)) \tag{4.8}$$

4. A 'delta-log-pitch' feature computed from the unnormalised log pitch utilising ± 2 frames of context.

The Kaldi pitch tracker possesses two main parameters, maximum pitch value (max-f0) and low-pass frequency cut-off (lowpass-cutoff), tuned by default for spoken speech as $400\,Hz$ and $1000\,Hz$, respectively. These parameters were tuned for sung speech using the MIR-1K pitch annotated sung speech dataset (Hsu and Jang, 2010). MIR-1K is a collection of 1000 male and female song clips totalling 133 minutes, extracted from 110 karaoke songs selected from 5000 Chinese pop songs. The MIR-1K provides manually annotated pitch values.

A grid-searching technique was employed to find the best values for the max-f0 and lowpass-cutoff Kaldi speech parameters for sung speech. The grid consisted of seven values for max-f0 (frequencies between 400 and $1000\,Hz$, with $100\,Hz$ step) and three values for lowpass-cutoff frequencies (1000, 1500 and $2000\,Hz$). The performance of the pitch estimation under the different parameters was assessed by using the gross pitch error (GPE) (Drugman and Alwan, 2011) and fine pitch error (FPE) (Asgari and Shafran, 2013) measurements. GPE calculates the proportion of frames classified as voiced by both the estimation and ground truth, where the estimated pitch deviates more than one semitone from the ground truth (Babacan et al., 2013). However, as the Kaldi pitch tracker does not perform a hard voiced decision, the GPE was calculated over all the ground truth's voiced frames. FPE is defined as the mean absolute error derived from the voiced frames where the reference deviates less than the GPE threshold. The results of the search are shown in Table 4.4.

Table 4.4 Fine Pitch Error (FPE): Mean absolute error in cents.

(a) FPE error.

max-f0	lowpass cutoff		
	1000	**1500**	**2000**
400	1.624	**1.622**	1.633
500	1.623	**1.619**	1.630
600	1.620	**1.618**	1.629
700	1.619	**1.617**	1.629
800	1.623	**1.621**	1.632
900	1.620	**1.619**	1.631
1000	1.622	**1.621**	1.632

(b) GPE error.

max-f0	lowpass cutoff		
	1000	**1500**	**2000**
400	5.801	5.814	5.806
500	2.225	2.259	2.277
600	1.889	1.907	1.912
700	1.881	1.899	1.904
800	1.880	1.891	1.900
900	1.863	1.885	1.890
1000	1.863	**1.882**	1.893

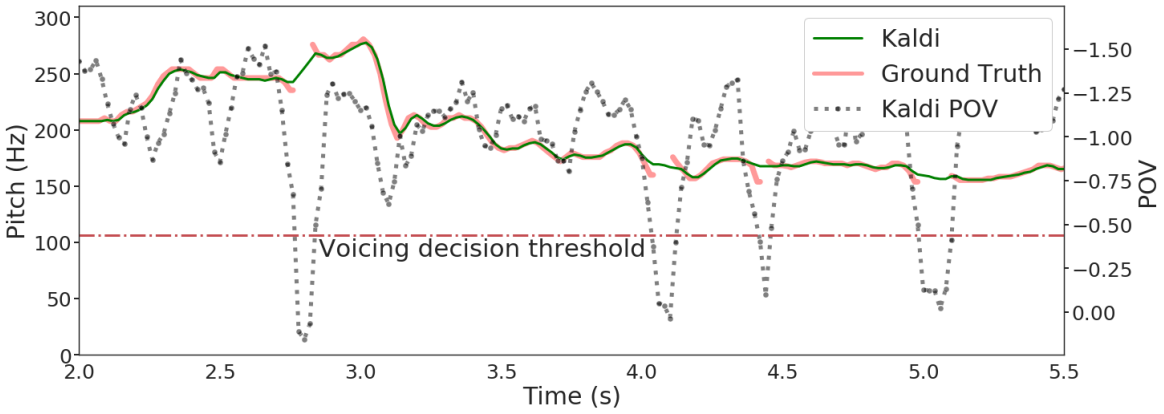


Figure 4.4 Kaldi pitch tracker's pitch and POV estimation contrasted with the ground truth for a 3.5 seconds excerpt from one MIR-1K sample.

The max-f0 and lowpass-cutoff parameters were set to 1000 Hz and 1500 Hz, respectively. The lowpass-cutoff consistently showed a lower FPE error at 1500 Hz, regardless of the max-f0. Also, for a lowpass-cutoff of 1500 Hz, the GPE error is lower at a max-f0 of 1000 Hz. Figure 4.4 shows an example of the predicted pitch by the Kaldi pitch tracker using these parameters, compared to the ground truth. Notice that the Kaldi pitch tracker predicted the values with high accuracy, and it interpolates between voiced frames to assign a pitch to the unvoiced ones. Additionally, this plot shows the POV estimation (right y-axis with inverted order); notice that in unvoiced areas, the POV value shows valleys to values below a voicing decision threshold estimated from MIR-1K.

4.4.2 Voice Source Quality

Three voice quality features (VQ) were evaluated: jitter, shimmer and harmonic-to-noise ratio (HNR). To extract these features, the Parselmouth (Jadoul et al., 2018) Python module was used. Parselmouth is a Python library that provides an interface to the Praat software (Boersma and Weenink, 2021) by directly accessing their C/C++ libraries. This work uses Parselmouth version 0.4.0, which interfaces Praat version 6.1.38.

Chapter 2.3 describes jitter and shimmer measurements as the voice perturbation in period and amplitude, respectively. These two parameters are computed by first detecting the glottal pulses and then obtaining the average between 2 consecutive periods for jitter or the amplitude between two consecutive periods for shimmer. However, the precision of these parameters depends on the accurate detection of the glottal pulses. In Praat, glottal pulses are measured using the "waveform-matching" method that attempts to determine at what time distance two consecutive waveforms look maximally similar. Titze and Liang (1993) describes that the procedure starts by searching for the location of the absolute minimum between the first two negative zero-crossings (a point $P(1)$). A point $P(2)$ between the second and third negative zero-crossings is found so that the mean square error between the two waveforms is minimal. This procedure is repeated until all cycles are processed. Titze and Liang (1993) and Boersma (2009) reported that the waveform-matching method is robust in detecting glottal pulses in the presence of additive noise, leading to more accurate perturbation measurements. This was especially true for normal voices with jitter lower than 1.03%, in contrast to methods like "peak peaking", which looks for time location where the waveform is at maximum.

The jitter measurement was obtained using the *Get jitter (local)* Praat method. This method takes a 'PointProcess' object (a Praat object containing the glottal pulses) from a signal and computes the local jitter using Equation 2.2 from page 22. The shimmer parameter was computed using method *Get shimmer (local)*. This method takes the same PointProcess computed for jitter but returns the average absolute difference between the amplitudes of consecutive periods (Equation 2.3 on page 22). In both cases, the computation is performed for the whole track, setting the parameters: the shorted period duration to 0.0001 seconds, the longest period duration to 0.02 seconds and the largest possible difference between consecutive periods to 1.3.

The HNR parameter is obtained from the signal's autocorrelation as described by Boersma (1993) (Equation 2.4 on page 23). The measurement is obtained by first computing the degree of the acoustic periodicity of the signal using the *To Harmonicity (ac)* Praat method with the parameters to default. Then, the HNR score is computed by averaging the harmonicity output.

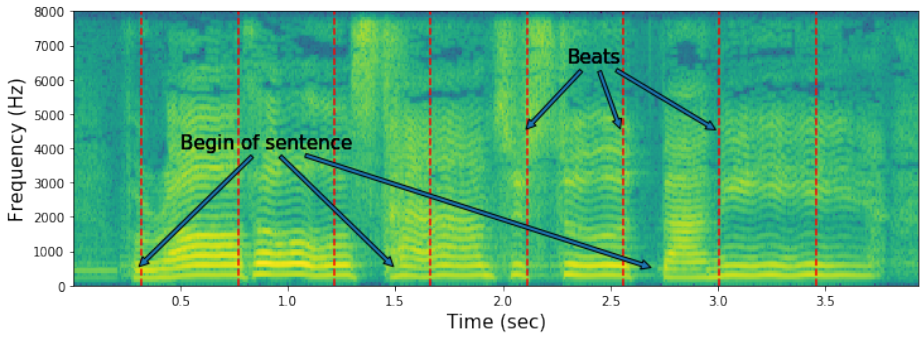


Figure 4.5 Spectrogram with the beat time and words boundaries extracted from a four seconds sample from the DSing development set.

The computation of each VQ feature is usually performed in a large window of 5 seconds in length to capture the variations from cycle to cycle. The parameters were computed from the whole utterance for these experiments, obtaining a single value for jitter, shimmer, and HNR. These parameters were presented to the model by concatenating the values with the MFCC vector, repeating the value on each frame.

4.4.3 Musical Beat

The musical beats were evaluated by informing the model with the distance to the closest beat. The beats location were extracted using the *LibRosa* Python module implementation of the dynamic programming beat tracking approach proposed by Ellis (2007). The algorithm assumes that the beats are located in the position of the strongest note onsets (a single instant that marks the start of a musical note) and that the song's tempo is roughly constant. The system uses a cost function that maximises both assumptions (for details of the implementation refer to Ellis (2007)).

The beat tracking algorithm proposed by Ellis (2007) returns the location (frames number) of the estimated beats. However, a closer inspection of the results of the beat extraction from examples from the DSing development set showed significant variability in the precision of the estimated beats, presenting several gaps resulting from undetected beats at the beginning or end of the audio. An attempt to correct the undetected beats were made by computing the average distance between the detected beats. The locations of the undetected beats were estimated using the average computed as the distance between them.

Note that, in some cases, estimated beats may not match with the relevant syllables. This mismatch may result from extracting the beats from a recording of a poor singing proficiency singer, e.g., an insecure singer not stressing the correct syllables. Poor proficiency in singing can result in errors in estimating the correct beat location. These errors are less likely to occur

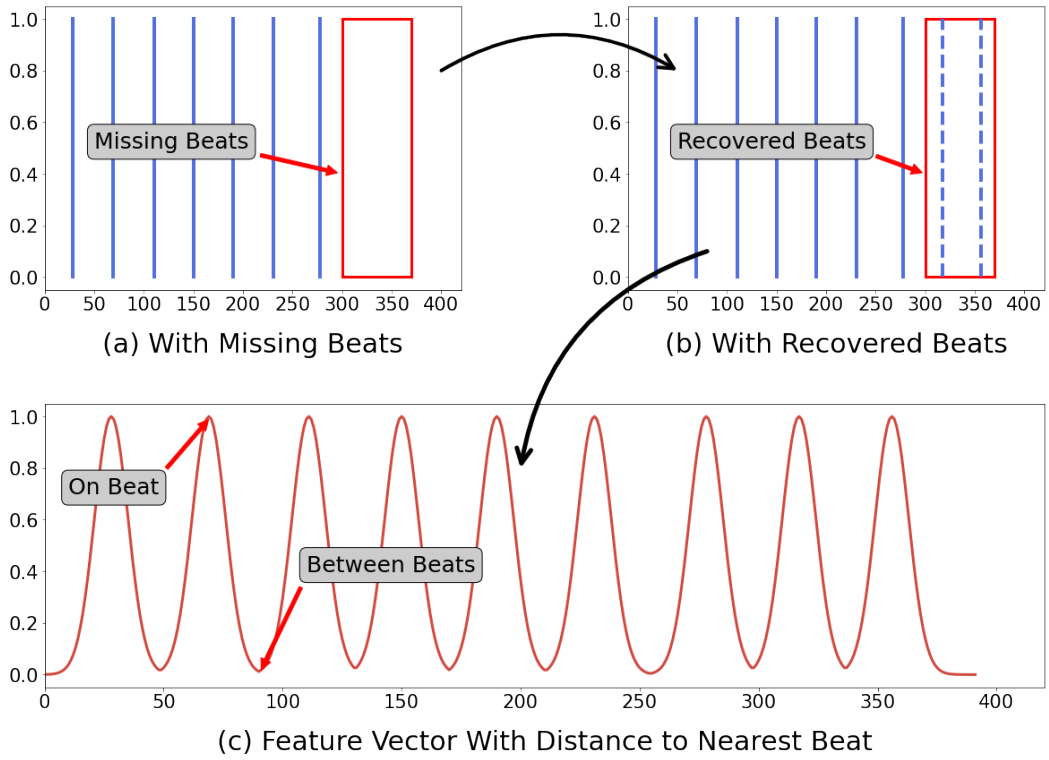


Figure 4.6 Illustration of the construction of the beat feature vector.

in the case of extracting the beats from an instrumental or an accompanied sung speech signal where more information for the beat estimation exists. Figure 4.5 shows the spectrogram of one excerpt from the DSing development set. The detected beats are marked by the vertical dashed red line. In this case, no clear correspondence between the beat location and the beginning of sentences is observable, e.g., the starting syllable of the third sentence occurs between two beats instead of on-beat.

Using the beats locations (the extracted plus the recovered undetected beats), a vector with the numbers of frames to the nearest beat is constructed. Then, the distance is updated using a normal distribution with mean zero and standard deviation equals to the deviation of the beats estimated. Finally, the vector is normalised to values between zero and one. The normalised vector shows the closeness to the nearest beat, being the maximum on-beat and the minimum between beats.

Figure 4.6 shows the construction of the beats feature vector for one sample in the DSing development set. The plot in Figure 4.6a shows the beats as obtained from the dynamic programming beat tracking approach. The blue lines mark the locations of the estimated beats. Note that there is a large segment at the end of the utterance, marked by the red rectangle, with undetected beats. The plot in Figure 4.6b shows the beat extracted complemented

with the estimated undetected beats (dashed blue lines). The plot in Figure 4.6c shows the normalised feature vector where the closeness equals to one on-beat and zero half way between two beats.

4.4.4 Syllables per Second

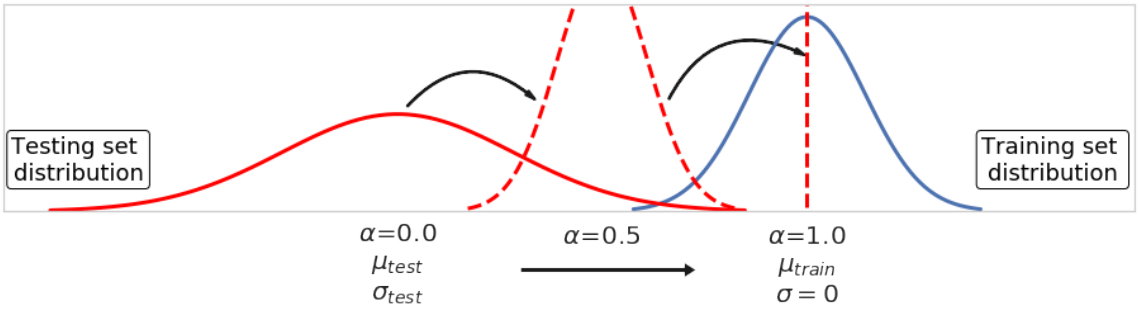
Additionally, the effect of normalising the testing set to the training speech rate was evaluated using the syllable-per-second ratio (SPS) measurement (for a complete description of SPS parameter refer to Appendix B). Two "speed perturbation factor" approaches were evaluated based on the training SPS per song (SPS_{song}).

The first approach attempts to 'move' the testing data towards the mean of the training data. This is the area where most of the data is located, and more robust patterns are expected to be learned. This approach assumes that the test data is not skewed and possesses a small variance, perturbing the data with a similar velocity without significantly distorting some samples over others. However, if the data is skewed, the perturbation of the samples in the extremes would be higher, rapidly adding more distortion to those samples. Its formulation is given by Equation 4.9:

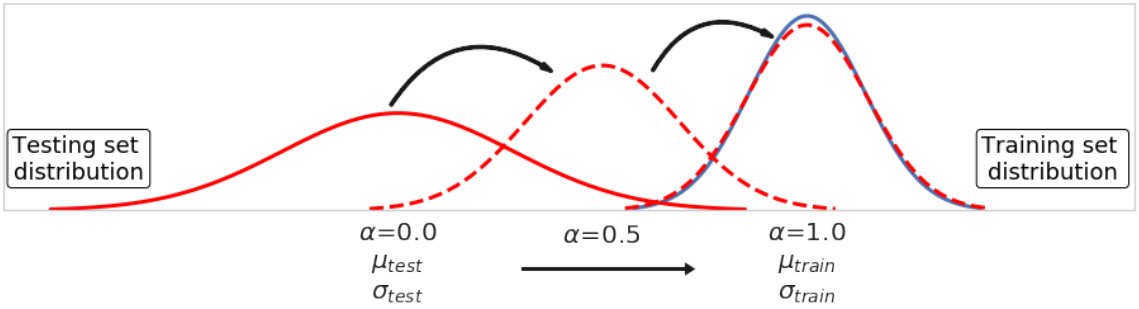
$$\varphi_1(test_i) = \frac{SPS_{test_i}}{(1-\alpha) \times SPS_{test_i} + \alpha \times \mu_{train}} \tag{4.9}$$

where $\varphi_1(test_i)$ is the normalisation factor 1 for the ith sample, SPS_{test_i} is the SPS_{song} for the ith sample, μ_{train} is the average SPS_{song} from the training data, and α is a weight factor valued between 0 and 1. When α is equal to zero, no perturbation is applied, keeping the speed of the test data unchanged. The effect of $\varphi_1(test_i)$ on the test distribution is illustrated in Figure 4.7a. In this figure, the distribution of the testing data moves from the original μ_{test} and σ_{test} when α is equal to zero, to μ_{train} and $\sigma = 0$ when α is 1.

The second approach attempts to move the testing data closer to a target distribution, i.e., it alters the testing data mean and variance scores. It first estimates a target distribution based on the training data, then perturbs the testing data moving it towards the new distribution. Due to this approach matching the testing distribution to a target one, the velocity of the perturbation of the different samples is less sensitive to skewness in the data. The process firstly calculates the mean and variance from the training data distribution (μ_{train} and σ^2_{train}), and the testing data distribution (μ_{test} and σ^2_{test}). Then, using these scores, it estimates a new target distribution with a target mean (μ_{target}) and a target variance (σ^2_{target}). Using a weight factor (α) between 0 and 1, the target distribution is defined by:



(a) Effect of normalisation factor 1 (φ_1).



(b) Effect of normalisation factor 2 (φ_2).

Figure 4.7 Illustration of the effect on the testing data by normalisation factor 1 (φ_1) and normalisation factor 2 (φ_2) under different values of α.

$$\begin{aligned} \mu_{target} &= \mu_{test} + \alpha(\mu_{train} - \mu_{test}) \\ \sigma^2_{target} &= \sigma^2_{test} + \alpha(\sigma^2_{train} - \sigma^2_{test}) \end{aligned} \tag{4.10}$$

Given these target distribution scores, the normalisation factor 2 (φ_2) is obtained by Equation 4.11:

$$\varphi_2 = \frac{(SPS_{testi} - \mu_{test})\sigma^2_{target}}{\sigma^2_{test}} + \mu_{target} \tag{4.11}$$

where φ_2 is the normalisation factor 2 for the ith sample, SPS_{test_i} is the SPS_{song} for the ith test sample. Like in normalisation factor 1, when α is equal to zero, no perturbation is applied. However, unlike normalisation factor 1, when α is equal to 1, the training distribution becomes the target distribution. Figure 4.7b illustrates how the testing data distribution changes when using the normalisation factor 2, under incremented values of α. In this scenario, the testing data distribution changes from μ_{test} and σ_{test} when α is equal to zero, to μ_{train} and σ_{train} when α is equal to 1.

4.5 Experiments

In order to establish the significance of the results, experiments were repeated multiple times. In particular, acoustic model training depends on the *random* initialisation of parameters, the *random* presentation order of training data, etc. Each training, though equally valid, can produce WER results that vary appreciably, and the variation can be mistaken for genuine performance variations. Therefore, all experiments systems were trained eleven times allowing the evaluation of the model performances to be treated statistically. The number of trainings was selected taking into account training time constraints.

First, the baseline system training was replicated eleven times to calculate confidence intervals by using the mean and the standard error of the mean scores. After each training, the development set was used to select the LM-weight and words-insertion-penalty Kaldi's decoding parameters, and these parameters were then used to decode the evaluation set. These parameters were firstly estimated when decoding with the 3-gram LM and then when using the 4-gram LM for re-scoring.

Following the above procedure, experiments expanding the baseline MFCC + i-Vector feature vector using different voice source based feature combinations were performed. The first experiment, *Kaldi LN*, evaluated the effect of including both Kaldi pitch representations: the log pitch and the normalised log pitch. The second, *Kaldi L*, evaluated the effect of using the log pitch, without the normalised log pitch. The third, *Kaldi N*, evaluated the effect of utilising only the normalised log pitch. For the different pitch experiments, both the delta pitch and POV were included. Then, experiments to evaluate voice quality (*VQ*) features were performed by expanding the best combination of MFCC plus pitch features – obtained from the results when training with the smallest DSing1 – with the four VQ features. Next, experiments evaluating the effect of the beat features extracted from the singing (*Beat*) were performed by expanding the Kaldi LN + VQ feature vector further with the beat information described above.

Finally, the effectiveness of normalising the testing data using the SPS parameter was evaluated by perturbing the testing data and decoding utilising one of the eleven baseline systems, randomly selected. This process assumes that the testing data's transcription is known, which can only be true after an initial decoding step. However, for evaluation purposes, the gold standard transcription was utilised. Both factors were evaluated using α valued from 0.0 to 1.0 with 0.1 steps.

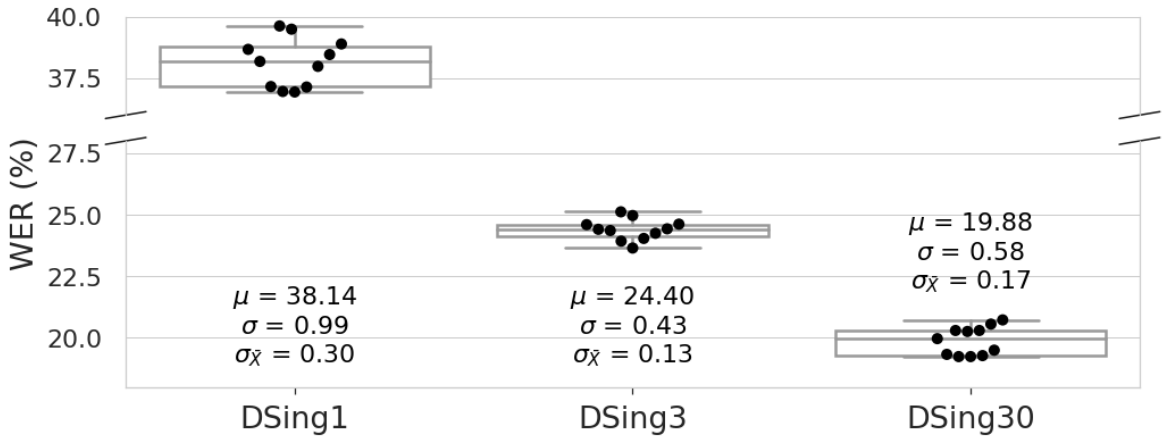


Figure 4.8 Box and whisker plots from eleven trains of the baseline system, detailed by the training set. The dots represent the results from individual experiments.

4.6 Results and Analysis

In an initial experiment, the acoustic model was trained on the *clean-100* LibriSpeech training acoustic material (100h of annotated audiobooks) (Panayotov et al., 2015) to test the performance of the recogniser when trained on well-labelled but out-of-domain spoken speech data. This mismatched baseline was compared with results achieved when training on the three karaoke DSing training datasets described previously, i.e., DSing1, DSing3 and DSing30. For each system, performance was measured using the gold standard development and evaluation sets described in Section 4.2.5.

Evaluation using the out-of-domain model resulted in 65.27% WER when evaluating the TDNN-F model with 3-gram LM and 64.81 when rescoring using the 4-gram LM. Figure 4.8 presents box and whisker plots with the 4-gram performances from the eleven runs of the baseline system trained with the in-domain datasets. The figure includes the mean (μ), standard deviation (σ), and standard error of the mean ($\sigma_{\bar{X}}$) WER scores for each training set. Training with in-domain datasets reported a significant improvement, resulting in a better performance of 38.14 ± 0.58, 24.40 ± 0.26 and 19.88 ± 0.34 WER for DSing1, DSing3 and DSing30, respectively.

The "Welch's t-test" statistic test was employed to measure the statistical significance of the improvements obtained from each experiment. Welsh's t-test is an independent two-sample t-test assuming unequal variance population. The test was employed to evaluate the significance of one system's mean being greater than another system's mean, i.e, one-tailed test.

Table 4.5 present the system performances (WER) along with the 95% confidence intervals for the baseline system and the experiments utilising the various combinations of

Table 4.5 WER (95% confidence interval) for experiments on systems trained using the DSing1 training set. P-values reported show the significance of the results of the experiment in relation to the baseline results. Values in bold represent the best performance for each training dataset.

Train Set	Experiment	3-gram	4-gram
DSing1	Baseline (B)	43.02 ± 0.55	38.14 ± 0.58
	B + Kaldi L (L)	$41.79 \pm 0.58 (p < .05)$	$37.30 \pm 0.52 (p < .05)$
	B + Kaldi N (N)	$41.67 \pm 0.37 (p < .05)$	$37.51 \pm 0.87 (p > .1)$
	B + Kaldi LN (LN)	$41.22 \pm 0.49 (p < .05)$	$36.81 \pm 0.45 (p < .05)$
	LN + Voice Quality (VQ)	$41.17 \pm 0.30 (p < .05)$	$\mathbf{36.70 \pm 0.46} (p < .05)$
	LN + VQ + Beat	$41.69 \pm 0.26 (p < .05)$	$37.17 \pm 0.22 (p < .05)$
DSing3	Baseline (B)	28.13 ± 0.14	24.40 ± 0.26
	B + Kaldi L (L)	$27.80 \pm 0.34 (p = .05)$	$24.35 \pm 0.27 (p > .1)$
	B + Kaldi N (N)	$27.79 \pm 0.27 (p < .05)$	$24.28 \pm 0.19 (p > .1)$
	B + Kaldi LN (LN)	$28.05 \pm 0.24 (p > .1)$	$24.27 \pm 0.21 (p > .1)$
	LN + Voice Quality (VQ)	$27.82 \pm 0.26 (p < .05)$	$\mathbf{23.76 \pm 0.27} (p < .05)$
	LN + VQ + Beat	$27.99 \pm 0.20 (p > .1)$	$24.33 \pm 0.32 (p > .1)$
DSing30	Baseline (B)	22.82 ± 0.21	19.88 ± 0.34
	B + Kaldi L (L)	$22.92 \pm 0.33 (p > .1)$	$19.72 \pm 0.25 (p > .1)$
	B + Kaldi N (N)	$22.95 \pm 0.15 (p > .1)$	$19.67 \pm 0.11 (p > .1)$
	B + Kaldi LN (LN)	$23.23 \pm 0.28 (p < .05)$	$19.87 \pm 0.12 (p > .1)$
	LN + Voice Quality (VQ)	$22.97 \pm 0.32 (p > 0.1)$	$\mathbf{19.60 \pm 0.21} (p > .05)$
	LN + VQ + Beat	$22.87 \pm 0.32 (p > .1)$	$19.71 \pm 0.15 (p > .1)$

pitch, voice quality and beat features for systems trained using the DSing1, DSing3, and DSing30 training set. The results correspond to performances obtained from the TDNN-F AM using both the 3-gram or 4-gram LMSmule+ LM. Significance of the experiments are reported in relation to the baseline results.

A significant improvement was obtained when training using the on-domain DSing1 training set compared with the system trained using the out-of-domain LibriSpeech data. The evaluation WER obtained using the 3-gram LM was 43.02%. This error is in the range of the performance reported for karaoke data by Kawai et al. (2017), without the need for any speech-to-singing adaptation technique. When re-scored using the 4-gram LM,

the performance increased, obtaining a 38.14% WER. Using the four Kaldi pitch features ('B + Kaldi LN' experiment) reduces the error by about 2.0% ($p < .05$) 3-gram and 1.4% ($p < .05$) 4-gram. A similar improvement is obtained by expanding Kaldi LN with the VQ features ($p < .05$). However, no significance error reduction was obtained from 'LN + VQ' experiment compared with the 'B + Kaldi LN' ($p > .1$).

Training the system using the DSing3 training set obtained a significant error reduction, decreasing the WER from 43.02% to 28.13% when using 3-gram LM. Rescoring with the 4-gram LM, the WER decreased from 38.14% to 24.4%. Even though the DSing3 introduced accent variability with recordings from Australia and the USA, increasing the training set from 15.4 hours to 44.7 hours has proved beneficial to improve the system's performance. As discussed in Chapter 2, native English singers tend to neutralise their accent when singing (Gibson, 2010) and there is a tendency to move towards US pronunciation (Konert-Panek, 2017). Pitch features alone did not produce a significant increment of performance. However, a reduction of 0.5% ($p < .05$) for 3-gram and 0.7% ($p < .05$) 4-gram was obtained when combining Kaldi LN with the VQ features. Notice that for 4-gram LM, a new significant improvement is obtained from expanding the 'B + Kaldi LN' feature vector with the VQ features ($p < .05$)

Training using the largest DSing30 led to a further increment in performance, decreasing the WER from 28.13% to 22.82% and from 24.40% to 19.88% for 3-gram and 4-gram LM, respectively. This increment in performance was achieved even though the 100 hours of singing from non-native English speakers introduced a considerable variability of pronunciations. As also discussed in Chapter 2, this might be possible by the tendency of non-native English speakers to neutralise their accent during singing (Hagen et al., 2011; Mageau, 2016). Neither pitch nor voice quality features helped to improve the models trained on the largest DSing30. However, a significant improvement is obtained when using the VQ to expand the 'B + Kaldi LN' ($p < .05$). A more detailed analysis of the benefit of using VQ features will be presented below.

The beat features did not significantly improve the performance over using the pitch and voice quality features for any training set size.

Note that the best results of 19.6% WER obtained in the 'LN + VQ' experiment trained on the DSing30 dataset represents a reduction of about 50% of error compared with the best results of 37% WER reported by Kawai et al. (2017) (Table 3.2).

Word error rates higher than 80% have been proved sufficient for music information retrieval applications such as keyword spotting systems (Kruspe, 2016a) and query-by-singing (Mesaros and Virtanen, 2010b). However, the recognition performance of 19.6% WER seems favourable for applications such as lyrics-to-audio alignment, where higher

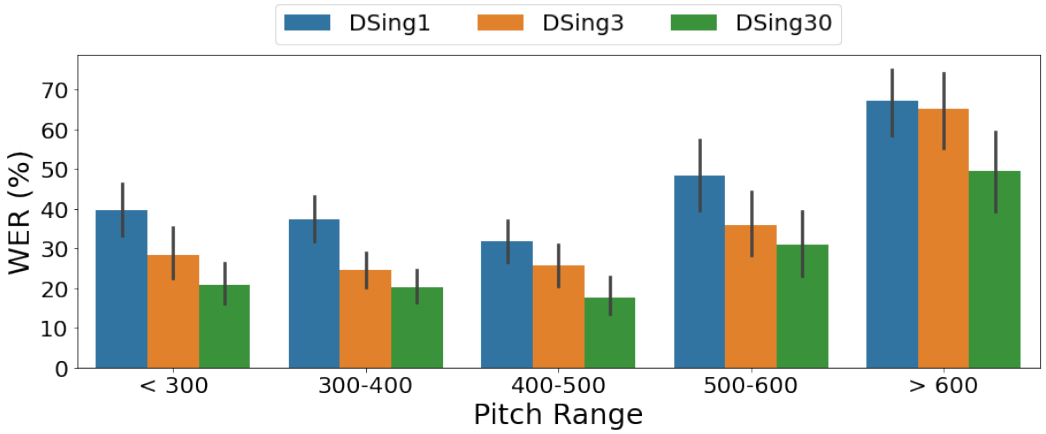


Figure 4.9 WER scores from the DSing development set using the 'Baseline' for DSing1, DSing3 and DSing30. Utterances are grouped into one of five pitch ranges by using the average pitch from the utterance.

recognition performances are required. However, recognition errors of 19.6% remain higher than state-of-the-art ASR performances in spoken tasks, e.g., 1.8% (Hsu et al., 2021) for the LibriSpeech dataset (Panayotov et al., 2015), 5.6% (Zhou et al., 2020) for the TED-LIUM dataset (Rousseau et al., 2012) and 2.9% (Hadian et al., 2018) for the WSJ dataset(Paul and Baker, 1992).

4.6.1 Analysis of WER

Figure 4.9 shows a bar plot with the WER scores from the development set using the baseline system. The height of the bars represent the mean WER, and the error bars represent the 95% confidence interval that provides some uncertainty around the mean. The utterances were grouped by the average pitch of the utterance (utt), constructing five groups: lower than 300 Hz (99 utt), between 300 Hz and 400 Hz (150 utt), between 400 Hz and 500 Hz (124 utt), between 500 Hz and 600 Hz (64 utt) and greater than 600 Hz (45 utt). As discussed before, high pitch singers shift their first formant to match the pitch frequency, increasing the energy in that area. However, this action (formant tuning) distorts the sound of speech, affecting the intelligibility. This reduction in intelligibility also affects the performance of speech recognition systems. As is shown in this figure, utterances with an average pitch lower than 500 Hz have similar performances. However, the performance is drastically reduced for utterances with pitches higher than 500 Hz. Note that the use of the largest DSing30 consistently introduced an improvement of about 15%-20% for all the pitch ranges.

Figure 4.10 shows the effect of the musically-motivated features for systems trained on DSing30 and grouped by pitch range using the same ranges as Figure 4.9. Like before, high

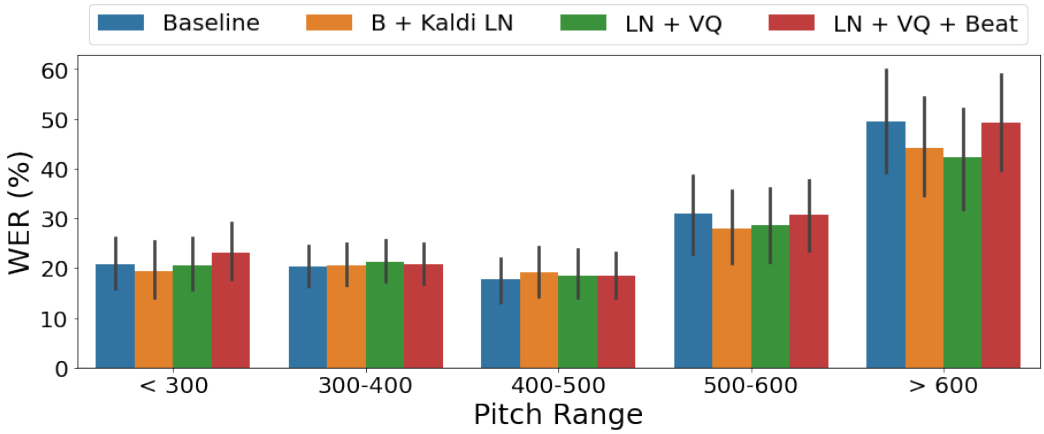


Figure 4.10 WER scores from the DSing development set for the 'Baseline', 'B + Kaldi LN', 'LN + VQ' and 'LN + VQ + Beat' systems trained on DSing30. Utterances are grouped into one of five pitch ranges by using the average pitch from the utterance.

pitch singing resulted in higher WER scores for all the experiments. As seen in this figure, pitch and VQ features seem to have a higher effect on high pitches singing than on lower pitches, especially for pitches higher than 600 Hz. Note that the matched pair test did not show significant improvement from results from the 'LN + VQ' system over results from the 'B + Kaldi LN' system. However, 80% of the utterances with pitches higher than 600 Hz obtained equal or lower error rates when using VQ features.

There is a clear benefit of using musically-motivated features, especially for high-pitched singing scenarios. It is expected that these features may also help to increase the distance between voiced and unvoiced pairs of consonants. Figure 4.11 shows a t-SNE plot of the [s]-[z] fricative and [p]-[b] plosive pairs of voiced and unvoiced sounds. The plot was constructed using the posterior probabilities from one model trained with DSing30. The subscript 0 represents the baseline model, subscript 1 Kaldi LN and subscript 2 Kaldi LN + VQ. In the baseline model there are separate clusters for the voiced and unvoiced fricatives and plosive pairs, but the clusters overlap. When the model is informed with pitch features and VQ information, the distance between classes becomes wider and the classes are more compressed. Best discrimination is seen when the VQ features are included, giving some hint that they may be useful even though improvements in WER were not significant.

4.6.2 Effect of Employing Syllables per Second Normalisation

Both speech normalisation factors were evaluated only utilising a system trained on the largest DSing30. The distribution of the training data is defined by $\mu_{train} = 2.68$ and $\sigma_{train} = 1.08$ (Appendix B). The development data distribution is defined by $\mu_{test} = 2.85$ and $\sigma_{test} = 1.00$.

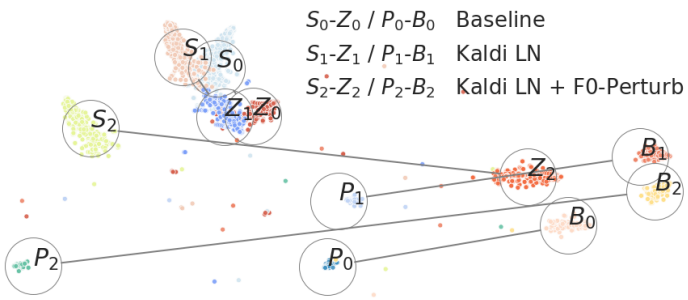


Figure 4.11 T-SNE constructed with the posterior probabilities from model trained on DSing30.

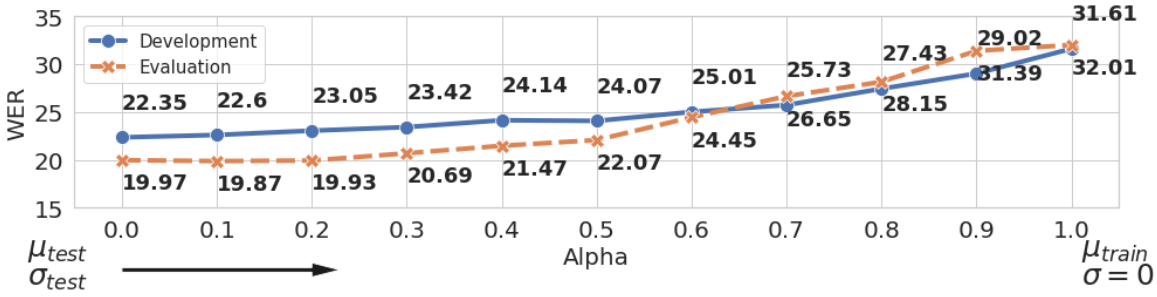


Figure 4.12 WER results after applying different α values for SPS normalising factor 1.

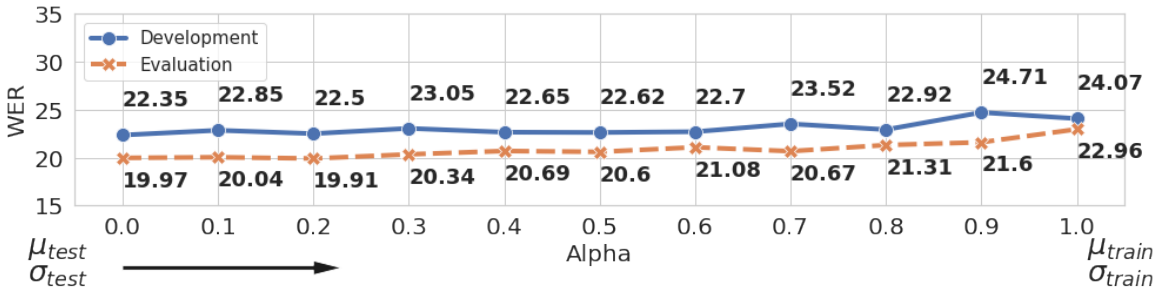


Figure 4.13 WER results after applying different α values for SPS normalising factor 2.

Figure 4.12 shows the results obtained after employing the normalisation factor 1 defined in Equation 4.9, with α values from zero to one and one-tenth step. When α starts to increase (starting from zero), the development performance starts to degrade rapidly, reaching the worst performance of 32.01% WER for development and 31.61% for evaluation. The poor performance obtained can be explained by observing the distribution of the development set SPS. As shown in Figure 4.14, the distribution of the development set does not follow a single Gaussian distribution, presenting two peaks at about 2.4 and 3.6, and some samples

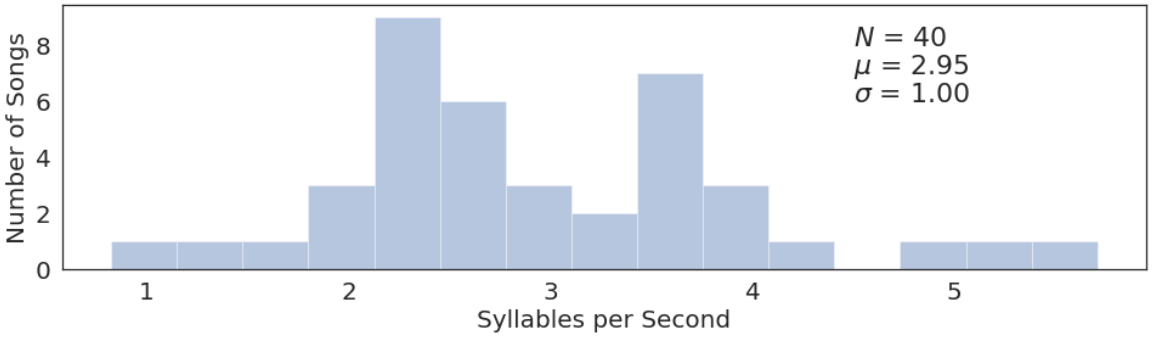


Figure 4.14 Histogram of the averages song syllables per second distribution from the DSing development set.

with very high SPS of about five syllables per second in contrast with the 2.68 for the training set.

Figure 4.13 shows the performance obtained when decoding the testing data normalised with normalisation factor 2. Slight performance degradation was obtained when this normalisation technique was employed. The development set performance varies from 22.35% WER when α equals zero to 24.07% WER when α is equal to one. This small degradation may indicate that the testing data SPS distribution was close enough to the training set distribution not to add any benefits.

4.7 Summary and Conclusion

This chapter focused on the study of acoustic modelling for unaccompanied sung speech. Differences in speech production, such as vowel duration, pitch range, pitch variation within a vowel, vowel space area and energy, act to make sung speech less intelligible than spoken speech. The task is made even more challenging by the lack of readily available unaccompanied sung speech data suitable for unaccompanied sung speech recognition study.

First, the chapter presented the novel DSing dataset, an annotated unaccompanied sung speech corpus constructed by processing the karaoke performances DAMP-MVP corpus sourced and released by Smule. The DSing corpus defines three training datasets: DSing1 constructed with English performances recorded in Great Britain; DSing3, which extends DSing1 with English recordings from Australia and the US; and DSing30 that extends DSing1 further with English performances from singers from non-English speaking countries. Testing data was constructed from Great Britain performances, using human annotators to correct the alignment timings and re-transcribe the speech. This ensures that accurate

recognition performances can be measured. The DSing dataset is the result of addressing the lack of a readily available dataset for sung speech ASR (**RQ.3**).

Next, a Kaldi-based benchmark system using a state-of-the-art TDNN-F acoustic model trained with LF-MMI was constructed using the DSing corpus. A 3-gram and 4-gram maximum entropy language model was built using an in-domain corpus composed of 1.7 million lines of lyrics text. This system has produced WERs of 43.02%, 28.13% and 22.82% when trained with DSing1, DSing3 and DSing30, respectively, using a 3-gram LM. When rescoring using the 4-gram LM, WERs fall to 38.14%, 24.40% and 19.88%, respectively.

Then, this chapter investigated the benefit of informing acoustic models with the musical properties of the singing (**RQ.4**). For this, musically-motivated features like pitch-based, voice source quality and beat features were evaluated. Using pitch and voice source quality features improved ASR performances, reducing WERs to 36.70%, 23.76% and 19.60% using the 4-gram LM for DSing1, DSing3 and DSing30, respectively, on an existing state-of-the-art baseline. However, this effect only improves performances for ASR systems trained with the smaller training sets. When using the more extensive DSing30 (149.1 hrs), the voice source-based features were, surprisingly, found to provide no significant benefit for ASR performance. This suggests that, with enough training data sets, systems are able to learn the phonetic cues being carried in the voice source in a less direct manner (e.g., via the temporal dynamics of the MFCC features). Beat features did not improve the performance for any training set size.

Finally, syllable-per-seconds normalisation experiments, to normalise the test set speech rate, were performed. Two normalisation factors were evaluated, but no improvement was obtained. This may be a result of the testing distribution being very close to the target distribution.

In this chapter we dealt with difference challenges faced by unaccompanied sung speech ASR systems. The next chapter will deal with the challenge of separating the singing from the background accompaniment. We will then return to the topic of acoustic modelling in Chapter 6 which will consider the architectures for combining the source separation front-end and the speech recognition back-end.

Chapter 5

Singing and Background Accompaniment Separation

5.1 Introduction

In the previous chapter, we dealt with recognising the lyrics from an unaccompanied sung speech signal. First, using the DAMP-MVP karaoke corpus released by Smule (Smule, Inc., 2018), the DSing dataset was designed. Three training datasets were constructed, starting with English recordings from Great Britain (DSing1), then adding English recordings from Australia and the US (DSing3), and finally adding singers from non-English speaking countries (DSing30). The DSing dataset was the result of addressing the lack of sung speech data for recognition (**RQ.3**). Next, it was explored how state-of-the-art systems designed for spoken speech perform on sung speech data and to what extent features motivated from the musical components of the singing can help improve recognition performances (**RQ.4**). Using the Kaldi toolkit (Povey et al., 2011), it was evaluated a state-of-the-art TDNN-F acoustic model architecture trained on the new DSing sung speech dataset. Also, it was built 3-gram and 4-gram maximum entropy language models trained on an in-domain lyrics corpus. The best performances were obtained using the 4-gram language model and musically-motivated features to inform acoustic modelling achieving 36.70%, 23.76% and 19.60% WER, for DSing1, DSing3 and DSing30, respectively.

However, in practical applications, singing is often accompanied by instrumental accompaniment, which makes recognising the sung speech even more challenging. This problem can be alleviated by separating the singing from the background before passing it to the speech recognition system. This chapter will deal with the challenge of separating the sung

speech from the instrumental background accompaniment while preserving the phonetic cues required for recognition of the lyrics, addressing **RQ.5**.

Sung speech separation is itself a particularly challenging task. In many spoken speech scenarios, e.g., recordings in a crowded street or multi-conversations in a cocktail party, the sources are independent, and they are uncorrelated in behaviour. In contrast, in music, the sung speech and instrumental accompaniment are parts of the same performance and, to some extent, singers try to blend their voice with the accompaniment. Deutsch (2013) suggested that Western music favours the twelve-tone equal temperament scale. Additionally, they suggest that pitches from different sources have a relationship of integer ratios (e.g., octaves, fifths, etc.). This relationship results in many harmonics from the singing overlapping with harmonics from the accompaniment. Moreover, the sung speech and accompaniment are often strongly correlated in onset times, i.e., they have synchronised note attacks and pitch changes. The result is that the sung speech and accompaniment share a lot of common properties; for example, they may have shared rhythm and melody. Therefore, time-frequency overlap is a significant issue in music separation. Further, many of the cues that spoken speech source separation techniques exploit, like differences in frequency and amplitude modulation, or harmonic filtering, are weakened.

The problem of musical sung speech separation has some obvious similarities to the speech source separation that has been widely studied in the context of distant microphone speech recognition. However, in distant microphone ASR the sources can be distinguished and separated by location, i.e., using microphone array processing such as beamforming (McCowan, 2001). This generally will not be an option in a digitally mixed recording, or even in a live recorded performance given that microphone arrays are not commonly used for recording.

The present chapter starts by proposing ways to use the novel Smule DAMP vocal separation dataset to train vocal source separation models in Section 5.2. Then, Section 5.3 presents a survey of different state-of-the-art vocal separation models and constructs a baseline using the proposed dataset. Next, Section 5.4 investigates the use of the STOI intelligibility loss function to improve the performance of the vocal separation models. Then, Section 5.5 employs speaker embedding ideas to investigate the effect of informing the separation models with an instrumental background embedding. Finally, Section 5.6 summarises the chapter.

5.2 Construction of the Corpus

Systems for robust speech separation (i.e., the separation of a speech signal from a noisy mixture) and speaker separation (i.e., separation of a speech signal from a mixture of two or more overlapping speakers) need to be trained on a large dataset with ground truth, i.e., datasets where the sources are provided in separated audio tracks. However, generating these kinds of corpora is very costly and time-consuming. For these reason, models tend to be trained using a small amount of real data, if any, and additional simulated data generated by mixing 'clean' speech with speech-free background recordings. For example, the training dataset for the CHiME 4 challenge (Vincent et al., 2017) is composed of 1600 real speech recordings (four speakers) in noisy environments ('cafe', 'street', 'on the bus' and 'pedestrian area') plus 7138 simulated recordings generated by using speech recordings from the World Street Journal dataset (Paul and Baker, 1992) and speech-free recordings from the same four environments used in the real data.

The use of simulated training data is not without its problems. Sentences can be randomly mixed into arbitrary backgrounds under the assumption that the speech and background are independent. However, even for spoken speech, this assumption is only an approximation. For example, it does not take into account the Lombard effect and other active speech adaptation where speakers tailor their speech to fit the acoustic environment and communicative setting. In contrast, sung speech separation models may require a greater amount of real data for training than for spoken models. This is because simulated musical mixtures where clean singing is mixed with a randomly selected instrumental accompaniment might not properly characterise real mixtures, affecting the generalisation of sung speech separation models.

Nowadays, several datasets are freely available for music source separation research, i.e., datasets which provide the separated sources and a mixture of them (Bittner et al., 2014, 2016; Hsu and Jang, 2010; Liutkus et al., 2014, 2017; Rafii et al., 2017; Smule, Inc., 2019). However, music source separation datasets tend to be relatively small and lack diversity, making most of them unsuitable for the supervised training of source separation models. Table 5.1 presents a list of the most common music source separation datasets available. The table includes the year when the dataset was released, the number of tracks, the total duration and an indication of whether the dataset contains full-length songs or song excerpts. MUSDB18 (Rafii et al., 2017) is currently the main dataset used for music source separation investigation since it is the corpus used in the SigSeg challenge (Stöter et al., 2018). Many models trained on MUSDB18 perform relatively well. However, several of the best systems trained tend to use additional private data to improve the system performances. This may be an indication that the size of the MUSDB18 dataset is insufficient for training robust models. For example, Défossez et al. (2019) reported an increment of 0.5 dB SDR (i.e. from 6.8 dB

Table 5.1 List of music source separation datasets. All datasets include a vocal source as one of the sources. Note that the number of tracks reported for MedleyDB excluded tracks that do not contain a vocal as one of the sources (i.e., instrumental songs). Note that the MUSDB18 dataset combines tracks from MedleyDB and DSD100.

Dataset	Year	Tracks	Duration (hrs)	Full song
MIR-1K (Hsu and Jang, 2010)	2010	1000	2.2	✗
ccMixter (Liutkus et al., 2014)	2014	50	3.2	✓
MedleyDB (Bittner et al., 2014, 2016)	2014	81	4.9	✓
DSD100 (Liutkus et al., 2017)	2015	100	7.0	✓
MUSDB18 (Rafii et al., 2017)	2017	150	9.8	✓
DAMP-VSEP (Smule, Inc., 2019)	2019	41,749	347.9	✗

to 7.3 dB) when augmenting the 100 songs from the MUSDB18 training set with 150 extra songs.

In 2019, Smule released the DAMP vocal separation dataset (DAMP-VSEP) (Smule, Inc., 2019), a large dataset for the vocal separation task generated by collecting several karaoke performances from the Smule karaoke application. This dataset conveniently provides both the vocal and the background recordings in separated audio files. However, this dataset presents several challenges (e.g., misalignment between the vocal and background sources) that must be addressed before it can be used for developing and evaluating source separation approaches.

This section presents the construction of a dataset for audio source separation experiments based on the 'DAMP-Vocal Separation dataset' (DAMP-VSEP) (Smule, Inc., 2019). DAMP-VSEP is a corpus originated and released by Smule in October 2019 for vocal separation research. It is the most extensive real-world karaoke dataset currently available for vocal separation research. It offers various singing styles and proficiency levels, matching those from the DAMP-MVP corpus used to construct the DSing dataset, i.e., the dataset used to train the acoustic models described in Chapter 4.

5.2.1 Description of the DAMP-VSEP dataset

DAMP-VSEP contains 41000 30-second length audio segments for singing separation research sourced from the Smule karaoke application. It includes performances from 6456 singers from 155 countries, covering 11,494 song arrangements and 36 different languages. DAMP-VSEP provides three tracks for each song segment, corresponding to the vocal performance, $\mathbf{v}$, the background, $\mathbf{b}$, and a mixture of the two, $\mathbf{mix_{damp}}$. About half of the song segments in DAMP-VSEP correspond to duet ensembles. In these cases, two separated

vocal tracks, one for each singer in the duet, are provided. The mixture for solo and duets ensembles at a sample **t** is expressed as:

$$\mathbf{mix_{damp}}(t) = \sum_{s} \left[\mathfrak{A}(\mathbf{v_s})\right](t) + \left[\mathfrak{A}(\mathbf{b})\right](t) \tag{5.1}$$

where $\mathfrak{A}$ denotes the possible use of non-linear operations, and $\mathbf{v_s}$ denotes the vocal performance for the singer $s \in \{1,2\}$.

The metadata information per performance includes the country where the singer was located during the recording, the language of the song, the type of ensemble ('solo' or 'duet'), and the sample rate of each track (44100 Hz for background and one of 22050 Hz, 44100 Hz and 48000 Hz for the vocal segments). In DAMP-VSEP, different arrangements from the same song may exist, differing mainly in the characteristics of the background provided. For example, some backgrounds may correspond to the original song accompaniment, while others may be generated by utilising a piano or guitar.

5.2.2 Challenges with the DAMP-VSEP dataset

The DAMP-VSEP dataset is a corpus collected for vocal separation research. However, as mentioned above, the dataset presents a number of challenges that must be addressed before it is to be used to evaluate source separation approaches fairly:

1. Like DAMP-MVP, there is no control over how the recordings were made. The singers may not be wearing headphones, or they may be located in noisy locations during the recordings, causing some vocal recordings to contain noise from the surroundings (also known as "bleeding").

2. Background recordings may not be purely instrumental, i.e., they may contain some sung speech. In these cases, the singing mainly corresponds to the singer's voice from the original composition, presumably included to help Smule users to perform better when using the application.

3. All the 30-second segments correspond to extracts obtained from seconds 60 to 90 from the karaoke performances; therefore, there is no guarantee about the amount of singing contained in the segment, or even that the segment contains any singing at all..

4. It is observed that the vocal segments and the background are not always aligned in time correctly. Some small offsets can be explained as the effect of the lag that different Android devices have. However, there are segments with offsets of up to 15 seconds, which may have resulted from errors during the data collection by Smule.

5. The volume level of the independent sources may not match the balance used in the provided mixtures, and the SNR is unknown.

6. The provided mixtures, $\mathbf{mix_{damp}}$, contain one of a number of proprietary studio effects (non-linear operation $\mathfrak{A}$), such as 'Magic', 'Super Harmony', 'Super Pop' and 'SF Opera'. Singers add these effects after finishing their recordings. The separated tracks do not include these effects and the dataset does not provide information on how to replicate them. Therefore:

$$\mathbf{mix_{damp}}(t) \neq \mathbf{v}(t) + \mathbf{b}(t) \tag{5.2}$$

7. The background source is provided for all performances, labelled using the performances ID, which means that for two or more performances of the same arrangement copies of the same background are provided using different IDs. This makes it difficult to identify the performances that use the same background.

8. The dataset contains some duplicated data: in the generation of duet ensembles, one singer may perform one part of the song, and other singers may perform the complementary part asynchronously. This results in several duets performances where the same singer, the second singer $\mathbf{v_2}$, is paired with many different first singers, $\mathbf{v_1}$. Several copies of the second singer track are provided for each duet performance to which that second singer is paired.

These challenges need to be carefully considered. Indeed, Wang et al. (2021) argue that the existence of bleeding of music in some of the vocal tracks (*challenge 1*) and bleeding of singing in some of the background tracks (*challenge 2*) make the DAMP-VSEP dataset unsuitable for supervised source separation training. Moreover, they reported a reduction in the separation performances when utilising the unprocessed DAMP-VSEP dataset to augment the MUSDB18 training set. However, it will be seen that if appropriate processing steps that deal with the abovementioned challenges are taken, the data can be made suitable for supervised vocal separation training.

Noisy and silent vocal tracks (*challenge 1* and *challenge 3*) can be easily detected by employing a voice activity detection (VAD) technique. If no silence fragments are detected in the vocal track (i.e., no singing stop in 30 seconds), the track can be regarded as noisy (e.g., the background accompaniment or ambient noise is bleeding in the vocal track). On the other hand, if no singing fragments are detected, the track can be considered silent. Noisy and silent vocal tracks can then be filtered out. A VAD technique might also be employed to detect the background tracks containing bleeding of singing (*challenge 2*). However, this

scenario is much more complicated to deal with than the previous two as it requires detecting singing over background accompaniments.

The alignment between the vocal and background tracks can be corrected by obtaining the lag where the cross-correlation between the vocal and the mixture maximises. This is possible because no misalignment exist between the mixtures and the background tracks. Therefore, any lag between the vocal and mixture tracks will be equal to the lag between the vocal and background.

Due to the lack of information about the SNR utilised to generate the provided mixtures (*challenge 5*), no actions were taken when using these mixtures as model input, i.e., the SNR of the separated sources will not match the SNR of the mixture when training using the provided mixtures. However, this is only an issue when using the provided mixtures and not when the mixtures are generated by linearly remixing the sources.

It is not possible to replicate the proprietary studio effects, $\mathfrak{A}$, included in the provided mixtures (*challenge 6*). This creates a mismatch between the provided mixtures, $\mathbf{mix_{damp}}(t)$, and the separated sources. In the following sections, experiments that attempt to deal with such difficulties will be presented.

The different copies of backgrounds, $\mathbf{b}$, (*challenge 7*) and second singers, $\mathbf{v_2}$, in duets (*challenge 8*) can be identified using their unique MD5 checksum. This step was sufficient to detect all the copies for the second singer in duets. However, an inspection of different arrangements for the same song showed that two or more different backgrounds, i.e., having different MD5 checksum, are, in fact, slightly different versions of the same background. This can happen when different users create a new arrangement for the same song utilising the same background but with different leading silence (time-shifted) or different volume level. These perceptually similar backgrounds can be detected by computing their cross-correlation score and grouping all backgrounds with a score higher than a certain threshold.

5.2.3 Defining Training and Test Set

This section describes steps to process and construct three training sets, the *English*, *English+Duets*, and *English+non-English*, and two testing sets for supervised vocal separation training based on the DAMP-VSEP corpus. The first training set, the 'English' set, is constructed by using the English solo-ensemble performances. The second training set, the 'English+Duets' subset, augments the 'English' set by using the English duets-ensemble performances. The third training set, the 'English+non-English' subset, augments the 'English' set by using all non-English solo-ensemble performances. Note that the 'English+non-English' set does not include the English duets-ensembles. Validation and evaluation sets are selected from the English solo-ensemble performances.

The first training set was constructed using the English solo-ensemble performances, which corresponds to 4607 recordings. As aforementioned, different recordings in the DAMP-VSEP corpus may correspond to different performances of the same arrangement. This means that a copy of the background is provided, using the performance id, for each performance, making it challenging to identify the performances that use the same background. Detecting the unique backgrounds is an important step to reduce the risk of assigning the same background to the train and test sets. For this, backgrounds were first grouped using their MD5 checksum. This process identified 1890 distinct backgrounds from the 4607 English solo-ensemble recordings. Then, "perceptually similar" backgrounds were detected using the cross-correlation score. All backgrounds with a cross-correlation score greater or equal to 0.9831 were grouped together. The threshold was selected empirically by generating the clusters using several threshold values and selecting the one where the number of generated clusters varies by more than five clusters. This process resulted in 1261 clusters of perceptually similar backgrounds. From the 1261 clusters, two hundred clusters with a single element (i.e., clusters conformed by a single background that does not correlate with another) were taken aside to construct the validation and evaluation sets. This process resulted in 4407 performances for the English training set, 100 for the 'validation' and 100 for the 'evaluation'. After manual analysis, eleven performances in the evaluation set that included singing in the background source – typically singing from the original singer – were discarded. Mixtures from the discarded performance contain two overlapping singers, a challenge for which the separation models will not be trained.

The second training set, the 'English+Duets' set, augmented the English set using all English duet ensembles performances. The duets were included by splitting the performances into two single ensembles, ensuring that no repetitions of the second singers were included. That is to say, duet songs composed of a background, $\mathbf{b}$, first vocal, $\mathbf{v_1}$, and second vocal, $\mathbf{v_2}$, recordings, are included as two single performances constructed as $\left[\mathbf{v_1}+\mathbf{b}\right]$ and $\left[\mathbf{v_2}+\mathbf{b}\right]$. There are 3839 English duets ensembles, but only 1006 unique second singers (i.e., the same second singer is paired with several first singers in asynchronous duet performances). This augmentation increases the 'English' training set from 4407 to 9252 performances.

The last training set, the 'English+non-English' set, augmented the 'English' set using all non-English single ensembles in DAMP-VSEP. Note that this set uses only the single ensembles performances, i.e., English duet ensembles are not included in this version. This augmentation increases the 'English' dataset from 4407 to 20,654 performances. Note that no 'similar backgrounds' analysis was necessary for any augmentations as they only increased the training set.

Table 5.2 Description of both training sets, the validation and the evaluation set.

Set	**Fragments**	**Hours**
English	17,510	14.6
English+Duets	29,882	24.9
English+non-English	79,808	66.5
Validation	264	0.42
Evaluation	258	0.37

After defining the training and testing sets, the performances were processed, in preparation for future experiments. As was mentioned before, the DAMP-VSEP's performances last 30 seconds and that they were extracted from seconds 60 to 90 of Smule's performances. From the 30-second performances, the first 10 seconds were taken aside for use in the background embedding experiments (see Section 5.5). The training and testing samples were generated from the last 20 seconds of the performances. This ensured that all experiments were trained using the same set of samples and that the embedding experiments used unseen background segments. It was assumed that from the 10 seconds excerpt, one could capture the characteristics of the song as it is very likely that all instruments will be active at some point. This assumption is supported by the fact that the DAMP-VSEP samples were extracted from seconds 60 to 90, which very likely corresponds to a verse or chorus. Additionally, leaving the rest 20 seconds for audio source separation training will provide enough data for robust training.

On the last 20 seconds of the performances, a voice activity detector (VAD) was used to detect the singing segments from the vocal recordings. For this, *Google's WebRTC voice activity detection*[1] (WebRTC VAD) system wrapped by the Py-WebRTCVAD[2] Python module was used. WebRTC VAD uses a GMM model, pretrained on several "noise" and "speech" classes, to compute the likelihood of a frame corresponding to a speech frame. Vocal tracks with no silence fragments were discarded as noisy recordings, i.e., they contain environmental noise or background accompaniment bleeding in the vocal track. Additionally, tracks with no speech fragments were also discarded as silent vocals. Training samples were generated using 3-second length windows with a shift of 1 second between (overlapping) windows from the speech segments for each training set. The final size of the training, validation and evaluation sets is presented in Table 5.2.

[1] https://webrtc.org/

[2] https://github.com/wiseman/py-webrtcvad

Table 5.3 Top four source separation ranking as of May 24, 2020. All models were trained using the MUSDB18 dataset and ranked by the average vocal SDR result.

Model	Year	Vocal SDR (*dB*)	# Params
Conv-TasNet (Luo and Mesgarani, 2019)	2019	6.81	5.1M
Meta-TasNet (Samuel et al., 2020)	2020	6.40	45.5M
Open-Unmix (Stöter et al., 2019)	2019	6.32	35.5M
DEMUCS (Défossez et al., 2019)	2019	6.84	1.5M

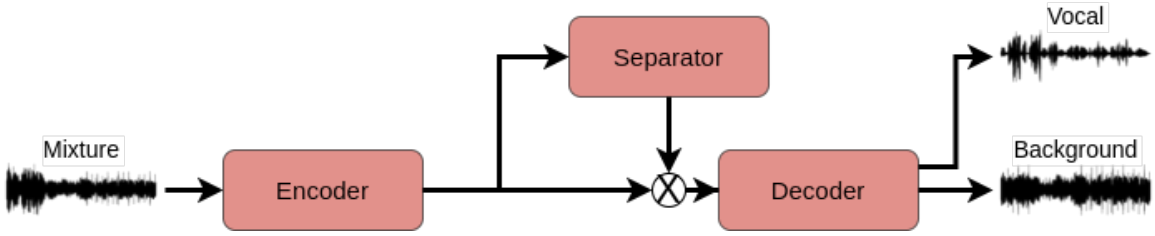


Figure 5.1 Block diagram of the TasNet architecture.

5.3 Music Audio Source Separation Baseline

The previous section described the procedure used to construct three datasets for supervised vocal separation training based on the DAMP-VSEP corpus. Chapter 3 defined audio source separation in music as the separation of existing audio sources into remixable elements. In this section, an audio source separation baseline system that separates the vocal from the background accompaniment is constructed. The system is based on the convolutional time-domain audio source separation network (Conv-TasNet) (Luo and Mesgarani, 2019) and trained using the 'English', 'English+Duets' and 'English+non-English' datasets.

The Conv-TasNet architecture was selected for the baseline construction from a ranking of several state-of-the-art music audio source separation approaches trained and evaluated on the MUSDB18 corpus (Rafii et al., 2017). The Conv-TasNet architecture is a fully convolutional architecture consisting of three modules, namely encoder, separator and decoder (Figure 5.1) (see Chapter 3.3 for details). The modular architecture of the Conv-TasNet architecture enables different experiments with minimal modifications to the system.

The models included in the ranking were obtained from the models reported in the *PapersWithCode* website[3] as of May 24, 2020, and ranked by the average vocal signal-to-distortion (SDR). Table 5.3 shows the ranking of the top four models. While the models like Meta-TasNet (Samuel et al., 2020), Open-Unmix (Stöter et al., 2019) and DEMUCS (Défossez et al., 2019) were originally designed for music source separation and audio

[3]https://paperswithcode.com/sota/music-source-separation-on-musdb18

sampled at 44.1 kHz, Conv-TasNet was designed for speech separation and audio sampled at 8 kHz. However, Défossez et al. (2019) adapted the Conv-TasNet for music separation. All performances reported in the ranking correspond to models trained using exclusively the MUSDB18 data, i.e., models trained using additional private training data were excluded from the ranking to keep the results comparable.

The baseline models were trained by generating the mixtures online by adding the vocal and background sources. For the largest 'English+non-English' dataset, additional experiments were performed by using the provided mixture, $\mathbf{mix_{damp}}$, to evaluate the models' performances in such extreme cases, i.e., experiments using a mixture including proprietary studio audio effects. The evaluations are measured in terms of the SDR, scale-invariant SDR (SI-SDR) and the short term objective intelligibility (STOI) and reported in terms of the improvement of the score.

5.3.1 Methodology

The Conv-TasNet baseline system employed in this section was implemented using the *Asteroid* framework (version 0.4.5) (Pariente et al., 2020), a PyTorch-based framework for audio source separation research. The use of Asteroid has several advantages:

- It provides well-tested implementations of several state-of-the-art source separation models, including the implementation of the Conv-TasNet model.
- Each experiment is organised in a recipe directory, a structure borrowed from the Kaldi toolkit (Povey et al., 2011).
- It provides data loaders for several common source separation datasets with a consistent structure and parameters.
- Pretrained models are shared using Hugging Face[4], an increasingly popular AI community[5].

Using the Asteroid framework also helps to increase the accessibility and reproducibility of this research as it facilitates the easy sharing of the experiment's recipe, the best-trained models and the data loaders for the processed versions of DAMP-VSEP.

Three models were trained by using the 'English', 'English+Duets' and 'English+non-English' sets and a remixing of the sources for model input, defined as:

[4]https://huggingface.co/

[5]The list of all models shared by Asteroid are found in https://huggingface.co/models?filter=asteroid

$$\mathbf{mix_{remix}}(t) = \mathbf{v}(t) + \mathbf{b}(t) \tag{5.3}$$

where $\mathbf{mix_{remix}}$ is the mixture, $\mathbf{v}$ is the target vocal source and $\mathbf{b}$ is the target background source.

Additionally, and exclusively for the largest 'English+non-English' training set, a model was trained using the provided mixture, $\mathbf{mix_{damp}}$, as input. This last model was trained to explore the complexities of the task presented by DAMP-VSEP, i.e., the mixture contains non-linear operations, $\mathfrak{A}$, not presented in the target sources. The provided mixture may be close to real music, where different electronic equalisation may be applied to the independent sources. The challenge presented in the DAMP-VSEP mixtures forces the separation models to learn how to filter out such non-linear operations. This can be regarded as a 'stressed scenario'. Therefore, this evaluation was not performed on the 'English' and 'English+Duets' training sets because the small size of the 'English' set makes it hard to generalise properly using the provided mixture. Second, the inclusion of the duets ensembles as two single performances in the 'English+Duets' set invalidates the use of this set for using the provided mixture as both singers may overlap (as illustrated in *utterance 4* in Figure 4.1 page 88).

In all models trained, the separator module consisted of 4 repetitions (R) of 10 (M) convolutional blocks. These parameters were chosen following the Défossez et al. (2019) adaptation of Conv-TasNet and confirmed after experimenting with different values for parameters R and M. The training used a batch size of 7 samples, selected to fit the GPU's memory limit. The objective function used was the mean absolute error criterion (MAE) using the Adam optimiser. Learning rates of 5×10^{-3}, 3×10^{-3}, 1×10^{-3}, 5×10^{-4}, 3×10^{-4}, 1×10^{-4}, and 5×10^{-5} were evaluated. In all cases, learning rates were halved after every five consecutive epochs of no improvement. Training stops after 10 epochs without improvement or after 50 epochs have been completed.

Note that no data augmentation techniques will be employed in this case. Although different data augmentation can help different music separation systems improve performances (Uhlich et al., 2017), preliminary experiments using the DAMP-VSEP dataset and the Conv-TasNet separation model showed no significant improvements.

5.3.2 Evaluation

Experiment performances are measured in terms of the SDR, SI-SDR and STOI for the vocal segments and the SDR and SI-SDR for the background, and reported in terms of their improvement, i.e., ΔSDR, ΔSI-SDR and ΔSTOI. All models trained are evaluated using, $\mathbf{mix_{remix}}$, a mixture generated by remixing the sources. However, the model trained using

Table 5.4 Performances obtained by the source separation baseline models. Values reported correspond to the improvement score. 'DAMP' mixture denotes the use of the DAMP-VSEP provided mixture that includes non-linear effects, $\mathbf{mix_{damp}}$. 'Remix' mixture denotes the use of a generated mixture by remixing the vocal and backgrounds sources, $\mathbf{mix_{remix}}$. Values are expressed in *dB*. Learning rates (LR) for the best models are also included for completeness.

Train Set	**Train Mixture**	**Eval Mixture**	**Lr**	**Vocal**			**Background**	
				ΔSDR	**ΔSI-SDR**	**ΔSTOI**	**ΔSDR**	**ΔSI-SDR**
English	Remix	Remix	5e-3	14.22	15.10	0.17	11.5	10.88
Eng+Duets	Remix	Remix	5e-4	14.84	15.73	0.18	11.95	11.37
Eng+nEng	Remix	Remix	5e-4	**16.46**	**17.43**	**0.21**	**13.4**	**12.92**

the 'English+non-English' data, and the DAMP-VSEP provided mixture, $\mathbf{mix_{damp}}$, will also be evaluated using the mixtures provided for the evaluation samples.

The statistical significance of the improvements obtained from each experiment is measured using the dependent two-sided t-test statistic test for paired samples. The reported p-values indicate the significance of the improvements in relation to the results obtained from the model trained using the 'English' data.

5.3.3 Results

Table 5.4 reports the performance improvements obtained by the three models trained using the 'English', 'English+Duets' and 'English+non-English' training sets and a remix of the sources, $\mathbf{mix_{remix}}$, for model input. Also, it includes the performance improvements of the model trained using the 'English+non-English' training data, and the DAMP-VSEP provided mixture, $\mathbf{mix_{damp}}$, for model input. The reported values correspond to the mean score computed across all samples in the evaluation set.

The input scores (i.e., the scores of the mixed signals) are reported to contextualise the improvements obtained for each model. For the $\mathbf{mix_{remix}}$, the scores for the vocal source are -3.19 *dB* SDR, -5.27 *dB* SI-SDR and 0.45 STOI, and the scores for the background source are 5.25 *dB* SDR and 5.14 *dB* SI-SDR. For the $\mathbf{mix_{damp}}$ mixture, the initial vocal scores are -6.75 *dB*, -25.14 *dB* and 0.44 for the SDR, SI-SDR and STOI, respectively, and -4.57 *dB* SDR and -9.90 *dB* SI-SDR for the background.

Training the model using the 'English' training set resulted in a ΔSI-SDR of 15.10 *dB* vocal and 10.88 *dB* background. When training using the 'English+Duet' training set, i.e., using the English duets performances for data augmentation, the ΔSI-SDR score increased by 0.63 *dB* vocal ($p < .05$) and 0.49 *dB* background ($p < .05$), reaching 15.73 *dB* and

11.37 dB, respectively. A greater improvement was obtained when using the non-English solo-ensemble performances to augment the 'English' training set; the ΔSI-SDR score increased by 2.33 dB vocal ($p < .05$) and 2.04 dB background ($p < .05$), reaching 17.43 dB and 12.92 dB, respectively. The ΔSTOI score obtained when training with the 'English' data was 0.17. A slight gain of 0.01 was obtained when training with the 'English+Duets' data ($p < .05$). When training with the largest 'English+non-English' data, the ΔSTOI increased from 0.17 to 0.21 ($p < .05$).

The last two rows in Table 5.4 report the performances obtained when training the model using the DAMP-VSEP provided mixture, $\mathbf{mix_{damp}}$. When evaluation using a remixing of the sources, the ΔSI-SDR score dropped from 15.10 dB to -21.74 dB vocal and from 10.88 dB to -6.65 dB background, and ΔSTOI dropped from 0.17 to -0.22 ($p < .05$), i.e., the separation failed and the processed signals had poorer scores than the original mixture. On the other hand, when evaluating using the 'DAMP' mixture, a reduction of 0.99 dB vocal SI-SDR was obtained; however, a large background ΔSI-SDR of 20.04 dB was obtained. Note that the performance obtained when evaluating the model using the 'DAMP' mixture is not comparable with experiments evaluated using the 'Remix' mixture due to the differences in the origin of the mixtures.

Figure 5.2 shows six spectrograms from a 2-second snippet from one evaluation sample to visualise the effect of using the different training sets. This sample was chosen as it seems to represent the general pattern seen across the dataset. In the top panel, this figure displays the spectrogram of the generated mixture, $\mathbf{mix_{remix}}$, generated by adding the vocal and background sources. The second panel from the top shows the spectrogram of the target vocal source. The next three panels display the spectrograms of the estimated vocal from the models trained on the 'English', 'English+Duets' and 'English+non-English' training data, respectively. The last panel displays the error of the estimation from the 'English+non-English' model obtained by decomposing the estimated vocal into 'true vocal' and 'estimation error' by using the methods from the bss_eval toolkit (version 4)[6]. The red boxes highlight 200 milliseconds of a purely instrumental fragment, i.e., no singing is present in that fragment. The models should identify this fragment as part of the background source and not include it in the vocal estimation. However, as show in the 'English' spectrogram, a large instrumentation residual is included in the vocal estimation. Using more training data increases the performance of the model; this is observable in the progressive reduction of the residual by the models trained on 'English+Duets' and 'English+non-English' data. The blue rectangles in the first and last panels highlight narrow band residual errors in frequencies

[6]https://github.com/sigsep/bsseval/blob/92535ea70e5c0864286ee5f0c5a4fa762de98546/bsseval/metrics.py#L421

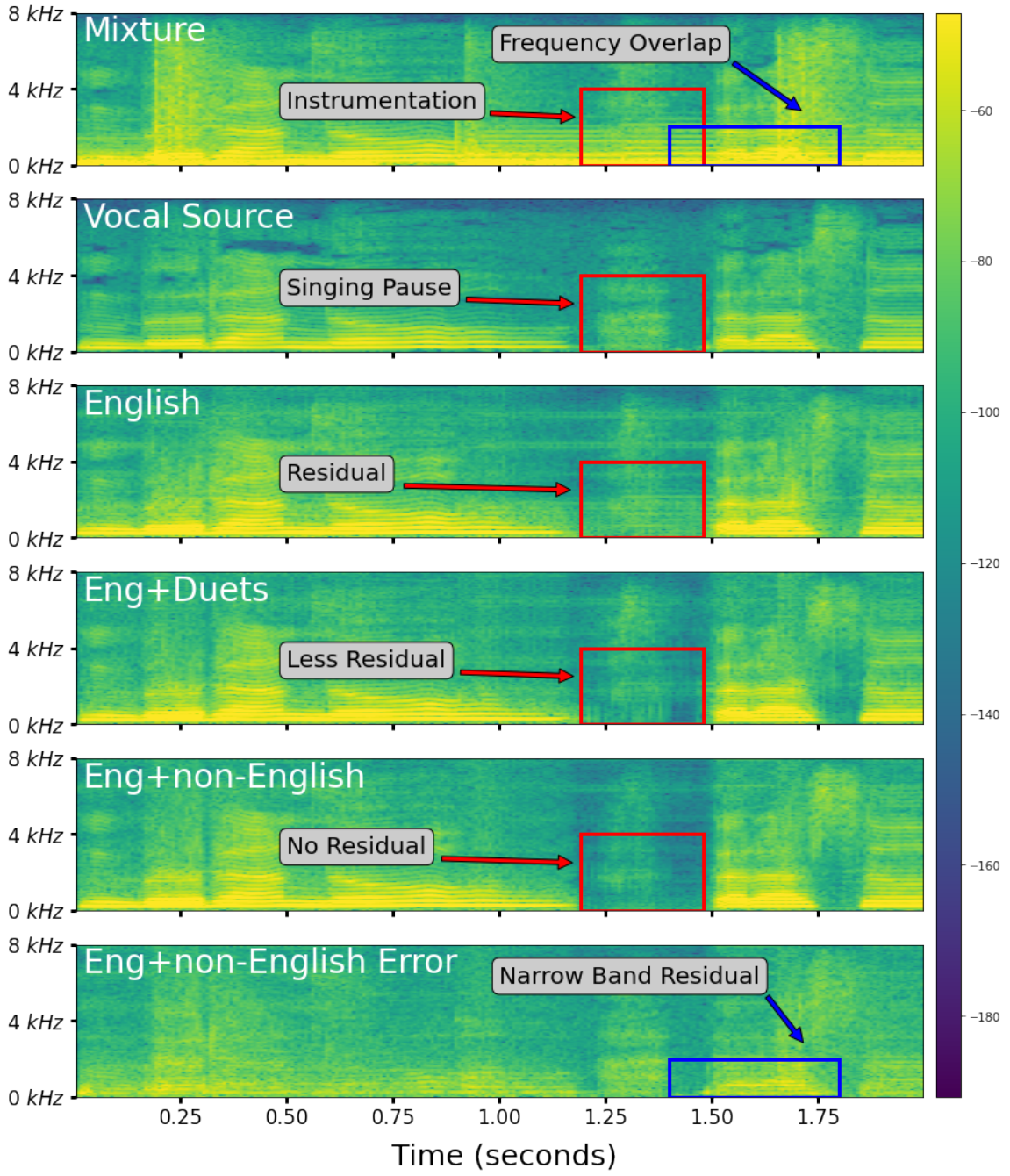


Figure 5.2 Example of separation performance from the baseline models trained using $\mathbf{mix}_{\mathbf{remix}}$ mixture. The figure shows six spectrograms of a 2-second snippet from one sample from the evaluation set. The top panel displays the mixture. The second panel displays the vocal target. The next three panels display the estimated vocal from the models trained using the 'English', 'English+Duets' and 'English+non-English' training datasets, respectively. The last panel displays the estimated error from the 'English+non-English' spectrogram.

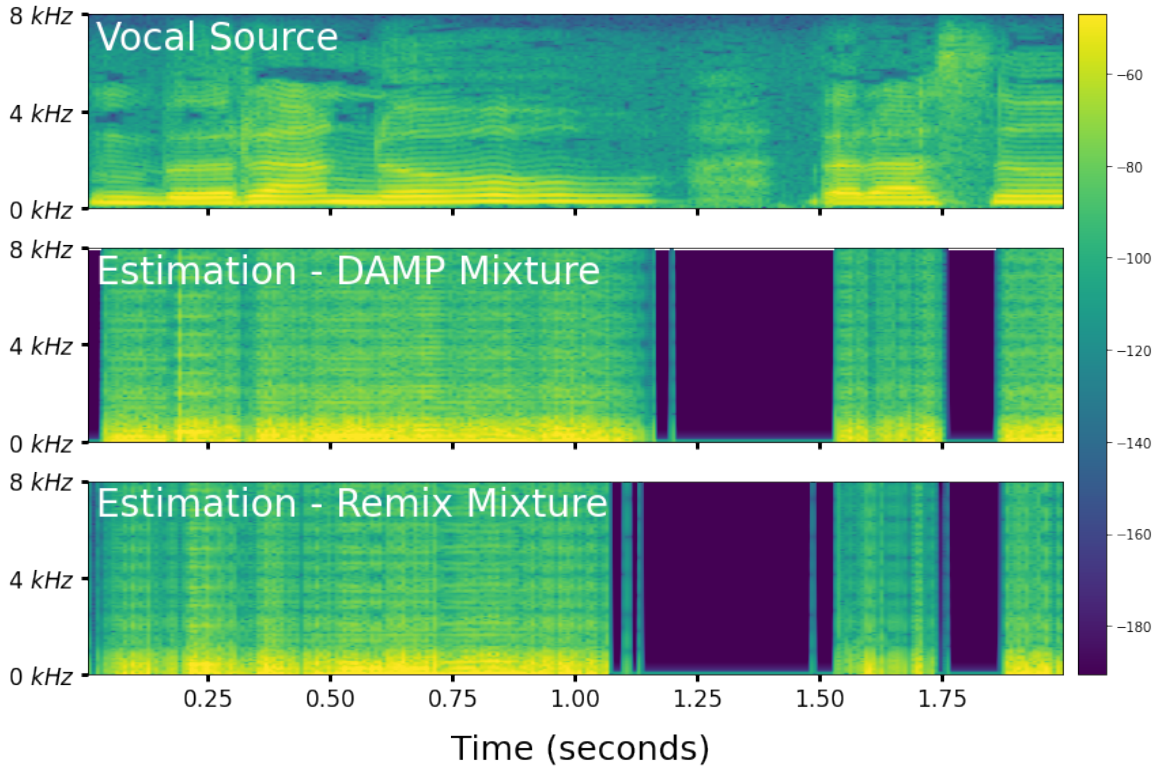


Figure 5.3 Example of separation performance from the baseline models trained using $\mathbf{mix}_{\mathbf{damp}}$ mixture. Figure shows three spectrograms from 2-seconds snippet from one sample from the evaluation set. The first panel displays the target vocal. The second panel displays the estimated vocal from the 'DAMP' mixture. The last panel displays the estimated vocal from the from the remixing of sources.

around $3500\,Hz$. The energy of this frequency belongs to the background sources and overlaps with some of the speech harmonics. Informing the training with the backgrounds structure may help them to better separate overlapped frequencies. This will be evaluated in Section 5.5.

Figure 5.3 shows three spectrograms of a 2-second snippet from the same sample used in Figure 5.2. The spectrograms correspond to the separated vocals from the model trained using the 'English+non-English' data and the DAMP-VSEP provided mixture, $\mathbf{mix}_{\mathbf{damp}}$, for model input. The top panel displays the spectrogram from the target vocal. The second panel shows the estimated vocal using the DAMP-VSEP provided mixture. The last panel shows the estimated vocal from the remixing of the sources. Note that this model completely fails to correctly identify and separate the vocal from the backgrounds for both types of mixtures inputs. The poor performance may be the results of the mismatch between the mixture, which includes non-linear operations, and the target sources that do not include such operations. As it can be seen in the spectrogram from both estimated vocals, the model fails to identify the

speech fragments. Training by focusing on the speech components may help the model to increase the separation performances. This will be evaluated in the next Section 5.4.

5.4 Composite Loss Function

As shown in the previous section, the model trained using the largest 'English+non-English' data and the DAMP-VSEP provided mixture, $\mathbf{mix_{damp}}$, for model input resulted in the poorest performance due to the inclusion of non-linear operations in the mixture. This section addresses the complexities of using the DAMP-VSEP mixture using a composite loss function. This loss function focuses on increasing the intelligibility of the vocal segment by using the short-term objective intelligibility loss function (STOI) (Taal et al., 2010) while using the MAE criteria to reduce the background distortion. The composite loss function is defined as:

$$\mathscr{L}_{composite} = \underbrace{(1-\alpha) \times MAE(\mathbf{b}, \mathbf{b}')}_{\text{Focus on reducing distortion errors}} + \underbrace{\alpha \times STOI(\mathbf{v}, \mathbf{v}')}_{\text{Focus on increasing the intelligibility}} \tag{5.4}$$

where α denotes the term weight in the range of 0 to 1 and, $\mathbf{b}$ and $\mathbf{v}$ are the background and vocal target signals, respectively; the prime symbol denotes the corresponding estimated signal.

Note that setting α equal to 0 is not equivalent to the loss function used in the baseline system. The baseline loss function computes the MAE over both sources, the vocal and background. However, MAE is only computed over the background source in the proposed composite loss function. This constraint is motivated by results from preliminary experiments. In these experiments, the MAE was computed over both sources and over the background source, obtaining better performances with the latter configuration.

Table 5.5 presents the performances obtained when utilising the composite loss function for training the model using the 'English+non-English' data and the DAMP-VSEP provided mixture, $\mathbf{mix_{damp}}$, for model input. The model was evaluated using both mixtures to evaluate the impact on both scenarios, i.e., the provided $\mathbf{mix_{damp}}$ and the sources' remixing $\mathbf{mix_{remix}}$. Experiments using a range of values for α showed that the best SI-SDR vocal performance for the 'DAMP' mixture is found when setting α equal to 0.5. When evaluating using the 'DAMP' mixture, a significant vocal ΔSI-SDR of 1.1 dB was obtained, increasing from -0.99 dB baseline to 0.11 dB ($p > .05$), However, this is still an insufficient improvement considering the baseline vocal SI-SDR of -25.14 dB. The STOI score resulted in a significant increment of 0.37 ($p < .05$). On the other hand, when evaluating using the remixing of sources, the vocal obtained a degradation of -3.29 dB, and the ΔSTOI increased from -0.22

Table 5.5 Evaluation results of the model trained using the composite loss function, the 'English+non-English' data and the mixture $\mathbf{mix_{damp}}$ for model input. Values reported correspond to the improvement scores. The first two rows correspond to the baseline scores obtained from the model trained using the MAE loss function, i.e., the scores reported in the last two rows from Table 5.4. All models using the $\mathscr{L}_{composite}$ objective function were trained using a learning rate of 5e-4. Values are expressed in *dB*.

α	Eval Mixture	Vocal			Background	
		SDR	SI-SDR	STOI	SDR	SI-SDR
-	Remix	-8.02	-21.74	-0.22	8.42	-6.65
	DAMP	-4.25	-0.99	-0.15	16.02	20.04
0.2	Remix	1.28	-4.69	0.03	8.08	-6.52
	DAMP	0.29	-0.06	0.23	15.44	19.32
0.3	Remix	1.21	-4.66	0.02	7.69	-6.72
	DAMP	0.27	-0.15	0.22	14.82	18.69
0.4	Remix	1.54	-2.7	0.04	7.22	-6.35
	DAMP	0.39	-0.34	0.22	14.27	17.95
0.5	Remix	1.31	-3.29	0.03	6.87	-6.51
	DAMP	0.3	0.11	0.22	13.94	17.54
0.6	Remix	0.82	-3.83	0.04	6.52	-6.4
	DAMP	-0.05	-0.4	0.22	13.43	17.13
0.7	Remix	0.53	-4.01	0.02	6.41	-6.34
	DAMP	-0.27	-1.1	0.22	13.39	16.88

to 0.03 ($p < .05$). Note that evaluating the remix mixture still results in degradation of the performance.

Figure 5.4 shows four spectrograms from a 2-second snippet of one evaluation sample (the same sample from Figure 5.3). The first panel displays the spectrogram of the 'DAMP' mixture, $\mathbf{mix_{damp}}$. The second panel displays the spectrogram of the target vocal. The third panel shows the estimated vocal from the baseline model (the same spectrogram as the second panel in Figure 5.3). The last panel shows the estimated vocal from the model using the composite loss function, $\mathscr{L}_{composite}$, with an α set to 0.5. The statistical test found a significant increment in the vocal SI-SDR. Moreover, due to the focus on the speech part of the mixture by the loss function, the singing fragments seems more clearly identified by the model. However, the estimated vocals retain a significant degree of residual, for example, the residual highlighted by the red rectangle in the last panel.

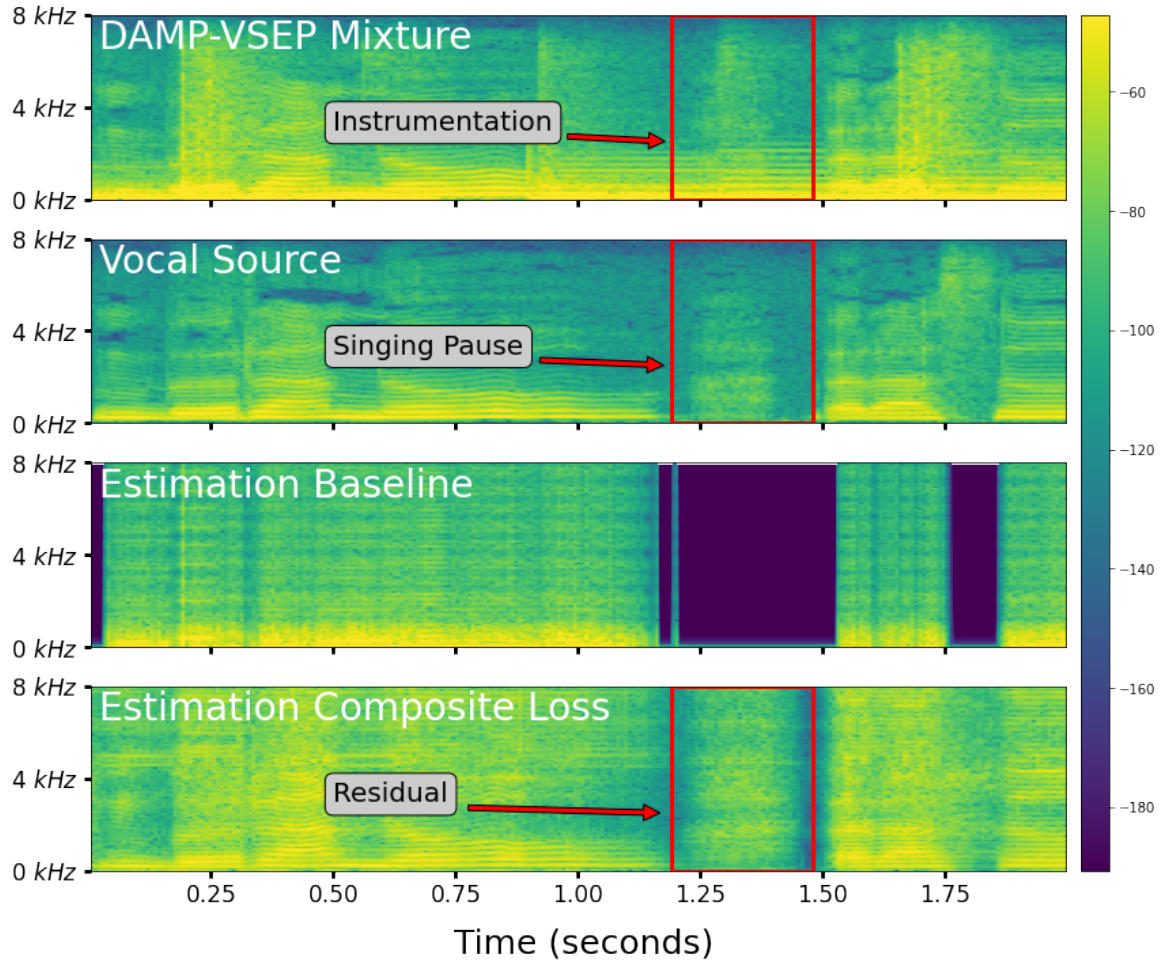


Figure 5.4 Example of separation performance from the model trained using $\mathbf{mix_{damp}}$ mixture and composite loss function. Figure shows four spectrograms from a 2-seconds excerpt sample from the evaluation set. The first two top panels display the $\mathbf{mix_{damp}}$ mixture and the target vocal, respectively. The third panel displays the spectrogram of the estimated vocal from the 'DAMP' mixture using the baseline model. The last panel displays the spectrogram of the estimated vocal using the model trained with the composite loss and α set to 0.5.

A significant performance reduction resulted from using the composite loss to train the model using the 'English+non-English' data and the remixing of sources, $\mathbf{mix_{remix}}$, for model input. The vocal ΔSI-SDR reduced from 17.43 dB baseline to 8.39 dB ($p < .05$) with α equal to 0.2 (best α when using $\mathbf{mix_{remix}}$) and a learning rate of 5×10^{-4}. This reduction is explained by a series of artifacts that the STOI loss function introduced in the estimated source (Figure 5.5).

Using a loss function that focuses on reducing the distortion and increasing the speech intelligibility has a clear benefit on improving the baseline performances when the mixtures

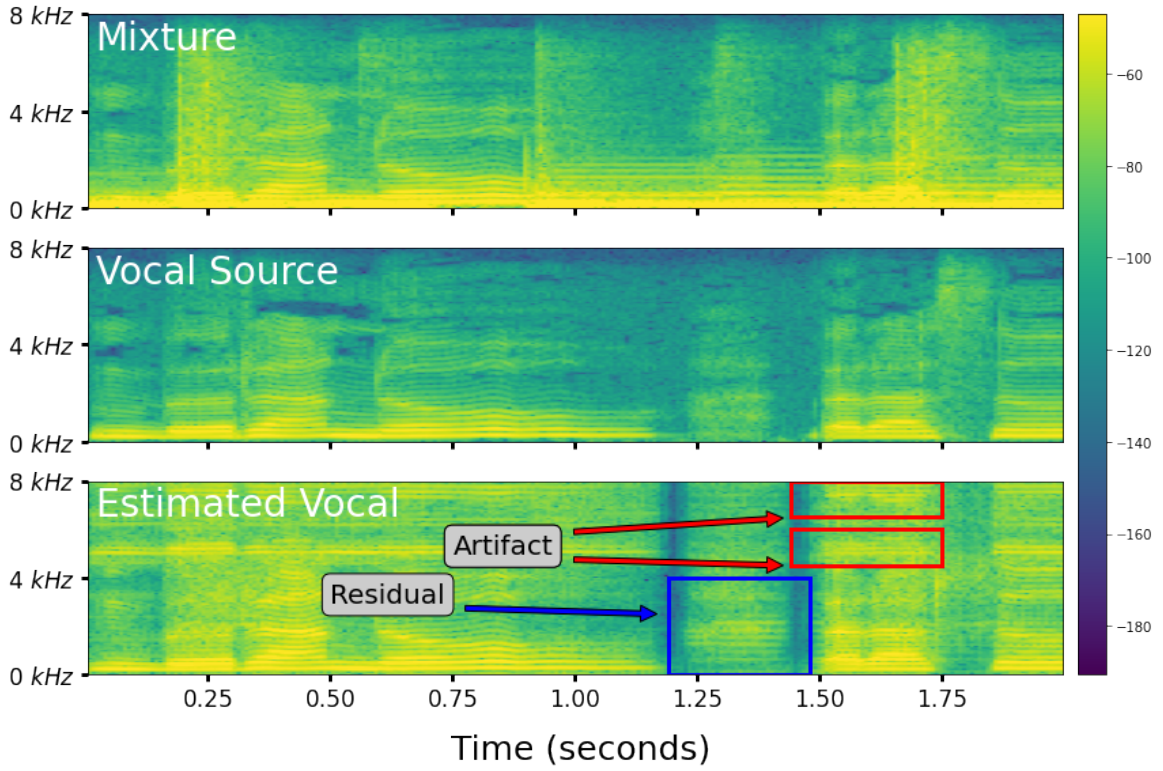


Figure 5.5 Example of separation performance from the model trained using $\mathbf{mix}_{\mathbf{remix}}$ mixture and composite loss function. Figure shows three spectrograms from a 2-seconds excerpt sample from the evaluation set. The first panel display the $\mathbf{mix}_{\mathbf{remix}}$ mixture. The second panel displays the target vocal. The last panel displays the spectrogram of the estimated vocal using the model trained with the composite loss and α set to 0.2.

include non-linear operations, $\mathbf{mix}_{\mathbf{damp}}$. The performance obtained may be sufficient for tasks like rebalancing the sound level between different sources. However, tasks like automatic speech recognition need higher quality separation where the sung speech is 'free' of distortions. The use of the STOI loss function, as presented in this chapter, results in insufficient performances for recognition and introduced several artifacts to the estimated vocal.

5.5 Musical Background Embedding

The baseline models reported in Section 5.3 resulted in high-performance separation with 15.68 dB vocal ΔSI-SDR when training using the largest 'English+non-English' data. However, narrowband instrumentation residuals remain in the estimated vocal. It is hypothesised that informing the model with the musical characteristics of the background would further

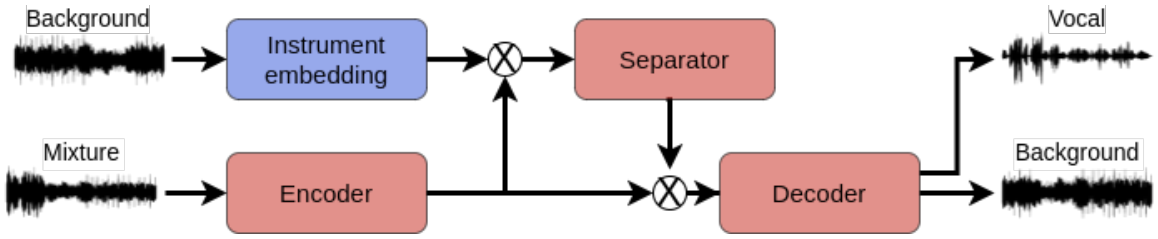


Figure 5.6 Block diagram of the Conv-TasNet architecture extended using the instrument embedding module.

improve the separation performances. For this purpose, an extension of the separation architecture is proposed by extending the Conv-TasNet architecture introducing a fourth module called *instrument embedding*, as is shown in Figure 5.6. We will refer to this model as 'ConvTasNet-extended'. The instrument embedding module extracts embedding information from the background source using a pretrained embedding model to inform the separator. The structure of the encoder, separator and decoder modules remain unaltered to ensure that any variation in performance results from the introduction of the instrument embedding and not from modifications to the baseline architecture, making the results comparable with the previous experiments.

Two embedding models were evaluated. The first model, called VGGish (Hershey et al., 2017), is an out-of-the-box pretrained convolutional neural network from Google based on the VGG architecture for image classification (Simonyan and Zisserman, 2014). The model is trained on the AudioSet (Hershey et al., 2017) dataset, a large audio events dataset sourced from over two million YouTube videos. The second embedding model evaluated was the X-vectors architecture for speaker recognition (Snyder et al., 2018) repurposed for instrumentation embedding. X-vectors consist of five time-delay layers that operate at frame-level, a statistic pooling layer that aggregates the frame-level representations and two linear layers that operate at segment level. The network is trained to classify each speaker in the training data. The instrumental X-vectors were trained using a corpus constructed from the duets ensembles from DAMP-VSEP and augmented using selected samples from the FMA dataset (Defferrard et al., 2017).

5.5.1 VGGish Embedding

For the initial embedding evaluation, the out-of-the-box "VGGish" embedding (Hershey et al., 2017) was used. VGGish is based on the VGG architecture developed by Simonyan and Zisserman (2014) from the Visual Geometry Group research lab at the University of Oxford. VGG is a convolutional network winner of the ImageNet Large-Scale Visual Recognition

Challenge (ILSVRC) (Russakovsky et al., 2014) for the year 2014. The size of the input image is fixed to 224×224 RGB. The image is passed to a five stacks of 3×3 convolutional layers plus a max-pooling layer. Then, the last max-pooling is connected to a final dense network of two fully connected layers with 4096 channels each and the output layer with 1000 channels (one for each class in ImageNet). Rectified linear activation unit non-linearity is applied after each hidden layer. The number of channels per convolutional layer begins at 64 and increases by a factor of two after each max-pooling layer, reaching 512. Simonyan and Zisserman (2014) presented six different variants of the architecture varying in the depth of the convolutional layers per stack. The different variants are explained in detail in their paper.

The VGGish embedding model is a variant of the VGG architecture 'variant A', consisting of 11 layers: eight convolutional and three fully-connected. However, and unlike VGG, VGGish input size is fixed to 96×64 for log Mel spectrogram audio inputs. VGGish drops the last convolutional stack, keeping four instead of five groups of convolutional plus max-pooling layers. The output channels are reduced from 1000 channels to 128 channels. The output layer acts as the embedding layer.

Simonyan and Zisserman (2014) trained the VGGish using the AudioSet corpus (Hershey et al., 2017), a large scale multi-class dataset for audio events. AudioSet consists of 632 audio events classes and a collection of over two million human-labelled 10-second clips drawn from YouTube videos. The classes cover many human and animal sounds, musical instruments and genres, and common environmental sounds. Each audio clip can be classified into one or more classes, with roughly 50% of the samples containing 'music'. The large scale of the AudioSet dataset makes the VGGish embedding a suitable model to evaluate the effect of instrumentally informed training of source separation models.

The VGGish embedding model returns a 128-dimensional vector for every 9.8 seconds of audio. This may be useful for online multi-class classification where different audio events can occur over time. However, the instrumentation and musical genre of a song remain invariant over time. Therefore, the resulting vectors from tracks longer than 9.8 seconds are flattened by averaging across the columns, keeping a single embedding vector for the song without losing information.

5.5.2 X-vectors Embedding

To contrast the VGGish embedding for audio events with a purely instrumentally trained embedding, the use of the "X-vectors" embedding (Snyder et al., 2018), repurposed for musical embedding, was evaluated. X-vectors are a robust DNN model designed for speaker embedding typically used in *speaker recognition* or *ASR* systems. The network architecture

consists of five time-delay neural network (TDNN) layers that operate at the speech frame-level. These layers progressively increase the temporal context starting with five frames in the first layer, 9 in the second and 15 from the third to the fifth layers. After the TDNN layers, a statistical pooling layer aggregates all frames by computing the mean and standard deviation. Then, the concatenation of these scores is propagated through two fully connected layers with 512 channels and a final softmax output layer where each class corresponding to a different speaker in the training data, i.e., the output size depends on the number of speakers. The embedding is extracted from the first fully connected layer before the non-linearity. The model is trained utilising 40 MFCCs with a frame length of 25 *ms* and a 15 *ms* overlap.

To repurpose the X-vectors for instrumental music embedding, an instrumental corpus was constructed. This corpus consists of the 1592 unique backgrounds from the DAMP-VSEP duets ensembles and augmented with 41,669 selected 30-seconds tracks from the *FMA* corpus (Defferrard et al., 2017), totalling 43,261 tracks. The FMA corpus is a dump of the available songs stored in the free and open musical library *Free Music Archive*[7] on April 1st 2017, totalling 106,574 polyphonic songs. The FMA dataset is distributed in various sizes of MP3 encoded audio collections. The most extensive set *FMA-full* contains the complete collection of 106,574 tracks, comprising 161 unbalanced genres. The second set, *FMA-large*, also includes the complete collection of tracks, but the audios are limited to 30-seconds excerpts extracted from the centre of the songs. The next set *FMA-medium* was designed for single-genre classification by selecting tracks with a single root genre (out of 16) and high popularity (as stored in the metadata), resulting in 25000 30-seconds excerpts with an unbalanced number of samples per genre. The last and smallest set *FMA-small* is a balanced subset of FMA-medium containing 8000 30-seconds tracks corresponding to the eight most popular genres and 1000 tracks per genre.

The 41,669 FMA samples used to train the X-vectors were selected from the FMA-large subset using the acoustic Latent Dirichlet Allocation (aLDA) data selection framework proposed by Doulaty and Hain (2019) and described in Appendix C. The data selection objective is to select the best-matching training data from a large pool of out-of-domain data to a set of representative samples sampled from a target domain. The duets backgrounds from the DAMP-VSEP corpus were used for the target domain, and the 106,574 samples from the FMA-large set are the pool of out-of-domain samples. Due to the fact that FMA tracks are a mixture of background and singing, backgrounds were separated from the vocals using the *Demucs-extra* (Défossez et al., 2019) pretrained source separation model. Demucs-extra is a source separation model trained on MUSDB18 plus extra data currently reporting the best overall speech separation performance on the MUSDB18 task. The final aLDA system

[7] https://freemusicarchive.org/

trained utilised 1664 Gaussian components trained on 40 MFCCs, 2048 audio topics and 1024 K-means clusters. Tracks were selected when the Euclidean and cosine distances between the audio topic of the sample and one of the K-means centroid were lower than a threshold of 0.2 and 0.15, respectively.

5.5.3 Training and Evaluating the ConvTasNet-Extended Model

In a speaker separation context, embeddings would be extracted from utterances where the target speaker is in isolation. An equivalent in music separation would be the use of solo instrumental sections, such as the introduction in some songs. One could argue that the first N seconds of a song are typically solo-instrumentation. However, there are no guarantees that all instruments will be activated at that time, and there are no rules that dictate for how long the instrumental introduction part would extend before the singing starts.

Therefore, for the training of the models, the embedding was extracted from the first 10 seconds of the background source, a segment previously taken aside for this purpose. This process ensured that every training sample generated from the same performance is using the same embedding vector.

For the evaluation, two processes were implemented. The first evaluation followed the same procedure as the training process, where the embeddings are extracted from the first 10 seconds of the backgrounds. In this case, all test segments generated from the same performance share the same embedding. This evaluation will be referred as 'Common Embedding' for the rest of the thesis. However, this evaluation is regarded as upper-bound result as, in practical applications, there is unlikely to be access to 10 second of unaccompanied background before each mixed segment. Therefore, a second evaluation using a 'two-pass' procedure (Figure 5.7) was implemented, This evaluation proceeds as follow. In the first pass, the background is estimated from the mixture by using the best baseline model trained on the 'English+non-English' dataset, presented above. Next, the second pass performs the source separation using the ConvTasNet-extended model, computing the instrumental embedding from the *estimated* background. Note that, in this case, the length of the audio used for computing the embedding is variable, i.e., the length depends on the audio segment instead of a fixed duration.

Implementing the two-pass process is motivated by two-stage source counting and source separation systems and iterative music separation methods. First, most audio source separation systems assume that the number of sources is known. However, in many practical situations, the number of sources is unknown. Two-stage methods aim to estimate the number of sources before the source separation (Mirzaei et al., 2015; Sgouros and Mitianoudis, 2020). Second, several music separation methods implement a combination of techniques that, in

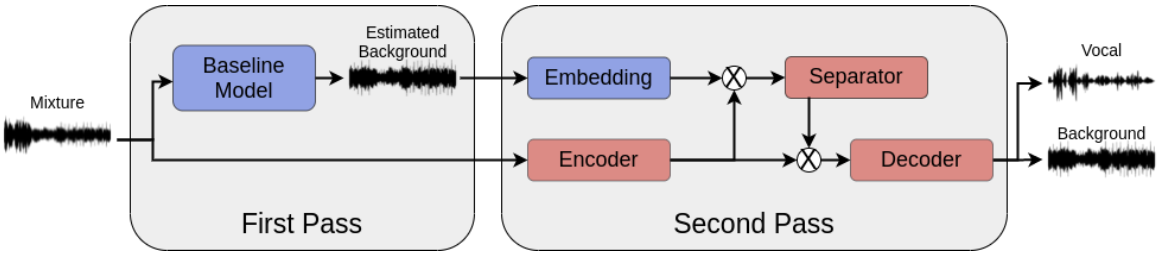


Figure 5.7 Diagram with the two-pass embedding evaluation procedure.

cascade or iteratively, leverage the benefits of one or more methods to improve performances (an overview of cascade and iterative methods was presented by Rafii et al. (2018)).

5.5.4 ConvTasNet-Extended Results

Table 5.6 reports the vocal ΔSI-SDR scores obtained from the evaluation of the ConvTasNet-Extended models. The table includes the delta performance, Δ, corresponding to the difference between the common embedding and two-pass evaluation procedures. Using the two-pass procedure for the models trained using the $\mathbf{mix}_{\mathbf{remix}}$ mixture and the VGGish embedding resulted in a reduction of vocal ΔSI-SDR of 0.24 *dB*, 0.19 *dB* and 0.27 *dB* for the models trained using the 'English', 'English+Duets and the 'English+non-English' datasets, respectively. On the other hand, models using the X-vectors embedding resulted in a much greater reduction when evaluating using the two-pass procedure of 0.87 *dB* for the 'English' dataset and up to 1 *dB*, for the 'English+Duets' and 'English+non-English' datasets.

Similar improvements are obtained from models trained using the $\mathbf{mix}_{\mathbf{damp}}$ mixture. When evaluating the DAMP mixture, a reduction of more than 1 *dB* was obtained for both embedding models when using the two-pass embedding procedure versus the common embedding. When evaluating the remix mixture, negative performances were obtained with both embedding models.

The results presented in Table 5.6 may indicate that the VGGish embedding is less sensitive than the X-vectors to the estimated background and more robust on computing song information from different parts of the song. This was an unexpected result as the X-vectors were primarily trained on separated backgrounds, which should have made it more robust for separated conditions.

Only the two-pass evaluation will be reported for the rest of the thesis as it describes a real separation scenario. However, when deemed relevant, the common embedding evaluation performances will also be included.

Table 5.7 shows the results of training the source separation models utilising the VGGish and X-vectors instrument embedding. The embedding results correspond to performances

Table 5.6 Comparison of the vocal ΔSI-SDR score obtained when using the common embedding and two-pass evaluations procedures to evaluate the ConvTasNet-Extended models. Common Embedding reports the score when computing the embedding from the leading 10-seconds. Two-Pass reports the score when using the two-pass evaluation procedure. Delta, Δ, reports the difference between the common embedding and two-pass evaluation procedures. Values are expressed in *dB*.

Train Set	Train Mixture	Test Mixture	Embed	Common Embedding	Two-pass	Δ
English	Remix	Remix	VGGish	14.84	14.60	-0.24
			X-vectors	14.72	13.85	-0.87
English +Duets	Remix	Remix	VGGish	15.72	15.53	-0.19
			X-vectors	15.52	14.54	-0.98
English +non-English	Remix	Remix	VGGish	17.47	17.20	-0.27
			X-vectors	17.58	16.58	-1.00
	DAMP	Remix	VGGish	-17.22	-16.90	0.32
			X-vectors	-20.29	-21.19	-0.90
	DAMP	DAMP	VGGish	0.57	-1.17	-1.74
			X-vectors	0.19	-1.41	-1.60

using the two-pass evaluation. The results from the baseline models trained on the 'English', 'English+Duets' and 'English+non-English' data and $\mathbf{mix_{damp}}$ mixture, and the best model trained using the composite loss function and the $\mathbf{mix_{damp}}$ mixture are included to facilitate the comparison.

Using the two-pass evaluation procedure, neither the VGGish nor the X-vectors resulted in statistically significant improvements. This is true for all training sets. However, using the common embedding evaluation, VGGish embedding resulted in a non-statistically significant vocal SI-SDR improvement of 0.02 *dB* increasing from 17.43 *dB* to 17.46 *dB*. On the other hand, X-vectors resulted in a increment of 0.15 *dB* ($p < .05$) increasing to 17.58 *dB*. However, no significance was obtained from the improvement of the X-vectors over the VGGish.

The common embedding evaluation procedure showed that the X-vectors instrument embedding can in fact add a slight improvement when training the models with large amount of data. However, the two-pass evaluation, which attempts to apply the embedding on real scenarios, fails on taking the advantages of the embedding model.

Table 5.7 Performances obtained by the source separation models using instrument embedding. The embedding results corresponds to performances using the two-pass evaluation. The values reported correspond to the improvement score. Baseline and composite loss models are included for comparison. Values are expressed in *dB*. Values in bold corresponds to the best performance per dataset, train and evaluation mixtures.

Train Set	Train Mixture	Eval Mixture	α	Embed	Lr	Vocal			Background	
						SDR	SI-SDR	STOI	SDR	SI-SDR
English	Remix	Remix	-	-	5e-3	**14.22**	**15.10**	**0.17**	**11.50**	**10.88**
			-	VGGish	5e-3	13.84	14.60	0.16	11.27	10.59
			-	X-vectors	1e-3	13.28	13.85	0.16	10.81	10.02
English +Duets	Remix	Remix	-	-	5e-4	**14.84**	**15.73**	0.18	**11.95**	**11.37**
			-	VGGish	1e-3	14.76	15.53	**0.19**	11.95	11.33
			-	X-vectors	1e-3	13.97	14.54	0.17	11.33	10.61
English +non-English	Remix	Remix	-	-	5e-4	**16.46**	**17.43**	**0.21**	**13.40**	**12.92**
			-	VGGish	5e-4	16.43	17.20	0.21	13.30	12.88
			-	X-vectors	1e-3	15.91	16.58	0.20	12.86	12.36
	DAMP	Remix	0.5	-	5e-4	1.31	**-3.29**	**0.03**	6.87	-6.51
			0.5	VGGish	3e-4	**1.76**	-17.05	0.03	**7.22**	**-6.27**
			0.5	X-vectors	5e-4	0.13	-10.82	-0.01	6.88	-6.4
	DAMP	DAMP	0.5	-	5e-4	0.30	**0.11**	**0.22**	**13.94**	**17.54**
			0.5	VGGish	3e-4	**7.41**	0.08	-2.99	13.84	-5.81
			0.5	X-vectors	5e-4	7.35	0.04	-2.22	13.46	-5.89

5.6 Summary and Conclusion

This chapter dealt with the task of separating the sung speech from the instrumental accompaniment from a polyphonic song. This problem was tackled by utilising a convolutional time-domain audio separation network (Luo and Mesgarani, 2019) (Conv-TasNet). Conv-TasNet is a well-known source separation model initially designed for speech separation (i.e., two overlapping speakers) and re-purposed to music separation by Défossez et al. (2019). Unlike Défossez et al. (2019) that uses the MUSDB18 dataset (Rafii et al., 2017), in this thesis, the model was trained by using data from the DAMP-VSEP corpus, which acoustic characteristics match with the corpus used for the sung speech acoustic modelling investigation in Chapter 4. Three training datasets were constructed. The first training set, the 'English' set, is composed of all English solo-ensemble performances. The second set, the 'English+Duets' set, augments the 'English' data with the English duets ensembles. The duets performances were included by splitting them into two single ensembles. The last training set, the 'English+non-English' set, augments the 'English' set with all non-English singles ensembles. Validation and evaluation sets were constructed from performances from the 'English' training set where the background is not repeated. A background was considered

repeated if two or more performances used the same or perceptually similar background. This process ensured that no backgrounds were overlapping between training and testing sets. The final size for the training datasets is 15, 25 and 66 hours for the 'English', 'English+Duets' and 'English+non-English' sets, respectively. The sizes of the validation and evaluation sets are 30 minutes each. Note that, for each performance, the DAMP-VSEP corpus provides a waveform for the vocal, background accompaniment and a mixture of both. The provided mixtures, mix_{damp}, may contain non-replicable non-linear operations that are not included in the isolated source.

Four baseline systems were constructed using the data described above. The first three systems were trained using the 'English', 'English+Duets' and 'English+non-English' training data, and a mixture generated online by remixing the vocal and background sources (mix_{remix}). The last model was trained using the 'English+non-English' training data, and the DAMP-VSEP provided mixture (mix_{damp}). This model evaluated the performance of the separation model on a stressed scenario where the model must learn how to filter out non-linear operations from the mixture. All experiments were evaluated in terms of the vocal ΔSDR, ΔSI-SDR and ΔSTOI, and background ΔSDR and ΔSI-SDR. For the models trained using the mix_{remix} mixture, the 'English' model resulted in a vocal SI-SDR improvement of 15.10 dB. This score increases by 0.63 dB to 15.73 dB when training the model using the larger 'English+Duets' dataset. When training with the largest 'English+non-English' dataset, the score increases by 2.33 dB, reaching 17.43 dB. For the model trained using the largest 'English+non-English' dataset and the mix_{damp} mixture, a reduction of -21.74 dB vocal SI-SDR was obtained when evaluating the mix_{remix} mixture and reduction of 0.99 dB when evaluating the mix_{damp} mixture.

The baseline results showed that the model trained using the mix_{damp} mixture failed to identify the singing areas. Therefore, experiments training the model using a speech intelligibility oriented loss function were carried out. The new objective function evaluated is a linear combination of MAE and STOI intelligibility loss functions. This model resulted in -3.29 dB vocal ΔSI-SDR when evaluating the mix_{remix} mixture (in contrast to -21.74 dB obtained from the baseline system), and 0.11 dB when evaluating the mix_{damp} mixture. Note that the baseline and the combined loss function experiments performed lower than not taking any action.

Further experiments evaluated the effect of informing the model with the background accompaniment characteristics, addressing **RQ.5**. This was done by extending the Conv-TasNet model using an embedding model. Two embedding systems were evaluated: the VGGish and X-vectors embedding. VGGish is a pretrained system trained on several audio events inspired by the VGG image classifier system. The X-vectors is a speaker

embedding system that was repurposed in this thesis for instrumental embedding. The model was trained on an extensive collection of solo-instrumental data composed of the DAMP-VSEP non-English duets backgrounds and augmented by selecting thousands of separated background samples from the FMA-large subset of the FMA corpus (Defferrard et al., 2017). Two evaluation procedures were implemented. The first evaluation, common embedding, employed the first ten seconds to compute the embedding, same as the training process. However, for practical applications, a two-pass procedure was implemented. The two-pass uses the best baseline system to estimate a background from which the embeddings are computed. No significant improvement was obtained with either embedding model when using the 'English' or the 'English+Duets' dataset. However, informing the model trained with the largest 'English+non-English' dataset and mix_{remix} mixture with the background characteristics through embedding was shown to increase the vocal SI-SDR improvement when using the common embedding procedure. However, improvements obtained from embeddings are lost when using the two-pass system.

The following chapter will address **RQ.6** by performing a full evaluation of the complete system, i.e., a system constructed by integrating the source separation model considered in this chapter with the sung speech ASR system investigated in Chapter 4. For this, it will use a broader set of datasets, such as the Hansen's dataset (Hansen, 2012), the Mauch's dataset (Mauch et al., 2012) and the Jamendo dataset (Stoller et al., 2019) used for different MIREX challenges evaluation.

Chapter 6

Polyphonic Lyrics Transcription System

6.1 Introduction

Chapter 4 investigated the challenge of recognising the lyrics from an unaccompanied sung speech recording. The chapter first addressed the lack of readily available sung speech recognition data (**RQ.3**) by presenting the new DSing sung speech data, a dataset with three increasingly larger training sets based on the English performances from the DAMP-MVP corpus (Smule, Inc., 2018). Using the DSing dataset and the Kaldi toolkit (Povey et al., 2011), it was addressed the question of how well a spoken speech designed ASR system can perform when trained using sung speech data and whether musically-motivated features can help improve the recognition performances (**RQ.4**). An ASR system based on the state-of-the-art TDNN-F acoustic model designed for spoken speech was constructed The model was trained using MFCC features plus different musically-motivated features. When using the largest DSing training dataset (about 149 hours), it was obtained an average WER of 19.60% computed across 11 runs.

Chapter 5 investigated the challenge of separating the singing from the background accompaniment. For this, work on processing the DAMP-VSEP corpus (Smule, Inc., 2019) to construct three increasingly larger training sets, again addressing **RQ.3**, was presented. Employing the Conv-TasNet separation system (Luo and Mesgarani, 2019), a source separation system that served as a baseline was constructed. Then, we employed embedding ideas from speaker separation systems to evaluate instrumental embeddings to inform the separation system with the background structure of the songs (**RQ.5**). Two embedding systems were considered for the embedding evaluation: the VGGish (Hershey et al., 2017), a pretrained audio events embedding system, and the X-vector embedding (Snyder et al., 2018) trained on estimated backgrounds from the FMA dataset (Defferrard et al., 2017). The evaluation procedure performed a two-pass evaluation procedure. In the first stage,

the background is estimated using the baseline separation system. In the second stage, the estimated background is used to compute the embedding.

When training the models using the largest training set (66.5 hours), the best baseline performances resulted in an improvement of 17.43 dB SI-SDR for the vocal source (improvement from the input vocal SI-SDR of -5.27 dB). Using the embedding model with a two-pass evaluation procedure was found to be unhelpful and the performance decreases to 17.20 dB ΔSI-SDR when using VGGish and to 16.58 dB ΔSI-SDR when using X-vectors.

Having trained an unaccompanied sung speech recognition system and a music source separation model, we can now investigate to what extent these two systems can be integrated to recognise the singing from a polyphonic song and how the mismatch between the 'clean' and estimated singing affects the recognition performances (**RQ.6**). This can be done by using the music source separation model as the front-end to separate the singing before passing it to a transcription back-end. However, considering that source separation models are not perfect, and residuals from one source are expected in the others, a simple connection of both systems might not be sufficient.

The present chapter will first combine the best sung speech recognition system from Chapter 4 with the best source separation model from Chapter 5 to construct an accompanied sung speech recognition system. However, in this simply-connected system, the ASR will be operating on mismatched singing characteristics, i.e., the separated singing contains distortions that are not present in the unaccompanied sung speech data used to train the acoustic model. Different distortions may favour different types of errors depending on the accompaniment's characteristics and the estimation's effectiveness. For example, a separated vocal may suffer from deletions if the residual background energy remains sufficient to mask the speech content of the signal. Alternatively, substitutions are likely where the separation algorithm has distorted the signal causing mismatch to the trained acoustic model. In situations where non-vocal instrumental segments have not been filtered to silence, insertions can occur, e.g. instruments triggering fully voiced words such as 'you', 'I', 'a', etc.

Transfer learning ideas can be employed to reduce the mismatch between the separated singing and the unaccompanied sung speech, i.e., to develop a model for a task by reusing a model pretrained on a different but related task as the starting point. Along these lines, this chapter will use three small datasets of separated singing – small in comparison with the 149 hours of unaccompanied sung speech data used for training the acoustic model – to adapt the acoustic model trained on unaccompanied sung speech. Reducing the mismatch between the separated and unaccompanied singing is expected to improve the recognition performances by reducing uncertainties in the acoustic model. The effectiveness of the adaptation may depend on the size and accompaniment characteristics of adaptation data.

The chapter is organised as follows. First, Section 6.2 presents several accompanied sung speech datasets that will be used for evaluation and adaptation in the rest of the thesis. Next, Section 6.3 evaluates how the recognition performances are affected by the mismatch between the separated sung speech, obtained from the separation model, and the 'clean' sung speech, used for acoustic modelling. Then, Section 6.4 evaluates the use of separated sung speech to adapt the acoustic model. Last, Section 6.5 summarises the key findings of this chapter.

6.2 Accompanied Sung Speech Datasets

This section presents a new version of the DAMP-VSEP validation and evaluation sets (sets introduced in Chapter 5) to be used for accompanied sung speech recognition evaluation. The segmentation made in Chapter 5 corresponds to segments generated by using a voice activity detector. However, as the recordings correspond to 30-second fragments extracted from seconds 60 to 90, there is no guarantee that each segment matches with correct lyric phrases.

Additionally, this section introduces five manually transcribed polyphonic music datasets that will be used for ASR evaluation or adaptation but are too small for training recognition systems. These datasets include Hansen's dataset (Hansen, 2012), Mauch's dataset (Mauch et al., 2012), the Jamendo dataset (Stoller et al., 2019), the ACOMUS dataset (Roa Dabike, 2016), the DALI dataset (Meseguer-Brocal et al., 2018) and the MUSDB18 dataset (Rafii et al., 2017). Note that MUSDB18 transcriptions were recently made available by Schulze-Forster et al. (2021).

6.2.1 DAMP-VSEP ASR Evaluation

Chapter 5 presented the construction of a vocal separation dataset based on the DAMP-VSEP corpus (Smule, Inc., 2019). In the separation dataset, the first ten seconds of the tracks were held back to be used for background embedding experiments, and the last 20 seconds were segmented by employing a voice activity detection system. The segments will generally contain some truncated sentences that do not match the structure of the lyric lines on which the language model has been trained. This mismatch will mean that these fragments will be poorly transcribed.

The use of these datasets enables three scenarios. First, the isolated vocal segment can be used for unaccompanied sung speech recognition evaluation, which corresponds to the upper bound performance for this data. Second, the accompanied sung speech (i.e., a mixture

between the vocal and background sources) can be used for computing the lower-bound recognition performance. Last, the sung speech can be estimated from the mixture using the music source separation model to evaluate how the recognition is affected by the distorted sung speech.

Therefore, a new segmentation of the DAMP-VSEP evaluation set was constructed that aims to include only the lyric lines contained within the 30-second performances, matching the structure used to train the language model in Chapter 6.5. This means that incomplete words or sentences from the borders of the 30-second segments were excluded. The new segmentation was performed using the whole 30-second performances, i.e., no segments were taken aside for embedding.

The segmentation and annotation process was divided into three steps. The first step aligned the accompaniment with the vocal sources, employing the same procedure described in Section 5.2 when dealing with DAMP-VSEP's challenge 4, that is, obtaining the lag where the cross-correlation score between the vocal and the mixture maximises.

Second, after aligning the sources, the vocal tracks were segmented into meaningful utterances (i.e., utterances matching lyric phrases) using the 'Audino' toolkit (Grover et al., 2020), an open-source audio annotation tool. The segments were constructed by taking each lyric phrase and manually finding the start and end times in the audio with a precision of two decimal points. The lyric texts were recovered from the Smule website. Next, the annotations were manually corrected for the cases where the singer replaced or omitted some words.

Last, each singing segment was labelled in terms of the language of the segment (i.e., English or non-English), the sample quality (i.e., clean segment or contains background bleeding or contains environmental noise) and if it is suitable for recognition evaluation. These labels were used to filter out segments that were not suitable for ASR evaluation.

Figure 6.1 shows a screenshot of the segmentation and annotation of one DAMP-VSEP evaluation sample using the Audino application. At the top, the application shows the audio track and the selected segments using boxes. In the middle, the application has a 'Reference Transcription' section that is meant for presenting the transcription of the whole audio. However, in this case, it was used to include the performance information to facilitate the recovering of the lyrics. Below the reference transcription, there is a box to edit the transcription of the currently selected segment. Finally, below the segment transcription, different personalised labels can be included, which, in this case, included the language of the segment, the audio quality and if the track would be suitable for ASR evaluation.

Note that a single annotator performed the new annotation. This was convenient for the small size of the dataset to annotate (about one hour of data). Using a single annotator has the advantage of having a single criterion when segments should be discarded. In contrast,

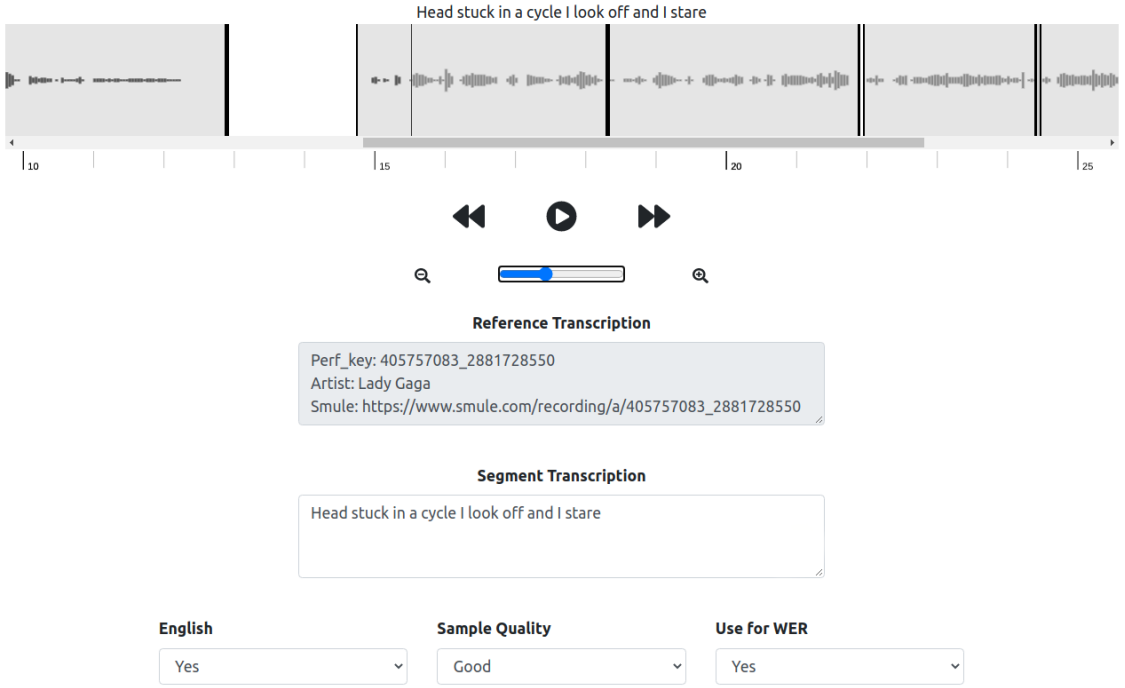


Figure 6.1 Screenshot of the segmentation and annotation on one sample using Audino application. At the top is showing the audio track and the segments. In the middle, the performance information is included in the 'Reference Transcription' section to facilitate recovering the lyrics from the Smule website. Bellow the reference transcription, there is a box to edit the transcription of the selected segment. The bottom of the image shows three combo boxes to label if the segment is in English, the audio quality, and if it is suitable for recognition.

when more than one annotator is required, it is common to have a third annotator check the annotations to ensure consistency.

From now on, these new segmented versions will be referred to as 'VSEP (eval)' and 'VSEP (valid)' sets. Note that, in this case, the '(eval)' and '(valid)' terms are used to indicate that the datasets correspond to the new version of the evaluation and validation sets used in Chapter 5, respectively.

The final size of the VSEP (eval) set, after filtering out six non-English performances, is 88 performances, segmented into 410 lyrics lines totalling 0.56 hours of singing segments. Similarly, five non-English songs were discarded from VSEP (valid) set, resulting in 87 performances, 406 segments, totalling 0.52 hours.

Table 6.1 Summary of polyphonic transcription evaluation sets. In the 'Isolated' column, the ✓mark indicates if, in addition to the the mixed signal, the dataset provides the singing in isolation that can be used for unaccompanied evaluation, and ✗ otherwise. The 'Recordings' column indicates the number of songs for the song-level datasets and the number of utterances for the utterance-level datasets.

Stage	Level	Eval Set	Isolated	Recordings	Size (hrs)
Adaptation	Song	MUSDB18 (train) (Rafii et al., 2017)	✓	96	6.1
		DALI (Meseguer-Brocal et al., 2018)	✗	91	6.1
	Utterance	ACOMUS (train) (Roa Dabike, 2016)	✗	3349	4.4
Evaluation	Song	Hansen (Hansen, 2012)	✓	10	0.6
		Mauch (Mauch et al., 2012)	✗	20	1.3
		MUSDB18 (test) (Rafii et al., 2017)	✓	45	3.1
		Jamendo (Stoller et al., 2019)	✗	20	1.2
	Utterance	ACOMUS (guitar) (Roa Dabike, 2016)	✗	687	0.9
		ACOMUS (piano) (Roa Dabike, 2016)	✗	787	0.9
		VSEP (eval) (Smule, Inc., 2019)	✓	410	0.6
		VSEP (valid) (Smule, Inc., 2019)	✓	406	0.5

6.2.2 Evaluation Sets

To make the results of this thesis comparable with other systems, a number of additional, preexisting datasets will be evaluated. The datasets include Hansen's dataset (Hansen, 2012), Mauch's dataset (Mauch et al., 2012), the Jamendo dataset (Stoller et al., 2019), the ACOMUS dataset (Roa Dabike, 2016), the DALI dataset (Meseguer-Brocal et al., 2018), the MUSDB18 dataset (Rafii et al., 2017) and the VSEP dataset (Smule, Inc., 2019).

From these seven datasets, the DALI dataset and the training set from the MUSDB18 corpus will be used for adapting the sung speech acoustic model to distorted sung speech. The rest of the datasets will be used for song-level evaluation (i.e. measuring ASR performance averaged across complete songs), except the ACOMUS and VSEP datasets that will be used for utterance level recognition evaluation.

The VSEP dataset cannot be used for song-level speech recognition performance as the recordings correspond to 30-second segments of the performances extracted from seconds 60 to 90. On the other hand, the DSing dataset corresponds to cover performances of popular songs – most of them made by amateur singers. Several of the recordings start with the singer speaking to introduce the video.

A brief description of these datasets is presented below.

Hansen (Hansen, 2012)

Hansen's dataset contains 10 full-length popular songs from the original 19 that com-

prised the dataset presented by Hansen (2012), e.g., *Beautiful Stranger* by *Madonna*, *I Kissed a Girl* by *Katy Perry* and *Umbrella* by *Rihanna*. The songs provided include recordings of polyphonic songs (i.e., accompanied singing) and of singing in isolation (i.e., unaccompanied singing). This subset of Hansen's dataset was used in the MIREX '2020:Singing Transcription from Polyphonic Music'[1] and the '2021:Automatic Lyrics Transcription' challenges[2]. Note that this dataset is not publicly available.

Mauch (Mauch et al., 2012)

Muach's dataset contains 20 full-length popular accompanied singing songs, including *Knowing Me Knowing You* by *ABBA*, *Ordinary World* by *Duran Duran* and *We are the Champions* by *Queen*. Mauch et al. (2012) initially presented this dataset for lyric alignment research, however, like Hansen's dataset, this dataset is also used for MIREX lyric transcription challenges evaluation. Note that this dataset is not publicly available.

ACOMUS (Roa Dabike, 2016)

The ACOMUS dataset contains 140 accompanied sung speech recordings of acoustic covers from popular songs sourced from YouTube. From the 140 recordings, 121 are accompanied by acoustic guitar, and 19 are accompanied by piano. The recordings were manually segmented and annotated, so each utterance matches a relevant lyric phrase, resulting in more than 4800 utterances. Note that ACOMUS relies on the availability of the videos on YouTube. The recordings used in this thesis were retrieved from YouTube in May 2020.

DALI (Meseguer-Brocal et al., 2018)

The DALI dataset is a collection of 5000 full-length polyphonic songs sourced from YouTube with synchronised audio, lyrics and notes. The dataset provides semi-automatically generated alignments using a teacher-student technique. 105 songs contain manually transcribed lyrics (corresponding to the ground truth set). In this thesis, only the ground truth will be used. Due to DALI relying on the availability of the videos on YouTube, 91 of the 105 songs of the ground truth set were found online when retrieving the recordings (Aug 20, 2020).

Jamendo (Stoller et al., 2019)

The Jamendo[3] set is composed of 20 freely available full-length songs of different

[1] https://www.music-ir.org/mirex/wiki/2020:Singing_Transcription_from_Polyphonic_Music
[2] https://www.music-ir.org/mirex/wiki/2021:Automatic_Lyrics_Transcription
[3] https://github.com/f90/jamendolyrics

genres (Pop, Rock, Hip-Hop, Indie, Reggae, RNB, Blues, Electronic and Metal) sourced from the Jamendo website[4]. The dataset contains the lyrics transcriptions and manual annotations indicating the start time of each word in the audio files. This dataset is the third dataset used for MIREX lyric transcription challenges evaluation.

MUSDB18 (Rafii et al., 2017)
The MUSDB18 contains 150 full-length songs recordings of different genres. It is a music separation dataset that provides independent tracks for the vocals, bass, drums and others (rest of accompaniment). This dataset was extended by Schulze-Forster et al. (2021) to include the lyric transcriptions. Only English lyrics were transcribed, resulting in 96 out of 100 tracks for the training set and 45 out of 50 for the testing set.

Table 6.1 summarises the effective size of the datasets used for adaptation and evaluation. All sets provide the mixed acoustic signal (i.e., the singer mixed with the accompaniment). However, when indicated, they also provide the singing in isolation. The isolated signal can be used to compute a lower bound speech recognition error for that set, making it possible to measure the impact of source separation error on the ASR performance. The ACOMUS (train), DALI and MUSDB18 (train) sets will be used for adapting the acoustic model from unaccompanied sung speech to separated (distorted) singing. Note that no overlap exists between the adaptation and evaluation sets.

6.3 Separated Trained Modules Evaluation

The previous section presented six datasets for polyphonic lyrics recognition evaluation, i.e., Hansen's dataset, Mauch's dataset, the MUSDB18 dataset, the Jamendo dataset, the ACOMUS dataset and the VSEP dataset. From these datasets, the first four correspond to sets containing full-length polyphonic songs. The ACOMUS and VSEP datasets are segmented datasets containing hundreds of transcribed utterances.

This section constructs the first polyphonic music lyrics transcription system investigated in this thesis using a music source separation front-end and an ASR system trained on unaccompanied sung speech. In this system, the ASR operates using separated singing, which possesses distortions resulting from imperfections in the separation. Despite the mismatch between the separated singing and the unaccompanied sung speech used to train the ASR system, it is expected to obtain recognition improvements from the separated singing compared with recognising directly from the mixture. It is hypothesised that recognising from the mixture will generate many deletions errors (missed words) due to the accompaniment

[4]www.jamendo.com

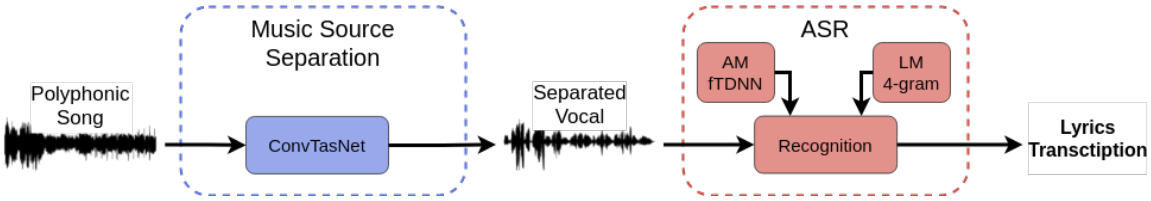


Figure 6.2 Diagram of the polyphonic lyrics transcription system. The system starts by estimating the singing segments from the mixture. Next, the lyrics are recognised from the estimated vocal.

masking effect and that the separation front-end will assist the ASR system in identifying better and recognising better those missed words. However, distortions in the sung signal may lead to an increment of insertion and substitution errors.

Due to the mismatch between the separated singing and singing used to train the acoustic model, from now on, the system used in this section will be referred to as a 'mismatched system'.

The mismatched system is composed of two independent modules connected in a pipeline: the music source separation and ASR module. Figure 6.2 shows a diagram of the transcription system.

For the music source separation module, a Conv-TasNet (Luo and Mesgarani, 2019) architecture was trained by using the Asteroid PyTorch-based audio source separation toolkit (Pariente et al., 2020). The model was trained using the largest 'English+non-English' training dataset (details of the dataset in Table 5.2 in Chapter 5.2 page 129) for 50 epochs using two GPUs, learning rate 5e-4 and the Adam optimisation algorithm (model reported in row 7 of Table 5.7 in Chapter 5, page 147).

For the ASR module, a hybrid DNN-HMM ASR system was built using the Kaldi ASR toolkit (Povey et al., 2011). Acoustic modelling consisted of a factorised time-delay neural network (Povey et al., 2018) with a lattice-free maximum mutual information loss function (Povey et al., 2016). Using the DSing30 dataset, the model was trained using a two frames context vector consisting of 40 MFCC, 4 pitch-based and 4 voice quality features, plus 100 i-vectors (Dehak et al., 2011). Language modelling consisted of a 4-gram model trained on a large collection of lyric texts sourced from the Lyrics Wiki website. This ASR system corresponds to the 'LN + Voice Quality (VQ)' experiment trained on DSing30 dataset (model reported in Table 4.5 in Chapter 4, page 113).

For the ASR decoding, the LM-weight and words-insertion-penalty Kaldi's decoding parameters used correspond to the best values for the respective dataset.

In the case of the utterance-level datasets, the evaluation proceeds as follows. First, utterances are generated from the polyphonic recording using the endpoints annotations.

Table 6.2 Recognition performances using the mismatched system. Scores are reported in terms of WERs. The Hansen's, Mauch's and Jamendo datasets also include the results obtained in MIREX 2020[6]. The ACOMUS dataset includes the performance reported in the MSc dissertation (Roa Dabike, 2016).

Level	Dataset	Previous Results	Evaluation		
			Clean	Mixture	Separated
Song	Hansen	83.4%	45.0%	86.1%	77.7%
	Mauch	85.0%	–	85.8%	78.4%
	MUSDB18 (test)	–	59.5%	88.2%	79.2%
	Jamendo	86.7%	–	83.6%	79.7%
Utterance	ACOMUS (guitar)	89.9%	–	65.3%	57.4%
	ACOMUS (piano)	84.6%	–	50.5%	40.2%
	VSEP (eval)	–	33.1%	82.8%	66.2%
	VSEP (valid)	–	31.3%	83.1%	68.3%
Average		–	–	78.2%	68.4%

Next, singing is estimated from each utterance. Then, the ASR operates on independent utterances to recognise the sentence text.

For the song-level datasets, the singing from the whole song is first estimated using the music source separation model. Then, the transcription is performed from the song-length singing segment. Note that no actions are taken to automatically discard non-singing areas before passing the recording to the recogniser. A voice activity detector trained to detect singing segments from the estimated singing signal may help to filter out non-singing areas. However, training such a model is outside of the scope of this thesis.

An initial evaluation using a mismatched system was submitted to the first polyphonic music singing transcription challenge (Roa Dabike and Barker, 2020) – the 'MIREX 2020:Singing Transcription from Polyphonic Music'[5]. The MIREX submission employed the same ASR system used in this thesis. However, the music source separation model employed was trained on an uncorrected version of the DAMP-VSEP, i.e., several of the challenges of the dataset were not addressed before training the music separation model.

Table 6.2 shows the performances of the mismatched system on the different evaluation datasets. For Hansen's, Mauch's and the Jamendo datasets, the scores obtained in the MIREX 2020 challenge are included for comparison. For the ACOMUS dataset, previous results correspond to the results reported in the MSc dissertation (Roa Dabike, 2016). All datasets with previous results obtained an improvement by the present system.

[5]https://www.music-ir.org/mirex/wiki/2020:Singing_Transcription_from_Polyphonic_Music

Table 6.3 Comparison of the WER details between the mixture and separated per evaluation dataset. The 'Audio' column indicates if the recognition was performed from the mixture signal or the separated singing. The columns 'Corr', 'Sub', 'Ins' and 'Del' show the number of corrects, substitutions, insertions and deletions words. The last column 'Total' shows the total number of words in the dataset.

Dataset	**Audio**	**# Words**	**Corr**	**Sub**	**Ins**	**Del**	**WER**
Hansen	Mix	2840	14.3%	23.6%	0.4%	62.1%	86.1%
	Separated		25.2%	37.5%	2.9%	37.3%	77.7%
Mauch	Mix	5061	14.6%	31%	0.4%	54.4%	85.8%
	Separated		25.7%	40.7%	4%	33.7%	78.4%
MUSDB18 (test)	Mix	11,885	12.3%	29.2%	0.5%	58.5%	88.2%
	Separated		24.7%	51.2%	3.9%	24.1%	79.2%
Jamendo	Mix	5688	17.3%	40.8%	0.9%	42.0%	83.6%
	Separated		23.3%	49.8%	3.1%	26.8%	79.7%
ACOMUS (guitar)	Mix	4905	39.3%	39.0%	4.6%	21.7%	65.3%
	Separated		51.4%	40.7%	8.8%	8.0%	57.4%
ACOMUS (piano)	Mix	4734	54.5%	31.7%	5%	13.8%	50.5%
	Separated		66.7%	27%	6.9%	6.3%	40.2%
VSEP (eval)	Mix	2704	18.2%	32.3%	2.5%	49.5%	84.3%
	Separated		39.6%	49%	6.5%	11.4%	66.9%
VSEP (valid)	Mix	2816	15.9%	23.6%	1.1%	60.5%	85.2%
	Separated		37.1%	49.4%	6.4%	13.5%	69.4%
Total	Mix	40,633	22.2%	32.0%	1.7%	45.8%	79.5%
	Separated		34.6%	44.4%	5.0%	21.0%	70.4%

Performing the sung speech recognition on the mixture signal without activating the source separation model (i.e., the upper bound score), the average error across all datasets was 78.2% WER. The ACOMUS sets showed the lowest error of 65.3% WER and 50.5% WER for the guitar and piano sets. These results are better than those achieved with other datasets suggesting that the singing in ACOMUS is less masked by the accompaniment than in the other datasets, allowing the singing to be more intelligible. The fact that there is less speech masking in the ACOMUS dataset was expected due to the less sophisticated background instrumentation compared with the other sets. Excluding the ACOMUS sets, the average error from the mixtures increased from 78.2% WER to 84.9% WER.

Table 6.4 Transcription sample of one segment from *Eternal Flame* by *The Bangles*. The first row shows the reference transcription. The second and third line show the hypothesis made by the ASR system from the mixture and the separated singing. The '***' symbols indicates a deletion error.

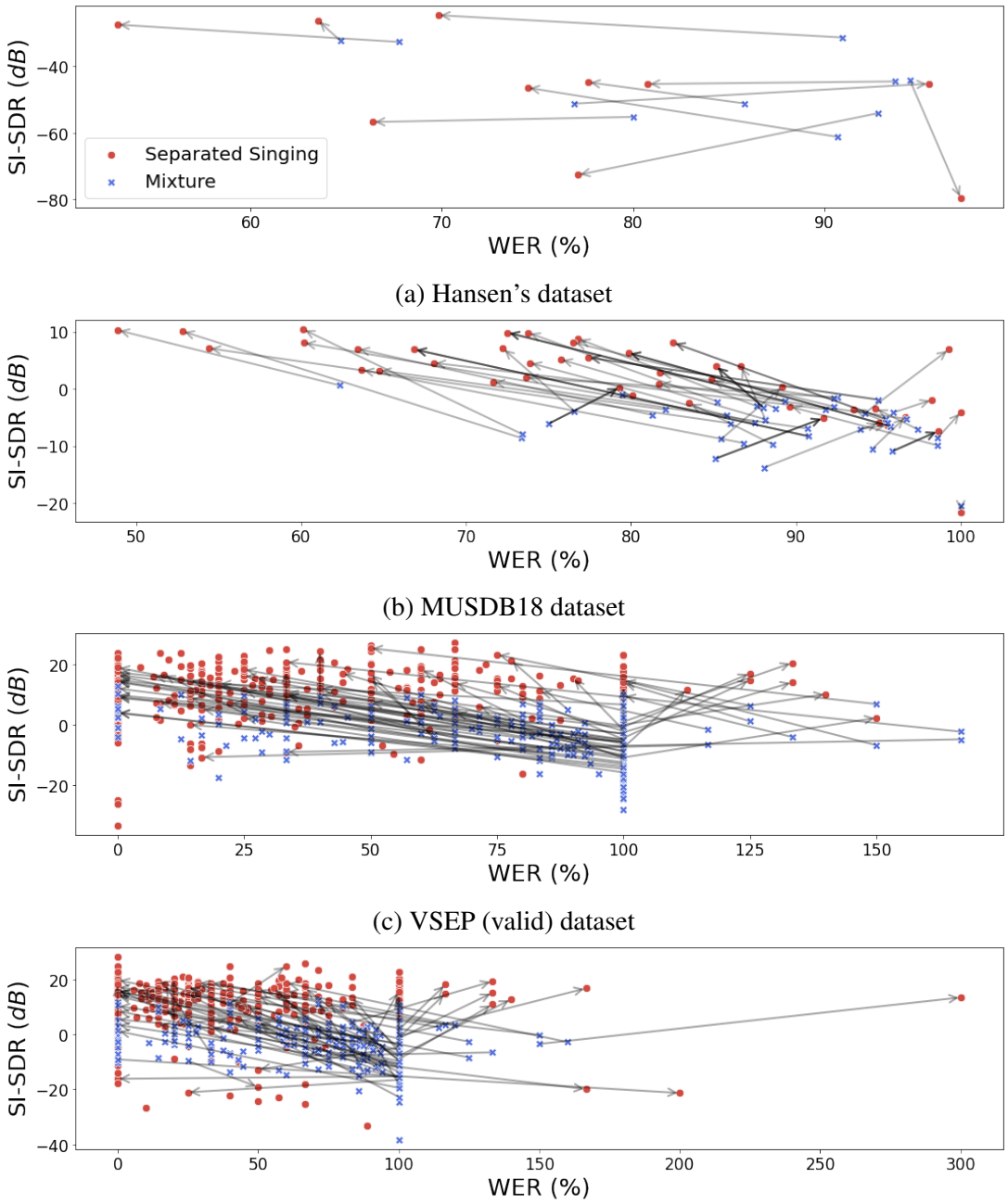
Reference	SAY	MY	NAME	SUN	SHINES	THROUGH	THE	RAIN
Mixture	SAY	MY	NAME	***	***	***	***	***
Separated	HEY	MY	NAME	***	SUNSHINE'S	THROUGH	THE	RAIN

After activating the source separation model, the average error across all datasets reduced from 78.2% WER for the mixtures to 68.4% WER for the separated singing (an average of 10% of absolute error reduction per dataset). The significance of the improvements was evaluated using the dependent two-sided t-test statistic test for paired samples. A significant increment in performance was obtained for all datasets ($p < .05$), except for the Jamendo ($p = .09$), MUSDB18 ($p > .1$) and ACOMUS (guitar) ($p = .06$) datasets which obtained a non-significant improvement.

Table 6.3 compares the sources of error between the recognition results from the mixture signal (accompanied sung speech) and the separated sung speech. The values are relative to the total number of words. The deletion errors were consistently halved across all datasets when transcribing from the separated sung speech. This means that words from the mixtures missed by the ASR system are identified when looking at the separated singing. The increment of substitution errors when transcribing from the separated singing may be related to the decrease in deletion errors, i.e., the now identified words may be transcribed erroneously.

Table 6.4 shows a snapshot of the transcription from the song *Eternal Flame* by *The Bangles* made by the ASR system (sample from Mauch's dataset). In this snapshot, the hypothesis from the mixture resulted in three correct words and five deletions (i.e., 62.5% WER). However, using the separation preprocess, the deleted words are recovered, reducing the deletions to one but increasing the errors in one insertion and one substitution, totalling three errors (i.e., 37.5% WER). However the errors in the separated singing are the result of ambiguity between 'SUN SHINES' ([sʌn] [shɑɪnz]) and 'SUNSHINE'S' ([sʌnshɑɪnz]). The WER counts two errors in the mixture and the separated singing. However, the phonetic transcription shows that the words SUN SHINES were correctly recognised in the separated signal. The phone error rate may be a more useful measure in many applications.

On the other hand, due to the distortion in the separation, correctly recognised words from the mixture can be misrecognised from the separated singing, such as the first word in the example where the sound [s] from 'SAY' is changed to [h] from 'HEY'.



(a) Hansen's dataset

(b) MUSDB18 dataset

(c) VSEP (valid) dataset

(d) VSEP (eval) dataset

Figure 6.3 Tendency of the WER on relation with the SDR. The blue crosses represent the samples when recognising from the mixture signal. The red dots represent the scores for the separated singing. The arrows show the direction of the scores' variation. (a) and (b) shows the tendency of the songs in Hansen's and the the MUSDB18 dataset, respectively. (c) and (d) show the tendency for the utterances in the VSEP (eval) and VSEP (valid) datasets. Samples with mixture WER greater than 250% were excluded to improve the visibility.

Figure 6.3 shows four scatter plots with the relationship between the SI-SDR and WER scores from Hansen's, MUSDB18, VSEP (valid) and VSEP (eval) datasets. The crosses represent the scores computed from the mixture, i.e., the vocal SI-SDR and WER from the mixture. The dots represent the SI-SDR and WER from the separated singing. Arrows are included to show the tendency of the variations from the mixture to the separated singing scores. To improve the visualisation of the plots, a maximum of 50 arrows are included. Outliers with a mixture's error greater than 250% WER are excluded from the plots. As expected, arrows from all four plots tend to point towards the top left of the plots, i.e., from higher WER to lower WER and from lower to higher SI-SDR. However, the slope of the arrow varies in some cases depending on the characteristics of the mixtures. This is particularly true for Hansen's dataset that despite the slight SI-SDR variation from several samples, the WER tends to reduce. The SI-SDR score was computed using the full extension of the recordings, and songs with long non-singing segments may affect the score. This is because the reference will be silent, and the estimation will be pure distortion for those segments. Note that the SI-SDR scores for these datasets were possible to compute because they provide the singing in isolation needed for the computation.

6.4 Distorted Sung Speech Acoustic Modelling Adaptation

The previous section evaluated the effectiveness of a source separation front-end for improving the performance of accompanied sung speech ASR when trained on unaccompanied song recognising separated singing using an ASR system trained on clean speech. Eight evaluation sets (namely Hansen's, Mauch's, MUSDB18 (test), Jamendo, ACOMUS (guitar), ACOMUS (piano), VSEP (eval) and VSEP (valid) datasets) were employed to test the system under different singing conditions – four sets with full-length songs and four segmented into utterances. In this system, the lyrics were recognised from estimated singing using a music source separation model, resulting in an absolute improvement of about 10% WER across all datasets compared with recognition performances obtained when operating directly on the mixture, decreasing from an average of 79.0% WER to 68.9% WER. Perhaps, the most significant improvement when operating on the separated singing compare with the mixture is the reduction in the number of deletion errors, i.e. meaning that words missed by the ASR system when processing the mixture are now identified. However, the ASR misclassified many of the recovered words, increasing the substitution error.

The system presented above works on mismatched singing conditions, i.e., the ASR system was trained on unaccompanied singing, but it is being evaluated using singing estimated from polyphonic songs using a music source separation model. It is hypothesised

that adapting the acoustic model to separated singing conditions using a music source separation model will increase the classification performance by reducing substitution errors. This section will extend the system on mismatched singing by adapting the acoustic model to recognise separated singing better using the DALI, ACOMUS and MUSDB18 (train) datasets. The different mixtures' characteristics of these datasets may help the model to adapt to different distortions, generalising better.

6.4.1 Adaptation Methodology

The adaptation procedure uses transfer learning ideas. In particular, this thesis employ weight transfer (Ghahremani et al., 2017) where the general idea is that a model is pretrained on a large dataset. Then, using a smaller target dataset, a number of layers are frozen and only the parameters of the layers nearer that output are updated. A comprehensive study on transfer learning is presented by Zhuang et al. (2021). Wang and Zheng (2015) presents a good survey in transfer learning applied to speech and language processing.

Ghahremani et al. (2017) showed better performances using a weights transfer technique by adapting the DNN model using a very small learning rate for the early layers and a larger learning rate for the output layer.

Following the procedure presented by Ghahremani et al. (2017), the adaptation takes the pretrained unaccompanied sung speech acoustic model presented in Chapter 4. The output layers are trained using a learning rate of 5e-5, corresponding to half of the last learning rate used during the unaccompanied singing training. Early layers are updated using a learning rate of 25% of the output's learning rate. The model is then continued to be trained for four more epochs.

For the adaptation, the DALI and MUSDB18 (train) datasets are segmented into utterances by using the timings from the transcription, matching the structure of the ACOMUS dataset. Having the three dataset at utterances levels simplifies the combination of them for the adaptation procedure. Table 6.5 shows the details of the adaptation datasets. Note that the size corresponds to the sum of the utterances duration, which is expected to be less than the duration of the whole songs previously reported in Table 6.1.

Three adapted acoustic models will be generated separately by performing the adaptation with each adaptation dataset. Results from these three systems will show the independent effect in the recognition by the datasets. Then, the adaptation datasets will be merged into one dataset to leverage the independent benefits of each dataset.

Note that one can train an acoustic model from scratch using separated vocals. A large annotated dataset of separated vocals would be necessary in this case. This can be done by estimating the singing from a large polyphonic singing dataset (e.g., the Dali dataset).

Table 6.5 Details of the separated singing adaptation sets. The 'Utt. Size' column corresponds to the size of the utterances in hours.

Dataset	Recordings	Utterances	Utt. Size (hrs)
MUSDB18 (train)	96	1940	3.8
DALI	91	4962	3.6
ACOMUS	120	3349	4.4

However, different audio source separation models may result in different kinds of distortions, with some of the distortions not appropriate for acoustic model training. The adaptation procedure adopted here ensures that the acoustic model starts from robustly learned patterns from clean singing before adapting to distorted conditions.

6.4.2 Adaptation Results

The adapted model was evaluated using the same evaluation sets used to assess the mismatched system, ensuring a fair comparison between the mismatched and the adapted systems. Table 6.6 reports the recognition performances from the system adapted using the different datasets. The Mismatch shows the recognition performances obtained from the separated singing by the mismatched system (i.e., these are the same scores from the column 'Separated' in Table 6.2).

Using the ACOMUS and DALI datasets resulted in a reduction of the recognition errors for almost all the evaluation datasets. Adapting the AM using the ACOMUS dataset resulted in a 3% average WER reduction, from 68.6% WER for the mismatched system to 65.9%. A slight improvement of 0.3% average WER across all datasets was obtained with the DALI adaptation. For the VSEP (eval) dataset, the adapted system's error was higher than that of the mismatched system, regardless of the adaptation set. Perhaps, in this case, the characteristics of the singing distortion from the adaptation sets do not match with the characteristics in the VSEP (eval) set.

Adapting using the MUSDB18 (train) dataset resulted in poorer performances for all evaluation sets. Inspection of several samples from MUSDB18 (train) showed that the source separation model failed to separate the singing in this dataset. Using a data selection process may help to filter out any highly distorted separated samples before performing the adaptation; however, this is out of the scope of this thesis. Due to the poor adaptation results using the MUSDB18 (train) datasets, this dataset was excluded from the adaptation from the merged datasets.

Table 6.6 Recognition performances using the distorted singing adapted systems. Scores are reported in terms of WERs.

Level	Dataset	Mismatched	Adapted System			
			ACOMUS	DALI	MUSDB18	ACOMUS + DALI
Song	Hansen	77.7%	76.1%	76.9%	87.3%	**74.2%**
	Mauch	78.4%	76.2%	76.4%	86.1%	**75.0%**
	MUSDB18 (test)	79.2%	74.8%	75.1%	84.8%	**73.8%**
	Jamendo	79.7%	77.1%	79.1%	85.1%	**76.7%**
Utterance	ACOMUS (guitar)	57.4%	50.1%	56.0%	81.0%	**49.9%**
	ACOMUS (piano)	40.2%	36.0%	44.2%	77.1%	**34.8%**
	VSEP (eval)	**66.9%**	68.4%	70.0%	86.8%	67.1%
	VSEP (valid)	69.4%	68.1%	68.4%	88.7%	**65.9%**
Average		68.6%	65.9%	68.3%	84.6%	**64.7%**

Table 6.7 Accumulated error details across all evaluation datasets.

System	# Words	Corr	Sub	Ins	Del	WER
Mismatched	40,633	14,065	18,034	2034	8534	70.4%
ACOMUS + DALI		15,280	14,224	1452	11,129	66.0%

Further improvements were obtained by using the combined effect of adapting the acoustic model using the ACOMUS and DALI datasets. The average WER across all datasets reduced from 68.6% WER from the mismatched system to 64.7% WER. Table 6.7 shows the accumulated error details across all evaluation datasets. The improvement in the adapted model was obtained due to a decrease in substitution and insertion errors. Note that there is also a slight increment in deletion errors.

Coming back to the unaccompanied sung speech evaluation set, the adapted model resulted in a performance reduction, increasing the error from 19.6% WER to 34.9% WER. Adapting the acoustic model to separated speech proved beneficial in improving the recognition performances for separated singing. However, these improvements come at the cost of performance reduction for unaccompanied sung speech. This reduction in performance is not unexpected as the model was adapted to recognise distorted sung speech better, conditions that differ from the clean sung speech.

Table 6.8 summarises the results from all the MIREX 2020 challenge submissions. It includes the performances from the 'ACOMUS + DALI' adapted system for comparison. The adapted system resulted in a 10% WER reduction for Hansen's, Mauch's and the Jamendo datasets. However, these performances are still lower than the best performances obtained by

Table 6.8 Comparative performance of the mismatched system with the MIREX 2020:Lyrics Transcription. The RB1 system corresponds to the performances reported as previous results in Table 6.2.

System	Hansen	Mauch	Jamendo
GGL1 (Gao et al., 2020a)	**47.9%**	**47.3%**	**61.0%**
GGL2 (Gao et al., 2020a)	47.9%	49.5%	62.5%
DDA2 (Demirel et al., 2020b)	74.8%	75.4%	72.1%
DDA3 (Demirel et al., 2020b)	77.4%	80.7%	73.1%
RB1 (Roa Dabike and Barker, 2020)	83.4%	85.0%	86.7%
ACOMUS + DALI	74.2%	75.0%	76.7%

Gao et al. (2020a) of 47.9% WER, 47.3% WER and 61.0% WER, respectively. Gao et al. (2020a) implemented a factorised TDNN acoustic model using the Kaldi toolkit, a similar architecture to the one used in this thesis. They used 3913 songs from DALI available to download in Singapore, plus a proprietary set of 517 popular songs for training data. However, acoustic modelling was performed directly on the mixed audios instead of separating the singing from the accompaniment. Their results suggest that given enough data, the acoustic models may be able to classify phonemes under different masking conditions.

6.5 Summary

This chapter evaluated the recognition of accompanied sung speech by integrating the best singing separation model from Chapter 5 with the unaccompanied sung speech ASR system resulting from the work in Chapter 4 (**RQ.6**).

Several evaluation sets were used to evaluate the recognition performances under different mixture conditions, namely Hansen's (Hansen, 2012), Mauch's (Mauch et al., 2012), Jamendo (Stoller et al., 2019), MUSDB18 (Rafii et al., 2017), ACOMUS (Roa Dabike, 2016) and VSEP (Smule, Inc., 2019). Hansen's, Mauch's, and the Jamendo datasets correspond to the three datasets used for evaluation in different MIREX challenges. The MUSDB18 dataset corresponds to the transcribed tracks from the MUSDB18 test set (Schulze-Forster et al., 2021). The ACOMUS set contains two evaluation sets differing on the background accompaniment: acoustic guitar and piano. The VSEP dataset corresponds to a transcribed version of the validation and evaluation sets used in Chapter 5.

Initial evaluation recognised the lyrics from the mixture recordings, resulting in 79.0% WER. A WER improvement of about 10% absolute was obtained by adding a speech separation front-end to the speech recogniser, reducing the average error to 68.6% WER. However,

recognition errors are still very high compared with unaccompanied performances. For example, in the particular case of Hansen's dataset, where we have access to the unaccompanied version of the songs, the error increases from 45.4% WER for the unaccompanied singing to 77.7% WER for the separated singing. Detailed analysis of the errors showed that applying speech recognition to separated singing results in many substitution errors. This was not unexpected because the acoustic model is making classification decisions on signals that have acoustic characteristics different from the ones used for training.

To address the mismatch problem, the unaccompanied singing acoustic model was adapted to distorted singing by using separate speech adaptation data and applying standard transfer learning techniques. Experiments were performed using three different adaptation sets: the ACOMUS training set, the MUSDB18 training set and the ground truth set from the DALI dataset (Meseguer-Brocal et al., 2018). It was found that recognition improvements were obtained when using the ACOMUS and DALI datasets, but a significant performance degradation occurred when adapting the acoustic model using the MUSDB18 dataset. Inspections of several samples showed that the source separation model performed poorly on the MUSDB18 dataset, resulting in higher distortions, which, in turn, introduced noise to the acoustic model affecting the recognition performances.

To leverage the independent benefit of adapting using the ACOMUS and DALI datasets, a last adaptation was performed by merging these datasets. The combined effect resulted in further recognition improvements reducing the average error across all evaluation datasets from 68.6% to 64.7%. This improvement was the result of reduction of insertions and substitution error, but at the cost of increased deletion errors.

Adapting the acoustic model trained on unaccompanied sung speech to separated singing proved beneficial for increasing separated singing recognition performances. However, as one may expect, the performance will highly depend on the characteristics of the accompanying singing recording and the capacity of the audio source separation model to separate the vocal from those recordings. Additionally, the improved performance on separated singing recognition comes at the cost of reducing unaccompanied singing recognition performance.

Chapter 7

Conclusions and Scope for Future Work

This thesis has investigated whether new deep learning techniques designed for automatic spoken speech recognition can be adapted to work for sung speech recognition. Recognising the lyrics from a sung signal poses several challenges that make this task more challenging than that of *spoken* speech recognition. Some of these challenges originate from the fact that, in sung speech, the intelligibility of the message is often of secondary importance to the artistic performance, resulting in several acoustic differences between both types of speech. This problem becomes even more challenging in the case of accompanied singing because the accompaniment can mask the speech signal. In a musical mixture, the different sources in the signal are designed to complement each other and hence can be hard to separate. The problem was divided into three more constrained sub-tasks to tackle the various challenges. The first task focused on unaccompanied sung speech recognition employing hybrid ASR architectures. The second task focused on separating the singing segment from the accompaniment's musical background. The last task considered the integration of the outcomes from the previous tasks to recognise the lyrics from a polyphonic song.

Six research questions were presented dealing with the tasks abovementioned. The questions will now be restated, along with a summary of how this thesis addressed these questions.

(RQ.1) What are the differences in the speech production mechanisms between sung and spoken speech, and how are these differences reflected in the acoustic signal?

Chapter 2 presents a study of how sung and spoken speech differ in the use of speech production mechanisms. Understanding how both speech styles differs helped contextualise the challenges of sung speech. Perhaps, the most evident differences between sung and spoken speech are larger pitch ranges, greater energy levels and larger syllables duration

presented in the sung speech. However, other less apparent differences may be equally significant. For example, when singing at a high pitch, singers tend to increase the jaw opening to tune the first formant – adjusting their frequency – to move it closer to the pitch frequency, increasing their energy. This process results in a reduction of the distance between the first two formants (de Julián, 2016). Other less apparent differences include the pitch variation within a vowel, the reduced glottal noise in sung speech and the formation of a special formant at 3500 Hz in operatic male singers.

(RQ.2) What is the masking effect of the musical accompaniment, and how does this masking impact the intelligibility of the sung speech signal?

The music structure highly influences singing. Composers argue that strong beats should match stressed syllables, and a bar should encompass a lyric sentence (Perricone, 2018), resulting in onset overlap between the singing and accompaniment. Additionally, Western music favours the twelve-tone equal temperament scale (Deutsch, 2013), resulting in frequencies from different sources overlapping. Chapter 2 analysed the local masking effect of the background accompaniment over the sung speech by employing the 'glimpse proportion' measurement. Results showed that the level of masking is highly dependent on the music genre. For example, Pop/Rock music SNR ranges between -16.5 dB and 1.35 dB with an average of -4.8 dB, representing a 23% of glimpse proportion.

(RQ.3) What datasets exist that can be shaped for modelling sung speech recognition and vocal source separation?

To deal with the lack of sung speech corpus for lyrics transcription modelling, Chapter 4 presented work to shape the novel karaoke singing DAMP-MVP dataset (Smule, Inc., 2018) to sung speech recognition. This work resulted in the DSing dataset (Roa Dabike and Barker, 2019). DSing provides three increasingly larger training sets: DSing1, DSing3 and DSing30, and two test sets. Dsing1 is composed of 15 hours of singing from singers located in Great Britain. DSing3 extends DSing1 with 30 hours of singing from Australia and the US, totalling 44 hours of singing. DSing30 extends DSing1 further by including all English singing performances from DAMP-MVP, totalling 149 hours of singing.

Since 2018, the MUSDB18 dataset (Rafii et al., 2017) has been the main dataset used for music separation. It is a well-constructed dataset for separating four different sources, i.e., the vocals, drums, bass and other instrumentation. However, recent research has shown that this dataset's size remains insufficient and more extensive datasets are needed. The recent release of the DAMP-VSEP corpus (Smule, Inc., 2019) by Smule enables the option of using a large

number of accompanied singing performances for source separation evaluation. However, this dataset presents several challenges that make it challenging to use out-of-the-box. This thesis presents work dealing with these challenges, resulting in three increasingly large training sets. The first set comprises 14 hours of solo-singing English performances. The second set extends the previous set using 10 hours of duets-English performances converted to singles. The third and largest training set extends the first set by using all single performances from different languages, totalling 66 hours of training data.

(RQ.4) How can we best re-adapt spoken speech ASR systems to meet the demands of unaccompanied sung speech recognition? In particular, how much benefit can be gained by re-considering the features on which the acoustic models are trained to capture the musical properties of the signal better?

Chapter 4 evaluated the effect of using musically-motivated features for acoustic modelling. A hybrid ASR baseline system was first constructed using a TDNN-F acoustic model (Povey et al., 2018) trained using 40 MFCCs, 100 i-vectors and LF-MMI loss function (Povey et al., 2016), and a 4-gram language model trained on seven million lyrics sentences. Three baseline systems were trained using the three training sets from the novel DSing corpus (Roa Dabike and Barker, 2019). The models trained obtained performances of 38% WER, 24.4% WER and 19.88% WER when trained using the DSing1, DSing3 and DSing30 dataset, respectively.

Then, the baseline systems were extended by using different musically-motivated features, such as pitch, voiced degree, and voice quality measurements. These features were computed at frame level and used to extend the MFCC feature vector. For the smallest DSing1 dataset, the model obtained a statistically significant error reduction from 38.14% to 36.7% WER. A similar reduction was obtained from the larger DSing3 dataset, reducing from 24.4% to 23.76% WER. For the largest DSing30 dataset, a non statistically significant error reduction of 0.2% WER was obtained. It was hypothesised that, given sufficient data, models could learn voice source phonetic cues in a less direct manner, e.g., via the temporal dynamic of the MFCCs. However, it was found that musically-motivated features help acoustic models to normalise singing sounds at high pitches.

(RQ.5) How can spoken source separation approaches be adapted to exploit the musical signal's constraints better? In particular, can recent speaker embedding approaches be used to characterise and filter out instrument-specific musical accompaniment?

This question was addressed in Chapter 5. This chapter first surveyed different audio source separation techniques trained and evaluated on the MUSDB18 corpus (Rafii et al., 2017). From these systems, the Conv-TasNet (Luo and Mesgarani, 2019) was selected for constructing the baseline. The Conv-TasNet is the only system from this survey that was not originally designed for music separation. However, Défossez et al. (2019) reported good performance adapting this model to the music scenario. Additionally, the modular structure of Conv-TasNet (i.e., it is composed of an encoder, separator and decoder networks) makes it possible to integrate different extensions without altering any of the encoder, separator or decoder.

The baseline system was trained using the processed version of the DAMP-VSEP dataset (Smule, Inc., 2019). The best performances was obtained when training the model with the largest training set (66 hours), resulting in an increment of the vocal SI-SDR of 17.43 dB, from -5.14 dB input SI-SDR to 12.29 dB.

Then, the baseline system was extended to investigate the effect of presenting the separator network with embedding computed from the background accompaniment. For this, the Conv-TasNet architecture was extended by introducing an instrument embedding module, ensuring that neither the encoder nor the separator was modified. Two different embeddings were explored. The first system consisted of the VGGish (Hershey et al., 2017), a pretrained network trained on several audio events. The second system consisted of the X-vectors network (Snyder et al., 2018) trained on several separated musical background accompaniments. The introduction of the embeddings resulted in a best vocal ΔSI-SDR of 17.47 dB VGGish and 17.58 dB X-vectors. However, these results were obtained from a known solo-background segment, and a different evaluation strategy was needed for practical applications. Therefore, a cascade evaluation of two steps was implemented. The first step estimates the background from the mixture, and then embeddings are computed from the separated background. The use of the two-step strategy resulted in a loss of the benefit introduced by the embeddings.

(RQ.6) How severely does the separation distortion impact the speech recognition performance, and to what extent can the recognition systems be adapted to accommodate distortion?

Chapter 6 investigated the problem of recognising the lyrics from an accompanied sung speech signal. Six evaluation datasets were employed to evaluate the performances under different accompanied conditions: Hansen's (Hansen, 2012), Mauch's (Mauch et al., 2012), Jamendo (Stoller et al., 2019), MUSDB18 (Rafii et al., 2017), ACOMUS (Roa Dabike, 2016) and VSEP (Smule, Inc., 2019).

First, an accompanied recognition system was constructed using a music source separation front-end and an unaccompanied ASR system back-end. Recognition performances were computed with and without activating the source separation module. These results showed that, using ASR sung speech recognition trained on unaccompanied singing, recognising the lyrics from separated singing resulted in an average WER improvement of about 10% across all evaluation sets over recognising directly from the mixture, reducing from 79% WER to 68.6% WER. Most of the improvements come from reducing deletion errors. However, several substitution errors occur due to the mismatch between the separated singing and the unaccompanied sung speech used to train the acoustic model.

Then, using separated singing from three adaptation datasets (i.e., the training set from ACOMUS, the training set from MUSDB18 and the ground truth from the DALI dataset (Meseguer-Brocal et al., 2018)), the acoustic model was adapted to distorted conditions using 'weight transfer' learning approaches. The leverage effect of adapting using the ACOMUS and DALI dataset resulted in further improvement from 68.6% WER to 64.7% WER. However, these improvements come at the cost of reducing recognition performances of unaccompanied sung speech.

7.1 Scope for Future Work

This thesis dealt with the sung speech recognition challenge by separating the problem into three more constrained subtasks: unaccompanied sung speech recognition, music audio source separation, and accompanied sung speech recognition. This approach allowed the focused study of the subtasks' challenges (challenges described in the corresponding chapters). However, this approach implies that for each subtask specific constraints must be defined.

Chapter 4 investigated the use of musically-motivated features to improve ASR performances. These features were extracted directly from the singing signal. However, several

musical characteristics are better extracted from the accompaniment than from the singing segment, such as beat tracking and tempo estimation.

Chapter 5 presented a background embedding network to inform a music audio source separation model with the musical accompaniment characteristics. This approach resulted in slight but significant improvements when applied in controlled conditions. However, the improvement was lost when it was attempted to employ the background embedding in practical applications, limiting its application.

Chapter 6 presented an accompanied sung speech recognition system composed of the best audio source separation model obtained in Chapter 5 and the best ASR system obtained in Chapter 4. In this connected system, the separation model estimates the singing segment before it is presented to the ASR system. This chapter employs an adaptation technique to adapt the ASR to separated singing conditions. However, as was shown in Chapter 6, adapting the ASR system using separated singing is not always possible. The adaptation depends on the characteristics of the adaptation dataset and the ability of the separation model to estimate the singing from that dataset.

Given the constraints described above, a number of suggestions for future investigation are presented below:

Background informed ASR

The accompanied sung speech recognition system developed in Chapter 4 can be further improved by informing the acoustic model with background characteristics. For example, it was shown that the use of beat-based features computed from the singing reduced the performance of the recognition from unaccompanied sung speech. It is hypothesised that the failure of these features resulted from the beat extractor being unable to detect beats from low proficiency singers correctly. However, using a source separation model, the background could be estimated and the beat-based features extracted from the background estimation.

Singing Data Augmentation

The DSing dataset presented in this thesis provides up to 149 hours of unaccompanied sung speech recordings with their corresponding transcriptions. However, this dataset's size remains smaller than spoken datasets like LibriSpeech (Panayotov et al., 2015), which contains 1000 hours of training data. Simulated singing data generated by adjusting the pitch and duration of spoken speech recordings can be a mechanism to increase the singing data, therefore, increasing acoustic model performances. Kruspe (2015b) has presented work in

this line with small improvements and, more recently, Zhang et al. (2021) with promising results.

Different procedures to apply the background embedding

The use of background embedding, presented in Chapter 5, resulted in variable performances depending on the procedure used to employ the embeddings during inference. During training, the embedding was computed from the first ten seconds of the background source. This ensured that all training segments from the same performances used the same embedding vector. This procedure resulted in a small but significant improvement in performance over the system without the embedding. However, for inference of polyphonic songs, the training procedure is not applicable as there is no guarantee of having ten second solo-instrumental segment. Therefore, a two-pass procedure was employed, where, in the first step, the background is estimated using the baseline model. The embedding vector is then computed from the estimated background in the second pass. The benefit of using the embedding vector was no longer present when using the two-pass procedure. It is hypothesised that using variable length of instrumentation (the length of the mixture used) weakened the computation of the embedding vector. Employing a different procedure for inference may help to retain the benefits. For example, by using a voice activity detection one can estimate the solo-singing segments and join them together to compute the embedding. Another option is to use use the estimated background from the whole song and select a random ten second segment to compute the embedding.

Multitask training

The polyphonic sung speech recognition system presented in Chapter 6 is based on a traditional ASR architecture where the system is composed of several models trained separately, such as the audio source separation model, the acoustic model and the language model. However, given enough data, novel end-to-end ASR system are able to jointly optimise the recognition and the separation components. An audio source separation model can benefit from the lyrics transcriptions when the separation is jointly optimised with lyrics alignments (Schulze-Forster et al., 2021). Similarly, an ASR system trained on polyphonic music can benefit by jointly optimising the recognition error and the vocal separation. Further, constructing a transcribed sung speech dataset with pitch annotations could be used to jointly optimise the recognition and pitch estimations, which may be particularly beneficial for high pitch singing recognition.

References

Abberley, D., Renals, S., and Cook, G. (1998). Retrieval of broadcast news documents with the THISL system. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 1998)*, pages 3781–3784.

Abercrombie, D. (1967). *Elements of General Phonetics*. Edinburgh University Press.

Antipolis, S. and Woodland, P. C. (2001). Speaker Adaptation for Continuous Density HMMs: A Review. *ITRW on Adaptation Methods for Speech Recognition*, pages 11–19.

Arnold, A. E. and Kramer, J. C. (2016). *What in the world is music?* Routledge, New York, NY.

Asgari, M. and Shafran, I. (2013). Improving the accuracy and the robustness of harmonic model for pitch estimation. In *Proceedings of Interspeech 2013*, pages 1936–1940.

Atal, B. S. and Schroeder, M. R. (1967). Predictive coding of speech signals. In *Proceedings of IEEE Conf Commun Process 1967*, pages 360–361.

Baayen, R. H., Piepenbrock, R., and Gulikers, L. (1995). "The CELEX Lexical Database" (Linguistic Data Consortium, University of Pennsylvania, Philadelphia, PA, 1995), (CD-ROM).

Babacan, O., Drugman, T., d'Alessandro, N., Henrich, N., and Dutoit, T. (2013). A comparative study of pitch extraction algorithms on a large variety of singing sounds. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2013)*, pages 7815–7819.

Baldwin, E. D., Cournand, A., and Richards, D. W. (1948). Pulmonary insufficiency; physiological classification, clinical methods of analysis, standard values in normal subjects. *Medicine*, 27(3):243–78.

Barker, J. and Cooke, M. (2007). Modelling speaker intelligibility in noise. *Speech Communication*, 49(5):402–417.

Barker, J., Marxer, R., Vincent, E., and Watanabe, S. (2017). The Third 'CHiME' Speech Separation and Recognition Challenge: Analysis and Outcomes. *Computer Speech and Language*, 46:605–626.

Bartholomew, W. T. (1934). A Physical Definition of "Good Voice-Quality" in the Male Voice. *The Journal of the Acoustical Society of America*, 6(1):25–33.

Bell, P., Gales, M. J. F., Hain, T., Kilgour, J., Lanchantin, P., Liu, X., McParland, A., Renals, S., Saz, O., Wester, M., and Woodland, P. C. (2015). The MGB challenge: Evaluating multi-genre broadcast media recognition. In *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU 2015)*, pages 687–693.

Bella, S. D., Berkowska, M., and Sowiński, J. (2015). Moving to the beat and singing are linked in humans. *Frontiers in Human Neuroscience*, 9(DEC):1–13.

Bittner, R., Salamon, J., Tierney, M., Mauch, M., Cannam, C., and Bello, J. P. (2014). MedleyDB: A Multitrack Dataset for Annotation-Intensive MIR Research. In *International Society for Music Information Retrieval Conference (ISMIR 2014)*, pages 155–160.

Bittner, R. M., Wilkins, J., Yip, H., and Bello, J. P. (2016). Medleydb 2.0: New Data and a System for Sustainable Data Collection. In *Late breaking/demo extended abstract, International Society for Music Information Retrieval Conference (ISMIR 2016)*.

Blaauw, M., Bonada, J., and Daido, R. (2019). Data Efficient Voice Cloning for Neural Singing Synthesis. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2019)*, pages 6840–6844.

Blei, D. M., Ng, A. Y., and Jordan, M. I. (2003). Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Boersma, P. (1993). Accurate short-term analysis of the fundamental frequency and the harmonics-to-noise ratio of a sampled sound. *Proceedings of the institute of phonetic sciences*, 17(1193):97–110.

Boersma, P. (2009). Should Jitter Be Measured by Peak Picking or by Waveform Matching? *Folia Phoniatrica et Logopaedica*, 61(5):305–308.

Boersma, P. and Weenink, D. (2021). Praat: doing phonetics by computer [Computer program]. Version 6.1.38, retrieved 2 January 2021 http://www.praat.org/.

Bogert, B. P., Healy, M. J. R., and Tukey, J. W. (1963). The Quefrency Analysis of Time Series for Echoes: Cepstrum, Pseudo-Autocovariance, Cross-Cepstrum, and Saphe Cracking. In *Proceedings of Symposium on Time Series Analysis*, pages 455–493.

Borch, D. Z. and Sundberg, J. (2002). Spectral distribution of solo voice and accompaniment in pop music. *Logopedics Phoniatrics Vocology*, 27(1):37–41.

Bouhuys, A., Mead, J., Proctor, D. F., and Stevens, K. N. (1968). Pressure-flow events during singing. *Annals of the New York Academy of Sciences*, 155(1):165–176.

Bouhuys, A., Proctor, D. F., and Mead, J. (1966). Kinetic aspects of singing. *Journal of Applied Physiology*, 21(2):483–496.

Brown, G. J. and Cooke, M. (1994). Computational auditory scene analysis. *Computer Speech and Language*, 8(4):297–336.

Brown, W., Rothman, H. B., and Sapienza, C. M. (2000). Perceptual and acoustic study of professionally trained versus untrained voices. *Journal of Voice*, 14(3):301–309.

Chandna, P., Blaauw, M., Bonada, J., and Gomez, E. (2019). WGANSing: A Multi-Voice Singing Voice Synthesizer Based on the Wasserstein-GAN. In *Proceedings of European Signal Processing Conference (EUSIPCO 2019)*, pages 1–5.

Chorowski, J. and Jaitly, N. (2017). Towards Better Decoding and Language Model Integration in Sequence to Sequence Models. In *Proceedings of Interspeech 2017*, pages 523–527.

Condit-Schultz, N. and Huron, D. (2015). Catching the Lyrics. *Music Perception*, 32(5):470–483.

Conlen, M. M. (2016). A Linguistic Comparison: Stress-timed and syllable-timed languages and their impact on second language acquisition. In *Honors College Theses. 30*. https://digitalcommons.wayne.edu/honorstheses/30.

Cooke, M. (2006). A glimpsing model of speech perception in noise. *The Journal of the Acoustical Society of America*, 119(3):1562–1573.

de Julián, P. P. (2016). Modificación o aggiustamento de las vocales españolas en el canto lírico. *Estudios de fonética experimental*, 25:263–293.

Defferrard, M., Benzi, K., Vandergheynst, P., and Bresson, X. (2017). FMA: A Dataset for Music Analysis. In *Proceedings of International Society for Music Information Retrieval Conference (ISMIR 2017)*, pages 316–323.

Défossez, A., Usunier, N., Bottou, L., and Bach, F. (2019). Music source separation in the waveform domain. *arXiv preprint arXiv:1911.13254v1*.

Dehak, N., Kenny, P. J., Dehak, R., Dumouchel, P., and Ouellet, P. (2011). Front-end factor analysis for speaker verification. *IEEE Transactions on Audio, Speech and Language Processing*, 19(4):788–798.

Demirel, E., Ahlbäck, S., and Dixon, S. (2020a). Automatic Lyrics Transcription using Dilated Convolutional Neural Networks with Self-Attention. In *Proceedings of the International Joint Conference on Neural Networks (IJCNN 2020)*, pages 1–8.

Demirel, E., Ahlbäck, S., and Dixon, S. (2021a). Computational Pronunciation Analysis in Sung Utterances. In *Proceedings of European Signal Processing Conference (EUSIPCO 2021)*, pages 186–190.

Demirel, E., Ahlbäck, S., and Dixon, S. (2021b). MSTRE-NET: Multistreaming Acoustic Modeling for Automatic Lyrics Transcription. In *Proceedings of International Society for Music Information Retrieval Conference (ISMIR 2021)*, pages 151–158.

Demirel, E., Ahlbäck, S., and Dixon, S. (2020b). A Recursive Search Method For Lyrics Alignment. In *Proceedings of Music Information Retrieval Evaluation eXchange (MIREX 2020)*.

Deng, C., Yu, C., Lu, H., Weng, C., and Yu, D. (2020). Pitchnet: Unsupervised Singing Voice Conversion with Pitch Adversarial Network. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2020)*, pages 7749–7753.

Deutsch, D., editor (2013). *The Psychology of Music*. Elsevier, third edition.

Di Persia, L., Milone, D., Rufiner, H. L., and Yanagida, M. (2008). Perceptual evaluation of blind source separation for robust speech recognition. *Signal Processing*, 88(10):2578–2583.

Doulaty, M. and Hain, T. (2019). Latent Dirichlet Allocation Based Acoustic Data Selection for Automatic Speech Recognition. In *Proceedings of Interspeech 2019*, pages 3228–3232.

Dressler, K. (2013). Sinusoidal extraction using an efficient implementation of a multi-resolution FFT. In *Proceedings of International Conference on Digital Audio Effects (DAFx 2013)*, pages 247–252.

Drugman, T. and Alwan, A. (2011). Joint robust voicing detection and pitch estimation based on residual harmonics. In *Proceedings of Interspeech 2011*, pages 1973–1976.

Duan, Z., Fang, H., Li, B., Sim, K. C., and Wang, Y. (2013). The NUS sung and spoken lyrics corpus: A quantitative comparison of singing and speech. In *Proceedings of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC 2013)*, pages 1–9.

Ellis, D. P. W. (2007). Beat Tracking by Dynamic Programming. *Journal of New Music Research*, 36(1):51–60.

Ellis, D. P. W., Cotton, C. V., and Mandel, M. I. (2008). Cross-correlation of beat-synchronous representations for music similarity. In *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 57–60. IEEE.

Farrus, M., Hernando, J., and Ejarque, P. (2007). Jitter and Shimmer Measurements for Speaker Recognition. In *Proceedings of Interspeech 2007*, pages 778–781.

Fine, P. A. and Ginsborg, J. (2014). Making myself understood: Perceived factors affecting the intelligibility of sung text. *Frontiers in Psychology*, 5(SEP):1–15.

FitzGerald, D. (2010). Harmonic/percussive separation using median filtering. In *Proceedings of International Conference on Digital Audio Effects (DAFx 2010)*, pages 1–4.

FitzGerald, D. (2012). Vocal Separation using Nearest Neighbours and Median Filtering. In *Proceedings of IET Irish Signals and Systems Conference (ISSC 2012)*, pages 1–5.

Fujihara, H. and Goto, M. (2012). Lyrics-to-Audio Alignment and its Application. *Dagstuhl Follow-Ups*, 3:23–36.

Fujihara, H., Kitahara, T., Goto, M., Komatani, K., Ogata, T., and Okuno, H. G. (2005). Singer identification based on accompaniment sound reduction and reliable frame selection. In *Proceedings of International Society for Music Information Retrieval Conference (ISMIR 2005)*, pages 329–336.

Fuller, C., Mallinckrodt, L., Maat, B., Başkent, D., and Free, R. (2013). Music and Quality of Life in Early-Deafened Late-Implanted Adult Cochlear Implant Users. *Otology & Neurotology*, 34(6):1041–1047.

Fuller, C. D., Galvin, J. J., Maat, B., Başkent, D., and Free, R. H. (2018). Comparison of Two Music Training Approaches on Music and Speech Perception in Cochlear Implant Users. *Trends in Hearing*, 22.

Gales, M. and Woodland, P. (1996). Mean and variance adaptation within the MLLR framework. *Computer Speech & Language*, 10(4):249–264.

Gao, X., Gupta, C., and Li, H. (2020a). Lyrics Transcription and Lyrics-to-audio alignment with music-informed acoustic models. In *Proceedings of Music Information Retrieval Evaluation eXchange (MIREX 2020)*.

Gao, X., Tian, X., Das, R. K., Zhou, Y., and Li, H. (2019). Speaker-independent Spectral Mapping for Speech-to-Singing Conversion. In *Proceedings of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC 2019)*, pages 159–164.

Gao, X., Tian, X., Zhou, Y., Das, R. K., and Li, H. (2020b). Personalized Singing Voice Generation Using WaveRNN. In *Proceedings of Odyssey 2020 The Speaker and Language Recognition Workshop*, pages 252–258.

Garofolo, J. S., Auzanne, C. G. P., and Voorhees, E. E. (2000). The TREC Spoken Document Retrieval Track: A Successful Story. *Text REtrieval Conference (TREC-9)*, 8940.

Garofolo, J. S., Lamel, L. F., Fisher, W. M., Fiscus, J. G., Pallett, D. S., and Dahlgren, N. L. (1983). TIMIT Acoustic-Phonetic Continuous Speech Corpus. In *Linguistic Data Consortium, Philadelphia*.

Ghahremani, P., Babaali, B., Povey, D., Riedhammer, K., Trmal, J., and Khudanpur, S. (2014). A Pitch Extraction Algorithm Tuned for ASR. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2014)*, pages 2494–2498.

Ghahremani, P., Manohar, V., Hadian, H., Povey, D., and Khudanpur, S. (2017). Investigation of transfer learning for ASR using LF-MMI trained neural networks. In *Proceedings of IEEE Automatic Speech Recognition and Understanding Workshop (ASRU 2017)*, pages 279–286.

Gibson, A. (2010). *Production and perception of vowels in New Zealand popular music*. Mphil thesis, Auckland University, New Zealand.

Goto, M., Hashiguchi, H., Nishimura, T., and Oka, R. (2002). RWC Music Database: Popular, Classical, and Jazz Music Databases. In *International Society for Music Information Retrieval Conference (ISMIR 2002)*, pages 287–288.

Gould, W. (1977). The effect of voice training on lung volumes in singers and the possible relationship to the damping factor of Pressman. *Journal of research in singing and applied vocal pedagogy*, 1:3–15.

Grabe, E., Post, B., Nolan, F., and Farrar, K. (2000). Pitch accent realization in four varieties of British English. *Journal of Phonetics*, 28(2):161–185.

Graves, A., Fernández, S., Gomez, F., and Schmidhuber, J. (2006). Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks. In *Proceedings of International Conference on Machine Learning 2006*, pages 369–376.

Grover, M. S., Bamdev, P., Kumar, Y., Hama, M., and Shah, R. R. (2020). audino: A Modern Annotation Tool for Audio and Speech. *ArXiv*, abs/2006.05236.

Gruhne, M., Schmidt, K., and Dittmar, C. (2007a). Detecting phonemes within the singing of polyphonic music. In *Proceedings of International Conference on Music Communication Science (ICoMCS 2007)*, pages 60–63.

Gruhne, M., Schmidt, K., and Dittmar, C. (2007b). Phoneme recognition in popular music. In *Proceedings of International Society for Music Information Retrieval Conference (ISMIR 2007)*, pages 369–370.

Hadian, H., Sameti, H., Povey, D., and Khudanpur, S. (2018). End-to-end Speech Recognition Using Lattice-free MMI. In *Proceedings of Interspeech 2018*, pages 12–16.

Hagen, M., Kerkhoff, J., and Gussenhoven, C. (2011). Singing Your Accent Away, and Why it Works. In *Proceedings of International Congress of Phonetic Sciences (ICPhS 2011)*, pages 799–802.

Hakes, J., Shipp, T., and Doherty, E. T. (1988). Acoustic characteristics of vocal oscillations: vibrato, exaggerated vibrato, trill, and trillo. *Journal of Voice*, 1(4):326–331.

Han, K. J., Chandrashekaran, A., Kim, J., and Lane, I. R. (2017). The CAPIO 2017 Conversational Speech Recognition System. *CoRR*, abs/1801.00059.

Hansen, J. K. (2012). Recognition of phonemes in A-cappella recordings using temporal patterns and Mel frequency Cepstral coefficients. In *Proceedings of Sound and Music Computing Conference (SMC 2012)*, pages 494–499.

Hermansky, H. (1990). Perceptual linear predictive (PLP) analysis of speech. *The Journal of the Acoustical Society of America*, 87(4):1738–1752.

Hermansky, H. and Sharma, S. (1998). TRAPS - Classifiers of Temporal Patterns. In *Proceedings of International Conference on Spoken Language Processing (ICSLP 1998)*, pages 289–292.

Hershey, S., Chaudhuri, S., Ellis, D. P. W., Gemmeke, J. F., Jansen, A., Moore, C., Plakal, M., Platt, D., Saurous, R. A., Seybold, B., Slaney, M., Weiss, R., and Wilson, K. (2017). CNN Architectures for Large-Scale Audio Classification. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2017)*, pages 131–135.

Hirahara, T. and Kato, H. (1992). The Effect of F0 on Vowel Identification. *Speech perception, production and linguistic structure*, pages 89–112.

Hochreiter, S. and Schmidhuber, J. (1997). Long Short-Term Memory. *Neural Computation*, 9(8):1735–1780.

Holmberg, E. B., Hillman, R. E., and Perkell, J. S. (1988). Glottal airflow and transglottal air pressure measurements for male and female speakers in soft, normal, and loud voice. *Journal of the Acoustical Society of America*, 84(2):511–529.

Horii, Y. (1979). Fundamental frequency perturbation observed in sustained phonation. *Journal of Speech and Hearing Research*, 22(1):5–19.

Hosoya, T., Suzuki, M., Ito, A., and Makino, S. (2005). Lyrics recognition from a singing voice based on finite state automaton for music information retrieval. In *Proceedings of International Society for Music Information Retrieval Conference (ISMIR 2005)*, pages 532–535.

Hsu, C. and Jang, J. R. (2010). On the Improvement of Singing Voice Separation for Monaural Recordings Using the MIR-1K Dataset. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 18(2):310–319.

Hsu, W.-N., Bolte, B., Tsai, Y.-H. H., Lakhotia, K., Salakhutdinov, R., and Mohamed, A. (2021). HuBERT: Self-Supervised Speech Representation Learning by Masked Prediction of Hidden Units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460.

Hu, D. and Saul, L. K. (2009). A probabilistic topic model for unsupervised learning of musical key-profiles. In *Proceedings of International Society for Music Information Retrieval Conference (ISMIR 2009)*, pages 441–446.

Hu, Y. and Loizou, P. C. (2006). Evaluation of objective measures for speech enhancement. In *Proceedings of Interspeech 2006*, pages 1447–1450.

Huang, X. and Lee, K. (1993). On speaker-independent, speaker-dependent, and speaker-adaptive speech recognition. *IEEE Transactions on Speech and Audio Processing*, 1(2):150–157.

Huang, X. D. (1991). A study on speaker-adaptive speech recognition. In *Proceedings of Workshop on Speech and Natural Language - HLT 1991*, pages 278–283.

Ioffe, S. and Szegedy, C. (2015). Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. In *Proceedings of International Conference on Machine Learning (ICML 2015)*, page 448–456.

Jadoul, Y., Thompson, B., and de Boer, B. (2018). Introducing Parselmouth: A Python interface to Praat. *Journal of Phonetics*, 71:1–15.

Jansson, A., Bittner, R. M., Ewert, S., and Weyde, T. (2019). Joint Singing Voice Separation and F0 Estimation with Deep U-Net Architectures. In *Proceedings of European Signal Processing Conference (EUSIPCO 2019)*, pages 1–5.

Jansson, A., Humphrey, E., Montecchio, N., Bittner, R., Kumar, A., and Weyde, T. (2017). Singing voice separation with deep U-Net convolutional networks. In *Proceedings of International Society for Music Information Retrieval Conference (ISMIR 2017)*, pages 23–27.

Jurafsky, D. (2009). *Speech and language processing : an introduction to natural language processing, computational linguistics, and speech recognition*. Prentice Hall series in artificial intelligence. Pearson/Prentice Hall, Upper Saddle River, N.J., 2nd ed. edition.

Kadandale, V. S., Montesinos, J. F., Haro, G., and G'omez, E. (2020). Multi-channel u-net for music source separation. *ArXiv*, abs/2003.10414.

Kamath, U., Liu, J., and Whitaker, J. (2019). *Deep Learning for NLP and Speech Recognition*. Springer Publishing Company, Incorporated, 1st edition.

Kawai, D., Yamamoto, K., and Nakagawa, S. (2016). Speech analysis of sung-speech and lyric recognition in monophonic singing. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2016)*, pages 271–275.

Kawai, D., Yamamoto, K., and Nakagawa, S. (2017). Lyric recognition in monophonic singing using pitch-dependent DNN. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2017)*, pages 326–330.

Kim, S., Sundaram, S., Georgiou, P., and Narayanan, S. (2009). Audio Scene Understanding Using Topic Models. In *Proceedings of Neural Information Processing System Workshop (NIPS 2009)*, pages 1–4.

Konert-Panek, M. (2017). Overshooting americanisation. accent stylisation in pop singing – acoustic properties of the bath and trap vowels in focus. *Research in Language*, 15:371–384.

Kruspe, A. M. (2014). Keyword Spotting in A-capella Singing. In *Proceedings of International Society for Music Information Retrieval Conference (ISMIR 2010)*, pages 271–276.

Kruspe, A. M. (2015a). Keyword spotting in singing with duration-modeled HMMs. In *Proceedings of European Signal Processing Conference (EUSIPCO 2015)*, pages 1291–1295.

Kruspe, A. M. (2015b). Training phoneme models for singing with "songified" speech data. In *Proceedings of International Society for Music Information Retrieval Conference (ISMIR 2015)*, pages 336–342.

Kruspe, A. M. (2016a). Bootstrapping a system for phoneme recognition and keyword spotting in unaccompanied singing. In *Proceedings of International Society for Music Information Retrieval Conference (ISMIR 2016)*, pages 358–364.

Kruspe, A. M. (2016b). Retrieval of textual song lyrics from sung inputs. In *Proceedings of Interspeech 2016*, pages 2140–2144.

Kruspe, Anna Marie (2018). *Application of Automatic Speech Recognition Technologies to Singing*. PhD thesis, Technische Universität Ilmenau.

Ladefoged, P. (1961). Sub-glottal activity during speech. In *Proceedings of International Congress of Phonetic Sciences (ICPhS 1961)*, pages 258–261.

Ladefoged, P. and Johnson., K. (2015). *A course in phonetics.* Cengage Learning, Australia, seventh ed edition.

Lane, H. and Tranel, B. (1971). The Lombard Sign and the Role of Hearing in Speech. *Journal of Speech and Hearing Research*, 14(4):677–709.

Lea, C., Vidal, R., Reiter, A., and Hager, G. D. (2016). Temporal convolutional networks: A unified approach to action segmentation. In *Proceedings of European Conference on Computer Vision 2016*, pages 47–54.

Levenshtein, V. I. et al. (1966). Binary codes capable of correcting deletions, insertions, and reversals. *Soviet Physics Doklady*, 10(8):707–710.

Lieberman, P., Knudson, R., and Mead, J. (1969). Determination of the Rate of Change of Fundamental Frequency with Respect to Subglottal Air Pressure During Sustained Phonation. *The Journal of the Acoustical Society of America*, 45(6):1537–1543.

Lindblom, B. E. F. and Sundberg, J. E. F. (1971). Acoustical Consequences of Lip, Tongue, Jaw, and Larynx Movement. *The Journal of the Acoustical Society of America*, 50(4B):1166–1179.

Liu, H.-M., Tsao, F.-M., and Kuhl, P. K. (2005). The effect of reduced vowel working space on speech intelligibility in Mandarin-speaking young adults with cerebral palsy. *The Journal of the Acoustical Society of America*, 117(6):3879–3889.

Liu, S., Cao, Y., Hu, N., Su, D., and Meng, H. (2021). FastSVC: Fast Cross-Domain Singing Voice Conversion with Feature-wise Linear Modulation. In *Proceedings of Interspeech 2021*, pages 1–6.

Liutkus, A., Fitzgerald, D., Rafii, Z., Pardo, B., and Daudet, L. (2014). Kernel Additive Models for Source Separation. *IEEE Transactions on Signal Processing*, 62(16):4298–4310.

Liutkus, A., Rafii, Z., Badeau, R., Pardo, B., and Richard, G. (2012). Adaptive filtering for music/voice separation exploiting the repeating musical structure. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2012)*, pages 53–56.

Liutkus, A., Stöter, F.-R., Rafii, Z., Kitamura, D., Rivet, B., Ito, N., Ono, N., and Fontecave, J. (2017). The 2016 Signal Separation Evaluation Campaign. In *Proceedings of International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA 2016)*, pages 323–332.

Lundy, D. S., Roy, S., Casiano, R. R., Xue, J. W., and Evans, J. (2000). Acoustic analysis of the singing and speaking voice in singing students. *Journal of Voice*, 14(4):490–493.

Luo, Y. and Mesgarani, N. (2017). TasNet: time-domain audio separation network for real-time, single-channel speech separation. *CoRR*, abs/1711.00541.

Luo, Y. and Mesgarani, N. (2019). Conv-tasnet: Surpassing ideal time–frequency magnitude masking for speech separation. *IEEE/ACM transactions on audio, speech, and language processing*, 27(8):1256–1266.

Lycke, H. and Siupsinskiene, N. (2016). Voice Range Profiles of Singing Students: The Effects of Training Duration and Institution. *Folia Phoniatrica et Logopaedica*, 68(2):53–59.

Mageau, M. (2016). *Foreign Accents in Song and Speech*. Mphil thesis, Carleton University, Canada.

Mainka, A., Poznyakovskiy, A., Platzek, I., Fleischer, M., Sundberg, J., and Mürbe, D. (2015). Lower vocal tract morphologic adjustments are relevant for voice timbre in singing. *PLoS ONE*, 10(7):1–19.

Manilow, E., Seetharaman, P., and Pardo, B. (2018). The Northwestern University Source Separation Library. In *Proceedings of International Society for Music Information Retrieval Conference (ISMIR 2018)*, pages 297–305.

Mauch, M., Fujihara, H., and Goto, M. (2012). Integrating Additional Chord Information Into HMM-Based Lyrics-to-Audio Alignment. *IEEE Transactions on Audio, Speech, and Language Processing*, 20(1):200–210.

McCowan, I. (2001). Microphone arrays: A tutorial.

McCowan, I., Lathoud, G., Lincoln, M., Lisowska, A., Post, W., Reidsma, D., and Wellner, P. (2005). The AMI Meeting Corpus. In *Proceedings of International Conference on Methods and Techniques in Behavioral Research 2005*.

McDermott, H. J. (2004). Music Perception with Cochlear Implants: A Review. *Trends in Amplification*, 8(2):49–82.

McVicar, M., Ellis, D. P., and Goto, M. (2014). Leveraging repetition for improved automatic lyric transcription in popular music. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2014)*, pages 3117–3121.

Merrill, J. and Larrouy-Maestri, P. (2017). Vocal features of song and speech: Insights from Schoenberg's Pierrot lunaire. *Frontiers in Psychology*, 8(JUL).

Mesaros, A. and Virtanen, T. (2009). Adaptation of a Speech Recognizer for Singing Voice. In *Proceedings of European Signal Processing Conference (EUSIPCO 2009)*, pages 1779–1783.

Mesaros, A. and Virtanen, T. (2010a). Automatic Recognition of Lyrics in Singing. *EURASIP Journal on Audio, Speech, and Music Processing*, 2010.

Mesaros, A. and Virtanen, T. (2010b). Recognition of phonemes and words in singing. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2010)*, volume 2010, pages 1–11.

Meseguer-Brocal, G., Cohen-Hadria, A., and Peeters, G. (2018). DALI: a large Dataset of synchronized Audio, LyrIcs and notes, automatically created using teacher-student machine learning paradigm. In *Proceedings of International Society for Music Information Retrieval Conference (ISMIR 2018)*, pages 431–437.

Mezzedimi, C., Spinosi, M., Massaro, T., Ferretti, F., and Cambi, J. (2018). Singing voice: acoustic parameters after vocal warm-up and cool-down. *Logopedics Phoniatrics Vocology*, pages 1–9.

Mirarchi, D., Vizza, P., Tradigo, G., Lombardo, N., Arabia, G., and Veltri, P. (2017). Signal Analysis for Voice Evaluation in Parkinson's Disease. In *Proceedings of 2017 IEEE International Conference on Healthcare Informatics (ICHI 2017)*, pages 530–535.

Mirzaei, S., Norouzi, Y., and Van Hamme, H. (2015). Two-stage blind audio source counting and separation of stereo instantaneous mixtures using Bayesian tensor factorisation. *IET Signal Processing*, 9(8):587–595.

Mohri, M., Pereira, F., and Riley, M. (2002). Weighted finite-state transducers in speech recognition. *Computer Speech & Language*, 16(1):69–88.

Morozov, V. P. (1965). The Intelligibility in singing as a function of fundamental voice pitch. *Soviet Physics-Acoustics*, 10:279–283.

Munro, M. J. and Derwing, T. M. (1995). Foreign Accent, Comprehensibility, and Intelligibility in the Speech of Second Language Learners. *Language Learning*, 45(1):73–97.

Nachmani, E. and Wolf, L. (2019). Unsupervised Singing Voice Conversion. In *Proceedings of Interspeech 2019*, pages 2583–2587.

Novak, J. R., Minematsu, N., and Hirose, K. (2015). Phonetisaurus : Exploring grapheme-to-phoneme conversion with joint n-gram models in the WFST framework. *Natural Language Engineering*, 22(6):907–938.

Olsen, W. O. (1998). Average Speech Levels and Spectra in Various Speaking/Listening Conditions. *American Journal of Audiology*, 7(2):21–25.

Orlikoff, R. F. and Kahane, J. C. (1991). Influence of mean sound pressure level on jitter and shimmer measures. *Journal of Voice*, 5(2):113–119.

Panayotov, V., Chen, G., Povey, D., and Khudanpur, S. (2015). Librispeech: An ASR Corpus Based on Public Domain Audio Books. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210.

Parekh, J., Rao, P., and Yang, Y.-H. (2020). Speech-To-Singing Conversion in an Encoder-Decoder Framework. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 261–265.

Pariente, M., Cornell, S., Cosentino, J., Sivasankaran, S., Tzinis, E., Heitkaemper, J., Olvera, M., Stöter, F.-R., Hu, M., Martín-Doñas, J. M., Ditter, D., Frank, A., Deleforge, A., and Vincent, E. (2020). Asteroid: the PyTorch-based audio source separation toolkit for researchers. In *Proceedings of Interspeech 2020*, pages 2637–2641.

Patel, A. D., Wong, M., Foxton, J., Lochy, A., and Peretz, I. (2008). Speech intonation perception deficits in musical tone deafness (congenital amusia). *Music Perception*, 25:357–368.

Paul, D. (1989). The Lincoln robust continuous speech recognizer. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 1989)*, pages 449–452.

Paul, D. B. and Baker, J. M. (1992). The Design for the Wall Street Journal-based CSR Corpus. In *Proceedings of DARPA Speech and Natural Language Workshop*, pages 357–362.

Pearsons, K., Bennett, R., and Fidell, S. (1976). Speech levels in various environments. *Bolt Beranek and Newman*, Report No.(May):Canoga Park, CA.

Perricone, J. (2018). *Great Songwriting Techniques*. Oxford Univesity Press.

Pike, K. L. (1945). *The Intonation of American English*. Ann Arbor: University of Michigan Press.

Povey, D., Cheng, G., Wang, Y., Li, K., Xu, H., Yarmohammadi, M., and Khudanpur, S. (2018). Semi-Orthogonal Low-Rank Matrix Factorization for Deep Neural Networks. In *Proceedings of Interspeech 2018*, pages 3743–3747.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., and Vesely, K. (2011). The Kaldi Speech Recognition Toolkit. In *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU 2011)*.

Povey, D., Peddinti, V., Galvez, D., Ghahrmani, P., Manohar, V., Na, X., Wang, Y., and Khudanpur, S. (2016). Purely Sequence-Trained Neural Networks for ASR Based on Lattice-Free MMI. In *Proceedings of Interspeech 2016*, pages 2751–2755.

Proctor, D. F. (1980). *Breathing, speech, and song*. Springer-Verlag, Wien, New York.

Rafii, Z., Liutkus, A., Stöter, F.-R., Mimilakis, S. I., and Bittner, R. (2017). The MUSDB18 corpus for music separation. https://doi.org/10.5281/zenodo.1117372.

Rafii, Z., Liutkus, A., Stoter, F. R., Mimilakis, S. I., Fitzgerald, D., and Pardo, B. (2018). An Overview of Lead and Accompaniment Separation in Music. *IEEE/ACM Transactions on Audio Speech and Language Processing*, 26(8):1307–1335.

Rafii, Z. and Pardo, B. (2011). A simple music/voice separation method based on the extraction of the repeating musical structure. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2011)*, pages 221–224.

Rafii, Z. and Pardo, B. (2012). Music/voice separation using the similarity matrix. In *Proceedings of International Society for Music Information Retrieval Conference (ISMIR 2012)*, pages 583–588.

Randel, D. M. (1986). *The New Harvard Dictionary of Music*. Belknap Press.

Roa Dabike, G. (2016). *Automatic Speech Recognition in Music*. Unpublished Master's dissertation, The University of Sheffield, UK.

Roa Dabike, G. and Barker, J. (2019). Automatic Lyric Transcription from Karaoke Vocal Tracks: Resources and a Baseline System. In *Proceedings of Interspeech 2019*, pages 579–583.

Roa Dabike, G. and Barker, J. (2020). The Sheffield University System for the MIREX 2020: Lyrics Transcription Task. In *Proceedings of Music Information Retrieval Evaluation eXchange (MIREX 2020)*.

Roa Dabike, G. and Barker, J. (2021). The Use of Voice Source Features for Sung Speech Recognition. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2021)*, pages 6513–6517.

Rossi, S., Gugler, M. F., Rungger, M., Galvan, O., Zorowka, P. G., and Seebacher, J. (2020). How the Brain Understands Spoken and Sung Sentences. *Brain Sciences*, 10(1):36.

Rousseau, A., Deléglise, P., and Estève, Y. (2012). TED-LIUM: an Automatic Speech Recognition dedicated corpus. In *Proceedings of International Conference on Language Resources and Evaluation (LREC 2012)*, pages 125–129.

Roux, J. L., Wisdom, S., Erdogan, H., and Hershey, J. R. (2019). SDR - half-baked or well done? In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2019)*, pages 626–630.

Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A. C., and Fei-Fei, L. (2014). ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision*, 115(3):211–252.

Samuel, D., Ganeshan, A., and Naradowsky, J. (2020). Meta-Learning Extractors for Music Source Separation. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2020)*, pages 816–820.

Sandoval, S., Berisha, V., Utianski, R. L., Liss, J. M., and Spanias, A. (2013). Automatic assessment of vowel space area. *The Journal of the Acoustical Society of America*, 134(5):EL477–EL483.

Schulze-Forster, K., Doire, C., Richard, G., and Badeau, R. (2021). Phoneme Level Lyrics Alignment and Text-Informed Singing Voice Separation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:2382–2395.

Sgouros, T. and Mitianoudis, N. (2020). A novel Directional Framework for Source Counting and Source Separation in Instantaneous Underdetermined Audio Mixtures. *IEEE/ACM Transactions on Audio Speech and Language Processing*, 28:2025–2035.

Sharma, B., Gao, X., Vijayan, K., Tian, X., and Li, H. (2021). NHSS: A speech and singing parallel database. *Speech Communication*, 133:9–22.

Simonyan, K. and Zisserman, A. (2014). Very Deep Convolutional Networks for Large-Scale Image Recognition. In *Proceedings of International Conference on Learning Representations (ICLR 2014)*.

Singer, H. and Sagayama, S. (1992). Pitch dependent phone modelling for HMM based speech recognition. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 1992)*, pages 273–276.

Sisman, B. and Li, H. (2020). Generative Adversarial Networks for Singing Voice Conversion with and without Parallel Data. In *Proceedings of Odyssey 2020 The Speaker and Language Recognition Workshop*, pages 7749–7753.

Slyh, R., Nelson, W., and Hansen, E. (1999). Analysis of mrate, shimmer, jitter, and F0 contour features across stress and speaking style in the SUSAS database. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 1999)*, pages 2091–2092.

Smule, Inc. (2015). Vocal performances (multiple songs) dataset, https://ccrma.stanford.edu/damp/. (accessed July 2018).

Smule, Inc. (2017). Digital Archive of Mobile Performances - Smule Vocal Performances Balanced (Version 1.0.0) [Data set]. Zenodo. http://doi.org/10.5281/zenodo.2616690.

Smule, Inc. (2018). DAMP-MVP: Digital Archive of Mobile Performances - Smule Multilingual Vocal Performance 300x30x2 (Version 1.0.0) [Data set]. Zenodo. https://zenodo.org/record/2747436.

Smule, Inc. (2019). DAMP-VSEP: Smule Digital Archive of Mobile Performances - Vocal Separation (Version 1.0.1) [Data set]. Zenodo. http://doi.org/10.5281/zenodo.3553059.

Snyder, D., Garcia-Romero, D., Sell, G., Povey, D., and Khudanpur, S. (2018). X-Vectors: Robust DNN Embeddings for Speaker Recognition. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2018)*, pages 5329–5333.

Stevens, S. S., Volkmann, J., and Newman, E. B. (1937). A Scale for the Measurement of the Psychological Magnitude Pitch. *The Journal of the Acoustical Society of America*, 8(3):185–190.

Stolcke, A. (2002). SRILM — An Extensible Language Modeling Toolkit. In *Proceedings of Interspeech 2002*, pages 901–904.

Stoller, D., Durand, S., and Ewert, S. (2019). End-to-end Lyrics Alignment for Polyphonic Music Using an Audio-to-character Recognition Model. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2019)*, pages 181–185.

Stoller, D., Ewert, S., and Dixon, S. (2018a). Jointly Detecting and Separating Singing Voice: A Multi-Task Approach. In *Proceedings of International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA 2018)*, pages 329–339.

Stoller, D., Ewert, S., and Dixon, S. (2018b). Wave-U-Net: A multi-scale neural network for end-to-end audio source separation. *International Society for Music Information Retrieval Conference (ISMIR)*, pages 334–340.

Stöter, F.-R., Liutkus, A., and Ito, N. (2018). The 2018 Signal Separation Evaluation Campaign. In *Proceedings of International Conference on Latent Variable Analysis and Signal Separation (LVA/ICA 2018)*, pages 293–305.

Stöter, F.-R., Uhlich, S., Liutkus, A., and Mitsufuji, Y. (2019). Open-Unmix - A Reference Implementation for Music Source Separation. *Journal of Open Source Software*, 4(41):1667.

Strube, H. W. (1980). Linear prediction on a warped frequency scale. *The Journal of the Acoustical Society of America*, 68(4):1071–1076.

Sundberg, J. (1975). Vibrato and vowel identification. *STL-QPSR*, 16(2-3):49–60.

Sundberg, J. (1977a). Singing and timbre. In *Proceedings of Music Room Acoustics, Stockholm: Royal Swedish Academy of Music 1977*, pages 57–80.

Sundberg, J. (1977b). The Acoustics of the Singing Voice. *Scientific American*, 236(3):82–91.

Sundberg, J. (1987). *The science of the singing voice*. Northern Illinois University Press.

Sundberg, J. (2001). Level and Center Frequency of the Singer's Formant. *Journal of Voice*, 15(2):176–186.

Szepannek, G., Gruhne, M., Bischl, B., Krey, S., Harczos, T., Klefenz, F., Dittmar, C., and Weihs, C. (2010). Perceptually Based Phoneme Recognition in Popular Music. *Studies in Classification, Data Analysis, and Knowledge Organization*, pages 751–758.

Taal, C. H., Hendriks, R. C., Heusdens, R., and Jensen, J. (2010). A Short-Time Objective Intelligibility Measure for Time-Frequency Weighted Noisy Speech. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2010)*, pages 4214–4217.

Takahashi, N., Singh, M. K., and Mitsufuji, Y. (2021). Hierarchical disentangled representation learning for singing voice conversion. In *Proceedings of The International Joint Conference on Neural Networks (IJCNN 2021)*, pages 1–7.

Talkin, D. and Kleijn, W. B. (1995). A robust algorithm for pitch tracking (RAPT). *Speech coding and synthesis*, 495:518.

Titze, I. R. and Liang, H. (1993). Comparison of F0 Extraction Methods for High-Precision Voice Perturbation Measurements. *Journal of Speech, Language, and Hearing Research*, 36(6):1120–1133.

Tsai, C. P., Tuan, Y. L., and Lee, L. S. (2018). Transcribing Lyrics from Commercial Song Audio: The First Step Towards Singing Content Processing. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2018)*, pages 5749–5753.

Uhlich, S., Porcu, M., Giron, F., Enenkl, M., Kemp, T., Takahashi, N., and Mitsufuji, Y. (2017). Improving music source separation based on deep neural networks through data augmentation and network blending. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2017)*.

Vaglio, A., Hennequin, R., Moussallam, M., Richard, G., and d'Alché Buc, F. (2020). Audio-Based Detection of Explicit Content in Music. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2020)*, pages 526–530.

Vetterling, W. T., Press, W. H., Teukolsky, S. A., and Flannery, B. P. (1992). *Numerical recipes in C: the art of scientific computing*. Cambridge University Press, Cambridge, 2nd ed. edition.

Vijayan, K., Gao, X., and Li, H. (2018). Analysis of Speech and Singing Signals for Temporal Alignment. In *Proceedings of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC 2018)*, pages 1893–1898.

Vijayan, K., Li, H., and Toda, T. (2019). Speech-to-Singing Voice Conversion: The Challenges and Strategies for Improving Vocal Conversion Processes. *IEEE Signal Processing Magazine*, 36(1):95–102.

Vincent, E., Gribonval, R., and Févotte, C. (2006). Performance measurement in blind audio source separation. *IEEE transactions on audio, speech, and language processing*, 14(4):1462–1469.

Vincent, E., Virtanen, T., and Gannot, S. (2018). *Audio Source Separation and Speech Enhancement*. Wiley.

Vincent, E., Watanabe, S., Nugraha, A. A., Barker, J., and Marxer, R. (2017). An Analysis of Environment, Microphone and Data Simulation Mismatches in Robust Speech Recognition. *Computer Speech and Language*, 46:535–557.

Vos, P. G. and Troost, J. M. (1989). Ascending and descending melodic intervals: Statistical findings and their perceptual relevance. *Music Perception: An Interdisciplinary Journal*, 6(4):383–396.

Waibel, A., Hanazawa, T., Hinton, G., Shikano, K., and Lang, K. (1989). Phoneme recognition using time-delay neural networks. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(3):328–339.

Walsdorff, M., Van Muylem, A., and Gevenois, P. A. (2015). Effect of total lung capacity and gender on CT densitometry indexes. *The British Journal of Radiology*, 89.

Wang, C. K., Lyu, R. Y., and Chiang, Y. C. (2003). An Automatic Singing Transcription System with Multilingual Singing Lyric Recognizer and Robust Melody Tracker. In *Proceedings of EUROSPEECH 2003*.

Wang, D. and Zheng, T. F. (2015). Transfer Learning for Speech and Language Processing. In *Proceedings of Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC 2015)*, pages 1225–1237.

Wang, Z., Giri, R., Isik, U., Valin, J.-M., and Krishnaswamy, A. (2021). Semi-Supervised Singing Voice Separation With Noisy Self-Training. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2021)*, pages 31–35.

Watanabe, K. and Goto, M. (2020). Lyrics Information Processing: Analysis, Generation, and Applications. In *Proceedings of Workshop on NLP for Music and Audio (NLP4MusA 2020)*, pages 6–12.

Wennerstrom, A. (2001). *The music of everyday speech: Prosody and discourse analysis*. Oxford University Press.

Whistler, K. (2021). UAX #15: Unicode Normalization Forms.

Wu, D.-Y. and Yang, Y.-H. (2020). Speech-to-Singing Conversion Based on Boundary Equilibrium GAN. In *Proceedings of Interspeech 2020*, pages 1316–1320.

Young, S. J. (1993). The HTK hidden markov model toolkit: Design and philosophy. *Technical Report, University of Cambridge, Department of Engineering Cambridge, Cambridge, UK.*

Zhang, C., Yu, J., Chang, L., Tan, X., Chen, J., Qin, T., and Zhang, K. (2021). PDAugment: Data Augmentation by Pitch and Duration Adjustments for Automatic Lyrics Transcription. *ArXiv*, abs/2109.07940.

Zhou, W., Michel, W., Irie, K., Kitza, M., Schluter, R., and Ney, H. (2020). The RWTH ASR System for TED-LIUM Release 2: Improving Hybrid HMM with SpecAugment. In *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP 2020)*, pages 7839–7843.

Zhuang, F., Qi, Z., Duan, K., Xi, D., Zhu, Y., Zhu, H., Xiong, H., and He, Q. (2021). A comprehensive survey on transfer learning. *Proceedings of the IEEE*, 109(1):43–76.

Appendix A

ASR Results

Table A.1 Independent results of Baseline experiments per DSing training set.

Run	DSing1				DSing3				DSing30			
	Dev		Test		Dev		Test		Dev		Test	
	3-gram	4-gram	3-gram	4-gram	3-gram	4-gram	3-gram	4-gram	3-gram	4-gram	3-gram	4-gram
1	45.84	40.07	42.46	37.14	33.35	29.19	28.12	24.25	25.66	22.57	22.98	20.73
2	45.32	40.74	42.04	36.96	32.68	29.32	28.06	24.6	25.68	23.15	22.81	19.5
3	46.67	41.24	42.95	36.94	33.23	29.17	28.19	24.36	26.01	22.52	23.26	20.26
4	47.56	42.06	43.93	39.62	33.28	29.49	28.36	24.97	25.88	22.97	22.27	19.33
5	48.03	42.68	44.03	38.18	33.82	29.62	27.97	24.41	26.46	23.57	23.02	20.3
6	46.69	41.59	43.51	38.89	32.58	28.62	28.56	24.43	26.18	23	22.65	19.24
7	47.51	42.83	44.36	39.49	32.8	28.82	28.06	24.62	25.78	22.35	22.37	19.28
8	47.36	41.41	43.6	38.46	33.87	28.94	28.3	25.12	25.78	22.27	23.04	20.3
9	44.72	39.85	41.79	37.16	33.6	28.94	28.1	24.04	25.78	22.57	23.28	20.56
10	46.57	41.91	42.84	38.67	33.47	29.54	28.1	23.93	25.24	21.88	22.35	19.24
11	46.99	42.21	41.76	37.98	33.92	29.99	27.6	23.65	25.21	22.35	23.02	19.97
μ	46.66	41.51	43.02	38.14	33.33	29.24	28.13	24.4	25.79	22.65	22.82	19.88
e_μ	0.6	0.58	0.55	0.58	0.28	0.24	0.14	0.26	0.22	0.28	0.21	0.34

Table A.2 Independent results of Kaldi N experiments per DSing training set.

Run	DSing1				DSing3				DSing30			
	Dev		Test		Dev		Test		Dev		Test	
	3-gram	4-gram	3-gram	4-gram	3-gram	4-gram	3-gram	4-gram	3-gram	4-gram	3-gram	4-gram
1	46.69	40.49	41.92	38	31.76	27.8	27.6	24.04	24.96	21.83	22.42	19.89
2	44.95	38.85	41.72	37.07	31.58	27.43	28.32	24.41	26.03	22.77	23.19	19.56
3	45.72	40.34	41.05	35.06	32.83	28.37	27.99	24.25	25.91	22.03	23.22	19.65
4	46.19	41.66	41.61	38.11	32.65	28.2	27.52	24.1	25.71	22.87	22.91	19.65
5	45.27	39.92	40.9	36.45	32.35	28.12	28.1	24.47	24.96	22.08	23.09	20
6	45.27	40.04	42.61	37.4	32.65	28.72	27.76	24.41	25.44	22.6	23.07	19.65
7	45.55	40	41.2	37.33	31.86	28.67	27.52	24.41	25.54	22.55	22.83	19.71
8	45.22	39.87	41.53	36.68	32.95	28.02	28.32	24.82	25.61	22.08	22.94	19.52
9	46.69	41.89	42.91	41.12	32.43	27.97	27.43	23.91	25.56	22.62	22.61	19.37
10	45.5	40.89	41.61	37.74	32.6	28.52	28.23	24.54	25.76	22.8	23.22	19.89
11	45.32	40.44	41.31	37.66	32.8	28.45	26.89	23.69	26.01	22.7	22.91	19.46
μ	45.67	40.4	41.67	37.51	32.41	28.21	27.79	24.28	25.59	22.45	22.95	19.67
e_μ	0.35	0.5	0.37	0.87	0.28	0.23	0.27	0.19	0.22	0.22	0.15	0.11

Table A.3 Independent results of Kaldi L experiments per DSing training set.

Run	DSing1				DSing3				DSing30			
	Dev		Test		Dev		Test		Dev		Test	
	3-gram	4-gram	3-gram	4-gram	3-gram	4-gram	3-gram	4-gram	3-gram	4-gram	3-gram	4-gram
1	45.05	40.82	42.07	37.31	32.53	29.12	28.62	24.95	23.67	20.58	21.38	18.98
2	46.64	41.61	42.33	39.04	32.11	28.4	27.32	23.97	25.93	22.75	23.28	19.24
3	44.38	39.97	40.96	36.6	32.83	28.07	28.56	24.43	25.66	22.5	23.48	20.32
4	45.72	39.97	41.29	37.16	32.78	28.82	28.3	25.1	26.08	22.35	23.11	19.67
5	46.64	41.54	43.9	38.95	32.38	28.27	26.78	24.36	25.29	22.03	22.83	19.5
6	45.5	40.04	41.55	36.84	32.55	28.1	28.06	23.74	25.86	23.25	22.63	20.19
7	45.62	40.47	41.5	37.33	32.38	28.94	27.69	24.56	25.19	22.55	23.11	20.19
8	45.32	39.85	41.89	36.62	31.63	28.37	27.39	23.67	25.86	22.45	22.91	19.87
9	45.42	40.59	40.64	37.14	32.95	28.62	27.8	24.58	25.88	22.9	23.13	19.82
10	44.67	40.57	40.7	36.55	32.21	28.5	27.37	24.25	25.24	22.2	23.32	19.82
11	45.55	40.64	42.84	36.81	32.3	28.45	27.95	24.21	25.49	22.47	22.96	19.3
μ	45.5	40.55	41.79	37.3	32.42	28.51	27.8	24.35	25.47	22.37	22.92	19.72
e_μ	0.41	0.36	0.58	0.52	0.22	0.2	0.34	0.27	0.4	0.4	0.33	0.25

Table A.4 Independent results of Kaldi LN experiments per DSing training set.

Run	DSing1				DSing3				DSing30			
	Dev		Test		Dev		Test		Dev		Test	
	3-gram	4-gram	3-gram	4-gram	3-gram	4-gram	3-gram	4-gram	3-gram	4-gram	3-gram	4-gram
1	44.43	39.85	40.99	36.77	31.48	27.9	27.73	24.28	25.34	22.52	22.76	19.67
2	44.38	40.57	40.45	36.38	32.03	27.68	28.15	23.69	25.09	21.65	23.74	19.52
3	45.55	41.36	41.81	37.18	32.18	28	27.76	23.71	24.81	21.38	22.63	20.08
4	44.47	40.32	40.4	35.54	32.65	28.97	28.19	24.43	26.18	22.75	23.69	19.91
5	45.15	40.12	42.33	37.2	32.6	28.52	27.95	24.47	25.01	22.35	23.39	20.06
6	44.7	39.35	40.7	36.64	32.58	28.92	28.25	24.32	25.88	22.6	23.24	19.65
7	45.72	40.57	41.25	36.06	31.88	27.82	28.12	24.32	24.99	21.95	22.96	19.82
8	45.4	40.74	40.79	36.62	31.83	28.15	27.41	24.82	25.29	22.62	23.65	20.17
9	46.32	41.66	42.97	38.54	32.18	28.07	28.36	24.43	25.66	21.95	23.93	19.84
10	45.69	40.77	41.14	36.81	32.65	28.47	27.76	24.54	25.93	22.72	23	19.91
11	44.82	40.29	40.55	37.12	32.21	27.6	28.92	23.95	25.06	22.22	22.59	19.89
μ	45.15	40.51	41.22	36.81	32.21	28.19	28.05	24.27	25.39	22.25	23.23	19.87
e_μ	0.38	0.38	0.49	0.45	0.23	0.28	0.24	0.21	0.27	0.27	0.28	0.12

Table A.5 Independent results of Kaldi LN + VQ experiments per DSing training set.

Run	DSing1				DSing3				DSing30			
	Dev		Test		Dev		Test		Dev		Test	
	3-gram	4-gram	3-gram	4-gram	3-gram	4-gram	3-gram	4-gram	3-gram	4-gram	3-gram	4-gram
1	45.42	40.89	42.04	37.31	32.06	27.35	28.06	23.99	25.14	22.45	22.61	19.26
2	45.15	40.32	41.31	37.22	32.35	27.68	28.43	24.54	25.76	22.22	22.63	19.37
3	45.5	40.27	40.73	36.12	32.23	28.2	27.89	23.97	25.01	22.67	22.7	19.15
4	45.72	40.52	40.9	36.34	32.11	27.87	27.43	23.97	25.56	22.37	24.32	19.91
5	44.92	39.92	41.14	36.81	31.46	27.1	27.89	23.56	25.86	22.8	22.76	19.33
6	44.77	39.37	41.27	36.66	31.21	27.53	27.52	23.74	25.11	21.8	23.65	19.93
7	46.07	40.29	42.09	37.96	30.91	27	27.02	23.54	25.56	22.47	22.68	19.39
8	44.95	39.92	40.81	35.3	31.58		27.67	23.19	25.54	22.65	22.96	19.97
9	45.02	39.85	40.83	37.38	31.46	27.8	27.86	23.45	25.86	22.03	22.78	19.54
10	44.7	39.37	41.25	36.77	32.43	27.15	27.69	23.07	25.16	22.08	22.94	19.52
11	44.33	39.25	40.49	35.78	31.73	27.97	28.56	24.38	25.21	21.98	22.65	20.23
μ	45.14	40	41.17	36.7	31.78	27.57	27.82	23.76	25.43	22.32	22.97	19.6
e_μ	0.3	0.31	0.3	0.46	0.29	0.25	0.26	0.27	0.19	0.19	0.32	0.21

Table A.6 Independent results of Kaldi LN + VQ + Beat experiments per DSing training set.

Run	DSing1				DSing3				DSing30			
	Dev		Test		Dev		Test		Dev		Test	
	3-gram	4-gram	3-gram	4-gram	3-gram	4-gram	3-gram	4-gram	3-gram	4-gram	3-gram	4-gram
1	45.99	40.44	42.61	36.94	31.63	27.6	28.17	25.25	25.36	22.47	23.56	19.76
2	45.42	40.87	42.2	37.61	32.13	27.73	27.76	24.25	26.33	22.9	22.2	19.56
3	45.22	39.7	41.22	37.25	32.55	28.92	28.15	24.04	24.69	22.03	22.2	19.61
4	46.09	41.04	41.81	37.92	32.33	28.02	27.43	24.92	25.09	22.5	23.09	19.43
5	45.57	39.77	41.27	36.9	31.51	27.25	28.56	24.43	25.83	21.78	22.31	19.58
6	45.25	41.04	41.74	37.48	32.23	27.78	27.39	23.8	26.38	23.32	22.63	20.17
7	45.79	40.99	41.72	37.16	31.81	28.12	27.97	23.58	25.49	22.03	23.45	19.97
8	45	39.82	41.68	36.92	31.43	28.1	28.1	23.82	25.14	22.03	23.63	19.67
9	44.77	40.69	41.83	36.64	32.25	27.9	28.06	24.84	25.09	21.88	23.26	19.97
10	45.32	40.77	41.48	36.84	32.38	28.07	28.21	23.99	25.46	22.57	22.52	19.84
11	45.15	40.82	41.07	37.22	32.45	27.97	28.12	24.69	25.78	22.47	22.72	19.28
μ	45.42	40.54	41.69	37.17	32.06	27.95	27.99	24.33	25.51	22.36	22.87	19.71
e_μ	0.24	0.31	0.26	0.22	0.24	0.24	0.2	0.32	0.31	0.28	0.32	0.15

Appendix B

Robustness and Variability in Singing Speed

This Appendix presents the definition of syllables per second measurement and its possible uses informing sung speech recognition systems.

Introduction

In music, timing and rhythm are essential components that can be described in terms of the musical beat measured in beats per second. However, the quality of the beat extraction when it is estimated directly from singing speech is highly sensitive to the singing proficiency.

The singing speed may be another way to obtain temporal information on singing. This speed can be estimated in terms of the number of **syllables per second** (**SPS**) of the song. This parameter could inform models of how fast or slow is the speech. In contrast with the beats per second measurement, the syllables per second are not affected by the singing proficiency because it solely measures the time used to sing a sequence of syllables.

The syllables per second computation involve the counting of the syllables in the transcription reference and the length of the utterances. This dependency on the transcription reference signifies that it cannot be directly used for feature vector augmentation as the testing sets transcription is the target of the system. However, this feature is explored to examine the potential of the singing speed features in 'ground truth' conditions. Potentially, the SPS information can be utilised as a mean to filtering out poor data, i.e. long utterances with a small number of syllables in the transcription reference. Additionally, knowing the SPS distribution of the training data, it can be utilised as an approach for informing training data augmentation.

Table B.1 Representation of four words from Celex and CMU dictionaries. In Celex the words are split into syllables, and in CMU the words are divided into phonetic units.

Word	Syllables	Celex	CMU
Love	1	[lVv]	L *AH* V
Yeah	1	[jE@]	Y *AE*
Baby	2	[beI][bI]	B *EY* B *IY*
People	2	[pi:][pl,]	P *IY* P *AH* L

The syllables per second parameter was explored at two different levels. Initially, it was calculated at the utterance level to analyse the merits of the parameter at this level, such as an objective way to filter poorly segmented utterances. Then, it was explored at a song level to determine if exist advantages when the song speed is used to normalise the data.

Section B presents the definition and computation of the SPS parameter. Section B presents and analysis of the possibles benefits of use the SPS for sung speech ASR task.

The syllables per second parameter

The syllables per second parameter (SPS) is defined as the average number of syllables sung per second over the duration of the utterance. The algorithm is divided into two steps. It first counts the number of syllables in the utterance using the transcription and a pronunciation dictionary. Then, the algorithm estimates the duration of the utterance by summing the duration of all the singing periods in the utterance, i.e., not including the pauses.

Two dictionaries and a rule-based algorithm are employed to obtain the **number of syllables** from each word in a sentence. The rules are applied one word at a time, as follows: The process starts by searching for the word in the Celex Lexical Database (Celex) (Baayen et al., 1995), a dictionary that provides a syllabification of words. If the word is included in Celex, the process returns the number of syllables from this database. Otherwise, the search continues using the CMU Pronunciation (CMU) dictionary. If the word is found in CMU, the algorithm determines the number of syllables by counting the vowels. Table B.1 illustrates the different way that Celex and CMU represent the words by presenting four examples of common words used in lyrics.

Finally, if the word is not found in neither Celex nor CMU dictionaries, a set of syllabification rules are applied;

1. Starts the count of syllables as zero.

2. Increment the count of syllables in one when:

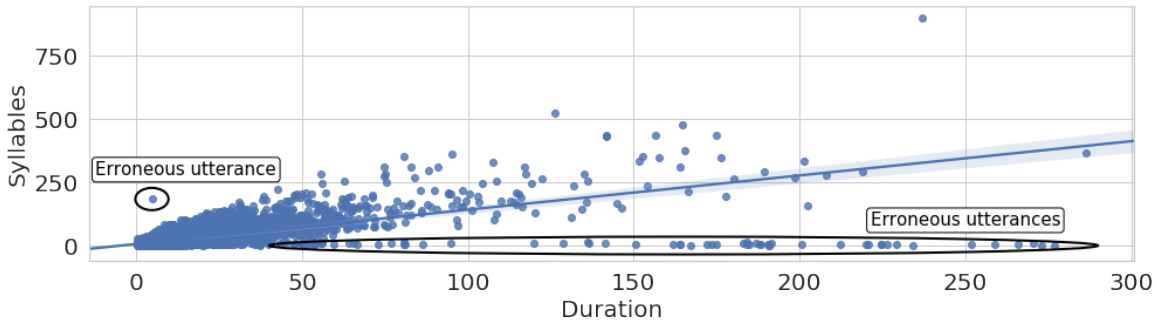


Figure B.1 Correlation between the number of syllables and the duration from the DSing30 dataset.

(a) The first letter is a vowel.

(b) A consonant precedes a vowel.

3. Subtract one syllable if the last letter is an *E*, except if an *L* precedes that *E*.

After the number of syllables is calculated, the algorithm starts with the estimation of the *duration* of the singing. The algorithm firstly applies the silence filtering out process used during the DSing dataset construction (Section 4.2). This step aims to avoid overestimating the duration, by excluding the silence fragments. Then, the duration is obtained by adding up all the non-silences segments.

DSing dataset and the usefulness of the SPS parameter

In this section, the SPS usefulness is evaluated for DSing task. The syllable count is obtained by using the procedure described above and the transcription of the utterances. The singing duration is estimated by reapplying the silence filtering out process to avoid overestimating the duration, by excluding the silence fragments added by the final re-alignment step from the DSing construction.

Figure B.1, shows a strong correlation between the number of syllables of the transcription and the duration of the utterances. However, this figure also shows several samples with a considerable duration but a small number of syllables. After a direct analysis of these samples, it was found that these erroneous samples are extreme overestimation of the duration resulting from samples containing different kinds of background noise, such as the musical accompaniment, or noise from the environment.

The data from this figure was then clustered by using a six components Gaussian mixture model (GMM) to group the noisy samples into one cluster. The number of components

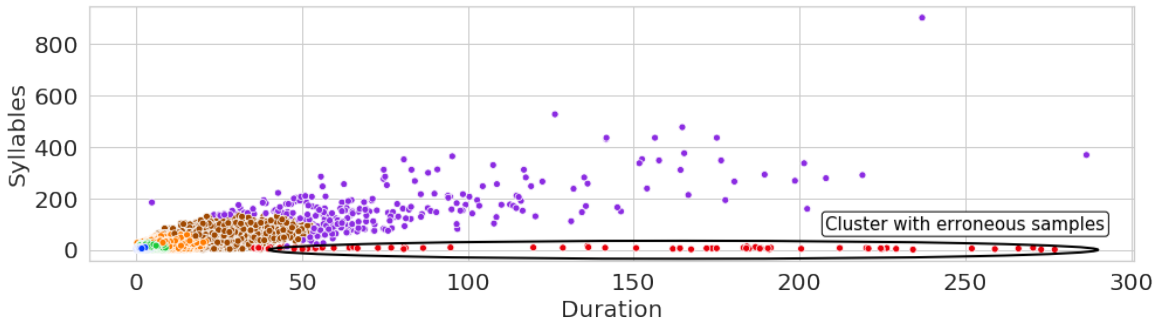


Figure B.2 Result of the GMM classification of DSing30 to identify the noisy samples.

selected was determined by experimenting with different numbers of clusters and utilising the Bayesian information criterion (BIC) to assess the different models. The BIC is minimised, and its gradient flattens at six components. Figure B.2 shows the resulting clusters from the SPS data estimated from the DSing30 dataset. DSing30 was selected because it is the most extensive set in the DSing corpus. It contains the DSing3 and DSing1, making the results of the clustering valid for all three sets. In the figure, most of the noisy samples were classified together in one cluster, providing a convenient way to filter out these samples for the SPS analysis.

After filtering out the noisy utterances using the *noisy samples cluster*, the utterances were grouped by song to calculate a song-level SPS estimate (SPS_{song}). Care was taken to reduce the impact of small segments in cases where their SPS deviated widely from the mean SPS of the containing song. This deviation can occur where noisy segments have not correctly filtered from the analysis. Figure B.3 shows one example of the resulting SPS_{song} of one song with two singers, one female and one male. This plot contains a graphical representation of the utterances interpreted by each singer plotted as a line. The location of the lines match with the moment in time that segment was interpreted, the length indicates the duration of the utterance, and the high indicates the estimated SPS. The mean SPS_{song} for this example is 2.9 syllables per second, represented by the dashed line. The initial segments in this example have a large SPS highly deviated from the average. These segments are not necessarily wrong utterances; instead, these are utterances with a different local speed than the rest of the song. The effect of these utterances can be reduced by dividing the total syllables from all sentences from one song, by the total duration of all the utterances. Equation B.1, present the formula for the SPS estimation for a song with N utterances.

$$SPS_{song} = \frac{syl_1 + syl_2 + ... + syl_N}{dur_1 + dur_2 + ... + dur_N} \tag{B.1}$$

This process resulted in 780 singing rates estimates. A histogram with the SPS_{song} from the DSing30 training set is presented in Figure 1, with an average (μ) of 2.08 SPS, and variance (σ) of 1.08.

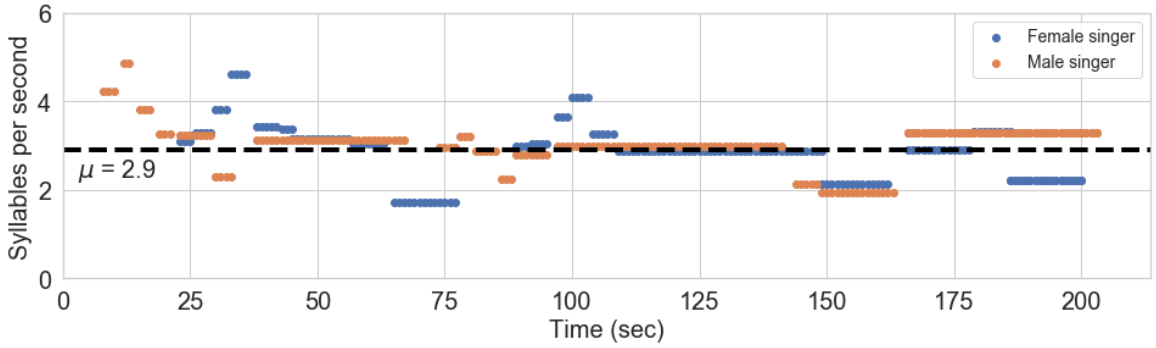


Figure B.3 Example of the song level syllables per second from one song, calculated from each utterance using Equation B.1.

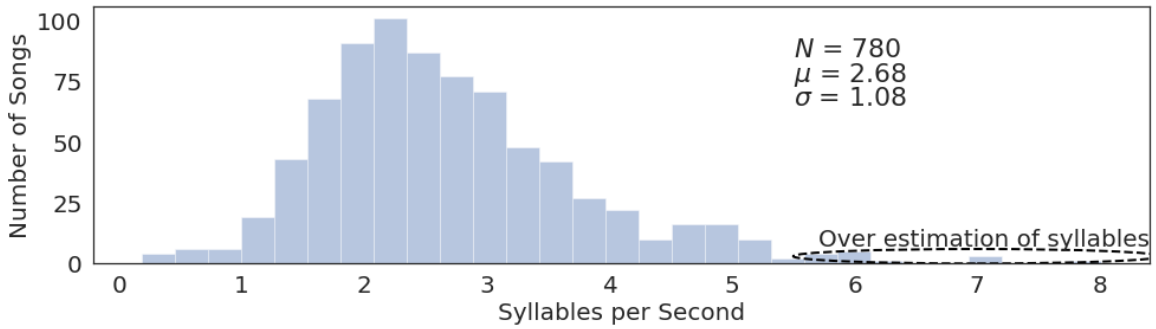


Figure B.4 Histogram of the averages syllables per second distribution of DSing30 training set.

Appendix C

Acoustic Latent Dirichlet Allocation

The acoustic Latent Dirichlet Allocation (aLDA) technique is based on the Latent Dirichlet Allocation algorithm, a statistical approach to discover latent topics in a collection of documents in an unsupervised manner (Blei et al., 2003), successfully used for acoustic scenarios by Kim et al. (2009) and Hu and Saul (2009).

To train the acoustic LDA system, it must first represent continuous audio data into a discrete representation called acoustic words. These acoustic words represent the vocabulary, and audio tracks represented using the audio words are the acoustic documents. These acoustic documents and words are used to train an LDA model. The topics' posterior Dirichlet probabilities from the acoustic documents are used to train a K-means model. In the selection stage, audio segments from a target domain are selected when the distance between the topics' posterior Dirichlet probability and any K-means centroid is lower than a defined threshold.

Defining the acoustic words is a crucial and non-trivial task (Kim et al., 2009). In this case, a Gaussian mixture model with 1664 Gaussian components was trained using a combination of 40 MFCCs and five perceptual features. The frames are then represented by the index of the component with the highest posterior probability.

Using the discrete audio representations, an LDA model was trained using 2048 acoustic topics. With the Dirichlet posterior from the LDA model, it was trained a K-means model with 1024 clusters. The Euclidean and cosine distances were used for the data selection with a threshold of 0.2 and 0.15, respectably.

Figure C.1 shows a diagram with the process used for learning the aLDA model for audio data selection.

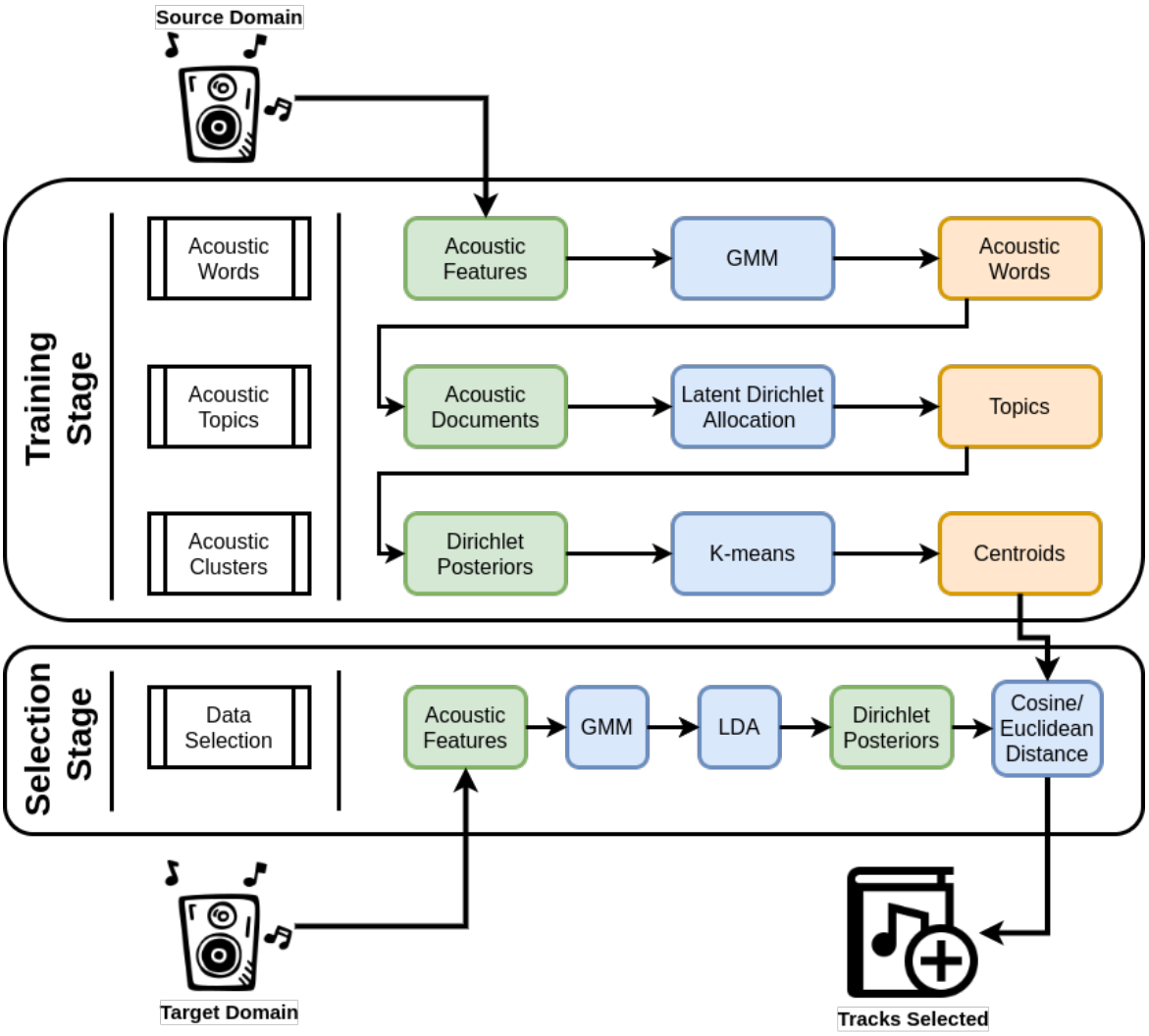


Figure C.1 Diagram of the process for audio data selection using acoustic latent Dirichlet allocation.