Speech Synthesis from Found Data

Pallavi Baljekar

CMU-LTI-18-004

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Alan W. Black (Chair), LTI, CMU
Florian Metze, LTI, CMU
Louis-Phillippe Morency, LTI, CMU
Heiga Zen, Google London

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy.

Keywords: Text-to-speech, Found Speech, Low-resource languages, Un-transcribed Audio, Prosody, Long-form Audio

To my Parents for their love, support and for being such inspiring role models,
To the magic of Alan W. Black (#blackmagic),
To Subhodeep and Gundappa, my two hiking poles in my hike through PhDLand.

Abstract

Text-to-speech synthesis (TTS) has progressed to such a stage that given a large, clean, phonetically balanced dataset from a single speaker, it can produce intelligible, almost natural sounding speech. However, one is severely limited in building such systems for low-resource languages where there is a lack of such data and there is no access to a native speaker.

Thus, the goal in this thesis is to use the data that is freely available on the web, *a.k.a, "Found Data"* to build TTS systems. However, since this data is collected from different sources, it is noisy and contains a lot of variations in terms of speaker, language and channel characteristics as well as prosody and speaking style. Conventional TTS systems on the other hand, require a large collection of clean, phonetically balanced, single-speaker data recorded specifically for the purposes of building TTS systems. This presents us with a number of challenges in using found data for building TTS systems within the current pipeline.

In this thesis, we address three of these challenges. First we look at data selection strategies to select good utterances from noisy found data which can produce intelligible speech.

Second, we investigate data augmentation techniques from cleaner external sources of data. Specifically, we study cross lingual data augmentation techniques from high resource languages. However, often found audio data is untranscribed. Thus, we also look at methods of using untranscribed audio along with unrelated text data in the same language to build a decoder for transcription. Furthermore, we address the issue of language, speaker, and channel variations, by training multi-language, multi-speaker models, in a grapheme based neural attention framework.

Lastly, since most of the speech data available on the web is in the form of audiobooks and podcasts, we explore iterative methods of learning a good prosody labelling for long form audio as well as learning prosody embeddings which mimic metrical structure of utterances in an unsupervised fashion to better match the acoustics of the data. In addition, we investigate methods of learning a set of *"prominence weights"* in attention based neural models, with the goal of improving prosody as well as overall quality of synthesized speech.

Acknowledgments

I have been extremely privileged to have had really supportive and inspirational teachers, mentors, friends and well-wishers who have supported and inspired me, provided me a ear to crib to and a shoulder to cry on. This thesis would not have been possible without their love, support and encouragement.

First and foremost I would like to thank my advisor for all his support and guidance and the great discussions that made my thesis possible. It was a real privilege being advised by him and being a witness to all of his magic. I would like to thank him for giving me the freedom to explore various topics, the freedom to break things and providing all of the necessary resources to run large-scale experiments and listening tests. It has truly been a magical experience working with him.

I am very grateful to my committee, Florian, LP and Heiga for being very supportive and available for discussion as well as for their very valuable time and feedback with my thesis. They were also very prompt with feedback and scheduling the defense dates which helped me finish my thesis on time. I would also like to thank Robert and Stacey for all of their support in ensuring things went smoothly and allowing me to finish my thesis on time.

I would also like to thank my masters advisors, Rita and Bhiksha, for their support during my masters and getting me acquainted with research at CMU. I would never have gotten to graduate school if it wasn't for my undergraduate professors, Kumara Shama, Anathakrishna and Hemant Patil, for inspiring the love of speech and signal processing and teaching me what research is all about. For this I am very grateful to them.

I am also indebted to all of the great collaborators and mentors I have had in CMU as well as during my internships. A very special thanks to my two Google internships in the Sound Understanding team and Text-to-Speech teams. My intern hosts, RJ Ryan and Heiga Zen provided two very fun internship experiences at Google. They are exceptional researchers and mentors, with a great work ethic one can only aspire to emulate. I am also very grateful to the other incredible members of the Sound Understanding and TTS teams at Google, Yuxuan, Ying, Daisy, Joel, Rif, Rob, Vincent, Yannis, Hanna, Markus and Chunan for making my internships at Google really really memorable and inspiring and helping me grow by providing me a different perspective on TTS, and inspiring many of the ideas in this thesis.

A very very special thanks to all of the members in Alan's group, Sai, Sunayana, Prasanna and Alok, for their help in my research and fruitful discussions. I am very grateful to them also for reading my thesis document and attending my practice talks! It has been a privilege being a part of the same group. I am also very grateful to all of my friends in CMU as well as Santosh Uncle and Prapti Aunty, for making my time at CMU and Pittsburgh really fun!

Last but not the least I would like to thank my family, Gundappa, Sub-

hodeep, my parents and grandparents for their love, support and inspiration, this thesis would not be possible without you. It would not be possible without the support that you gave me and the opportunities that it afforded without having to worry about backup plans and allowing me the freedom to dream up whatever future I wanted. Thank you! I feel extremely lucky, privileged and thankful to have the family that I have! I love you all!

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In text-to-speech synthesis systems, we model three parts of the vocal production mechanism. The movement of the vocal chords (as the pitch model), which models the source of excitation, and the movement of the vocal tract and articulators (the tongue, teeth and mouth) as the vocal tract acoustics and the duration models. The parameters for these models are learned from a clean, phonetically balanced set of recordings from a single speaker. The goal in text-to-speech synthesis is to combine these models to be able to predict natural sounding, intelligible speech from text.

However, if we want to build such a system as part of a speech-to-speech translation system to be used in a disaster relief effort in areas such as Nepal, Orissa or Pakistan where the languages spoken are dialects of Nepali, Oriya and Pashto, then we are severely limited in our ability to build such systems because we do not have access to a native speaker who speaks the language and is available for recording. On the other hand, we might want to build a personalized voice for a target speaker who has limited ability to produce speech such as people afflicted with motor neuron disease or vocal cord paralysis. Such a voice would need to match the physical and dialectical characteristics of the speaker. Both of these scenarios are examples where we would like to build TTS systems but are limited by an access to a large, clean phonetically balanced dataset of recordings from a single speaker, that matches our requirements.

To alleviate this problem, we propose to build systems from the abundant but noisy data available on the web. Speech data, especially from low resource languages, is becoming increasingly available on YouTube and other streaming services in the form of news and radio broadcasts, speeches, demo-videos, etc. In addition, with the growing popularity of neural networks and other big machine learning models, large multi-speaker datasets mainly for the purposes of speech recognition, have become available. Even if such data is unavailable on the web it can be crowd sourced via the web [Caines et al., 2015], examples of such corpora include the Babel corpus [Harper, 2011] provided by IARPA for multiple low-resource languages and the NCHLT Xitsonga corpus [Barnard et al., 2014] collected by the South African Centre for Digital Language Resources. In theory, this large variety of data available on the web would allow us to build voices matching the dialectical and physical characteristics of a particular speaker and also give us access to a huge variety of languages allowing us to build TTS systems for

low-resource languages.

Thus, the question is can this large quantity of phonetically imbalanced, noisy, multi-speaker corpora be used to build better synthesis systems that are understandable. Furthermore, can this data be integrated with existing clean data to give more natural sounding systems?

1.1 Thesis Statement

Thus, in this thesis, we will try to answer these two questions. We will explore techniques for building voices from found speech, *i.e.*, speech which was not recorded specifically for the purposes of TTS and is available freely on the Internet. we will further divide this problem into three parts:

- **Data Selection :** In this part of the thesis we will look at methods of selecting an appropriate subset of data and techniques for noise removal. For dataset selection, the goal is to select new utterances to insert into a small subset of seed utterances, such that it it is both representative of the target voice and informative to the existing model built with the seed utterances. Informativeness and representativeness measures will have to span categories such as acoustic (Mel cepstral features, durations, pitch, speaking rate, energy, etc), linguistic (linguistic features and units such as phone distribution, etc), speaker (age, gender, dialect) and channel (various noise types). In this regard, we will look at various subset selection techniques, such as rank-and-select based methods for different metrics such as mean cepstral distortion [Kominek et al., 2008] (a measure of acoustic prediction for TTS) and duration and find which measures can automatically allow us to select sentences to build intelligible TTS systems.

- **Data Augmentation:** This part of the thesis explores cross-lingual and cross-corpus adaptation techniques to leverage and use external, cleaner sources of data as well as other higher resource languages to augment the more noisy found data. We consider two cases. In the first case we assume the availability of transcribed training data. Thus, the main goal here is to first identify good sources of external data for augmentation and then identify how to integrate this in the TTS pipeline, after normalizing across channel, speaker and language.

  However, it might not always be the case that we have transcribed audio. This is often the case for low resource languages where you can find translated audiobooks in the language but it is hard to find transcriptions. If one has access to an ASR model in that language, we can get transcription of the audio. To build a decoder, we need to define a phoneset for the language. Thus, in the second part, we will explore methods of building an acoustic unit detector, which will involve segmenting the acoustic stream into small phoneme like units and learning the phonetic inventory, which can then be used as transcriptions to train a TTS system for an

unknown language.

- **Models for Long form Audio:** Most of the data found on the web is in the form of audiobooks or podcasts which contain prosodically rich, long form audio, where the prosodic realization of a sentence needs to be synthesized by taking into account longer term global context over multiple sentences. Thus, in this part of the thesis we will first explore traditional frame based RNN models for prosody on audiobooks. In the second part we explore an iterative strategy to improve prosody labelling. We also look at end-to-end models and propose a multi-scale attention mechanism for better prosody modelling. Finally, we look at augmenting these end-to-end models with prosody embeddings guided by learning a metrical structure in an unsupervised manner.

1.2 Contributions in this Thesis:

Previous methods have looked at building speech synthesis systems from audio books [Prahallad and Black, 2011], [Stan et al., 2016], [Watts et al., 2013] and ASR corpora [Yamagishi et al., 2010]. These systems have only looked at systems for languages with a Latin script and which one would not necessarily categorize as being low-resource [Stan et al., 2013]. In this thesis, the goal is to look at a variety of languages, English, Hindi, Bengali, Marathi, Tamil, Pashto and a variety of data conditions such as audiobooks, speeches, telephone speech etc.

In [Braunschweiler and Buchholz, 2011], they show that diverse speech with different speaking styles contained in audiobook data can be detrimental to the quality of synthesized speech and they show that by using certain acoustic and linguistic based measures one can identify and remove sentences that might be detrimental for speech synthesis. Along similar lines, in results shown in [Cooper et al., 2016b], they use low-level acoustic features such as speaking rate, pitch and energy to select utterances to improve naturalness of synthesized speech. In addition, previous studies have been done for unit-selection synthesis [Black and Lenzo, 2001] optimizing for coverage of linguistic units and phone coverage. However, most of these techniques have looked at measures that perform well on relatively clean, single-speaker datasets such as audiobook data [Braunschweiler and Buchholz, 2011] and radio broadcast corpus (BURNC) [Cooper et al., 2016b]. In this thesis we would like to look at an incremental approach to select good utterances from a variety of single and multi-speaker datasets which are suitable for synthesis and identify metrics that can detect noise found in found corpora, specifically mis-alignment errors and errors due to channel noise.

In addition, most previous techniques have concentrated on building average voice models using a variety of adaptive training techniques. Moreover, most of these adaptation techniques such as shared decision tree context clustering [Yamagishi, 2006] and speaker adaptive training such as MAP and MLLR based methods [Yamagishi et al., 2009] have concentrated on HMM based synthesis models. In this thesis, we would like to explore newer neural network based models which do not need explicit adaptation and

can be trained on multiple speakers and multiple languages in an adaptive framework [Swietojanski and Renais, 2016],[Li and Zen, 2016],[Fan et al., 2015].

Furthermore, cross-lingual adaptation has been used so far as a voice conversion technique to improve speech-to-speech translation for a particular speaker and assumes access to trained models in both languages [Wester et al., 2010]. However, in this thesis we would like to look at cross-lingual techniques to augment low resource data and build a robust system in the low-resource target language from found data which may or may not have transcriptions available.

Thus, this thesis takes a first look at building TTS systems from found data. In this thesis, we consider the problem from the three aspects; the data, how much performance improvement can one squeeze out of the data itself by selecting and pre-processing it intelligently. Second, we look at how one can integrate external resources such as resources from other major languages as well as cleaner resources from the same language to adapt to the noisy dataset and finally we look at various models for better prosody prediction of long form audio and methods to improve its labelling. Concretely, the contributions in this thesis are as follows:

- **Chapter 3: Data Selection:** In this chapter we explore various metrics to rank utterances based on its suitability for speech synthesis. We look at metrics spanning categories including linguistic (frequency of occurrence of phones), acoustic (Mel cepstral distortion (MCD), modulation spectrum, instantaneous frequency), duration as well as correlation based features. We show that selecting utterances based on Mel Cepstral Distortion and duration give the best results in terms of synthesis quality on both artificially degraded corpus as well as two corpora of found speech, one a single-speaker corpus of public speeches and the other a multi-speaker corpus of telephone conversations.
- **Chapter 4: Data Augmentation:** In this chapter, we explore best methods to augment our target low-resource found speech corpora with external data. We consider two scenario. In the first scenario, we assume we have transcribed speech and text available. In Section 4.1, we first explore various methods of augmenting cross-lingual resources in an end-to-end neural attention based multi-speaker, multi-language model. We show that attention based neural models with multiple languages benefit from augmentation with higher resource languages belonging to the same language family and can be adapted on a new language with as little as half hour of data when fine-tuned with the target data. We also show that these models tend to over-fit to the data and learn super sentential context. We also explore the robustness of these models on two corpora of noisy found multi-lingual data of Bengali and Gujarati speech and show that these models are significantly better than the clustergen models when trained with noisy data.

  In Section 4.2, we consider the case of augmenting from cleaner resources of the same language, when we have data from speaker's with different pitch, accent and speaking style. We first consider the effect of augmenting the training data with data that matches the target speaker's characteristics in terms of pitch and accent. We show that domain adaptation to the target speaker's voice by modifying the

training data has little effect, while adapting solely on the target speaker's data has a much larger effect in producing good quality of speech. We also explore building models with multiple speaking styles and style transfer experiments from clean to other conversational styles as well as from noisy to clean. We show that it is possible to transfer from noisy to clean speech and produces much more intelligible speech, than when using the original speaking style.

The second scenario we consider is a zero-resource setting, where we only have the audio and no other language resources. For these experiments, which are described in Section 4.3, we explore cross-lingual methods of building an acoustic decoder, the first step of which is acoustic unit discovery. We consider both a clean and noisy multi-speaker dataset of Hindi and Xitsonga, respectively, and show that our method of cross-lingual *"inferred phoneme"* method out performs phonetic-decoding when using a clean multi-speaker corpus, but is not very well suited to a very noisy datasets such as crowd-sourced telephone conversations.

- **Chapters 5 and 6: Prosody Models for Long-form Audio:** In the final two chapters we explore neural network approaches to modelling long-form audio. In Section 5.2, we explore traditional frame-based approaches to modelling long-form audio with various RNN based sequence models. We show that these models are good at intelligibility, however, they fall short on modelling naturalness well. Thus, in Section 5.3, we look at methods to improve the labelling of prosodic features as well as adding in prosodic relation features inspired by learning metrical tree relations in order to match the text better to the tune of the acoustics that are actually present in the data.

  In Chapter 6, we explore unsupervised methods of improving prosody modelling in neural attention based models. First in Section 6.2, we explore two methods of introducing a notion of *"prominence weights"* in neural attention models. In the first method we use a multi-headed, multi-hop, multi-scale attention mechanism (3M attention) to learn *"where"* and *"when"* to focus and in the second we use the errors from the decoder to weights the inputs to the post-net, a mechanism we call *"error-feeding"*. In the second part of this chapter, we augment the neural attention model with another prosody encoder, that learns an utterance-level prosody embedding, guided by metrical theory. We show that even though addition of the prosody embedding does not improve synthesized prosody significantly, the embedding space learns some notion of quoted *vs.* narrated utterances for syllable inputs.

1.3 Thesis Organization

This thesis is organized as follows. Chapter 2 gives an introduction to characteristics of found speech and lays out the background of previous work done in this area. Chapter 3 discusses the data selection strategies. Chapter 4 discusses cross-lingual and same language methods of data augmentation for transcribed data and shows results on a number of noisy datasets. Furthermore, Chapter 4 also discusses the acoustic unit dis-

covery method for untranscribed audio. Chapters 5 and 6 discuss various methods of refining prosody models for both frame based and end-to-end neural frameworks respectively. Finally, Chapter 7 summarizes the work done in this thesis and lays out future directions and extensions.

Chapter 2

Found Speech in Speech Synthesis

This chapter first describes the TTS pipeline and then describes some of the adaptation techniques that have been used in speech synthesis so far. It then describes what found speech is and what are the characteristics of found speech.

2.1 Background

There are broadly two ways of synthesizing speech. The first is unit-selection speech synthesis, which mainly involves stitching together small chunks of speech units to render the waveform after applying various signal processing techniques. The second method which we will be concentrating on in this thesis involves training models to predict vocoder parameters which can then be synthesized into waveforms.

2.1.1 Traditional Statistical Parametric Speech Synthesis (SPSS) Pipeline

Traditional statistical parametric speech synthesis involves a training stage and a synthesis stage. Within the Festival system, we use the CLUSTERGEN framework [Black, 2006] for training and synthesis. The pipeline used in the Festival [Black et al., 2014] system is shown in Fig. 2.1. As can be seen, there are three main parts in this system.

- **Text Analysis:** This stage which also forms the front-end of the TTS pipeline, involves converting the raw text into a form that can be used to train acoustic and duration models. Thus, in this step, the raw text is converted into a heterogeneous relation graph (HRG) [Taylor et al., 2001] from which we can then extract linguistic features as desired. Each utterance in the training dataset is represented as an HRG. This utterance structure [Taylor et al., 1998], consists of relations and items. The *Relations* as defined in [Taylor et al., 2001] are graph structures which contain nodes and named edges. They can be trees, lists, or other structures. All "atomic linguistic entities such as words, syllables and phones are represented as *items*. They contain linguistic information in the form of features. New relations and items are added onto the utterance structure as the TTS processing proceeds until waveform generation.

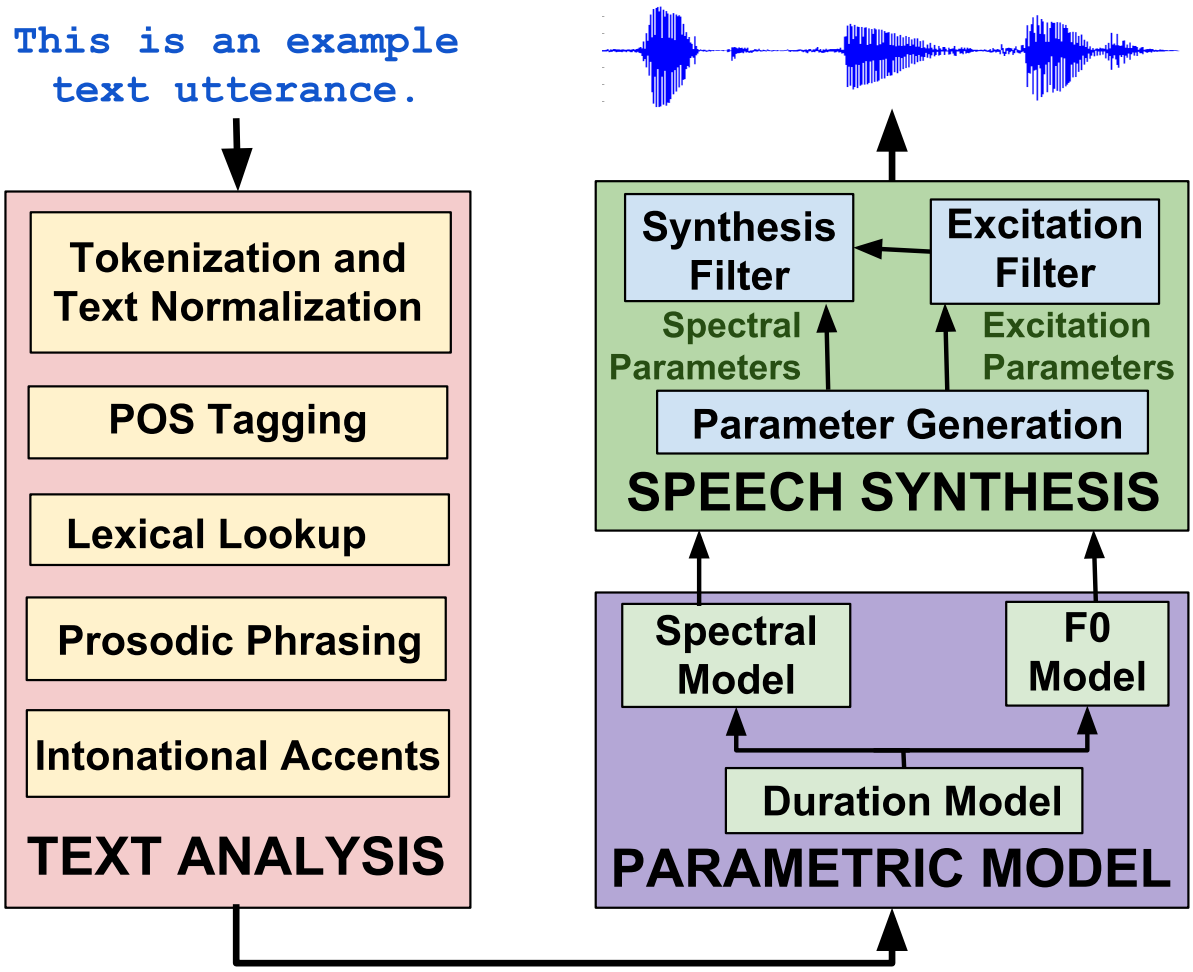


Figure 2.1: *Overview of the TTS Pipeline*

Thus, to get linguistic features, first the raw text is converted to tokens and normalized, where the abbreviations and numerical forms are expanded. Then, we extract POS tags. Then the characters are expanded to the phones using the lexical look-up if the word is present in the lexicon, otherwise we use letter-to-sound rules to expand the word [Black et al., 1998], [Sitaram, 2015]. Prosodic phrasing is then predicted based on punctuation rules and other data-driven approaches [Parlikar, 2013]. Pitch accents are predicted on top of these prosodic breaks, using simple rules or data-driven approaches [Anumanchipalli, 2013]. The linguistic features are then extracted from this hierarchical graph. These linguistic features contain both numerical questions such as identity of the linguistic unit, POS tag, intonational features as well as numerical values relating to the phone type, duration, lexical stress and other features relating to the characteristics of the articulators used in producing the phone.

- **Parametric Models:** After the linguistic features are extracted they are used by the parametric models to predict the vocoder parameters. The parametric models mainly consist of three models. The F0 model which predicts the F0 parameters, the acoustic model which predicts the acoustic parameters such as MCEPs, LSPs etc and the duration model which predicts the duration of each HMM state. Then the MCEP relation is added to the hierarchical relation graph and frame-wise linguistic features are extracted.

  Training: To train these models, we require both speech and its transcription. The audio is first segmented to get the sub-phonetic segmentation. This segmentation also serves as the output on which the duration model is trained. Then appropriate F0 and acoustic parametrization are derived on a frame level (at 5 *ms* frame shifts) and labelled according to the output of the decoder. These labelled frame-wise features then form the outputs for training. The text is made to undergo the text analysis and linguistic features are extracted on a per-frame basis. Given the frame-wise linguistic and acoustic features, CART trees[Breiman et al., 1984] are trained to cluster speech frames having similar linguistic contexts for each HMM state. The mean and variance of the pooled data at each node is calculated.

  Synthesis During synthesis, given the linguistic features, first the durations per HMM state are predicted from the duration model. Then the means for those segments are predicted from the acoustic model and the F0 model. Given these parameters for the utterance it is then sent to the next stage.

- **Waveform generation:** Since the means are predicted on a per HMM state basis, the contour that is generated, is very monotonous per HMM state with rapid changes at the boundaries. In order to make it more like how human speech is produced by smoothly changing articulator positions, speech parameter generation algorithm [Tokuda et al., 2000] is applied to the trajectories in order to make it smoothly varying within a state and not have rapid changes at the boundaries. Optionally, after this stage a post-filter such as described in [Toda and Tokuda, 2007],[Li and Atlas, 2005], [Yoshimura, 2002] can be applied to counter over-smoothing effects in the high frequency regions and then it is sent to the synthesis

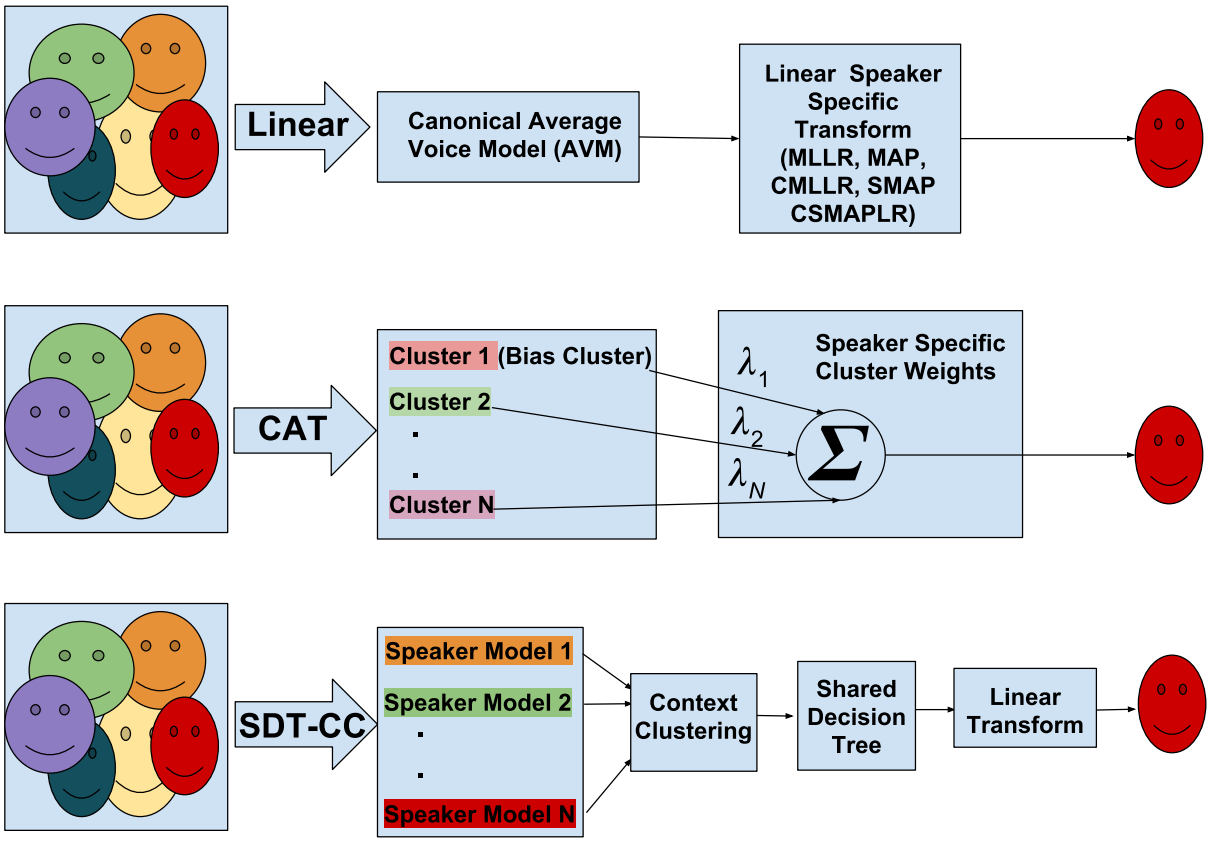


Figure 2.2: *Different Speaker Adaptive Training (SAT) Techniques.*

filter [Imai et al., 1983] to re-synthesize the waveforms.

More recent techniques use neural network models to model the F0, acoustics and duration, predicted either separately or jointly. This mainly involves replacing the CART trees with Neural network models [Zen, 2015]. The advantage of Neural Network models is that the data does not get split for each phone, since a single model is trained on all of the data.

2.1.2 Adaptation Techniques in SPSS

Adaptation techniques in TTS, can broadly be classified as feature based adaptation techniques and model based adaptation techniques. There is a large body of work that has been investigated previously in building average speaker independent models in TTS and applying different methods of both speaker and cross-lingual adaptation [Yamagishi, 2006], [Yamagishi et al., 2007a], [Yamagishi et al., 2003b], [Yamagishi and Kobayashi, 2007], [Yamagishi et al., 2007b], [Wester et al., 2010], [Yamagishi et al., 2009]. I describe some of these model based methods below.

Linear Methods

These methods are the most widely used methods for speaker adaptation in TTS [Yamagishi, 2006], [Yamagishi et al., 2008], [Yamagishi et al., 2007b]. It involves first building an average voice, by pooling all the speakers together and then applying some speaker specific linear transform to the model parameters to adapt the average canonical model to the speech of the specific speaker. Maximum Likelihood Linear Regression (MLLR) [Gales, 1998], involves transforming the means and sometimes variances of the model by applying an affine transformation to the mean/variance of the average model. There is a constrained version of this called CMLLR which constrains the variance to have the same form as the mean. Maximum aposterior adaptation technique is the other technique that is used which maximizes the posterior of the Gaussian parameters. However, MAP is local and applied to only the Gaussians seen in the training data. While MLLR applies the same transformation to all the tied states. There are other variants such as Structure MAP, which applies MAP to tied states and makes the prior the parent of the leaf node in the tree [Yamagishi et al., 2009]. Similar linear transforms can directly be applied to the features instead of the model.

Cluster Adaptive Training

This method involves learning a set of basis functions in order to describe the input space [Gales, 2000], [Zen et al., 2012], [Wan et al., 2014b], [Wan et al., 2014a]. The advantage of using cluster adaptive training is that it can factorize out the various conditions seen in the training data. Adapting a speaker primarily involves learning a set of cluster weights for each speaker and thus requires less adaptation data.

Shared Decision Tree Context Clustering

This involves building separate speaker specific models and then building a common tree with common questions from the speaker dependent models. This shared decision tree is adapted to the speaker and then speaker adaptive training to a specific speaker [Yamagishi et al., 2003a] is applied to this shared tree. The disadvantage in using this method is that it requires enough training data from each speaker to be able to build a speaker dependent voice.

2.1.3 Evaluation Metrics

In speech synthesis we use both objective and subjective measures for evaluation. Furthermore, the two main criteria that TTS systems are evaluated on is naturalness and intelligibility. In this thesis, we choose to use MCD as the objective measure to evaluate the intelligibility of the TTS system and subjective tests such as MOS and A/B testing to test user's preferences, which would depend on both the naturalness of the voice as well as how intelligible it is. Some of these measures that we use to determine the quality of our TTS systems are described below.

Objective Metric

We use Mean Cepstral Distortion [Kominek et al., 2008] as the objective metric. It is a weighted root mean square error between the predicted v^{pred} and the original acoustic vectors v^{ref}. For an utterance containing a sequence of frames $v_d(t)$, from 0 to T and feature dimension 0 to D, It is defined as:

$$MCD(v^{pred}, v^{ref}) = \frac{\alpha}{T} \sum_{\substack{t=0 \\ ph(t) \notin sil}}^{T-1} \sqrt{\sum_{d=s}^{D} (v_d^{pred}(t) - v_d^{ref}(t))^2}$$

where, $\alpha = 6.14185$ and T is the minimum number of frames in the shorter of the two files. In this thesis, we denote the MCD metric calculated using ground-truth durations as MCD. However, when we do not have a separate duration model as is the case with neural attention models, we directly compare the synthesized wavefile with the ground-truth wavefile. In this scenario, we use the dynamic time warping (DTW) algorithm to first make the sequences of acoustic vectors derived from the wavefiles to be of similar lengths and then calculate the MCD between these. In this thesis, this measure is denoted as DTWMCD. Both the MCD and DTWMCD metrics are dataset dependent and cannot be compared across datasets.

Subjective Metrics

Subjective results are mainly evaluated using A/B preference tests or Mean Opinion Scores (MOS). The MOS score is a number on a scale from 1-5 where the user is asked to rate how well they prefer a particular synthesized utterance. Natural vocoded speech is generally rated around 4.5. A/B preference tests are comparison tests that compare some baseline system with the new system.

2.2 Found Data

Any data that is available readily in the public domain is called *found data*. This includes data from audiobooks, public speeches, news and radio broadcasts, YouTube data and telephone conversations. This data has large variations in its type and characteristics.

2.2.1 Characteristics of Found Speech

Single speaker databases such as audiobooks, public speeches, channel broadcasts on YouTube, *etc.*, have the advantage that they are spoken by the same speaker. However in terms of diversity of the data they have their own challenges. Audiobooks for instance, have large variations in the prosody and intonation as well as the speech rate. Even though, these variations in prosody make for interesting listening, they are difficult to model in current speech synthesis systems. Moreover, since current systems are built using single isolated utterances, they do not account for long range dependencies as seen

in audiobooks over paragraphs and sections.

On the other hand public speeches and YouTube broadcasts have a lot of channel noise due to variations in recording conditions such as variations in microphone characteristics, room acoustics, distance from microphone, *etc*. These types of databases require better data pre-processing and feature adaptation and normalization techniques which will be robust to channel conditions.

Multi-speaker databases include news and radio broadcasts, telephone conversations and voice search data. These types of utterances are generally short, contain a lot of fillers and non-speech sounds and distortions.

News broadcasts generally have each segment consisting of multiple speakers and having many queuing audio sounds like sequences of music and other data. In terms of building TTS systems with such relatively clean, multi-speaker corpora we need to look at various voice averaging and voice conversion techniques to average over the different speakers in the database in addition to data pre-processing techniques in order to select an optimal subset for synthesis and remove speakers who are very different and not representative of our target speaker.

Telephone conversations on the other hand provide a very rich set of data with multiple issues. They have low sampling frequency and so are of low quality. Furthermore, they contain lots of channel distortion, background distortions and noisy utterances containing long periods of silences between utterances, which causes problems with labeling and alignment. The advantage of using this type of data is it can be easily obtained. This kind of data needs good pre-processing methods in order to be useful to build TTS systems which do not fail during training.

Thus in the subsequent chapters we will see ways of handling this variability in different types of found data, by selecting the right data (chapter 3), using better external resources (chapter 4) and using better prosody modelling techniques (chapters 5 and 6).

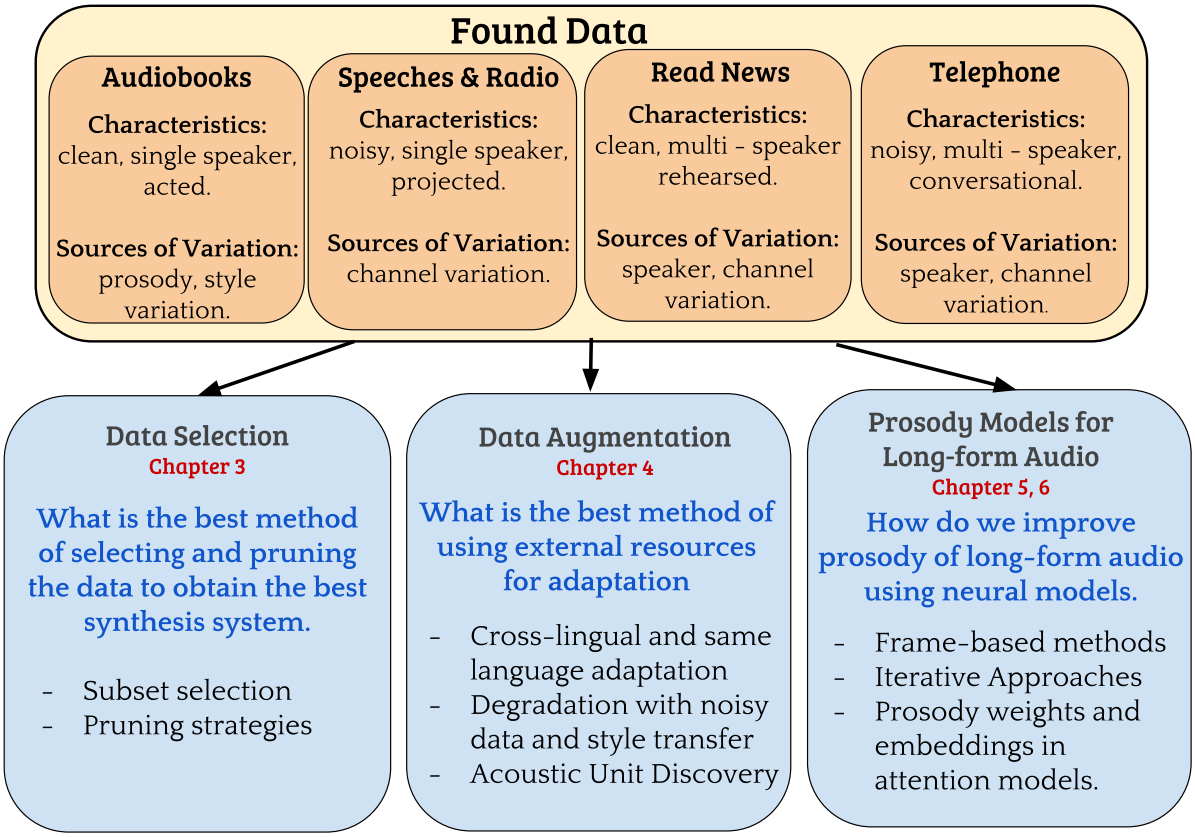


Figure 2.3: *Found Data*

Chapter 3

Data Selection

3.1 Introduction

In the last chapter we saw that the nature of found data is such that it contains a lot of channel and speaker variability. Most text-to-speech systems are ill-equipped to handle this variability and noise. For instance, it is hard to build an intelligible voice from a data pool of 30 hours of poorly recorded, multi-speaker telephone speech, however it is possible to build an intelligible voice from ten minutes of cleanly recorded, phonetically balanced data from a single speaker.

Thus, our hypothesis is that given a noisy dataset, there is an optimal subset of that data that will allow us to build the best TTS system as opposed to using all of the noisy data in the dataset. The goal in this chapter therefore, is to formulate methods and metrics in order to select appropriate data for training TTS systems.

Thus the main goals in this chapter are:

- Identify measures which are good selectors of the data to retain for synthesis.
- Find what is the best level at which to prune.

3.2 Related Work

Most of the work relating to data selection techniques for TTS has been investigated previously for unit selection synthesis, where the main goal is to optimize for phonetic coverage suited for unit selection. In [Black and Lenzo, 2001] they use a greedy, bootstrap process which weights clusters based on how frequently they are used, and then greedily selects utterances that have optimal coverage with respect to this frequency weighted cluster tree. In [Van Santen and Buchsbaum, 1997] they used a greedy approach to the set covering algorithm by iteratively selecting the sentence with the most diphone units and then removing those diphone units from the set and in this way, selecting the smallest set of utterances that cover the phonetic inventory at least once. In [François and Boëffard, 2001] they follow a similar greedy procedure, wherein they prune out units

which are rare and also repeated triphonemes which are frequent.

Other data selection methods in speech recognition involve active learning methods for selecting new data to be added to an existing model. These techniques are generally used to select utterances to be labelled by annotators as in [Hakkani-Tür et al., 2002], wherein utterances that generally give low confidence scores by the decoder are selected to be labelled by annotators. On the other hand, in some unsupervised settings, the recognizer is adapted by using the recognition results of sentences which have high confidence scores as in [Kemp and Waibel, 1999] or they can be combined in a semi-supervised setting as in [Tur et al., 2005], where the recognition results of the sentences with high confidence scores are combined with human annotated data for the ones with low confidence scores. However, these entropy based methods select for diversity, while confidence based measures are more geared towards representativeness. In [Itoh et al., 2012], they combine measures of informativeness along with representativeness to select a set of good utterances. They use N-best entropy for each utterance as a measure of informativeness and a TF-IDF similarity between phones as a measure of representativeness.

There have also been active learning based approaches applied to selecting a good subset for automatic speech recognition(ASR) and keyword spotting (KWS) for low resource languages. In [Wu et al., 2007], they try to obtain a uniform distribution over the linguistic units (words and phonemes) by maximizing the entropy over the distribution of these units. While in [Fraga-Silva et al., 2015], they used different variants of entropy based criterion on words, triphones, phones and HMM states. They find the best performance when using a mutli-stage approach, specifically, first using letter density to pre-select and then using acoustic entropy of HMM states for final selection.

Techniques involving maximizing a sub-modular function have recently become famous for data subset selection in ASR [Wei et al., 2014b], [Lin and Bilmes, 2009], [Wei et al., 2014a] and machine translation [Kirchhoff and Bilmes, 2014]. However the main goal here is to ensure minimum loss in performance while using much less data. In these methods they use an appropriate submodular function to measure the usefulness of each subset. In [Lin and Bilmes, 2009] they use a Fischer kernel based word similarity measure derived from acoustics. This maximizes for the representativeness. While, in [Wei et al., 2014b], they combine this representativeness measure which they call the facility function with a diversity reward function which basically splits all the utterances into K disjoint subsets and then it is formulated so as to maximize a function that encourages selecting from different blocks, thus also maximizing diversity in the subset.

The advantage of using submodular functions is that they have the nice property of *diminishing returns* and so implicitly are able to maximize for both representativeness, and diversity. However, since our goal is to minimize some objective measure which quantifies the quality of synthesized speech such as MCD, formulating it as a sub-modular function is difficult.

However, there has been very less work done in the area of selecting utterances from noisy speech for TTS. Some earlier work which was done in [Cooper et al., 2016b] and [Cooper et al., 2016a], focused on selecting utterances to get natural synthesized speech. Their selection metrics relied on low-level acoustic descriptors such as statistics of energy,

fundamental frequency, speaking rate and articulation. They found the level of articulation to be the best indicator and found that by removing hyper-articulated sentences and adapting to hypo-articulated sentences gave good results on female speech, which was not true for the male subsets. Their goal was to optimize for the naturalness of speech synthesized from the selected data.

The goal in this work is to select data to optimize for intelligibility, by measuring how well it does on MCD. Moreover, the goal in this work, is to use speech from low-resource and noisy datasets.

3.3 Experimental Framework

In this chapter, we would like to address the question as to whether all data is good data or can we find an optimal subset of the data that gives us the best system as measured in terms of Mean Cepstral Distortion (MCD) [Kominek et al., 2008].

For preliminary investigation, we broadly categorize the error types found in such data, into two main types, *misalignment errors* and errors due to variation in *channel conditions*. Misalignment errors occur because certain types of sounds are not described in the transcript such as claps, laughs, coughs, breaths, etc. The second type of error is caused due to varying microphone conditions, channel noise *etc*. Misalignment errors are never good for the system and we would like to detect and remove utterances containing such errors from our training data . However, errors caused due to channel variation and noise, might in some cases be good for training our models, adding to the diversity. Thus our goal here is to find good measures which detect for the misalignment errors and bad utterances which will be detrimental to the system, while retaining the sentences which even though they might not be representative of the training set, might still provide valuable and diverse characteristics to the training data.

We show results on experiments carried out on an artificially degraded dataset to address the issues of misalignment errors and errors caused due to large non-representative channel conditions. We also evaluate the best metric found on the artificially degraded corpus on two corpora of found data, a single speaker corpus of public speeches and multi-speaker dataset of telephone speech.

3.4 Data

3.4.1 Artificially Degraded Clean Corpus

Since the goal is to evaluate metrics to detect misaligned data and data that is very different from the majority of the target domain such as data recorded in very noisy environments we created this dataset to act as ground truth and provide us an insight into how well different metrics perform.

To simulate misaligned data we shifted 100 utterances in the ARCTIC dataset for speaker RMS. For instance, the acoustics of utterance 1 corresponded to the transcription

for utterance 10. To simulate varying channel conditions, we convolved another set of 100 utterances with an impulse response recorded inside a chamber. We then mixed the two sets to simulate having both misaligned data and channel noise in the data. The total duration of this data was about an hour of speech (1000 utterances) out of which ten minutes (100 utterances) were misaligned and the other 10 minutes had very different channel characteristics.

3.4.2 Single-Speaker Found Data

The single speaker dataset was created using speeches from the former American President, Barack Obama. We used subsets of three different speeches. The third speech was chosen to be very noisy with a lot of extraneous sounds such as claps and containing a lot of reverberation. The other two speech subsets were much cleaner in comparison. The total number of utterances in the dataset was 495 utterances totaling about 1 hour of training data. This subset of utterances was selected after running interslice [Prahallad and Black, 2011] on each of the speeches and selecting utterances that had a minimum of four words. The utterances in this dataset tended to be longer on an average than the ARCTIC dataset.

3.4.3 Multi-Speaker Found Data

For the multi speaker found corpus we used the CallHome dataset and created two gender specific sets. The CallHome corpus consists of 30 minutes of unscripted telephone conversations. For these experiments we used only the primary speakers from each of the conversations. We first created a clean set of about 400 utterances by removing sentences transcribed as containing laughs, breaths, fillers and other channel distortions. The noisy version consisted of about 900 utterances. The age range varied from 8 to 70 years for these conversations. The data was further pre-processed to remove too short or too long utterances.

3.5 Rank and Select Methods

Since the main aim in this chapter is to find a *measure of goodness* of an utterance to be selected for training, we investigate various metrics for selecting the best subset of utterances that will help us build a good TTS system. In addition, we investigate two scenarios with respect to our experiments. Scenario 1 is where there is a seed set of utterances available, while scenarios 2 involves building a model, assuming no seed set is available.

3.5.1 Seed data selection

For the artificially degraded corpus we assumed the phonetically balanced subset of 100 utterances of the ARCTIC RMS utterances to be the seed. For the noisy datasets, since

we did not have a seed dataset to begin with, we investigated different methods of finding this small seed subset of good utterances. We investigated using only acoustic measures such as the 100 best performing utterances in terms of MCD, and on the other hand purely optimizing for coverage of linguistic sub-units such as obtaining phonetically balanced subset by counting the frequency of occurrences for each HMM state. We also tried a combination of the two, however, we found that optimizing for linguistic coverage gave the best results in terms of MCD on a held out test set.

3.5.2 Voice Building

For all of the experiments in this chapter, we have used CLUSTERGEN [Black, 2006] for the parametric speech synthesis. In addition, we have only used the base voice building tools without using Move-Label [Black and Kominek, 2009] and Random Forests [Black and Muthukumar, 2015]. Using move-label or random forests might improve performance, however, just using the base is probably sufficient to deciding the best examples. Moreover, in [Black and Muthukumar, 2015] the results show that the performance improvement is uniform over the base model and the score for a single model is strongly correlated with the score for RFS models. Thus, we only use the base model as it is faster and the improvements by move-label and random forest will be uniform over the simpler models, so it will not make a difference with respect to data selection.

3.5.3 Metrics

We evaluated various metrics such as duration, spectral measures and other cross-correlation based measures that could be directly calculated from the synthesized wavefiles.

Utterance Mean Cepstral Distortion (UMCD)

The mean cepstral distortion [Kominek et al., 2008] is a weighted Euclidean distance between the true and the predicted Mceps and is evaluated for each predicted frame. We score each utterance by the frame-wise MCD averaged across the utterance. The main intuition in using the MCD was because this was a direct measure of the frequency content of the signal and thus a higher MCD would imply the synthesized wavefile is further away from the true wavefile in terms of predicted Mceps on the trained model. Thus, a really high MCD would imply that the acoustics in the utterance are not being modeled well and so would indicate misalignment errors and errors that are not representative of the majority of the training data. Note that this MCD is calculated on an utterance level, so as to select good utterances. This is different from the MCD calculated on the entire held-out test set which is used to evaluate how well the voice has been modelled as plotted in Figs. 3.1 - 3.5.

Duration

For durations, we used the root mean square error between the predicted duration for each senone in the utterance compared to the *true* label given to an utterance after training 30 iterations using Baum Welch. The predicted durations were obtained from two models, one was from the entire noisy dataset and the other was on a model trained with a small seed set of a 100 utterances. The main goal here was to eliminate the really bad utterances which would in turn have bad labeling. This measure was expected to give good results on the misaligned data.

Modulation Spectrum

Since the modulation spectral trajectories capture the temporal dynamics of components of the spectral envelopes [Hermansky, 1998], we decided to investigate it as a global indicator of differences in spectral dynamics between the true and synthesized wavefiles. Moreover, using the Modulation Spectrum as a postfilter has shown gains in synthesis [Takamichi et al., 2016]. Thus, we expected this metric to be an informative measure in capturing large channel differences between recordings. We scored each utterance with the mean error between the modulation spectrum of the true and the synthesized wavefile in order to obtain a score per utterance.

Global Variance

This was another global spectral measure we tried since it improves results when used as a post-filter [Toda and Tokuda, 2007]. The idea here was to investigate whether differences in global variance of the predicted Mceps as compared to the true ones are correlated to the noisiness in the data.

Cross-Correlation Based Measures

The main intuition in using this metric was to detect misaligned data. If the two sentences are similar they should have a high peak when the two wavefiles, the true training wavefile and the synthesized wavefile are cross-correlated. We experimented with three cross-correlation based measures. We simply cross-correlated the two wavefiles and used the maximum of the resulting cross-correlated sequence as the measure. We also experimented by cross correlating the Teager Energy operator of the two wavefiles and the Hilbert envelope. The Teager energy operator gives a running energy estimate [Kaiser, 1990] of the wavefile and is supposed to capture the energy of the system that produced the speech rather that the energy in the speech itself. Thus, we hoped that this metric would also be helpful in capturing channel distortions. The Hilbert envelope on the other hand, computes the discrete-time analytic signal of the real part of a complex signal. The intuition in using this measure was to make it easier to detect the differences in misaligned data, since it can be used as a correlate to the envelope of the speech signal.

Instantaneous Frequency

The instantaneous frequency is supposed to capture the the average of the sinusoids at each point in time in a signal. The utterances were scored by taking the mean of the absolute difference between the instantaneous frequencies in the synthesized and the true wavefiles. So we would expect that signals that are similar acoustically will have smaller differences in error between the true and synthesized waveforms while signals differing in acoustics will have higher errors.

3.6 Empirical Evaluation

This section discusses the results on artificially degraded speech as well as noisy speech. We first describe the evaluation of various metrics on the artificially degraded data for both types of errors and a combination of the two. These metrics are evaluated on how accurately they can detect the artificially misaligned sentences and sentences that have been degraded with noise. The best performing metric is then used to iteratively select 10% of the best utterances which are then used to re-train the models. The models can be retrained at each iteration by re-clustering the state models while assuming fixed segmentation or the models can be re-trained by re-aligning the models each time using only the selected utterances and then re-clustering based on the new segmentation obtained. The baseline is calculated as the average MCD error on a held out test set from a model trained on all of the noisy data. In addition, we also investigate how this metric scales with larger amounts of noisy data.

3.6.1 Metric Evaluation

We carried out two sets of experiments. The results in Table 3.1 show results when using synthesized wavefiles obtained from a system trained with a small almost phonetically balanced seed dataset of about 100 utterances (approximately 10 minutes of speech). The results in Table 3.2 show results assuming the wavefiles are synthesized from a model trained on the entire dataset. Since the goal is to find the best metric which can detect the noisy utterances, we have reported the detection accuracy of each metric in detecting the artificially degraded utterances. Thus an accuracy of 90 implies that 90 out of the worst 100 sentences by some metric were the artificially degraded sentences.

We see that the MCD and duration parameters outperform all of the other metrics on both misaligned data as well as channel noise. It makes sense given the fact that these are the two main parameters about the vocal production mechanism that we model. Moreover, we see that both of these measures do better on the model trained with the entire noisy dataset, because on the noisy dataset, the really bad utterances will be synthesized wrongly, while the good ones that fit the majority of the data will be synthesized nicely. However, it is not guaranteed that the model trained on the seed dataset encompasses all of the diversity in the data and so might even reject utterances which might provide it information and make it a better model.

Table 3.1: *Evaluation metrics on artificially degraded set assuming a model built from a* **small seed set** *of utterances. (% Accuracy of detection, here a 99% accuracy implies that 99 out of a 100 degraded sentences were detected as being noisy)*

Metric	Misalignment Noise	Channel Noise	Misalignment + Channel Noise
Utterance MCD	**99**	**88**	**95.0**
Duration	85	36	63.0
Modulation Spectrum	51	27	35.5
Global Variance	59	4	34.0
Cross-corr	87	1	32.5
TEO Cross-corr	20	43	51.5
Hilbert Env. Cross-corr	45	1	34.5
Instantaneous Freq.	53	2	17.5

Table 3.2: *Evaluation metrics on artificially degraded set assuming a model built from* **all** *utterances. (% Accuracy of detection, here a 99% accuracy implies that 99 out of a 100 degraded sentences were detected as being noisy)*

Metric	Misalignment Noise	Channel Noise	Misalignment + Channel Noise
Utterance MCD	**100**	**94**	**96.5**
Duration	87	43	67.0
Modulation Spectrum	21	25	31.5
Global Variance	86	4	47.0
Cross-corr	44	1	30.5
TEO Cross-corr	36	51	55.5
Hilbert Env. Cross-corr	39	1	33.5
Instantaneous Freq.	13	3	16.0

The cross-correlation metrics do much better on the model trained with the clean seed subset of about 100 utterances. We find that it is much easier to detect misaligned

data than it is to detect channel mismatch. The Teager energy operator is quite successful in detecting channel mismatch, as compared to all of the other correlation metrics.

3.6.2 Re-alignment vs. Re-clustering

The subset of utterances selected was used to either re-cluster and re-align the models to obtain labels suited to the iteratively changing data subset or fix the labels by training on the entire corpus of noisy data and then only re-cluster based on the utterances selected with the MCD metric. Figures 1-4 show plots of iteratively selecting the best 10% of utterances and using these utterances to either only re-cluster the state models (blue line) or re-align at each iteration and re-cluster based on new segmentation obtained (pink line).

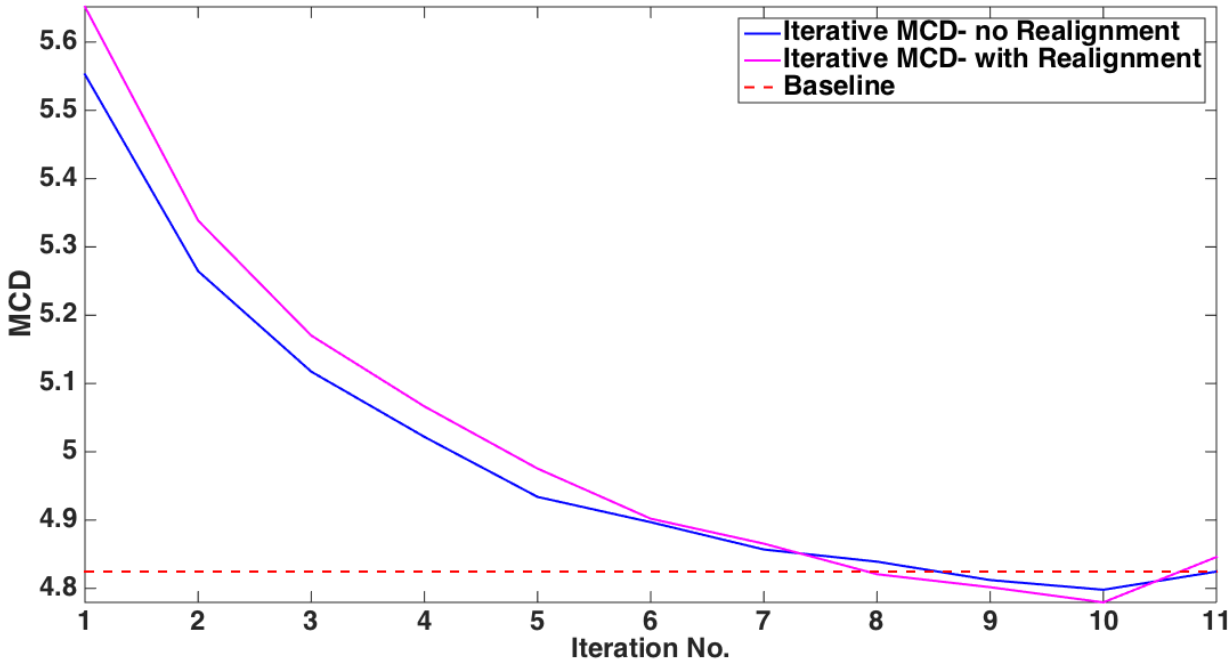


Figure 3.1: *Iterative MCD for artificially misaligned data*

We find that realigning the labels each time helps and results in a much lower MCD on both of the single speaker datasets as illustrated in Figures 3.1 and 3.2. However, the trend is opposite in case of the multi-speaker corpora for both males and females as shown in Figures, 3.3 and 3.4. This might be because each subset has a different set of speakers and having labels trained from the entire data set might be better than having labels from a smaller set which does not encompass all of the diversity in the data. In addition, we find that in case of the noisy datasets as seen in Figures 3.2, 3.3 and 3.4, we see that the results with realignment are not monotonically decreasing upto a certain point and have a slightly erratic behavior. This might be because at every iteration, the alignment shifts based on the limited amount of training data, which in some cases might be representative of the test set , while in some others it might not be. In contrast,

re-clustering the data, assuming fixed labels obtained by running Baum-Welch on the entire set always results in the MCD decreasing monotonically to a certain point and then increasing as bad data keeps getting added.

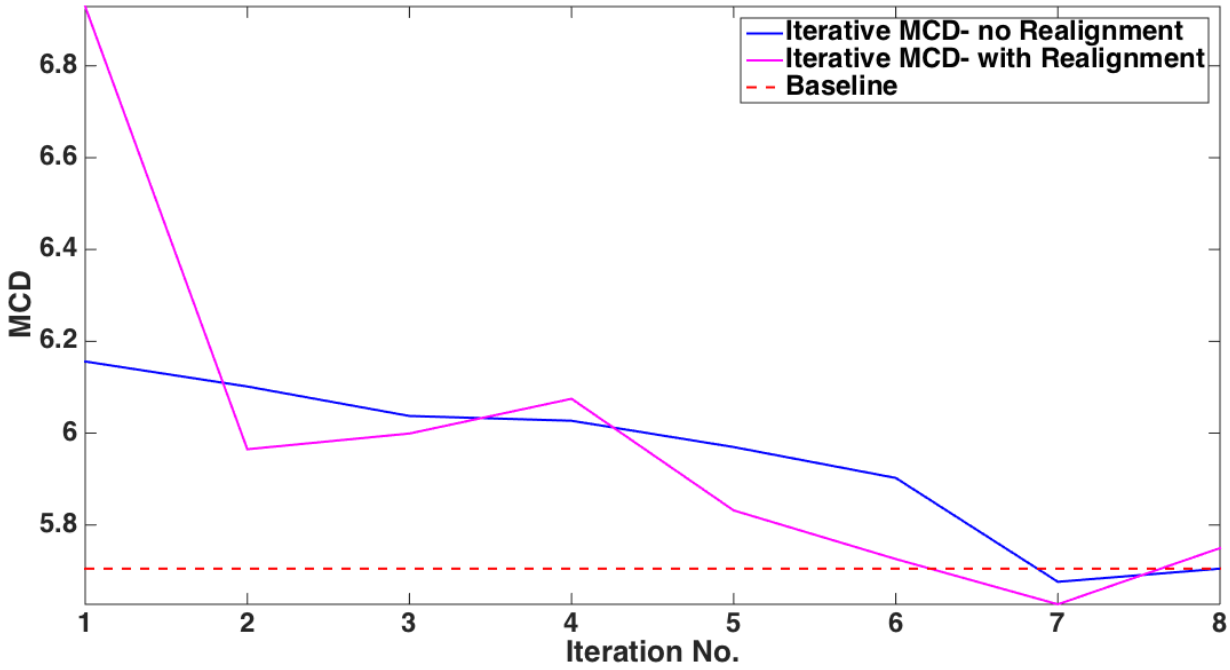


Figure 3.2: *Iterative MCD for Single speaker found data*

Thus, we find that even though realigning data gives a higher increase in MCD, it is also more time consuming. In addition, there is no clear trend as to where we need to stop and ignore the data, unlike the case when only re-clustering on the data, in which case there is a clear point after which performance of the system degrades. However in all of the four plots, Figures 3.1 - 3.4 we see that the MCD metric with re-clustering and no re-alignment, is in fact doing better than the baseline. Moreover, the lowest MCD obtained on the artificially misaligned data is the same as obtained when testing on a model trained only with the clean data. This shows us that this metric is indeed rejecting the noisy data which is not helpful to the system.

3.6.3 How does it scale?

The question then arises can this metric scale when larger amounts of data are misaligned and can this high performance in terms of accuracy hold up even when half the dataset has been artificially misaligned? From the figure, Fig. 3.5, we see that yes indeed, this method based on pruning out misaligned data does work even when more than half the data is misaligned. In fact, the gain obtained over the baseline, a change of 0.3 MCD is significant. A 0.12 MCD change has been shown to be a significant, almost equal to the improvements obtained by doubling the amount of training data [Kominek et al., 2008]. We also find that results with realignment which is almost 0.3 lower in MCD from the

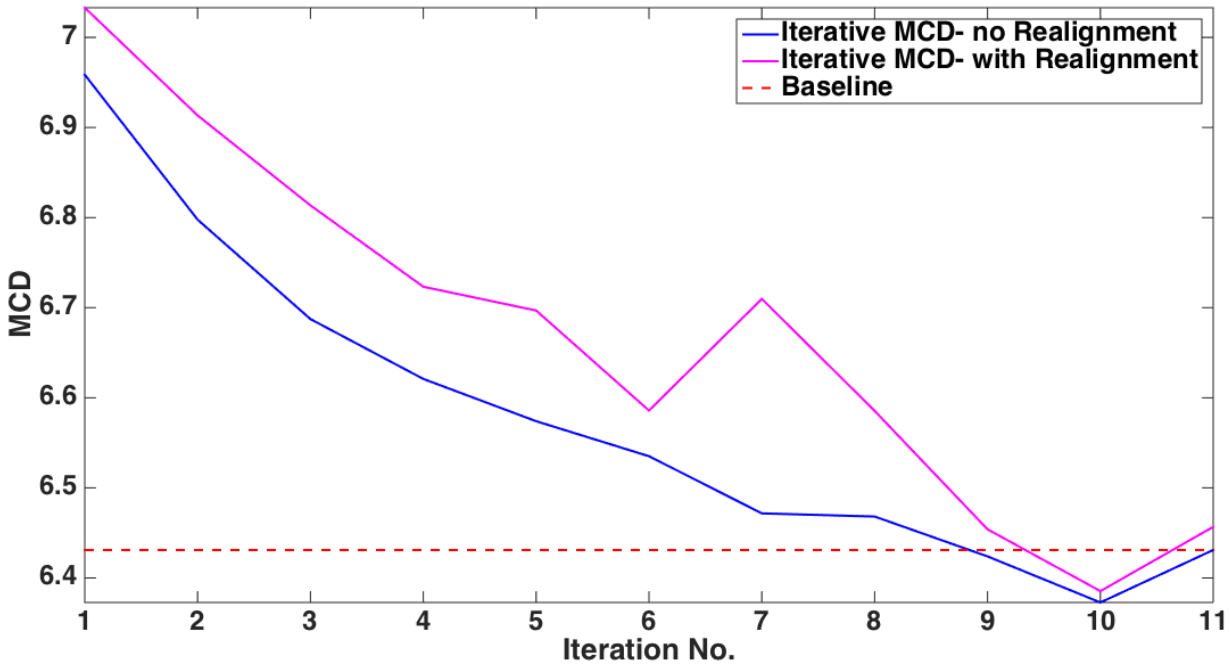


Figure 3.3: *Iterative MCD for multi-speaker found data-Male*

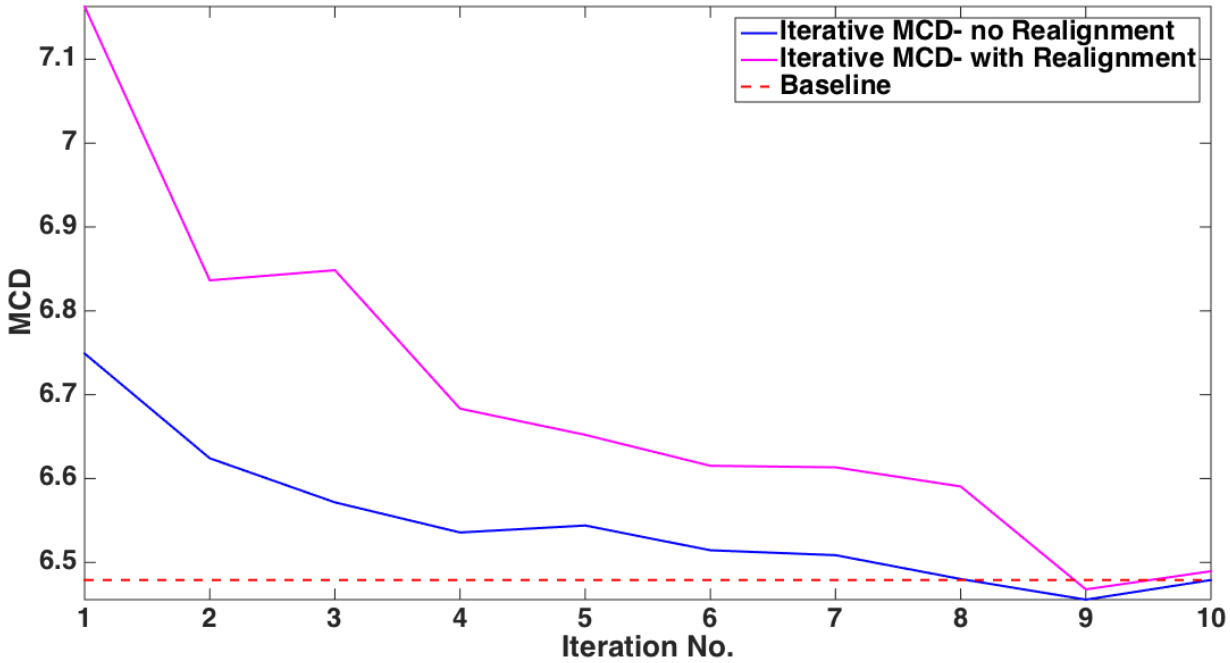


Figure 3.4: *Iterative MCD for multi-speaker found data-Female*

baselines is more beneficial in this case as compared to just re-clustering which is 0.14 MCD lower than the baseline.

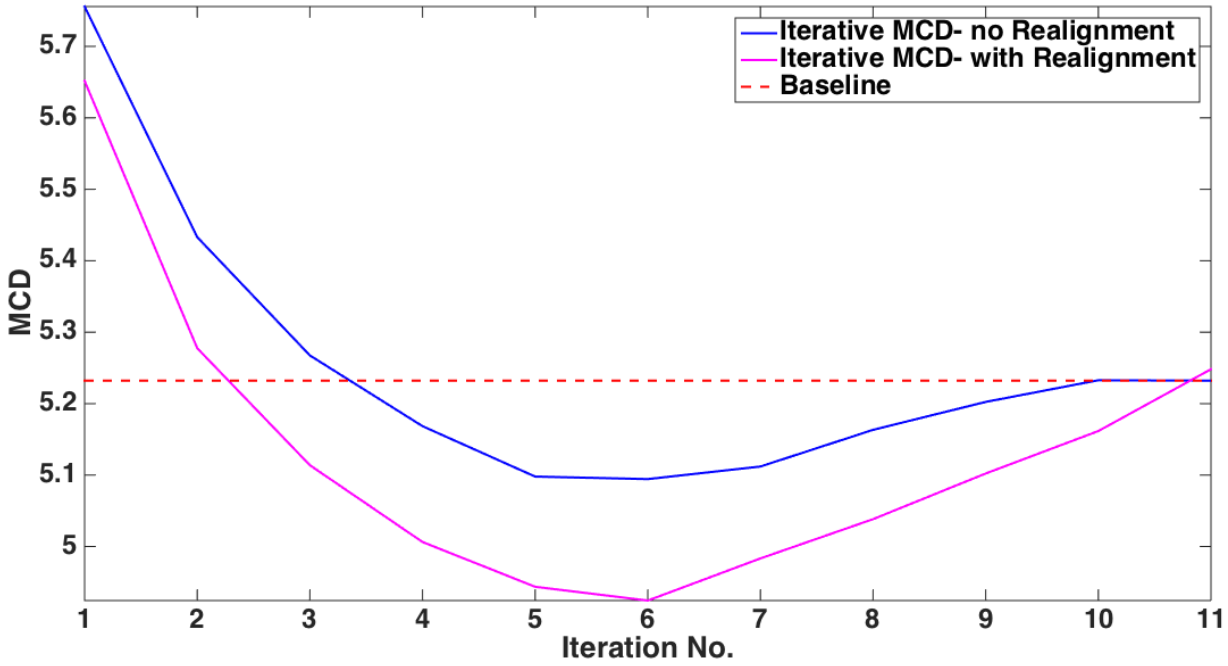


Figure 3.5: *Iterative MCD for ARCTIC RMS containing 50% misaligned data*

3.7 Summary

In this chapter, we show that our claim that not all data is good data holds. We see that selecting a smaller, cleaner subset for voice building is much better and less time consuming than building from the full noisy dataset.

We have explored various utterance level metrics which would be indicators of the *measure of goodness* of an utterance. We showed results considering both the availability of a small seed set of about 100 utterances and building from all of the noisy data without access to such a seed dataset. We find that there is no advantage in having access to a seed dataset and we can instead get similar if not better results by training our initial model on the entire dataset. From all of the measures we explored, we find that the mean cepstral distortion performs the best followed by the error in the duration prediction. We surprisingly find that the various cross-correlation based metrics are not good indicators of the presence of misaligned data. In addition, contrary to our expectations, we find the global spectral measures, *i.e.*, modulation spectrum and global variance perform poorly in detecting changes in channel conditions.

In terms of the errors we find that it is much easier to detect misaligned data than it is to detect noisy channel variations. We also find that the data selection does scale even when 50% of the dataset has been misaligned and yields a MCD which is 0.3 lower than the baseline. In addition, we find that re-clustering works better on the multi-speaker dataset while realigning works better and gives a lower MCD on the two single speaker corpora. However, when re-clustering on these datasets the MCD monotonically

improves upto a certain point, while the behavior of the average MCD with each iteration is erratic when also realigning each time.

We have shown results in this chapter on utterance selection. We show that it scales well on misaligned data and gives us significantly better results than using the entire subset of noisy data. Thus, the main takeaway from this chapter is that for noisy datasets, we can get a gain in MCD by removing at least the worst 10% of the data using MCD or duration as a scoring metric and retraining the clustering model with it. In the next chapter we will see how we can make our noisy data models better by augmenting using other external cleaner data and identify what sources of data are good for augmentation in found and noisy data scenarios.

Chapter 4

Data Augmentation

For synthesis we generally require recordings which have consistent characteristics in terms of speaker and channel. However, since we are dealing with found data, especially for low resource languages, it is difficult to find enough clean recordings from a single speaker, to be able to train an understandable speech synthesis system. One method to use this found data can be to augment it with cleaner data from another speaker of the same language or if one does not have access to a cleaner dataset of recordings in the same language, use some form of transfer learning from a higher resource language to augment the model.

In addition, it is often the case that for many low-resource languages, audio in the form of audio-books and podcasts are available without transcriptions. However, since, we require transcribed audio to train a system for speech synthesis, and there is large amounts of unrelated text data in the same language available in the form of Wikipedia text and text obtained from other regional websites, question is can we use this data to obtain transcriptions suitable for building speech synthesis models. Thus the goals in this chapter are two fold :

- To identify good sources of external data for augmentation to integrate in the TTS pipeline assuming we have transcribed data.
- Given un-transcribed audio data and unrelated text in the same language, how can we use this text data to obtain transcriptions suitable for training TTS systems.

In this chapter, we will first explore cross-lingual data augmentation methods for transcribed data as well as show results on various styles of transcribed English data and then explore methods for un-transcribed speech.

4.1 Cross Lingual Augmentation of Transcribed Text

The goal in this section is to investigate techniques to augment end-to-end neural models with data from a related higher-resource language. Depending on the amount of training data available in the low-resource language and its proximity to the higher resource language, this cross-lingual data could be integrated into the whole system or it could be used to only improve a part of the system. For instance, if we have limited training

data available in the desired low-resource language, we can train an entire system in the higher-resource language and use linear transforms to map the distributions to the lower resource language as was done in the context of cross lingual voice conversion in [Latorre et al., 2006]. On the other hand, in traditional models we could choose to improve a specific part of the TTS pipeline such as the feature extraction, the decoder for segmentation or the context clustering stage. The newer neural style systems [Wang et al., 2017], [Ping et al., 2018] [Li and Zen, 2016], allow us to pool all of the higher and lower resource languages to adapt the entire system. Thus, in this section, the goal is to investigate best methods of identifying good higher resource languages, given a target low-resource language and then integrate it into a neural seq-to-seq TTS pipeline.

4.1.1 Previous Approaches

Previous approaches in using cross-lingual techniques in speech synthesis have been primarily explored in the direction of cross-lingual speech synthesis [Peng et al., 2010] or polyglot speech synthesis, [Chen et al., 2014] and [Latorre et al., 2006], where the goal is to map the speaker's characteristics from one language to another. Most of these techniques assume parallel corpora from the speaker in order to learn a phone or state mapping as in [Wu et al., 2008], [Wu et al., 2009].

However, the goal in this section is to use cross-lingual resources to augment low-resource data in order to give better intelligible TTS. In this regard our goal is similar to the unsupervised cross-lingual techniques as in [Gibson, 2010], and polyglot synthesis techniques mentioned in [Latorre et al., 2006] and [Peng et al., 2010]. In [Gibson, 2010], they assume adaptation data to be coming from the same language as the language used to train acoustic models and so maps the adaptation acoustic data into the phone-set of the source language. It then uses a two-pass decision tree clustering stage to add specific features in the adaptation data in the second stage. While in [Peng et al., 2010] and [Latorre et al., 2006], they build polyglot synthesis models not assuming parallel corpora, using transforms such as MLLR and MAP instead to adapt model on the speaker's data. However, unlike both of these approaches, our goal is not to retain speaker characteristics, but rather to be able to synthesize intelligible speech in our target low-resource language, by augmenting with data from a more easily available higher-resource language. In addition, in this section we explore the newer seq-to-seq neural style models for multi-lingual synthesis, which to our knowledge has not been explored yet. In the next section, we briefly describe our factored embedding model and following that in section, Sec 4.1.3 we describe our experiments and results.

4.1.2 Factored Embedding Model

The model used for this set of experiments is a modified version of the convolutional model, with multi-step attention (dot-product attention at each decoder layer), as described in [Gehring et al., 2017] which was used for machine translation. This model was adapted as the DeepVoice3 model by Ping *et al* in [Ping et al., 2018] for multi-speaker

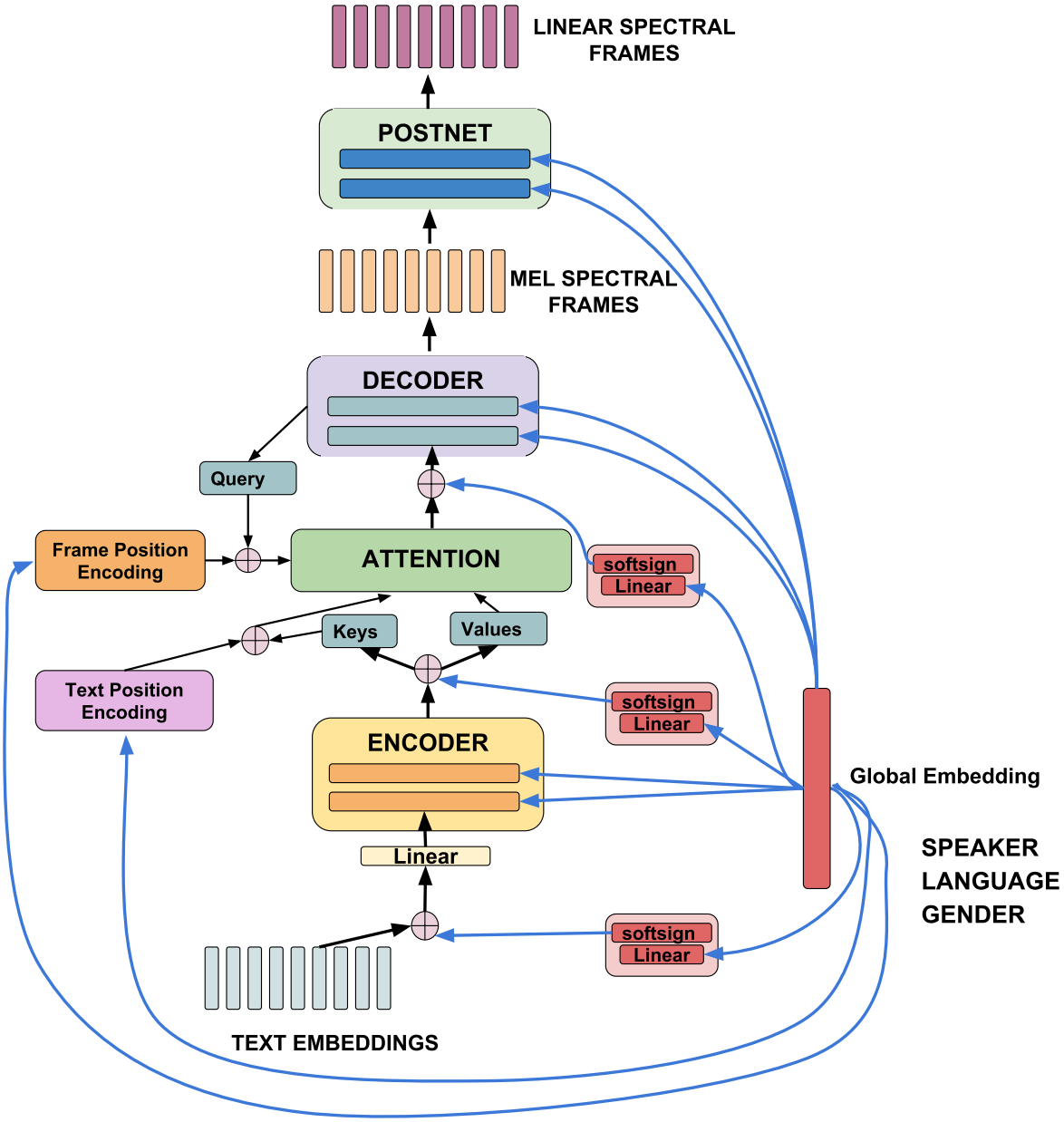


Figure 4.1: *Factored Embedding Model, with global attributes for each speaker (speaker id, gender, language), learned as a separate embedding.*

speech synthesis of English speech. Similar to most sequence-to-sequence models used for speech synthesis, it consists of four main blocks. The encoder block, which converts the input sequence of text units (phones or characters) into a sequence of linguistic embeddings, the attention block which produces a context vector at each time step in the sequence, a compact weighted representation of the linguistic input which is given to the decoder and post-net. The third block is the decoder, which uses the output from the encoder and the context vector of the attention block to predict a sequence of Mel frames, which are then converted to the wavfile using the postnet block, which converts the sequence of Mel frames to linear spectrograms via time-upsampling using multiple convolution layers as was done in [Tachibana et al., 2018] and finally synthesizes them using Griffin-Lim algorithm for phase reconstruction [Griffin and Lim, 1984].

Our model is based on the DeepVoice3 model described in [Ping et al., 2018]. In addition to using position embeddings for each speaker, the factored version of our model also includes global embeddings such as speaker, language, and gender which is given to each encoder, decoder and postnet convolutional layer, as well as added to the position embeddings and text embeddings. In addition, similar to the implementation in [Yamamoto, 2017], we use a guided-attention as well as a divergence multi-task loss as described in [Tachibana et al., 2018], in addition to the L1 Mel and Linear loss. The exact loss function used is described in more detail in 6.2.2 and the different components of the model architecture and its hyper-parameters is described in more detail in Appendix A. Since ours is a grapheme based system, we use the UniTran Sampa Table [Qian et al., 2010], to map Unicode characters to the Sampa symbol-set and we use this set of input symbols as the input to the system.

4.1.3 Experiments

In this set of experiments we seek to answer mainly the following questions:

- **Factored Embeddings:** Is there any advantage to factoring the data across the global attributes and more importantly whether factoring these global attributes will help share data across speakers and languages in data scarce scenarios.
- **Scaling to a new language:** What is the minimum amount of data required in a new language to be able to synthesize from a trained multilingual model and does adding other speakers from the same language improve performance.
- **Transferrability:** Are these global embeddings for gender, language and speakers transferable, *i.e.*, can we make *AXB*, an Indian female speaker speak like a man and can we make *AWB*, a Scottish male speaker talk in a different language such as Hindi.
- **Degradation with Noisy Datasets:**What is the performance degradation when we train on noisy multi-speaker datasets.

Factored Embedding Model *vs.* Un-factored Embedding Model

We first compare a multilingual factored embedding model *vs.* its unfactored counterpart for various languages.

Data: The data used to train the multilingual model consisted of multilingual datasets provided by Hear2Read and available on Festvox [Bazaj, 2017] as well as the Blizzard datasets provided by the Deity project from government of India [Baby et al., 2016]. These datasets range from 30 mins to 9 hours per speaker recorded in relatively clean conditions. We have listed the amount of each speaker's data in the DTWMCD results table, Table 4.1.

Table 4.1: *DTWMCD results on multi-lingual speaker Indic Datasets (Marathi, Hindi, English, Gujarati) example wavs.*

Language Speaker (Gender)	Hours	Unfactored (DTWMCD)	Factored (DTWMCD)
English AXB (F)	00 : 30	8.20 ± 0.44	8.75 ± 0.64
English SLP (F)	00 : 30	9.05 ± 0.58	9.90 ± 0.43
Gujarati IITM2 (M)	09 : 06	6.54 ± 0.60	7.03 ± 0.49
Hindi AXB (F)	01 : 55	7.51 ± 0.40	8.57 ± 0.91
Hindi IITF (F)	04 : 35	7.09 ± 0.40	7.57 ± 0.65
Hindi IITM (M)	04 : 30	6.62 ± 0.20	7.52 ± 0.25
Marathi AUP (M)	00 : 27	6.38 ± 0.28	7.35 ± 0.33
Marathi IITM1 (M)	03 : 03	6.48 ± 0.21	7.09 ± 0.35
Marathi SLP (F)	00 : 31	9.19 ± 0.40	10.23 ± 0.40

Table 4.2: *A/B Listening test results on multi-lingual speaker Indic Datasets (Marathi, Hindi, Gujarati), comparing factored vs. unfactored models example wavs.*

Language Speaker (Gender)	% Who prefer Unfactored	% Who prefer Factored	% Who find both same
Hindi IITF (F)	36.07	**57.14**	6.78
Marathi AUP (M)	47.05	**50.59**	2.94
Gujarati IITM2 (M)	**42.50**	39.17	18.33

Training: First we trained a multi-lingual, multi-speaker model on 7 speakers comprising Hindi, English, Marathi and Gujarati. The model was initialized with weights trained from the VCTK corpus [Veaux et al., 2017], since we found that this makes the model

converge faster as opposed to starting each training cycle from scratch. For the unfactored model, each language-speaker pair was treated as a separate embedding, while, for the factored model, we factored global attributes of speaker identity, gender and language separately as shown in figure, Fig. 4.1, so it could share similar speaker characteristics across languages, as well as use common language features across speakers.

Evaluation: To evaluate the factored embedding model, we use a 10% held-out dataset from the original speaker. Table 4.1, lists the DTWMCD values, comparing the synthesized wavefiles to the ground-truth. For three of these speakers, we also ran listening tests, comparing the factored *vs.* unfactored models. For the listening tests we used 30 random wavefiles, divided into 3 subsets of 10 files each. There were at least 25 listeners per subset. These results are reported in Table 4.2.

Results: From the results in Table 4.1 and 4.2, we see that:

- The results of the unfactored model seem to be better in all cases than the factored model in terms of the DTWMCD values. This is confirmed in most cases with the listening tests, however, we do find that for the Gujarati speaker, listeners prefer the factored model. Our hypothesis as to why the unfactored model performs better than the factored model in most cases is because not factoring the global attributes, helps the model to over-fit to the data better and this might be a good thing when synthesizing speech using end-to-end models. The case for Gujarati speaker is interesting as well, since this is the only speaker in the training data that does not have multiple speakers for the language, and our hypothesis is that the factored and unfactored models are thus very similar since they are sharing the same data per embedding. This can be seen by the marginal difference in DTWMCD, 6.54 *vs.* 7.03 as well as the fact that 18% people find no difference in between the factored and unfactored models and among those listeners, the factored model is preferred more than the unfactored by about 3% only.
- Speaker *SLP*, a female Marathi speaker has the worst performance in the multilingual model. Now it is surprising that male speaker *AUP*, who has the same amount of Marathi data as *SLP* performs much better. However, there is also another male Marathi speaker in the dataset with about 5 hours of data. So the question is, is it something about the speaker's characteristics and speaking style that make it a good voice for the neural network model to learn or can we improve the performance of *SLP*'s Marathi speech by augmenting it with another female Marathi speaker.
- The performance on English speech is very bad for both *AXB* and *SLP*. This might again be because of the paucity of the English data in the dataset. The multilingual dataset has only about 1 hour worth of English data coming from two female speakers. However, since the model was transferred from a model trained on VCTK corpus, a multi-speaker English corpus, it begs the question, whether the neural network works better with similar languages and tends to forget its previous weights, as shown in [Kirkpatrick et al., 2017] or can the problem be solved

by adding more data to it. Thus to this end we look at one speaker *SLP*, and try to improve her performance on English speech with experiments described in Table 4.5

How does it Scale to a New Language and Many New Speakers

Data and Training: To the model mentioned in Section, 4.1.3 which produced results in Table 4.1, we added 7 more speakers and one more language, Bengali. The details of the training data in terms of speaker, gender and hours of data are shown in Table 4.3. We wanted to see if the results degrade when adding more diverse data, showing the case for over-fitting to improve results. The details of the data that have been added are shown in Table 4.3.

Results: We see that the degradation in performance if at all is not much. In fact, some unfactored model results improve marginally for a few speakers. The results for most factored models degrade but only very marginally. From the listening tests comparing the old unfactored model *vs.* the new unfactored model, as shown in Table 4.4 we see that in case of all of the three speakers of Hindi, Gujarati and Marathi, listeners prefer the new model, showing that even with the addition of new languages and external speakers, performance of the model tends to improve.

Case Study:Marathi-English speaker *SLP*

As mentioned in the previous section, we wanted to see if the model improves by adding additional external data from another speaker of the same language, or is it better to fine-tune the multilingual model on a single speaker of the target language. To this end, we present various experiments in Table 4.5 in trying to improve the performance of speaker *SLP*'s English as well as Marathi synthesized speech.

Data and Training: To the model described in Table 4.1, we either add more data from other speakers of the same language and retrain this model, or adapt it on our target data, which in this case is half hour of speaker *SLP*'s Marathi speech, and half hour of the same speaker's (*SLP*) English speech. We also further fine-tune the data augmented model with *SLP*'s English and/or Marathi speech. To augment the models we use the 4 hours and 20 mins of female Marathi speaker (*IITF4*) and 6 hours of Indian English female speaker (*IITF*).

Results: Table 4.5 agrees with Table 4.3 in that the unfactored model's performance is much better than the factored model. Some observations from the results in Table 4.5 are as follows: Adaptation or fine-tuning the model on our target data seems to produce some improvements in quality of speech as measured by the DTWMCD metric calculated on a 10% held out test set from speaker *SLP*. Surprisingly, we did not see as much gain in adding more data, and in fact find that the DTWMCD metric shows a slight degradation in quality, when augmenting with additional data from external speakers. In addition,

Table 4.3: *DTWMCD results on multi-lingual speaker Indic Datasets (Marathi, Hindi, English, Gujarati and Bengali) example wavs.*

Language Speaker (Gender)	Hours	Unfactored (DTWMCD)	Factored (DTWMCD)
Bengali IITF3 (F)	01 : 34	7.30 ± 0.39	7.64 ± 0.39
Bengali IITM3 (M)	08 : 01	6.79 ± 0.56	7.28 ± 0.64
English AXB (F)	00 : 30	8.08 ± 0.47	8.70 ± 0.72
English SLP (F)	00 : 30	9.46 ± 0.65	9.95 ± 0.54
English IITF (F)	06 : 14	7.78 ± 0.43	9.17 ± 0.42
English IITM (M)	06 : 00	7.93 ± 0.47	9.00 ± 0.492
Gujarati AD (M)	00 : 54	6.74 ± 0.37	7.71 ± 0.77
Gujarati DP (F)	00 : 48	7.45 ± 0.58	9.21 ± 0.87
Gujarati IITM2 (M)	09 : 06	6.85 ± 1.02	8.27 ± 0.66
Gujarati IITF2 (F)	05 : 07	6.79 ± 0.48	7.52 ± 0.63
Gujarati KT (F)	00 : 26	6.44 ± 0.39	9.36 ± 1.44
Hindi AXB (F)	01 : 55	7.47 ± 0.394	8.65 ± 0.49
Hindi IITF (F)	04 : 35	6.96 ± 0.380	7.72 ± 0.58
Hindi IITM (M)	04 : 30	6.48 ± 0.22	8.32 ± 0.32
Marathi AUP (M)	00 : 27	6.34 ± 0.30	7.68 ± 0.49
Marathi IITF4 (F)	04 : 19	7.32 ± 0.44	8.41 ± 0.40
Marathi IITM1 (M)	03 : 03	6.38 ± 0.20	7.35 ± 0.54
Marathi SLP (F)	00 : 31	9.62 ± 0.55	10.37 ± 0.48

Table 4.4: *A/B Listening test results on multi-lingual speaker Indic Datasets (Marathi, Hindi, Gujarati) comparing the big unfactored model vs. smaller model example wavs.*

Language Speaker (Gender)	% Who prefer Old Model	% Who prefer New Model	% Who find both same
Hindi IITF (F)	37.08	**52.50**	10.40
Marathi AUP (M)	36.40	**43.20**	20.40
Gujarati IITM2 (M)	31.42	**57.14**	11.42

Table 4.5: *DTWMCD results on Marathi-English speaker SLP example wavs.*

Adaptation / Data Added	Unfactored (DTWMCD)		Factored (DTWMCD)	
	SLP Marathi	SLP English	SLP Marathi	SLP English
Baseline Multi-lingual Indic Model	9.19 ± 0.40	9.05 ± 0.58	10.23 ± 0.40	9.90 ± 0.43
Adapting base model on SLP	8.81 ± 0.43	8.46 ± 0.42	9.15 ± 0.35	8.77 ± 0.37
Adapting base model on SLP Marathi	8.80 ± 0.36	8.70 ± 0.38	9.28 ± 0.40	9.31 ± 0.42
Adapting base model on SLP English	8.90 ± 0.37	8.42 ± 0.39	9.16 ± 0.31	8.74 ± 0.34
External Marathi Female	9.36 ± 0.41	9.09 ± 0.49	10.49 ± 0.44	10.02 ± 0.53
External English Female	9.30 ± 0.43	9.13 ± 0.53	10.26 ± 0.36	10.08 ± 0.45
External Marathi + English Female	9.42 ± 0.40	9.29 ± 0.54	10.13 ± 0.44	9.88 ± 0.41
Adapting external Marathi female speech model on SLP Marathi	8.85 ± 0.42	8.74 ± 0.33	9.17 ± 0.41	9.07 ± 0.37
Adapting external English and Marathi female speech model on SLP	9.14 ± 0.42	8.70 ± 0.37	9.35 ± 0.40	8.84 ± 0.37

we see that fine-tuning the model after augmenting with more data is not as good as just fine-tuning alone. This again seems to show that over-fitting on our target dataset seems to be desirable for synthesis. If you consider the model adapted on only English data, *vs.* the model adapted on Marathi only data, we see that it is getting confused between the acoustics of English and Marathi speech. One hypothesis for the failure of the model to converge is that the model might be getting confused with English and Marathi from the same speaker. This can be seen from the attention plots in Figure, 4.2 which fail to produce good alignments, even though the model converges very fast. As such if we compare results for English speech across all speakers, we see that it finds it difficult to synthesize English well across all four speakers.

Case Study: *IITF2*-Gujarati - How does the Model Perform on a New Language?

The next question we wanted to answer was how much data in a new language is good enough to adapt the multilingual model to the new target language and speaker. We also wanted to compare the performance improvements (if any) when we add other external speakers in the same language and whether similarity of the speakers in terms of gender matters, if at all. First we looked at how augmenting with data in the same language but from other speakers effects performance and how much of a role gender of the external speaker plays in improving results. These are described in Table 4.6.

Effect of External Speakers in the Same Language:
Data: The baseline single speaker model was trained on English audiobook data of

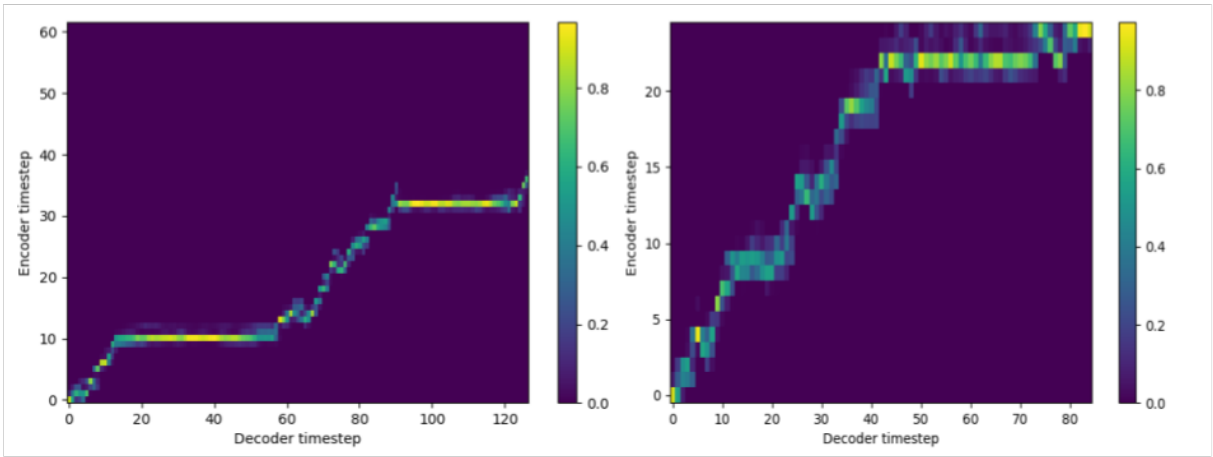


Figure 4.2: *Failed Attention Plots for Model Adapted on SLP's Marathi Data*

23 hours (*LJSpeech* dataset) [Ito, 2017] and then fine-tuned on our target Gujarati female speaker *IITF2*. Our baseline multilingual model which we abbreviate MLMSI-NG or Multi-lingual, Multi-speaker, Indic model with non Gujarati data includes all of the subsets from Table 4.3, without the Gujarati speakers. To this model we add an external Gujarati male (*IITM2*) dataset of 9 hours which we call (Big) and a smaller male dataset is that of speaker *AD* totalling just under 1 hour of data. Similarly the external female data which is not the target speaker is Gujarati speaker *DP*, including just less than an hour (50 mins) of data. Finally, the multi-speaker Gujarati corpus includes 4 other Gujarati speakers, *AD*, *DP*, *IITM2*, and *KT* apart from the target speaker, totalling about 16 and half hours of total Gujarati data. Table 4.6 presents these results.

Results: We see that augmenting the model with external data at least for the case of Gujarati, does not give expected gains in synthesis quality, when comparing results augmenting with a multi-lingual model. We do see gains when going from a single speaker English model to a multi-speaker, multi-lingual model, however, the presence of the target language in the training data does not make significant differences in synthesis quality. As expected, there are some gains when augmenting with a speaker of the same gender, but if we compare the quality of synthesis obtained from adding 9 hours of external male Gujarati data *vs.* adding only 1 hour of external male Gujarati data, we do not see much difference in improvement. One reason for this might be that 5 hours of target speaker data is enough to train the model and it is not getting any gains from the other speakers of the same language. Thus, in the next experiment we wanted to see whether this had any effect if we reduced the size of the training data available for the target speaker.

Effect of Size of Training data of Target Speaker, and External Data for Augmentation:

Data and Training: To understand the effect of the amount of data needed to make a

Table 4.6: *DTWMCD results introducing a new speaker with different types of external data in another language or the same language but from an external speaker example wavs.*

External Data Added	Hours-Guj. Data (hh:mm)	DTWMCD
Adapt on English model (Baseline)	$05:07$	7.90 ± 0.38
MLMSI-NG	$05:07$	7.96 ± 0.50
MLMSI-NG + Guj. male (Big)	$14:13$	7.96 ± 0.53
MLMSI-NG + Guj. male (Small)	$06:01$	7.90 ± 0.55
MLMSI-NG + other Guj. female	$05:55$	6.96 ± 0.69
MLMSI-NG + Multispeaker Guj.	$16:34$	7.52 ± 0.63

model synthesize a new speaker's voice in a new language, we made 4 subsets of the female Gujarati (*IITF2*) data. These included subsets of 100, 250, 500 and 1104 utterances each with durations as mentioned in Table 4.7, going from about half hour of data to almost five hours, and doubling in size with each subset. We assumed three different adaptation/initialization conditions in training these models. In the first case, we directly adapted the model on the target speaker, by transferring weights from English audiobook data trained model on *LJSpeech*. In the second case, we trained a multilingual model with all of the speakers mentioned in Table 4.3 without the other Gujarati speakers (*AD*, *DP*, *KT* and *IITM2*). For the last model, we trained with all of the speakers from 4.3. Table, 4.7 lists these results.

Evaluation: For calculating the DTWMCD, we use a 10% held out test-set from the original speaker. For the Mean Opinion Score (MOS) listening tests reported in Table 4.8, we asked listeners to rate 30 examples of each subset of 100 and 1000 utterances on a scale of 1 to 5. The listeners where not asked to directly rate the utterances on a scale, instead the prompts they were asked were:

- *It is natural* (5)
- *It is somewhat natural* (4)
- *It is neither natural nor robotic* (3)
- *It is somewhat robotic* (2)
- *It is very robotic* (1)

Because of these prompts, we find that the listeners had a propensity to rate it generally higher than if they would be asked to rate it on a a numerical scale. However, these numbers still give us an idea of which systems are preferred over other systems, and are a good complement to the objective results reported in Table 4.7.

Results: As you can see the DTWMCD numbers don't paint a very good pattern of the data being synthesized as compared to the listening test results reported in Table 4.8. From the listening test results, one can clearly see that starting from a multi-lingual model irrespective of whether it contains Gujarati speech helps to synthesize a better Gujarati model on target speaker when the subset size is small. In addition, we again see that we get significant gains by fine-tuning to the training data, and show that we do not need a lot of data to train an already trained multi-lingual model, irrespective of whether it contains external Gujarati data or not. Thus, these experiments clearly show that initialization matters, when it is being initialized from totally different speech data, and this is especially true when the size of the data is small.

Table 4.7: *Performance of the model as calculated by DTWMCD on subsets of a new target speaker of a new language, when transferring either from a multilingual model or a monolingual model example wavs.*

Data	100 (00:37:21)	250 (01:26:40)	500 (02:52:42)	1104 (05:07:29)
Model adapted from single speaker English model	8.10 ± 0.42	8.39 ± 0.43	8.13 ± 0.43	7.90 ± 0.38
Model Adapted from non Gujarati Multilingual Indic Model	7.20 ± 0.76	6.97 ± 0.57	6.90 ± 0.54	7.96 ± 0.50
Model Adapted from Multilingual Indic Model including multiple Gujarati speakers	8.50 ± 0.74	8.17 ± 1.26	8.14 ± 0.84	7.52 ± 0.63

Table 4.8: *MOS Listening test results on different subsets of Gujarati female data adapted on different models example wavs.*

Model used for adaptation	100 utterances	1000 utterances
Single speaker English Model	3.46	3.92
Multilingual model with no Gujarati Data	4.07	4.13
Multilingual model with external Gujarati speakers	4.03	4.20

Transferrability

One advantage that we get with using factored embeddings is that it allows more control on the gender, speaker and language. Thus, with this model it is possible to get a speaker to speak a language for which we do not have any training data. We did some cross-lingual and cross-gender synthesis. The results can be found at this link. We wanted to

check what information the different global embeddings capture, and how transferable they are. Thus we did three sets of experiments for both male (*AUP*) and female (*IITF*) speaker.

- **Change of Speaker:** In the first experiment we changed only the speaker ID for scripts of language where there was no training data from the speaker in that language. Thus for Gujarati speaker *IITM2*, Hindi speaker *IITM* and Bengali speaker *IITM3*, we changed the speaker id to that of the Marathi speaker *AUP*, to see if the speaker characteristics transfer. Similarly we did the same with female Gujarati and Marathi speakers, replacing the speaker id with female Hindi speaker *IITF*. Indeed, we see that the speaker characteristics transfer well to other languages.
- **Change of Gender:** We changed the gender for both males to females and vice-versa and synthesized for all four languages. We see that apart from the Marathi speaker *AUP*, the synthesized file still carries the original gender and is not transformed much.
- **Change of Language:** To see how much information the language embedding carries, we changed the language ID for speaker *AUP* from Marathi to Bengali, Gujarati and Hindi and find that again this does not change the pronunciation very much.

Thus, we see that most of the information is carried in the speaker ID and it is hard to factorize out the gender from it. The language embedding by themselves also have no effect. They need the right sequence of characters to synthesize the target language and thus do not transfer well.

Performance on Noisy Datasets

We also wanted to see how robust the model is to noisy conditions, especially given the fact that the attention mechanism of this end-to-end model, is notorious for being brittle in noisy conditions or even failing on utterances that have excessive silences [Taigman et al., 2018]. Thus, we tested our multilingual model on two sets of noisy datasets as described below.

Data: We used two sets of noisy datasets in two languages. The first was a multi-speaker Bengali dataset provided by DARPA as part of the Babel IARPA program [Harper, 2011]. It was collected via the telephone, and included 308 conversations in very noisy conditions totalling about 11.30 hours of data from which we had about 7 minutes of maximum data for the majority speaker. The second set that we used was a Gujarati ASR corpus collected in clean conditions by SpeechOcean.com and distributed by Microsoft as part of the Low Resource ASR Challenge for Indian Languages at Interspeech 2018. However, it did not include any speaker information, so the number of speakers in unknown. We used a 5000 utterance subset of this data which was approximately about 8 hours of speech totally.

Training: For the Bengali data since we had speaker ID information, we trained a fac-

tored as well as an unfactored model. In case of the Gujarati corpus, since we did not have any speaker-ID information, we trained a single speaker model with the multi-speaker dataset. Both models were either initialized with wieghts from an English multi-speaker model trained on VCTK or a multi-lingual model trained on cleaner data including Bengali and Gujarati.

Evaluation: We used 10% of the utterances as held-out test set to calculate DTWMCD values as reported in Table 4.9. For the listening test results, we used 30 utterances divided over 3 subsets and required at least 15 listeners per subset. For each language we ran two A/B tests. The first compared a multilingual model to the CLUSTERGEN model and the second compared the multilingual model to a model initialized with a trained multi-speaker English dataset (VCTK).

Results: For the Bengali data we see that we get significant performance gains over our baseline CLUSTERGEN model. In addition, we again see that the unfactored model produces much better results than the factored model and it produces almost intelligible speech, a huge improvement over the traditional HMM based system. For the Gujarati speech we find that even though we treat the data as a single speaker, it ends up learning a multi-speaker model. In addition, we find that when synthesizing from this model, it randomly samples a speaker per utterance and the speaker remains consistent over the entire utterance, when synthesizing on random text utterances. On the other hand, if we synthesize from a held-out test set with similar context as what the speaker was talking, we see that it synthesizes an utterance with a speaker ID very similar to the original speaker. This results again shows how powerful these models are at learning super-sentential context. Thus, we see that even in the absence of speaker ID information, these models are good at picking up and modelling speaker information as a global attribute over multiple sentences. The listening test results reported in Table 4.10 also corroborate our findings that the Neural model is significantly better than the multi-lingual model and it does help to initialize with a multi-lingual model as compared to a multi-speaker English model in terms of perceptual quality.

4.1.4 Summary

In this section, we explore convolution attention based models for multi-speaker, multi-lingual speech synthesis of Indian languages. We explored these models withe respect to mainly four criteria:

- **Factored Embeddings:** With respect to factoring global attributes across speaker, language and gender, we show that the factoring does not improve performance over the unfactored model, where each language-speaker pair is represented by a separate embedding. We see that this degradation in performance is greater if there are more speakers of the same language, showing us that the unfactored model allows it to over-fit to speaker characteristics and thus is better for our case of speech synthesis.

Table 4.9: *DTWMCD results on noisy datasets for Bengali and Gujarati (with missing labels) example wavs.*

Dataset	Speaker	Model	Unfactored	Factored
Babel Bengali	Average (All)	Baseline CLUSTERGEN	9.56 ± 3.98	-
		Adapted from VCTK	6.51 ± 2.13	7.33 ± 2.64
		Adapted from Multilingual Model	6.52 ± 2.15	7.35 ± 2.64
MSR Gujarati	Avg (All)	Baseline CLUSTERGEN	7.63 ± 1.22	-
		Adapted from VCTK	6.65 ± 1.67	-
		Adapted from Multilingual Model	6.60 ± 1.65	-

Table 4.10: *A/B Listening test results on noisy multi-lingual datasets comparing CLUSTERGEN, English and multi-lingual models example wavs.*

Language	Model	% Who prefer CLUSTERGEN/VCTK	% Who prefer Multi-lingual	% Who find both same
Bengali	CLUSTERGEN	11.50	**73.00**	15.50
	VCTK	23.71	**51.14**	25.14
Gujarati	CLUSTERGEN	13.68	**81.58**	4.73
	VCTK	40.00	**52.86**	7.14

- **Scalability and Effect of External Speakers:** We also explore how well these models scale when we add many more new speakers and languages. We see that the performance generally improves in terms of both subjective and objective metrics. We also show that initialization of the model makes a difference when we compare initialization with an English model *vs.* a multi-lingual model. However, we find that it does not make much of a difference whether this multi-lingual model contains other external speakers of the target language. We find that as long as we initialize with similar multi-lingual languages and fine-tune on our target data we get good performance as shown both through objective as well as subjective scores.
- **Transferrability:** We also show an analysis of what these embeddings learn in terms of speaker, gender and language. We see that when we change speaker ID for data that the model has never seen a speaker speaking that transferred language, it generally transfers well, and we can make a speaker speak an unknown language. However, when we try to transfer gender or language, we see that these two attributes do not transfer well, and generally the source gender/language remains unchanged. Thus, we see that it is hard to factorize out gender and language from the speaker embedding, and the speaker embedding carries most of the information about the speaker.
- **Robustness to Noise:** We see that the neural models are very robust to noise and perform significantly better than the CLUSTERGEN models. We even show that with as little as 7 minutes of noisy speech, they can synthesize understandable speech for most utterances. In addition, we show that even with missing speaker labels, these models intrinsically learn a sense of global attributes such as speaker ID and end up learning super-sentential context on their own.

Thus, in this section we saw what these convolutional models are good at and cases where they fail for multi-lingual data. In the next section we will explore these models on a single language (English), but with a variety of found data spanning multiple speaking styles to see how robust they are to style transfer and domain adaptation.

4.2 Multi-speaker data augmentation for Found data in English speech

So far we have looked at relatively clean multilingual data. However, as mentioned in Chapter 2, found data is not always clean and can contain many styles, such as audiobooks (read, acted), public speeches (rehearsed, conversational), telephone speech (conversational), *etc.* Thus, the goal in this section is to explore if we can adapt the multi-lingual model from the previous section into a multi-style model, so that we can build better end-to-end neural multi-speaker models for found data. The goal is to be able to synthesize a target speaker's voice given only 20 minutes of speech. We will thus explore various types of data with different styles and noise levels including Telephone speech (CallHome), Public Speeches (TED), Audiobooks (Librivox), and Clean Speech (ARCTIC) and explore if it is possible to take really degraded data such as either public

speeches or telephone conversations and be able to synthesize them as relatively cleaner speech, while still retaining the speaker characteristics. In the the next section we briefly describe previous work that has been done in the domain of style transfer in speech synthesis and in the subsequent section, Section 4.2.2, we will briefly describe the experiments and results obtained on clean, phonetically balanced datasets and various noisy datasets spanning various speaking styles and experiments with style transfer.

4.2.1 Related Work

Style transfer work in speech synthesis has mainly involved transferring speaker characteristics either using parallel training data from two speakers, where the model learns to map explicitly from speaker A to speaker B [Toda et al., 2007], [Desai et al., 2010],[Mohammadi and Kain, 2016], or as multi-speaker voice conversion, with non-parallel data [He et al., 2012], [Song et al., 2013], [Hsu et al., 2016],[Nakashika et al., 2016]. There has also been some work from Microsoft Research [Qian et al., 2011], [He et al., 2012] in using frame replacement techniques to transfer accent characteristics. However, we are not aware of work that has explicitly looked at transferring style across datasets, this is more so since most of these voice conversion datasets have been explicitly recorded for the purposes of speech synthesis. Only in recent times has there been some effort in being able to synthesize from less than 5 minutes of data and using found data for speech synthesis [Mehri et al., 2017], [Taigman et al., 2018]. However, we are not aware of any work that explicitly tries to transfer across various speaking styles. In the next section we explore sequence-to-sequence based models for speech synthesis with noisy datasets and explore style transfer across various speaking styles.

4.2.2 Experiments

We first look at multi-speaker models of clean speech and then show results on noisy datasets.

Character *vs.* **Phone based multi-speaker model:**

First we looked at multi-speaker experiments with 11 ARCTIC voices [Kominek and Black, 2004], 7 Female, and 4 Male. Each speaker was identified by a unique speaker embedding. First we wanted to understand what would be a good input representation of text for multi-speaker models on clean data. Thus, we compared phone *vs.* character based models.

Results: We see that in all cases, the phone based models do better than the character based models.

Table 4.11: *DTWMCD results on multi-speaker clean ARCTIC Datasets using speaker dependent character and phone based models example wavs.*

Speaker	Gender (Nationality)	Char-Based (DTWMCD)	Phone-Based (DTWMCD)
AEW	Male (American)	6.686 ± 0.327	6.569 ± 0.349
AXB	Female (Indian)	8.622 ± 0.737	8.486 ± 0.742
AWB	Male (Scottish)	6.323 ± 0.265	6.239 ± 0.296
BDL	Male (American)	7.301 ± 0.376	7.268 ± 0.421
CLB	Female (American)	7.817 ± 0.402	7.635 ± 0.359
EEY	Female (American)	7.343 ± 0.662	7.008 ± 0.588
LJM	Female (American)	7.355 ± 0.574	7.100 ± 0.628
LNH	Female (American)	8.651 ± 1.690	8.513 ± 1.298
RMS	Male (American)	7.033 ± 0.344	7.041 ± 0.324
SLP	Female (Indian)	8.699 ± 0.729	8.380 ± 0.717
SLT	Female (American)	7.002 ± 0.368	6.897 ± 0.410

Case Study: American English Speaker *AEW*

From the multi-speaker results on the clean speech, we found that the quality of the synthesized speech for speakers particularly with low F0 was not of the best quality. So, we tried various adaptations, by switching the higher F0 speakers with lower F0 speakers as well as training an all male model. Results of these various adaptations with different subsets of training data with an overall F0 made to get closer to that of the target speaker in order to improve the speech synthesized for speaker *AEW* are described in Table 4.12.

Data and Training: We use a relatively cleanly recorded, phonetically balanced dataset. We start with an initial model trained with 23 hours of female audiobook data from speaker *LJSpeech* [Ito, 2017]. This data has a relatively high F0. Thus, we then decrease the overall average F0 of the training data, by introducing more male ARCTIC speakers [Kominek and Black, 2004]. Each arctic dataset has been collected to be phonetically balanced and contains ARCTIC set A and B for some speakers, or only ARCTIC set A for a few speakers. Thus, overall, there are about 1000 or 500 utterances per speaker. The speakers have varying accents. The goal here is to see how the average F0 of the training data affects the quality of synthesized speech and if we can manipulate the training data in order to improve synthesis on data that is not well represented in the dataset. We report results with adaptation, *i.e.*, after training the entire model, we do an additional fine-tuning of the model only on the target speaker's voice as well as without the adaptation.

Results: We see that changing the overall F0 of the training data has little effect on

Table 4.12: *DTWMCD results on AEW (F0:* 117.0 ± 31*) with various adaptations example wavs.*

Model	Num. of Males-Females	Mean & std F0 (Hz)	No Adaptation	With Adaptation
Model adapted from LJSpeech Baseline	0M 1F	200.0 ± 56.0	-	5.05 ± 1.83
Baseline Multispeaker Model	4M 7F	160.0 ± 52.0	6.57 ± 0.35	4.57 ± 1.39
Multispeaker Model with majority males	9M 2F	134.0 ± 37.0	6.52 ± 0.36	4.54 ± 1.36
Multispeaker Model with all males	11M 0F	129.0 ± 29.0	6.53 ± 0.34	4.58 ± 1.40

the quality of synthesis when we have multiple speakers. Adapting on the *LJSpeech* model does produce lower quality synthesis. We find that starting from a different style (audiobook) and gender has some effect on the final results, however, as long as we have a multi-speaker model, it does not matter what the average F0 of the training set is as long as we have the target speaker in the training data, we seem to get similar results. However, we see that fine-tuning on our target data gives us huge gains in quality for all models. Thus, given that adding similar training data to that of our target speaker does not seem to improve quality, this then begs the question of whether there are certain speaker characteristics that are better to model by these networks. This would be an interesting future direction of research to pursue.

Multi-Speaker Experiments with Noisy Found Data

We also wanted to see how robust these models are to noisy multi-speaker datasets. In this regard, we present results on various speaking styles of noisy English multi-speaker speech datasets.

Data: We used various subsets of 4 noisy English speech datasets collected in varying conditions and with different speaking styles. To get a wide coverage of speaking styles and accents, we looked at mainly 4 subsets of data. For each subset we have indicated the style, the number of speakers and the total time of the dataset. To evaluate, we used the maximum speaker per dataset one male and one female. The maximum data per speaker varied from a maximum of 20 mins to as little as 4 minutes for telephone speech. We also ran listening tests for each speaker, as reported in Table 4.14. Here we compare the neural model with the CLUSTERGEN model per subset. We have combined the rating over the majority male and female speaker per subset to report the A/B test numbers in Table 4.14.

Training: We trained a factored as well as an unfactored multi-style, multi-speaker model. For each target speaker, we ran a fine-tuning run to fine-tune to the target speaker. We also trained single speaker models for each target speaker, which was not augmented with any other data. However, we did transfer the weights from a previously trained ARCTIC multi-speaker model for faster convergence.

Results: Table 4.13, lists the DTWMCD results on various noisy datasets spanning multiple speaking styles. We have reported results for the majority male and female speaker for each dataset. We see that the the neural model performs significantly better in almost all cases as compared to the CLUSTERGEN models. The listening test results in Table 4.14, further corroborate the fact that the neural model significantly outperforms the CLUSTERGEN model across all speaking styles. However, in comparing the single model with the factored and unfactored multi-speaker models, we see no advantage in augmenting with multiple speaking styles and in case of the noisier datasets, with cleaner data. Neither do we see any advantage in factoring the embeddings. It seems like these end-to-end neural models are pretty good at over-fitting to training data, however noisy very fast. If one listens to the sample wavfiles, it is clear that these models also end up picking up the channel noise present in these datasets. Thus, in the next experiment we explore if we can remove this noise, by retaining speaker characteristics, but doing a style transfer to a more clean speaking style like either read news or clean speech such as ARCTIC.

Style Embeddings and Experiments with Style Transfer

Our goals for experimenting with style transfer were two-fold. First we wanted to see if we can synthesize really noisy data such as telephone speech into a much cleaner format such as clean speech, while maintaining speaker characteristics. Second, we wanted to see whether we can induce a more conversational tone into a clean speaker's speaking style by changing the style from say a monotonic single utterance read style to one that is more conversational.

Data and Training: Thus, we trained a single factored multi-speaker model with clean data comprising 6 ARCTIC speakers, (3 Males and 3 Females), along with all of the speakers and styles mentioned in Table 4.13. There were a total of 308 speakers across 5 speaking styles, Clean (ARCTIC), Audiobook (LibriVox), Public Speeches (TED Talks) and Telephone Conversations (CallHome). Three global attributes were used, style, speaker-ID and gender.

Results: The sample waves can be found at this link.

- **Transfer from Clean to other Styles:** We find that transferring from the clean speaking style to other speaking styles does not have an effect on the speaking style. Instead, what we see is that the style embedding is instead learning the channel characteristics. This is especially apparent with the telephone speech style

Table 4.13: *MCD results on different Styles of Found English Speech. Times for each subset are indicated as [hh:mm:ss] example wavs.*

Dataset (Style)	Speaker	Model	Single Model	Unfactored	Factored
WSJ (Read News) [08:45:34] (20 speakers)	S1 (Male) [00:21:56]	Baseline CLUSTERGEN	8.32 ± 0.48	7.71 ± 0.44	-
		Neural End-to-End TTS	5.45 ± 1.31	5.41 ± 1.31	6.32 ± 1.70
	S2 (Female) [00:24:07]	Baseline CLUSTERGEN	9.03 ± 0.53	8.60 ± 0.48	-
		Neural End-to-End TTS	5.63 ± 1.10	5.63 ± 1.10	6.44 ± 1.44
LibriVox (Audiobook) [06:21:10] (17 speakers)	S1 (Male) [00:22:58]	Baseline CLUSTERGEN	6.95 ± 0.67	7.80 ± 0.22	-
		Neural End-to-End TTS	9.03 ± 3.65	8.29 ± 4.144	9.03 ± 3.66
	S2 (Female) [00:22:14]	Baseline CLUSTERGEN	8.24 ± 0.36	9.10 ± 0.64	-
		Neural End-to-End TTS	9.67 ± 3.765	9.722 ± 3.73	9.83 ± 3.64
TED Talks (Public Speeches) [04:59:31] (32 speakers)	S1 (Male) [00:27:04]	Baseline CLUSTERGEN	8.04 ± 0.56	7.95 ± 0.44	-
		Neural End-to-End TTS	7.25 ± 3.32	7.14 ± 3.31	8.12 ± 3.35
	S2 (Female) [00:18:05]	Baseline CLUSTERGEN	7.47 ± 0.75	8.07 ± 0.86	-
		Neural End-to-End TTS	5.14 ± 2.45	5.13 ± 2.44	5.61 ± 2.70
CallHome (Telephone) [04:40:08] (80 speakers)	S1 (Male) [00:04:27]	Baseline CLUSTERGEN	8.71 ± 0.34	9.24 ± 0.50	-
		Neural End-to-End TTS	4.60 ± 2.15	4.72 ± 2.26	4.62 ± 2.18
	S2 (Female) [00:06:41]	Baseline CLUSTERGEN	9.06 ± 0.70	9.04 ± 0.55	-
		Neural End-to-End TTS	5.83 ± 2.07	5.74 ± 2.05	5.71 ± 2.06

where one can hear the channel noise as well as the reverberation in the public speeches style.

Table 4.14: *A/B Listening test results on English multi-speaker datasets with different speaking styles, comparing CLUSTERGEN and neural models example wavs.*

Style (Dataset)	% Who prefer CLUSTERGEN	% Who prefer Neural	% Who find both same
Read News (WSJ)	13.07	**83.84**	3.07
Audiobook (Librivox)	26.31	**66.31**	7.37
Public Speeches (TED)	23.12	**66.56**	10.31
Telephone Conversations (CallHome)	17.14	**65.35**	17.50

- **Transfer from Noisy to Clean style:** First we change the style embedding to be clean across all speaking styles. This is compared to the baseline speech synthesized with original style tags. We see that there is significant improvement in the understandability of the synthesized voice with clean style as compared to the baseline. However, we see that in some cases, there is a slight loss of speaker characteristics.

Thus, we see that in many cases the style transferred wavefile to clean style is much better sounding in terms of understandability. This increase in understandability is especially significant in the case of CallHome data. However, we do see that we lose some speaker characteristics. Going in the other direction, of trying to make the clean data sound more natural, we see that the style embeddings do not end up learning speaking style, and instead pick up on channel characteristics. The telephone style is in most cases not understandable and we can hear reverberation in the clean speech transferred to style of public speeches.

4.2.3 Summary

In this section, we showed experiments with clean multi-speaker English datasets as well as multi-style training with noisy English found speech spanning various speaking styles. We basically looked explored two aspects:

- **Domain Adaptation:** First we show experiments with 11 clean ARCTIC voices. We see that for certain speakers with a low F0, the synthesized speech is of lower quality. Thus, we try to improve on this by adapting the domain of the training data by shifting its composition with respect to gender and F0. However, our results show that shifting the composition of the training data has little effect in improving the quality of synthesized speech. However, we find significant gains when we fine-tune the model on the target speaker's data, irrespective of the composition of the training data.

- **Multi-Style Training and Style Transfer:** We also show results with multiple datasets of found speech spanning various speaking styles. We see that the neural models are much better than the CLUSTREGEN models in both objective and subjective metrics. We also compare a single model, an unfactored and factored model and show that in most cases, the single model and unfactored models perform almost similarly and are much better than the factored model, further strengthening our thesis that these models over-fit to speaker's characteristics and this is a good thing for our application of speech synthesis. With respect to experiments of style transfer we see that we can transfer from noisy to clean style and obtain much more understandable speech, however, going in the opposite direction, it is not possible to transfer a more conversational style to the clean speech, since it only transfer the channel characteristics.

In the next section we will explore the case when we do not have any transcribed data or language resource and consider a zero-resource setting, where we only have audio data available.

4.3 Unsupervised Acoustic Unit Discovery

The previous section explored cross-lingual augmentation assuming transcribed text. However, when we have untranscribed data, the question is how can we use this untranscribed data. For higher resource languages, where ASR models exists, we could use an ASR decoded transcript. However, for many low-resource languages, such ASR systems and ASR resources might not exist. Thus, in this section, we will consider a zero-resource scenario, where such an ASR system does not exist. The question then arises of how to build an ASR system in an unsupervised fashion. In the literature, this problem of building an unsupervised ASR can be broken down into two parts:

- Find good phonetic or sub-word units.
- Unsupervised word discovery and lexicon building given these phonetic or sub-word units.

In this thesis, we will consider only the first problem of finding a good inventory of acoustic units given only the audio data, and how we can use cross lingual resources to help find these acoustic units in a zero-resource scenario. We will show these methods on one example of a South African Bantu language called Xitsonga.

The main goal of this work is to find the basic units in language in an unsupervised fashion directly from the acoustics. These smaller units could either be sub-word units, phonemes or even sub-phonetic units. The goal is to find this auditory unit inventory such that the different auditory units are good enough to build TTS systems, *i.e.*, they are discriminable enough to be used as the building blocks to generate speech from. In this section we build on previous work done in deriving these phonetically motivated representations of speech [Metze and Waibel, 2002] that we refer to as "Articulatory Features" (AFs). AFs, which can be derived from arbitrary streams of recorded speech, and represent IPA-like phonetic features. These Articulatory features have been used

beyond speech recognition in representing expressive speech [Black et al., 2012] and cross-lingual voice conversion [Bollepalli et al., 2012]. Note our articulatory features might sometimes be called by others as Phonetic Features, and are not directly related to what we would call articulatory position features as might be discovered from an Electro-magnetic Articulograph. Our AFs are directly derived from speech in a language independent way, using standard software algorithms without any specialized hardware.

Finding a frame based AF representation is only the first part of our task. We then discover a segmental representation of the signal, that is phoneme-like derived from the AFs that is at least sufficient to reconstruct the signal using statistical parametric synthesis techniques.

The method used to derive these higher order segmental features uses the methods proposed in [Sitaram et al., 2013b] and [Sitaram et al., 2013a] and further developed in [Muthukumar and Black, 2014]. In [Sitaram et al., 2013b] and [Sitaram et al., 2013a], the authors built a text to speech system for languages without a writing system, which used *cross-lingual phonetic decoding* to come up with a phoneme-based written form for building TTS systems. However these techniques are still dependent on an originally seeded (cross-lingual) phonetic system. Since it inherently makes assumptions about the phoneme distribution in the original cross-lingually trained phonetic models. The more recent work that was proposed by Muthukumar *et.al* [Muthukumar and Black, 2014] derives segment-based "inferred phones" (IPs) using acoustically derived frame-based "articulatory features" (AFs) as illustrated in Fig. 4.3.

4.3.1 Relation to prior work

Most methods in this domain have looked at variety of methods related to either unsupervised pattern discovery or unsupervised acoustic modeling. The first set of methods treat it as a pattern recognition problem, by first finding repetitive patterns in the database and then using these patterns to build word based models [Jansen and Church, 2011] [Jansen et al., 2013] . In [Jansen and Van Durme, 2011] they use a hashing scheme to convert the raw input features to a binarized fixed length form and then do a clustering of these fixed length vectors, with the main goal of improving feature based term discovery.

The second set of methods include unsupervised acoustic modeling based approaches, wherein the speech is first segmented, then a clustering of these segments is done based on minimizing a certain objective measure and finally a re-training of the acoustic model is done. This process is repeated until convergence. In [Badino et al., 2014], the authors take a similar approach to subword modelling wherein, they train an auto-encoder to give encoder posteriors which are then binarised and clustered to a maximum of 64 units. These 64 units are then used to obtain a transcription of speech and based on this transcription the acoustic model is retrained and this process of segmentation, clustering and re-training continues until the model converges. In most systems these three subtasks of *segmentation*, *clustering* and *re-training* are carried out as independent tasks to one another.

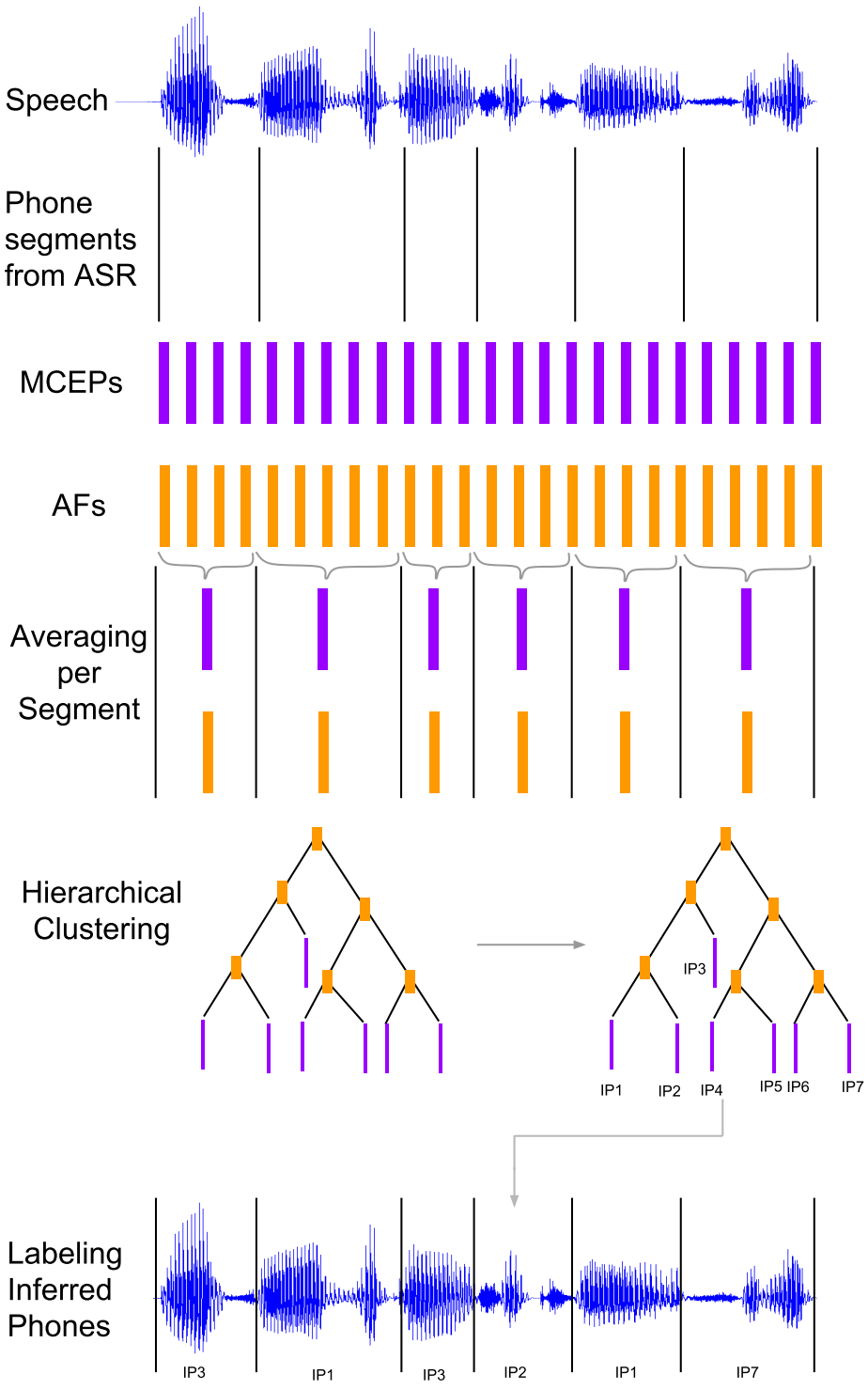


Figure 4.3: *Steps involved in extracting Inferred Phones.* taken from [Muthukumar and Black, 2014]

However, the authors in [Varadarajan et al., 2008] combine the segmentation step along with the clustering step by first starting out with a single HMM state to represent the entire dataset and then iteratively splitting these HMM states based on some objective measure. The authors in [Lee and Glass, 2012] go a step further, by jointly training an acoustic model using a nonparametric Bayesian model namely the Dirichlet Process mixture model. However, most of these methods approach the task of unsupervised unit discovery from an ASR perspective, with the objective of increasing the classification or discrimination ability of each unit, such that it is maximally distinguishable in an ABX test [Schatz et al., 2014][Schatz et al., 2013]. The ABX task measures how discriminable two different phones are when the occur in the same context, either spoken by the same speaker (within speaker) and spoken by different speaker (across speaker).

In our work, we approach the problem from a synthesis viewpoint and so are interested in finding the basic units in speech that are discriminable enough to be used to generate speech rather than classify between different phonetic or sub-word units. Thus the units our synthesis pipeline discovers are designed to be invertible and robust to speaker variation.

4.3.2 Experimental Methodology

Data and resources

The data that we used for our experiments was data provided as part of the Zero Speech challenge in English and Xitsonga. The English Buckeye database consists of 9 hours of data spoken by 12 speakers, with multiple speakers in the same audio file. Since the size of each audio file was many minutes long, we split the files into 10 second long files for some of our tasks and then recombined them during evaluation. For Xitsonga, we used the NCHLT Xitsonga Speech corpus [De Vries et al., 2014], which consists of 4.5 hours of speech by 24 speakers.

In addition, we used two other databases. The first was a combined database of the *RMS* and *SLT* ARCTIC data [Kominek and Black, 2004], which is around 2 hours of US English speech data and from one male and one female speaker. The second was a combined Hindi database with recordings from a local female speaker and the Blizzard challenge 2015 data [Black and Tokuda, 2005], consisting of recordings of one male speaker, giving a total of around 2.5 hours of data.

For all our experiments, we used the (US English) WSJ acoustic model distributed with the CMU Sphinx toolkit [Placeway et al., 1997] for cross-lingual phonetic decoding. We used a trigram German phonetic language model for decoding and performed multiple iterations of decoding and building targeted acoustic models from the decoded transcripts and the speech, as described in [Sitaram et al., 2013b].

We built all our models in the context of the Festival Speech Synthesis Engine [Taylor et al., 1998] and the Festvox voice building tools [Black and Lenzo, 2003], using CLUSTERGEN [Black, 2006] to build the Statistical Parametric Synthesis voices.

Feature Description

- **Baselines:** We used the MFCC features derived using the SPTK toolkit [Imai et al., 2017] and are 50 dimensional vectors (25 dimensions + Δ) which are used in synthesis and hence designed to be invertible.
- **Z-model Mceps:** The z-model Mceps are speaker normalized Mceps. Each speaker's Mceps are mean and variance normalized to match the average across all speakers in the database.
- **Cross-lingual phonetic decoding:** We decoded the speech from all the databases cross- lingually using the WSJ model and obtained phonetic transcripts. This process is done iteratively, with a targeted acoustic model being created at each iteration that is used to decode speech at the next iteration. Typically, we build voices at each iteration and measure the MCD of the voices. This iterative process is carried out till the MCD converges and stops improving. The iteration that produces the lowest MCD is selected as the best iteration. From our previous work we have found that the best labels are obtained in around iteration 3, so we chose the labels of iteration 3 for all our databases. The choice of labels is not critical here, since we only use the timestamps of these labels for the inferred phonemes.
- **Raw Articulatory Features:** We trained a neural network on a large corpus of multi-speaker English speech [Paul and Baker, 1992]. This predicts a 26 coefficient vector of 0-1 values for phonetic features, such as voicing, nasality, place of articulation, etc, trained from the labeling derived by forced aligned models of the original WSJ data. This produces a frame-based labeling (from Mel-cepstrum features).
- **Inferred Phonemes:** Using these AF's, the next stage is to find similar varying length segments in our acoustics which we do with a cross-lingual phonetic recognizer. Then we take these segmentations and re-cluster them into similar segments based on their frame-level AFs. Thus a cross-lingual phonetic recognizer may label all */k/* like sounds together, this post recognition re-clustering may separate out different types of */k/* (e.g. aspirated and unaspirated) into different segment-types. We can control the number of segment-types to find the number of symbols that can best re-construct the signal using statistical parametric text to speech techniques. We refer to these segment-types as "inferred phones" (IPs).

Evaluation Metrics

In order to make our method comparable to other systems we use the ABX measure for evaluation (in cases where we use frame based features such as Mceps and AFs), in addition to using MCD as a metric. The ABX metric measures the discriminative power of the sub-word units within and across speakers. For measuring the discriminability of an acoustic unit across speakers, if we select an ABX triplet to be such that, A and B are triphones from the *same speaker*, having the same context, but varying in the middle phone, like *put* and *pat*, while X is the same as A except from a *different* speaker. The goal then is to find linguistic units, such that A and X are much closer than B and X. Similarly

in the within speaker task, the goal remains the same, except that X is another instance of A from the *same* speaker. The Mel Cepstral Distortion (MCD) is a weighted distance measure used to measure the quality of speech synthesizers as described in section 2. To calculate the MCD, we hold out 10% of the data and build a synthetic voice using the rest of the data. Then, we re-synthesize the held-out data and compare the Mceps of the synthesized speech to the Mceps of the original speech. Furthermore, for the voice built using the inferred phones, we also did a word based comparison, to see how distinguishable words were.

Results

Table 4.15: *ABX on Mceps and AFs (% Error rate)*

Method	Data	Within Speaker	Across Speaker
Mceps	Buckeye	16.69	29.50
Z-model Mceps	Buckeye	16.98	28.01
Raw AFs	Buckeye	18.35	29.84
Mceps	RMS+SLT	7.56	18.45
Z-model Mceps	RMS+SLT	7.54	17.62
Raw AFs	RMS+SLT	7.53	15.02
Mceps	Xitsonga	19.69	33.93
Z-model Mceps	Xitsonga	19.74	30.69
Raw AFs	Xitsonga	18.12	29.73
Mceps	Hindi	8.89	28.13
Z-model Mceps	Hindi	9.11	27.22
Raw AFs	Hindi	8.33	24.91

From Table 4.15 we see that the articulatory features perform much better than the z-model Mceps across all databases. This indicates that the AFs are doing speaker normalization implicitly, and are more robust to speaker variation. We also see that the within-speaker error rate for the z-models is slightly higher than the Mceps, which is expected, given that the z-models are doing speaker normalization.

Next, we used the raw AFs to create IPs as described earlier. Instead of using the raw AFs for each utterance as we had done before, we replace the IPs for the utterance with the vector of average value of the phoneme's AFs, calculated across the *entire* database. Since the ABX task was set up to be a frame based evaluation, we replicated this average value for each frame that the IP spanned.

Table 4.16 shows the ABX results on IPs of different sizes. The stop value was used to control the number of IPs that were inferred. Stop value of 1200, 1000 and 800 were

experimented with. For the same stop value, the exact numbers of IPs varied across databases as can be seen in Table 4.16. As we see, none of the IPs were able to do

Table 4.16: *ABX on IPs of different sizes (% Error rate)*

No. of IPs	Data	Within Speaker	Across Speaker
81	RMS+SLT	14.30	19.06
65	RMS+SLT	14.64	19.41
55	RMS+SLT	14.92	19.57
71	Xitsonga	42.84	46.03
57	Xitsonga	44.09	46.43
82	Hindi	20.83	29.62
55	Hindi	20.05	28.91

better than the AFs or the Mcep baseline. We hypothesize that the major difference in the performance of the IPs on the Xitsonga database as compared to its AF's is because of the nature of the Xitsonga database which contains a higher variability of speakers, has been recorded over the telephone and consists of short utterances, a combination of all of which does not allow us the benefit of having clean data to train the cross-lingual model to give suitable IP representations. The AF's do perform better because, they firstly are frame based features and secondly, because they implicitly do speaker normalization.

Since our work focuses in finding phoneme-like segments in untranscribed data, we would like to test these IPs within the ABX test framework used above. But that framework is not very appropriate for a sub-word segmental model. As it is tested against some phonetic-like truth, the segment size will be similar and thus our deduced segments will be about the same size (given some reasonable assumption about finding appropriate boundaries). Thus scores will be a simple 0 or 1, depending on whether it fits the frame exactly or not. Thus we also present some other measures that might better show our own contribution.

The main issue is measuring phoneme sized units against phoneme sized units when the boundaries are one of the key variances in such a model, thus it would be better to extend the size of the comparison units to something more like words (specifically multiple phoneme-like segments long).

We analyzed our data (for which we have true transcriptions) and looked for multi-syllable words that appear more than once. We then used these words as our test words. We then compare these words with each other within and across speakers using different measures. In the cases where we are comparing the same word the measure should be lower, and when they are different words the measure should be larger. We can do this with simple frame based parameterization (as done above) but as the words are longer we can also do this in the inferred phone domain too. Uniquely additionally we can also do this using synthesis, as we can generate an acoustic stream from the symbolic inferred

Table 4.17: *Word-based scores-Within Speaker (DTW cost)*

Method	Data	Keyword	Not Keyword
Mceps	RMS-SLT	2.55 ± 1.89	4.44 ± 0.04
Average AFs	RMS-SLT	0.59 ± 0.55	0.92 ± 0.02
Synthesis	RMS-SLT	3.30 ± 5.05	3.67 ± 0.11
Mceps	Hindi	7.27 ± 9.11	8.78 ± 0.22
Average AFs	Hindi	0.93 ± 1.49	0.97 ± 0.05
Synthesis	Hindi	3.79 ± 3.95	3.84 ± 0.08

Table 4.18: *Word-based scores-Across Speaker (DTW cost)*

Method	Data	Keyword	Not Keyword
Mceps	RMS-SLT	2.22 ± 1.46	4.58 ± 0.04
Average AFs	RMS-SLT	0.41 ± 0.35	0.94 ± 0.02
Synthesis	RMS-SLT	1.59 ± 3.67	3.68 ± 0.11
Mceps	Hindi	10.08 ± 8.93	12.44 ± 0.30
Average AFs	Hindi	0.88 ± 1.53	1.10 ± 0.05
Synthesis	Hindi	4.45 ± 4.83	5.27 ± 0.14

phone stream. Table 4.17 column 1 lists the average DTW cost across all instances of the same speaker saying the same keyword, i.e., we are finding an average cost of matching the keyword in one sentence to all other instances of it and calculating an average of this cost as compared to column 2 which represents the average of the cost when matched to other words apart from the keyword the same speaker said in the corpus.

Table 4.18 column 1 compares the cost of measuring the keyword said by speaker 1 to all instances of the same keyword said by speaker 2 in the database , *vs.*, column 2 which lists the average cost of comparing keywords said by speaker 1 to all non keyword instances spoken by speaker 2. These measures within and across have been done as overall cost measures for Festival Mcep (baseline), Average AF's (the vector representation of the IP) and synthesized Mceps after rebuilding the voice from the unsupervised IP units obtained. We see that the IP as a feature for doing keyword spotting is successful, since in both cases across and within speaker it is able to give a lower cost on a simple DTW euclidean distance metric. One interesting point to note from this result is that the variance is lower for AF's as compared to those of the Mceps and is again indicative of the speaker normalization that is happening implicitly in deriving this representation.

The motivation behind using the speech synthesis pipeline was to find a set of linguistic units which are good at representing the speaker agnostic, invertible sub-units in the speech corpus. The measure of how good these units are in generating speech can be

measured with the MCD. Since the MCD is a distance based metric, lower is better, and it is database-specific, so it cannot be compared across the different databases. Thus in addition to reporting the MCD scores for the voice built from our best IP, we have also reported scores from cross-lingual phonetic decoding from WSJ acoustic model. Since the MCD is database specific, we also give ground truth (Full TTS baseline) when transcripts were available for comparison. Table 4.19 lists the MCD of voices built with transcripts from cross-lingual phonetic decoding, our best IPs and the full knowledge-based speech synthesizer (Full TTS-groundtruth) – for comparison. An increase of 0.08 is found to be perceptually significant, while an increase of 0.12 is equivalent to doubling the data [Kominek et al., 2008].

Table 4.19: *MCDs of voices built with different transcripts*

Data	Transcript	MCD
RMS-SLT	Full TTS	4.97
RMS-SLT	Phonetic Decoding	5.51
RMS-SLT	IPs	5.86
Hindi	Full TTS	4.94
Hindi	Phonetic Decoding	6.60
Hindi	IPs	5.94

Here, we see that for English, the phonetic decoding MCD is better than the IP MCD. Although this may seem surprising, we must note that we used the WSJ acoustic model to decode the *RMS-SLT* voice, so this is not being done cross-lingually. So, the phones in the phonetic voice are appropriate for this voice, which results in a higher MCD. Both the MCDs are higher than the knowledge-based MCD, which is to be expected. For Hindi, the IP-based voice has a lower MCD than the cross lingual phonetic voice which indicates that the IPs are a better representation of the speech for Hindi.

4.3.3 Conclusions

In this work we present an alternative unsupervised linguistic unit discovery method to find speaker agnostic, invertible speech units which are optimized for speech synthesis. We have investigated these features as an alternative to unsupervised acoustic modeling and in the context of performing well on the ABX task.

However, since our proposed features do not fit well into the ABX framework, which requires the discovery of units which can fit within its framework of phoneme-sized ground truth, we have also reported MCD scores which measure how good the synthesis of the Inferred Phoneme (IP) based voices is, which in turn measures the discriminability of the IP representation.

Although our inferred phonemes give a good symbolic representation of the speech they are still not the most ideal representation. As the number of segments in an utter-

ance are initially derived from a cross-lingual phonetic recognizer, they most probably represent phoneme-sized units. It may be better to allow them to be split into multiple subsegments (the IP-based text to speech synthesizer does automatically model subphonetic segments).

We find that on clean datasets, with less number of speakers, our proposed method works well. However, on noisy datasets like the Xitsonga dataset, which consists of many speakers and short utterances recorded via a telephone, we find that our model fails to perform as well, which we conjecture is due to the lack of good data to adapt the baseline model to.

The work presented here is still preliminary, a more elaborate speaker specific adaptation technique may help – though we have found that AFs are typically a better speaker independent representation. However, when synthesizing templates for matching, adapting the acoustics toward the target speaker in the utterance will improve performance.

Also IPs alone probably do not give all the information useful for word level matching. We know in IP-based text to speech that addition of word boundary information helps synthesis and thus finding super segmental information about syllable and word (like) boundaries will probably help higher level matching too (and certainly the generation of synthesized acoustics for later matching). In the future, we would like to explore multiple subsegment IP based voices and further this work by also augmenting it with word and lexicon discovery.

4.4 Summary

In this chapter, we looked at various methods of augmenting our training data for the target speaker with external data either in the same language or another higher-resource language, for both transcribed as well as untranscribed speech.

In the first section, we looked at data augmentation techniques for transcribed speech. We looked at both multi-lingual and multi-style training of attention based neural style speech synthesis models. For both multi-lingual and multi-style models, we find that the factored-embedding model seems to perform worse than either a single model trained on a single style or language or an unfactored model, where we do not factor the embeddings across various global attributes. We hypothesize that the reason for this is that over-fitting to a particular speaker's characteristics is good for our case of speech synthesis, and factoring the embeddings makes it more general and degrades performance as measured by the DTWMCD. However, the factored model, for both multi-style and multi-lingual synthesis, affords us more control in terms of achieving cross lingual synthesis and style transfer.

In terms of transferrability of these embeddings, we see that the model is very good at modelling speaker characteristics and the speaker-ID tends to dominate. Just changing the gender does not have much effect and cross-gender transfer is still difficult, with the gender of speaker-ID dominating. This shows us that it is difficult to factor out gender from the speaker characteristics. In case of the style transfer experiments, we see that the style embeddings are transferable when going from noisy speech to clean, and are

most noticeable for telephone or public speeches, however, going from clean to noisy transfers only the channel noise and not the speaking style.

We also show multiple experiments in augmenting both English multi-speaker and multi-lingual models with data similar to the target speaker, as well as transferring model weights from models trained on various datasets. We find that for the multi-lingual models, augmenting with similar languages improves performance significantly as compared to using a single speaker model from English and this is especially true for smaller subsets of 100 and 250 utterances. In case of the multi-speaker experiments however, we see no improvements in terms of DTWMCD, when we shift the domain of the training data closer to that of the speaker, as we did with speaker *AEW*'s F0. We see that we can generate speech from 100 long utterances of a new language and new speaker without much loss when transferring from a multilingual, multi-speaker trained on closely related languages, though the effect of adding additional external speakers in the same language had no effect. This is a promising result for low resource languages, especially if we can only afford to record data for 30 minutes in a new language. We also show experiments with respect to robustness of these models in modelling noisy data across various languages. We see that these models are not as bad as the CART based CLUSTERGEN models in modelling extremely noisy data such as the Bengali Babel data. Furthermore, they are pretty good at modelling the much cleaner Gujarati ASR corpus and we show that even without providing speaker IDs one can sample random speakers consistently across a sentence when synthesizing from a model trained on a multi-speaker dataset with missing speaker-ID labels. We also show that in the absence of speaker labels, the model can quite accurately synthesize with the original speaker's identity on a held out set of utterances. This is further evidence that these models are good at capturing super-sentential context, even when trained utterance-wise.

In the second section, we explore data augmentation techniques in a zero resource setting where we only have un-transcribed data and do not have access to an ASR decoder for decoding. In this section, we only address the first issue of finding good acoustic units, which will at least allow one to obtain a phonetic transcript, with which one can build TTS systems. We build on top of the work proposed by Muthukumar *et. al* [Muthukumar and Black, 2014] and find features that are maximally distinguishable to be used as phonetic units. We find that the articulatory features are speaker agnostic and perform the best in ABX test across speakers. The cross-lingual inferred phone (IP) based voices do better than phone based voices, and our proposed features do well on relatively clean datasets with few speakers, but fail on noisy datasets with multiple speakers.

Thus, in this chapter we show that it is difficult to train attention based neural sequence models on very noisy multi-speaker data. However, we have shown that end-to-end neural models provide a promising direction for grapheme based synthesis of low resource languages, if one has access to a relatively small (about 30 minutes) clean dataset of recordings from a single speaker. One form of high quality, clean audio data that we can find online without the need to explicitly record a speaker is audiobooks and podcasts. Thus, in the next two chapters we will look at modelling this relatively clean form of found data, and explore better methods of modelling their long-form prosody using neural models.

Chapter 5

Frame Based Speech Synthesis Models for Long-Form Audio

5.1 Introduction

A large portion of speech data that is freely available online is in the form of audiobooks and podcasts, which are long-form, sometimes acted or conversational datasets often with rich varying prosody. Current speech synthesis systems are not entirely suited to model long paragraphs and large prosody variations, since many of them have been optimized on training data sets involving monotonous, short read utterances [Kominek and Black, 2004]. Thus, in this chapter we address some issues in modelling long form audio as well as looking at different ways of learning a better prosodic labelling.

Speech synthesis has progressed to a stage where the best examples are comparable to pre-recorded speech. All of the best performing techniques (unit-selection [Hunt and Black, 1996], statistical parametric synthesis [Zen et al., 2009], [Zen et al., 2013] and the latest neural deep learning techniques [Oord et al., 2016]) require corpora of natural speech to train from. The training stage in a statistical parametric speech synthesis system involves mapping text to natural speech via a complex labeling of the acoustic data. One common method, pioneered by the Festival Speech Synthesis System [Black et al., 2014], is to construct utterance structures from which labels are derived in order to train a mapping from phonetic context and the audio signal. However current systems do not often care enough about the quality of these labels, and in order to improve any corpus based machine learning technique we must care about the accuracy of the given labels.

One might argue that recent end-to-end speech synthesis models circumvent the necessity to label datasets with linguistic features. The advantage of this approach is that the quality of synthesized speech and its prosodic variations can be learned directly from data. However, letting the black-box neural network do all the learning, gives us less control on the "*knobs*" one can turn to control prosody.

Thus, in this chapter we first look at traditional frame-based methods of modelling long form audio with default prosodic labels. We then try to improve on these labels

using iterative approaches that try to match the labelling to the actual acoustics of the data. In the next chapter we will explore neural methods of improving prosody and adding in the prosody "*knobs*" for more control.

5.2 Modelling Long Form Audio using Linguistic Features

In this section we describe some of the traditional techniques of modelling long form audio using linguistic features derived from utterance structures. Concretely, we look at traditional RNN based methods which have shown good performance on other sequence tasks such as machine translation and music sequence modelling. Specifically, we look at three RNN based architectures, the Clockwork RNN (Cw-RNN), the quasi RNN (QRNN), and the recurrent highway network (RHN) as well as the baseline bidirectional LSTM model. We show results on the Blizzard dataset of children's audiobooks.

5.2.1 Data

For this set of experiments we use a set of recordings of readings of Children's audiobooks recorded by a native British female speaker. The data was produced by Usborne Publishing and distributed as part of the Blizzard Challenge 2017 and 2018 [Black and Tokuda, 2005]. It consists of about 6 hours of data, out of which about 4 hours is actual speech data. All of the data was segmented by line, and the front-end was Festival [Black et al., 2014].

5.2.2 Model Description

First we investigated various existing recurrent architectures for the task of modelling long form audio. The main motivation for using these recurrent architectures was because the audiobooks have very long term structure, and the realization of a particular prosodic delivery depends on a long term context spanning multiple sentences. Learning these long-term contexts is difficult in a vanilla RNN because of the vanishing gradient problem [Pascanu et al., 2013]. So we wondered if using one of the RNN models proposed in the literature specifically designed to handle longer sequences would do well on this task of speech synthesis of audiobooks. We will first briefly describe these models and the architecture that we used for each.

Clockwork RNN

The Clockwork RNN as described in [Koutnik et al., 2014], partitions the hidden layer of a RNN network into N partitions, with each partition tuned to a certain period of the input sequence. Thus it allows different parts of the RNN's hidden layer to be tuned to different times in the history and the hope is that this will allow it to capture both local and global effects, by tuning to different periods in the input feature spaces and thus

better model the harmonic structure in speech. More concretely, it modifies the vanilla RNN as follows. If the vanilla RNN is described as:

$$y_t = W_h * y_{t-1} + W_i * x_t$$

The clockwork RNN converts the W_h and W_i matrices into,

$$W_h = \begin{pmatrix} W_{h1} \\ W_{h2} \\ \vdots \\ W_{hT} \end{pmatrix} W_i = \begin{pmatrix} W_{i1} \\ W_{i2} \\ \vdots \\ W_{iT} \end{pmatrix}$$

such that at each time step, only the weights for t mod $T = 0$ are executed. In our model, we experimented with various time periods, the details of which are described in the experiments section.

Recurrent Highway Network

Recurrent Highway Networks [Zilly et al., 2017], extend highway network [Srivastava et al., 2015] to the recurrent networks, by basically adding a highway layer within the recurrent transition, to solve the vanishing gradient problem in recurrent neural networks. A highway layer extends the feed-forward layer by gating how much of the input is added to the non-linear transformation from the hidden layer. Thus, a feed-forward highway network can be written as:

$$\mathbf{y} = \mathbf{T} \cdot \mathbf{h} + \mathbf{C} \cdot \mathbf{x},$$

where $\mathbf{x}$ is the input, $\mathbf{y}$ the output and $\mathbf{h}$ the non-linear transformation from the hidden layer. The gating functions, $\mathbf{T}$ and $\mathbf{C}$ are referred to as the *transform* and *carry* gates respectively and are learned by the model. The carry gate controls the amount of input that is propagated to the next layer, while the transform gate does the same for the hidden activation. Thus, it can be thought of as a gated skip connection with partial input.
For a recurrent highway network, let us denote the output of the recurrent layer at time t at an intermediate recurrent depth d as $\mathbf{s_d^t}$. Then the output at a recurrent dept d at time t is,

$$\mathbf{s_d^t} = \mathbf{T_d^t} \cdot \mathbf{h_d^t} + \mathbf{C_d^t} \cdot \mathbf{s_{d-1}^t},$$

where,

$$\begin{aligned} \mathbf{h_d^t} &= tanh(\mathbf{W_H x_d} + \mathbf{R_H^d s_{d-1}^t} + \mathbf{b_H}), \\ \mathbf{T_d^t} &= \sigma(\mathbf{W_T x_t} + \mathbf{R_T^d s_{d-1}^t} + \mathbf{b_T}), \\ \mathbf{C_d^t} &= \sigma(\mathbf{W_C x_t} + \mathbf{R_C^d s_{d-1}^t} + \mathbf{b_C}). \end{aligned}$$

Thus, a RHN with recurrent depth 1 is just a gated variant of a recurrent network like an LSTM or GRU.

Quasi RNN

The Quasi RNN (QRNN) was proposed by Bradbury et al [Bradbury et al., 2017] to address the problem that LSTMs cannot generalize to long sequences and to address the inherent assumptions of time invariance that the the max-pooling and average-pooling operations in CNNs assume, which cannot take advantage of the full time sequence. It has shown good results on character level machine translation, document level sentiment classification and language modelling. Just like the CNN, the QRNN has two components. The convolutional component performs 1D causal convolutions in the time domain, mapping an n dimensional, length T sequence $\mathbf{X}$ in $\mathbf{R^n}$ to a m dimensional length T sequence $\mathbf{Z}$ in $\mathbf{R^m}$ thus accounting for the sequential nature which is suitable for our application of speech synthesis. The 1D convolution or the masked convolution is also used to compute the gating functions for the forget and output gates, $\mathbf{F}$ and $\mathbf{O}$ respectively, which are used in the second component. Specifically, the equations for the output from the convolutional layer, $\mathbf{Z}$ and gating functions $\mathbf{F}$ and $\mathbf{O}$ are as follows:

$$\mathbf{Z} = tanh(\mathbf{W_z} * \mathbf{X}),$$

$$\mathbf{F} = \sigma(\mathbf{W_f} * \mathbf{X}),$$

$$\mathbf{O} = \sigma(\mathbf{W_o} * \mathbf{X}),$$

The second component is a pooling layer which does a dynamic average pooling, where it gates the output from the convolutional layer and additively combines it with a gated portion of the previous state. More concretely the dynamic average pooling at time t is defined as:

$$h_t = f_t * h_{t-1} + (1 - f_t) * z_t$$

where f_t is the gating function representing the forget gate and z_t is the output from the convolutional layers at time t. One can optionally also have and extra output gate o_t and then the above equations become:

$$c_t = f_t * h_{t-1} + (1 - f_t) * z_t$$

$$h_t = c_t \cdot o_t$$

The advantage of the QRNN is that it can be easily parallelized, given the entire sequence during training, while also allowing to take into account sequence information through causal convolutions and gated mechanisms.

5.2.3 Model Architectures

Clockwork RNN (CwRNN)

We experimented with different time-periods. The longest time period we went upto was 512 timesteps, with a decrease of half upto 8. However, from our experiments we found that an architecture consisting of 4 alternating layers of LSTM and clockwork RNN's followed by a linear layer at the output did the best. The first clockwork RNN

had time-periods of 32 and 16 and the second one 8 and 4. It would be expected that the longer timesteps would do better, however, our hypothesis is that there were too many model parameters and too less data to train such a big model for the longer time clockwork RNN's. More experiments with a 20 hour dataset would give us a better idea if this hypothesis is true. The clockwork RNN was by far the worst of the three recurrent models.

Recurrent Highway Network (RHN)

For the Recurrent Highway Network (RHN), we tried 2, 4 and 6 variants with alternating LSTM layers. We found the system with 6 RHN recurrence depth to be the best. This was also CMU's submitted system to the Blizzard Machine Learning Challege 2017, the results of which are described in Section 5.2.5.

Quasi RNN (QRNN)

For the QRNN we tried different stride lengths starting from 5 in steps of 5 upto 30. We also alternated with multiple QRNN layers increasing stride length and decreasing stride length. Specifically we used 3 QRNN layer with 5, 15 and 30 but didn't find much difference in using different stride lengths. Moreover, the performance of the QRNN model overall was not as good as the RHN.

Multiple layer combination

We also tried a combination of various layers as described below.

1. CwRNN-RHN-LSTM-Linear: In this model, the Clockwork RNN (CwRNN) had a time-period of 512, The RHN had a recurrence depth of 4, followed by an LSTM of 128 hidden units feeding into a linear layer at the output.
2. RHN-LSTM-RHN-CNN-RNN-Linear: In this model each RHN layer had a recurrence depth 4. The CNN had a stride length of 3. The penultimate layer had 64 units in the hidden layer as compared to all of the other layers which were 128 dimensional. We did not find much of a difference as compared to the 6 layer RHN.

Other Experimentation

- **Zoneout:** Zoneout is similar to dropout, except that instead of making random activations zero, it propagates the previous state of the RNN. It was shown to be a good regularizer [Chung et al., 2017]. However, we found minimal differences with zoneout applied to an LSTM and preferred the LSTM without zoneout. Thus, in all subsequent models we decided not to use zoneout.
- **Overlap and add:** We also tried predicting the context (future and past frames) around the current frame in a multi-stream setting, and then using an overlap and add method to get the final sequence. However, we found that the advantages of

this model were minimal compared to the time it took to train the model. We tried windows of 3 frames and 5 frames. We believe that since we were already using a bidirectional RNN with 300 BPTT size, this did not make much of a difference.

- **Weighted Loss:** We weighted the loss according to the Mel scale, given that humans are perceptually more sensitive to lower frequencies than higher frequencies. However, we found that this did slightly worse than giving all of the frequency bands the same weight. One hypothesis for this is that, as such the neural networks do worse in modelling higher frequencies and giving it lesser weight makes the performance worse.

5.2.4 Observations

From all of our experiments these are our observations:

- Gated skip connections, make a difference, given that the RHN was the best system followed by the QRNN.
- BPTT size made a big difference, especially going from 20 BPTT steps to 300 BPTT steps.
- Bi-directionality improves performance slightly as compared to the unidirectional models.
- Surprisingly we did not find the advantages of zoneout that other works have found.
- Weighting the cost function made the results worse.
- We did not find much advantage in predicting the context around the current frame and synthesizing with overlap add.

5.2.5 Results

Objective Results

In terms of the final Mean Cepstral Distortion (MCD) from the neural networks and informal listening tests that we performed, the network with 1 LSTM embedding layer feeding a recurrent highway network with a recurrence depth of 6, followed by a linear output layer performed the best. Like all of the networks it was used with the Adam optimizer, l2 weight regularization and Glorot weight initialization. The network was implemented in Keras [Chollet, 2015] with a Theano backend [Team et al., 2016]. The results comparing the different models with MCD scores are reported in Table, 5.1.

Blizzard 2017 Listening Test Results

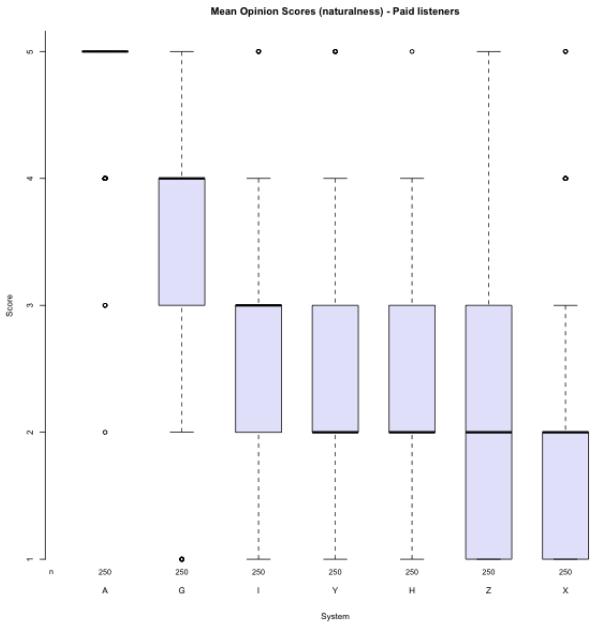
Apart from natural speech which was system A, 3 baselines were used. Systems X and Y were DNN based benchmarks predicting vocoder parameters, while system Z was a Wavenet style benchmark predicting waveforms directly. Ours was system H.

Table 5.1: *MCD results on Blizzard Dataset with Traditional frame-based RNN Models example wavs.*

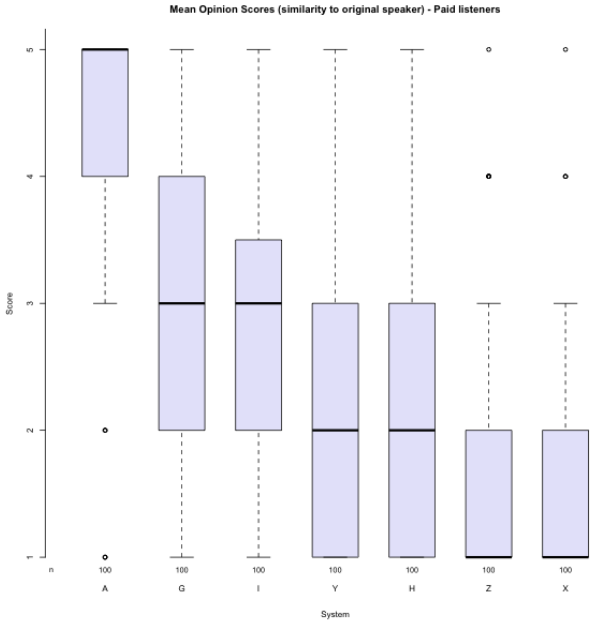
Experiment	Model	MCD
Baseline	CLUSTERGEN	5.60 ± 2.13
RNN Models	Clockwork RNN	5.76 ± 0.46
	Quasi - RNN	5.60 ± 0.45
	LSTM	5.45 ± 0.45
	Recurrent Highway Network	$\mathbf{5.43 \pm 0.44}$
Combined Models	CwRNN-RHN-LSTM	5.48 ± 0.38
	LSTM-RHN-CNN-RNN	5.44 ± 0.55
Other System Combinations	LSTM with Weighted Loss	5.54 ± 0.52
	LSTM with Zoneout	5.43 ± 0.47
	LSTM with Overlap Add	5.42 ± 0.44

Evaluation Methodology: The listening test evaluations for frame based and waveform based tasks were combined into one listening test. 50 paid listeners who are native speakers of English were used for evaluation. System orderings were varied by using a Latin Square design and listeners heard one system in each part of each section. There were a total of 11 sections 5 measuring naturalness (MOS), 4 measuring intelligibly (SUS) and 2 measuring speaker similarity (SIM). Thus, totally, there were 250 judgments per system for measuring naturalness, 100 judgments per system for measuring speaker similarity and 200 sentences per system for measuring intelligibility. The results of the listening test are as follows:

- **Naturalness:** In each part listeners listened to one sample and chose a score which represented how natural or unnatural the sentence sounded on a scale of 1 (Completely Unnatural) to 5 (Completely Natural) which is reported as the MOS score in Fig. 5.1a. As can be seen our system performs significantly worse than the other participating teams and is only significantly better than system Z in naturalness.
- **Similarity to original speaker:** In each part listeners could play 4 reference samples of the original speaker and one synthetic sample. They chose a response that represented how similar the synthetic voice sounded to the voice in the reference samples on a scale from 1 (sounds like a totally different person) to 5 (sounds like exactly the same person). As can be seen from Fig. 5.1 b, our system does significantly worse than the 2 other participating teams, but is comparable to baselines.
- **Intelligibility:** Listeners heard one utterance in each part and typed in what they heard. Semantically unpredictable sentences (SUS) designed to test speech synthesis intelligibility was used. Listeners were allowed to listen to each sentence



(a) Naturalness Listening Tests



(b) Speaker Similarity Listening Test

Figure 5.1: Naturalness and Speaker Similarity.

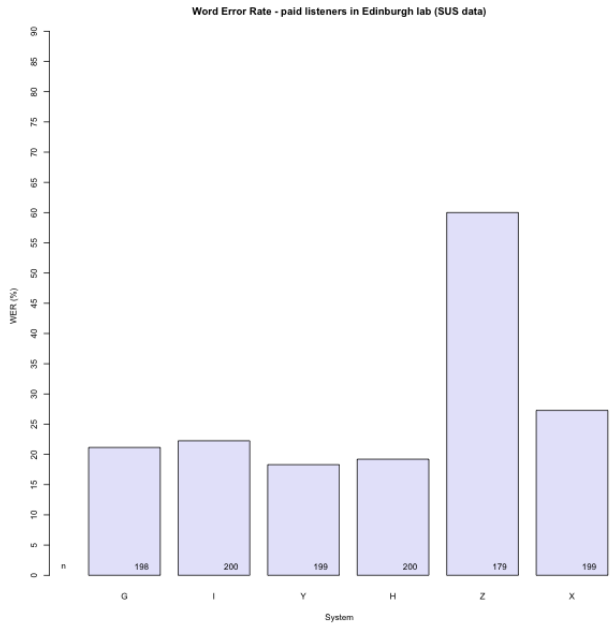


Figure 5.2: Intelligibility: Word Error Rate SUS.

only once. The results are reported as the Word Error Rate (WER). As can be seen in Fig. 5.2, our system does slightly (but not significantly) better than the other participating teams.

5.2.6 Summary and Conclusions

In this section, we showed a comparison of the performance of audiobook datasets on various recurrent model architectures built to model long sequences which have previously shown good results on sequential data. We find that these models are quite intelligible, however, they suffer in terms of naturalness. However, we see that though our model is as intelligible as the other benchmarks, it does not do very well in naturalness and speaker similarity compared to the other teams. One way of improving this might be to look at improving the front-end labelling of prosody. In the next section, we will look at one iterative approach of improving the prosody labelling, namely the phrase breaks and linguistic stress assignment and adding extra features that can explicitly capture this inherent prosodic structure in some way. In addition, given the trend in recent literature to get rid of recurrent models [Vaswani et al., 2017], [Gehring et al., 2017], in the next chapter we show results on the same dataset obtained using attention based sequence-to-sequence models with causal dilated convolutions.

5.3 Hierarchical Prosody Labeling using an Iterative Approach

In the previous section we showed results with traditional RNN based models on various datasets. We saw that the synthesized speech samples using traditional linguistic features was intelligible, however, the prosody wasn't always the best. The main problem with the current labeling techniques is that they do not rely on acoustics of the training data to predict prosodic events. They are good at finding phoneme mismatches in quality, however, the alignment of intonation events comprised of phrase breaks and prominence is still predicted directly from text and does not match the actual delivery of the utterance or the speaker's characteristics, which can be obtained from the training data.

In Festival for instance, if we consider the same utterance spoken by two different speakers, the assignment of phrase breaks and prominence remains the same in both cases irrespective of how the acoustics might differ. Fig. 5.3, illustrates for a short sentence *arctic_a0015 "It's the Aurora Borealis."* , the intensity and pitch contours along with the waveform for two male speakers having very different accents. On top one can see the intonational labeling assigned to both sentences. Vertical lines (|) indicate phrase breaks, colon (:) indicates pauses and brackets indicate syllables. The stressed syllables are shown in a bold dark colour while the unstressed syllables are shown in a lighter colour. As can be seen from the F0 contour the stress is assigned wrongly for the Indian Male speaker (*ksp*), since the last word is mostly said in a monotonous voice, with no break after *r* and Borealis is unstressed in this case which is not reflected in the labeling. The Scottish male (*awb*) on the other hand has a short break after the *ax* of aurora, which is not reflected in the assignment of prosody in this case. Moreover, until recently most corpora for speech synthesis [Kominek and Black, 2004] were designed to be short neutral utterances which did not have much intonational complexity. However, as the demand for prosodically richer speech corpora increases we need to find better methods of labeling prosody which is derived from acoustics and closely matches acoustics present in the training data.

In this section, we look at a new incremental method of finding a better prosodic labeling which matches the acoustics of the training data more closely. We propose to use the ideas from metrical structure, and intonational theory [Liberman, 1975; Liberman and Prince, 1977; Pierrehumbert, 1980; Ladd, 2008] to provide a more detailed labeling of natural speech, and use feedback from acoustic data to incrementally improve the accuracy of the labeling. We are following the work of Anumanchipalli *et. al* [Anumanchipalli et al., 2011, 2013; Anumanchipalli, 2013], in his Statistical Phrase Accent Models, where he hypothesized many different labeling of intonational phrases within larger prosodic phrases and chose the segmentation that allowed for the best intonational phrase model. In this work, we propose taking such work further and finding the best intonational phrases, and the best metrical tree structure (a binary-branching tree structure labeled with weak/strong) for an utterance [Ladd, 2008]. We will define *"best"* structures as those that allow the best improvements in synthesis of acoustics, duration and F0 parameters.

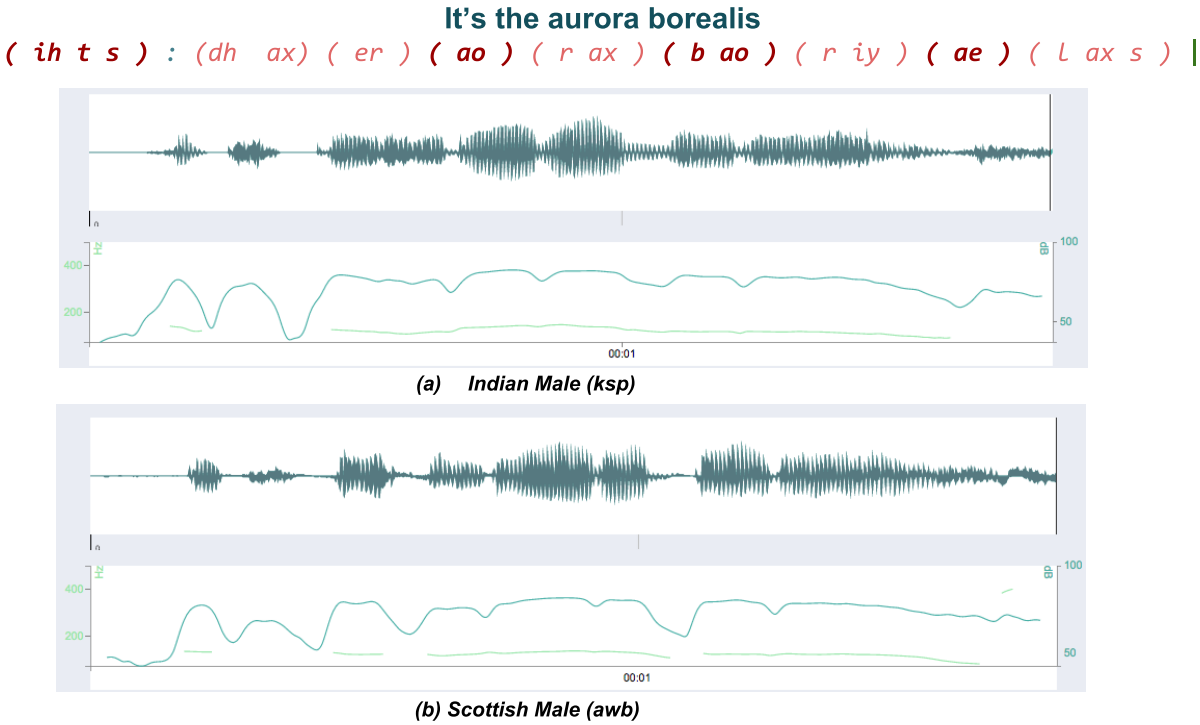


Figure 5.3: *Waveform, F0 (light green) and Intensity (dark green) plots for 2 males (ksp and awb) speaking the same utterance from the ARCTIC corpus, plotted using the Prosody Tagger [Domınguez and Wanner, 2016]*

We are building on top of existing synthesis structures, assuming prosodic phrases are bracketed by short acoustic silences. But we go further and appeal to the theories of intonation, and metrical foot structure to define sub-phrases that need not be rendered with acoustic silences, but still affect F0 contours, duration, and acoustic-phonetic realization. But as such intonational theories do not yet have well-defined automatic parsers, we will derive these labels incrementally changing basic labeling to find which labels lead to the best prediction models. Having obtained the best labeling using the incremental approach, we will then use the trained parser to be able to predict the metrical parses for unseen text.

Although English intonation modelling may already be at a plateau that will be hard to beat for simple declarative sentences, we wish to apply this labeling and modelling technique to less well modeled texts. For English we wish to consider audio-books which have more interesting intonational structures. We believe that providing an accurate hierarchical intonation structure labeling automatically on large natural speech corpora that models the true delivery of the speech will aid all types of corpus-based synthesis from unit-selection to end-to-end waveform generation systems like WaveNet.

Data driven approaches for hierarchical prosodic labeling has been explored before. Ostendorf and Veillieux [Ostendorf and Veilleux, 1994] show the effectiveness of a hierarchical phrase model, but derive it from hand-labelled data, and stop short of defining

a deeper metrical structure within intonation phrases. Badino [Badino et al., 2012] improves upon Festivals pitch accent predictor by adding in more syntactic and semantic features. However, the subjective evaluation results showed little difference between the proposed system and the baseline Festival system, which was attributed to over-flattening the complex hierarchical structure of prosodic prominence. This result reaffirms the fact that the hierarchical prosodic structure [Ladd, 2008] is indeed very informative and there is information in the structure that is lost when the tree is flattened. Anumanchipalli [Anumanchipalli et al., 2013] takes a first step in finding this hierarchical structure implicitly by first using a data driven approach to grouping syllables into accent groups and then re-estimating the tilt parameters which can be thought of as implicitly learning the hierarchical structure between the different accent groups. He showed about 80 % improvement overall in preference scores. His method optimizes for F0 (not duration and acoustic-phonetics) and does not address metrical structures within intonational phrases. More recently Lenzo in his thesis [Lenzo, 2017] uses a twiddling approach to incrementally *twiddle* utterances to find the best labeling. He shows results on the ARCTIC set which is prosodically neutral. In this work, the goal is to improve upon the model proposed in [Anumanchipalli, 2013] and the incremental approach taken by Lenzo [Lenzo, 2017] and explore better seeding techniques using a different twiddling approach than [Lenzo, 2017] to converge to the best hierarchical intonation structure as determined by the scoring function. We explore various scoring functions which depend on F0, acoustics as well as duration. Furthermore, our method makes no assumptions about the language and thus allows us to apply it to other languages and datasets that have no prosodic labeling. In the next section, we will briefly describe the theory behind Metrical Phonology and in the subsequent section explain our iterative approach.

5.3.1 Metrical Phonology

The theory of Metrical Phonology was put forward in the Liberman's thesis [Liberman, 1975] as a theory to describe *tunes*. He wanted to describe a universal way of annotating different prosodic realizations of the same sentence which might give rise to different meanings. In his thesis he argues that there are three independent phenomena, the *stress*, the *tune* and the *phrasing* of the utterance that work together to produce a certain prosodic realization of the same string of characters that can convey different messages.

- **Differences in Stress:** The primary stress on a segment of text can be determined by rules based on the context and part of speech of the words in the sentence. Differences in primary stress can lead to different meanings of the same utterance. Take for example, the sentence *An English Teacher*. The meaning of the sentence changes depending on which word the primary stress is attached to. If the primary stress is on the word *English*, it means the teacher who teaches English, while if it is on the word *teacher*, it means a teacher who is English [Liberman, 1975].
- **Differences in Tune:** In his thesis, he considers the tune as different from the stress. The same sentence with same primary stress, such as *an English* ***Teacher*** with the primary stress on teacher can be realized in four different ways by varying

the tune and lead to differences in meaning. These four different variations are depicted in Fig. 5.4, taken from [Liberman, 1975]. The first contour is a neutral realization. The second is a question and shows the pitch contour rising at the end. The third is an incredulous statement and shows a very high rise and fall on the word English. The final realization is one where the speaker is stating a fact that is very obvious. Thus, we see that even when we fix the primary stress on a sentence, by varying the tune, one can convey a different message or emotion.

- **Differences in Phrasing:** Finally, the phrasing or the breaks, as depicted by commas and pauses in sentences also yield different meanings. For example, consider the sentences, *Mary finds joy in cooking her family and her dog*, *vs.* the sentence *Mary finds joy in cooking, her family, and her dog*. The first one implies Mary likes cooking her family and dog, while the latter implies she really loves her family and dog.

Thus, Liberman wanted to put forward a theory of how the tune relates to text and how this tune is changed given some lexical stress and phrase breaks, and how exactly can this be integrated into the linguistic structure. This theory of metrical phonology as a described in [Liberman, 1975] is a theory of *relative hierarchical prominence* between tonal segments, where each tonal segment is built up from its children in a hierarchical fashion. Since the tune, lexical stress as well as phrasing together define the F0 contour, in this theory, the tonal segments are built up starting with syllables annotated with lexical stress and built upto the phrase level. Each node of this binary tree has a strong or weak label denoting the relative prominence between its two children. Figure 5.5 denotes the metrical trees for the same prosody contours depicted in Fig. 5.4. The leaves here denoted in green are dark for the syllables having lexical stress 1 and light green for syllables having stress 0. The primary stress here is on the first syllable of the word *teacher*, denoted by a red bounding box. Thus, given that a tree of relative prominences is built up starting with an initial labelling of lexical stress over syllables, and ends at the phrase level, we need these markings of relative prominence and phrase breaks to be accurate in order to learn a good metrical tree relation that matches the acoustics. Thus, our goal is to improve this labelling and in doing so improve the metrical trees learned, with the goal of improving the final TTS quality. In the next section, we describe our nomenclature for various terms and our base assumptions from which we work.

5.3.2 Method

Nomenclature

- **Phrase:** Utterances are made up of phrases, which are chunks of the utterance surrounded by silences. These big phrases are denoted as BB by Festival. An utterance might have one or more of these phrases.
- **Intonational Phrase (IP):** Each of these phrases is further divided into one or more intonational phrases, which we assume is not separated by a silence. These need to be discovered through the algorithm. We assume that the intonational phrase

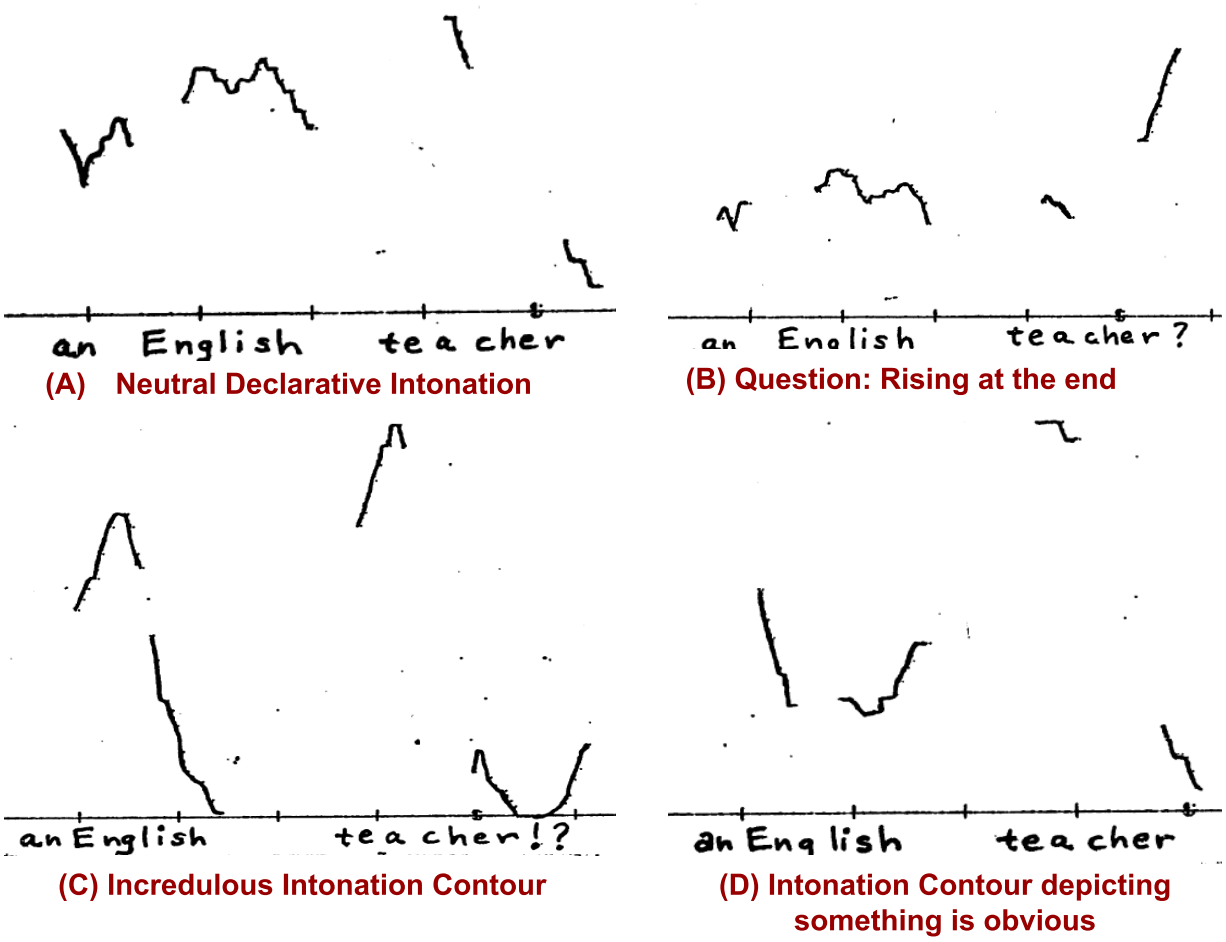


Figure 5.4: *Four different variations of the sentence "An English **Teacher**", depicting differences in tune, taken from [Liberman, 1975]*

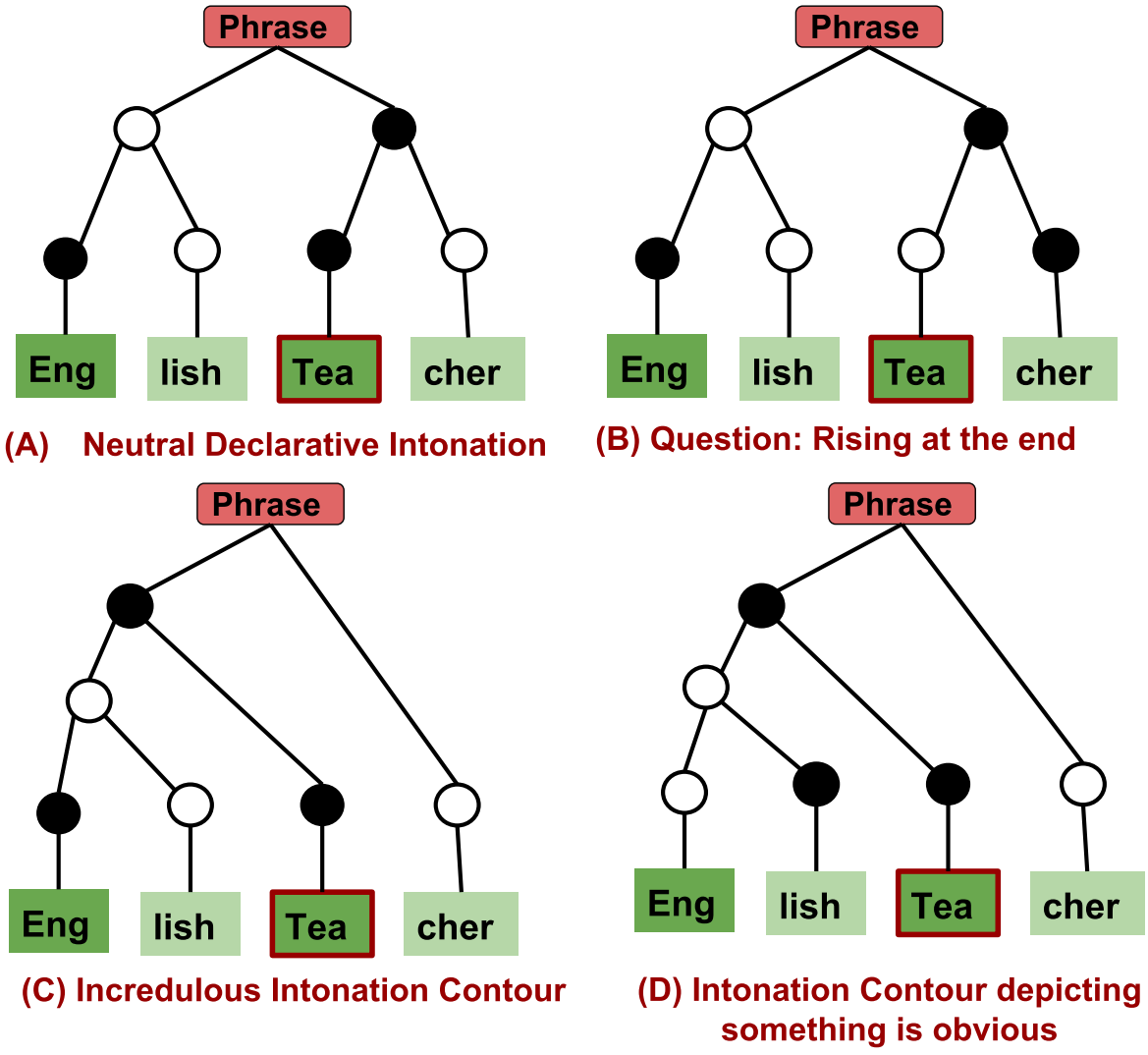


Figure 5.5: *Metrical Trees for the sentence "An English* ***Teacher****", depicting differences in tune.*

boundary can only occur on the end syllable of a word.

- **Hierarchical Intonation Structure:** We assume that similar to the metrical structure, our proposed hierarchical structure is binary, rooted at the IP and with syllables at the leaves. Furthermore, we assume each node in this structure is labelled with a strong (S) or weak (W) label.

Steps

The steps followed to obtain this hierarchical intonation model is illustrated in Fig. 5.6 and is as follows:

- **Initialize:** First initialize the phrase breaks and prominence on syllables using a labeling either obtained from Festival or Prosody tagger [Domınguez and Wanner, 2016] as explained in Section 5.3.4 under initial seed labelling.
- **Train an Initial Grammar:** Given these S-expressions of utterances represented as a sequence of strong/weak syllables, train a metrical grammar using an SCFG parser. We start with an initial grammar, with equal probability given to each production.
- **Train and Synthesize with Metrical Tree:** Use this metrical grammar obtained from the SCFG parser to parse data, flatten the trees and add them as additional features. Retrain the acoustic and duration models and synthesize and calculate utterance-wise scores using a scoring metric as described in Section, 5.3.4 under scoring function.
- **Twiddle breaks:** Twiddle the breaks, by randomly moving breaks in a range of 3 syllables before or after from the current syllable and retrain the grammar.
- **Train and Synthesize with new breaks:** Given the new *twiddled* breaks, retrain the grammar and using the new parsed utterance structures, retrain the acoustic and duration model.
- **Re-score and Retain:** Obtain utterance-wise scores and only retain those utterances whose score has improved.
- **Twiddle Prominence:** Now twiddle the stress on each syllable with a probability of 0.5 that it will change from its current value.
- **Train and Synthesize with new prominence:** Given these new *twiddled* stress values on syllables, retrain grammar, parse utterances with the new grammar and retrain acoustic and duration models.
- **Re-score and Retain:** Score each utterance on the new model and only retain sentences that have improved.

5.3.3 Data

The experiments were carried out on primarily two corpora of audiobooks and one corpus of ARCTIC sentences [Kominek and Black, 2004]. The first corpus is a 6 hour 45

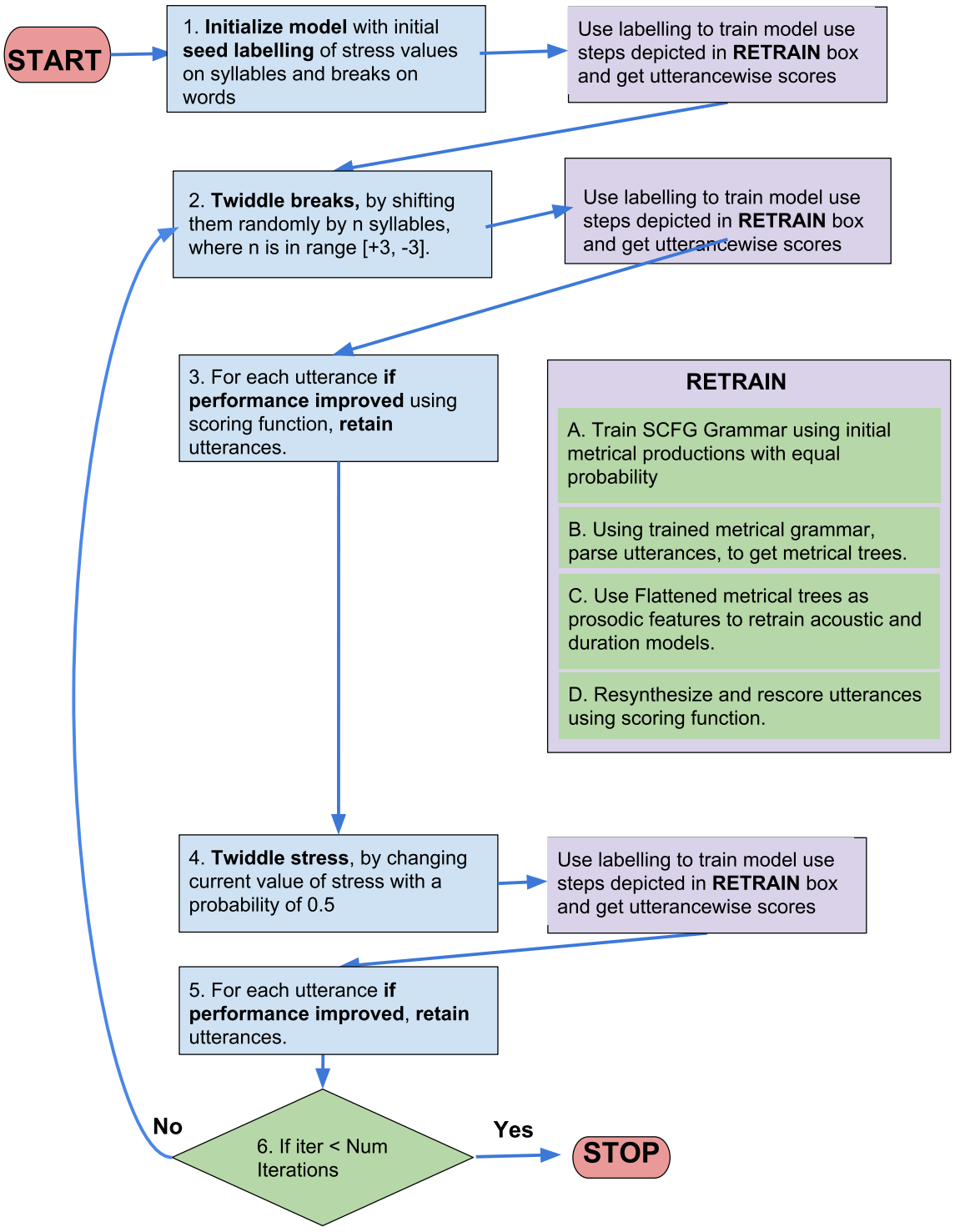


Figure 5.6: *Flowchart of steps followed in iterative process to obtain better prominence and phrase break labelling*

mins corpus of a US male speaker reading the The Adventures of Tom Sawyer (TATS), distributed as part of Blizzard Challenge 2012 phase 1. The third audiobook corpus is a larger 16.36 hour corpus of a US English female provided by Lessac Technologies as part of Blizzard Challenge in 2011, which we call the Nancy corpus. The final corpus is a one hour corpus of an American male speaker (RMS) recording a set of phonetically balanced utterances distributed as part of the ARCTIC corpus. All of the data was segmented into utterance level chunks, which are still much longer than the average utterances in the ARCTIC corpus. The front-end was Festival [Black et al., 2014].

5.3.4 Experiments and Results

Initial Seed Labeling:

We explored various seed labeling strategies to seed the iterative procedure of twiddling breaks and stress. These included:

- **Default (D):** This was the default assignment of stress and breaks as given by front end.

- **Single-syllable Closed Class stress modified (C):** This included converting all closed class single syllables to be unstressed, which were originally assigned a strong stress by the front-end. This was inspired by the finding in [Lenzo, 2017] where only twiddling the stress values on function words yielded some improvements in performance.

- **Prosody Tagger (PT):** Prosody Tagger [Domınguez and Wanner, 2016] includes an automatic way of finding the prominence and breaks in a utterance using raw speech. It does this by extracting features from the raw audio waveform including pitch, duration and intensity peaks and valleys, which are then used to predict prominence and breaks. We used these predicted features as the stress and break values is Festival. The stress values in this case tend to be mostly unstressed except for the prominent points.

- **Default stress with Prosody Tagger breaks (DPT):** Since the breaks in the front-end are assigned based on pauses and Prosody Tagger does the break prediction using only acoustics of the raw waveform, in this formulation, we combined the default stress assignments from the front-end with the breaks obtained based on acoustics directly from Prosody Tagger.

The results with initial seed labeling are shown in Fig. 5.7. We show MCD, RMSE of F0 and duration on the three voices with different seed initialization. Here B denotes the baseline results without adding metrical tree features. We see that Prosody Tagger does better than all the other techniques in case of F0. We cannot derive much insight from MCD, since the changes in MCD are not very significant and neither in case of duration,

since it is not consistent across the three voices.

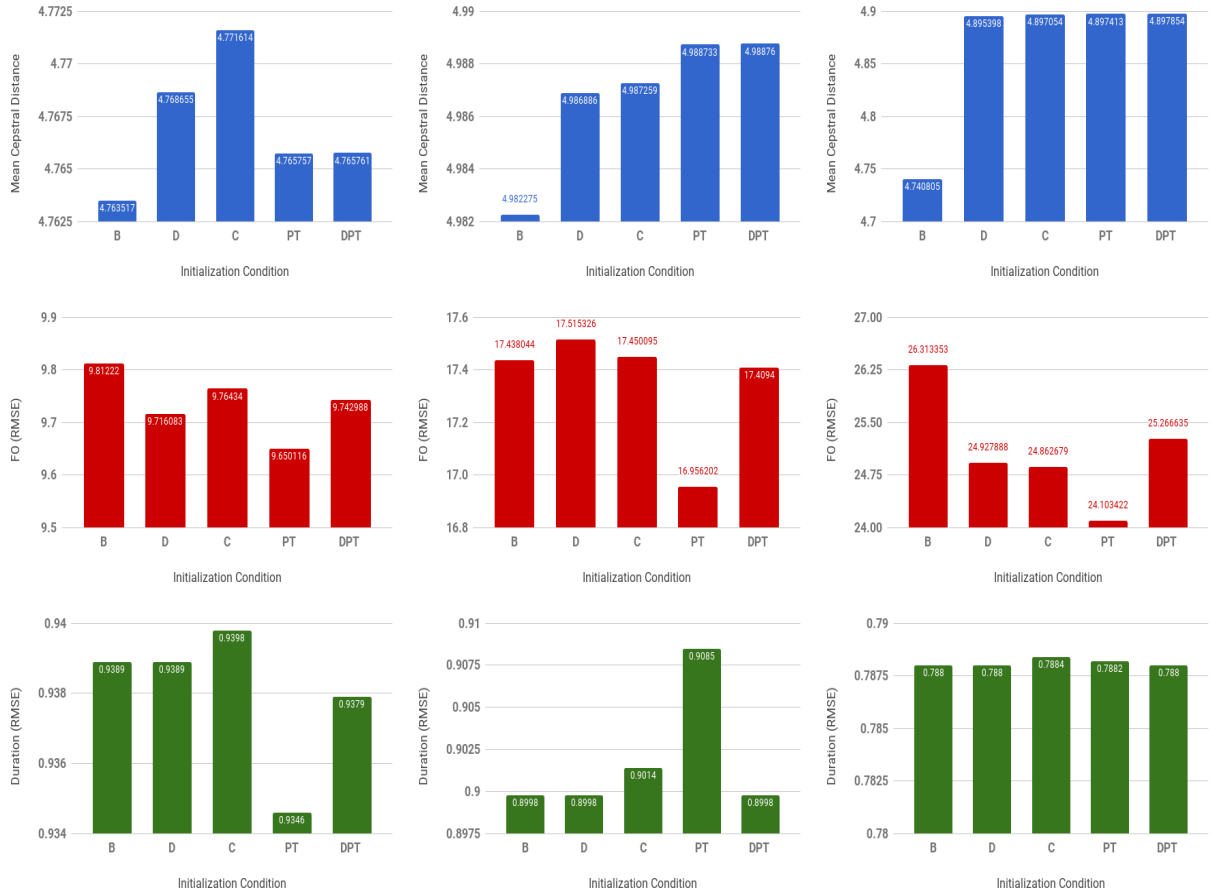
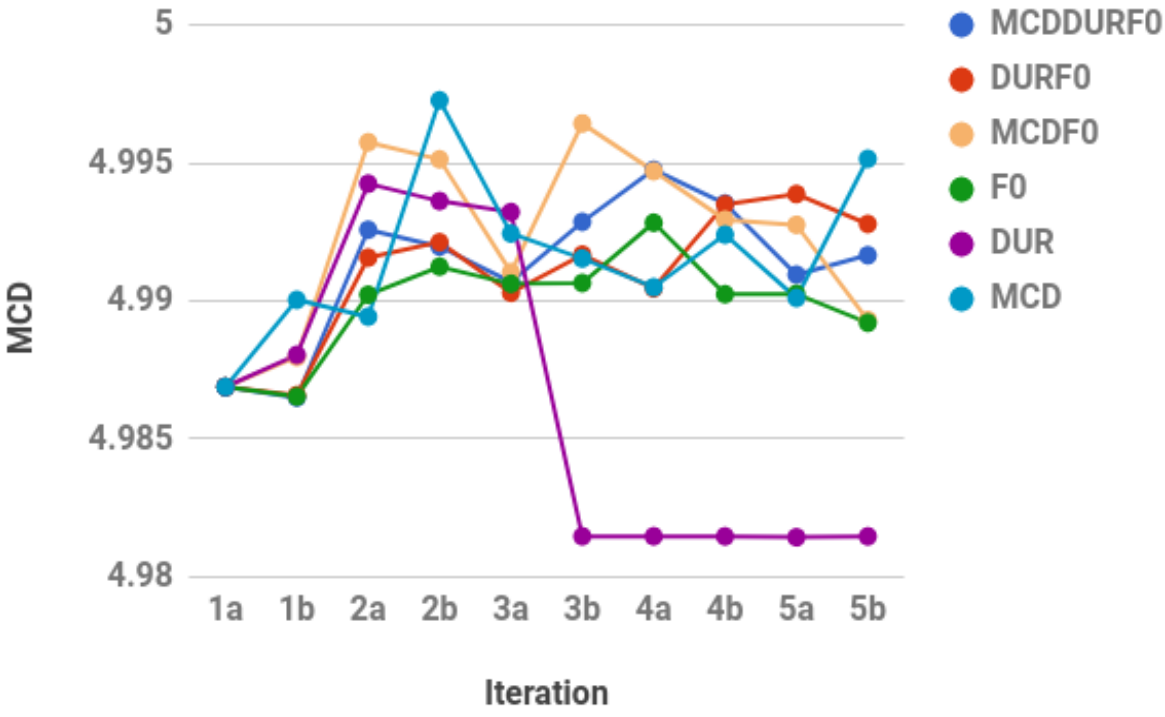


Figure 5.7: *Initialization results for RMS (col1), TATS (col2) and NANCY (col3), for MCD, F0 and Duration. B denotes Baseline and the rest are as described in Sec. 5.3.4*

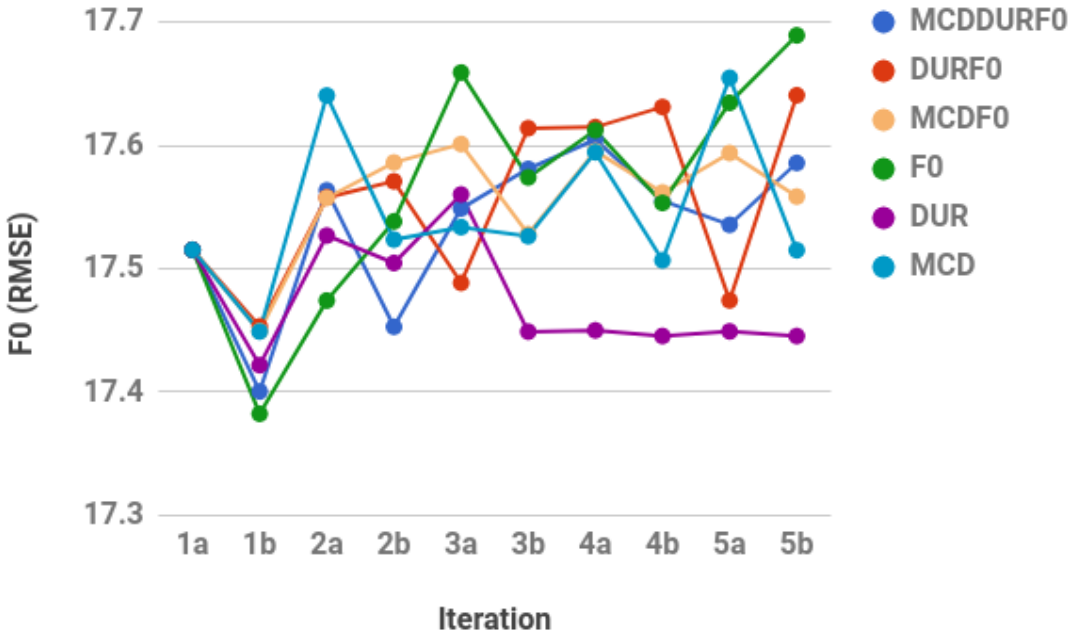
Scoring Function:

Since it is difficult to know what factors determine prosody, we explored various scoring functions that would determine if an utterance with a new labeling is retained. Since prosody can broadly be said to be influenced by acoustics, duration and the fundamental frequency, we looked at Mean Cepstral distance (MCD) of acoustic features (MCEPS) and root mean squared error (RMSE) of F0 and Duration as cost functions as well as a combination of them.

Figure, 5.8 illustrates the MCD and F0 scores obtained on a held out test set on the TATS corpus over 5 iterations, when using different scoring functions. We found that RMSE of z-scored state level duration as a cost measure seems to converge fastest within couple of iterations. However, we found the performance on a held out test set in terms of MCD and F0 scores, did not change much with different scoring functions and the performance over iterations also was not very different either. In addition, we noticed



(a) *Iterative MCD scores on TATS*



(b) *Iterative F0 scores on TATS*

Figure 5.8: Iterative MCD and F0 scores on speaker *TATS*.

that when we used MCD as a scoring function, the MCD and F0 scores for the same iteration have the opposite trend-wise direction. This also points to the fact that MCD might not be a suitable objective measure for this task.

5.3.5 Conclusions

In this section we explored an incremental method of learning a metrical tree grammar to learn a hierarchical intonation model of prosody to better match the actual delivery of the acoustics. We show that using raw acoustics to influence initial labeling improves F0 prediction, but does not have the same effect in terms of MCD and duration. In terms of cost function, we see that RMSE of duration converges fastest and we also show that the trends in iterative MCD and F0 scores when using MCD as a cost function move in opposite directions, which might imply that MCD might not be the best measure for this task and we need to look into better objective metrics which can capture the subtleties in the F0 and intensity contour as well as take into account duration. One possibility would be working in an embedding space of F0 contour combined with a RMSE over duration, or using reinforcement learning techniques or human in the loop to train a prosody model instead of using MCD or some aggregate mean statistic as a measure of the goodness of a time varying quantity.

5.4 Summary

In this chapter we looked at frame-based methods for learning better prosody models on long form audio such as audiobooks and podcasts. In the first section we explored traditional frame-based RNN based models on the Blizzard dataset. We found that even though these models are good at understandability, they fall short of modelling prosody well in terms of naturalness. We hypothesized that one reason this models might not be doing so well was because of the poor labelling of lexical stress and prosodic phrase breaks which are agnostic to the actual delivery of the utterance. Thus in section 2 of this chapter, we try to improve upon this labelling using an iterative technique, at each time "twiddling" the assigned labelling in order to learn a latent metrical tree over the utterance which we use as features in synthesis and optimize it using a a scoring function over the acoustics. We explored various methods of initialization and scoring functions using these iterative methods. We found that we get some improvement in using the breaks provided by the prosody tagger in terms of F0, however, in terms of duration and MCD, we do not see much change with a change in labelling. Moreover, we find that MCD and F0 go in opposite directions after each iteration. The problem is that we need to find a better metric of measuring prosody, some metric that can take into account the time varying nature of "relative prominence". Thus, in the next section, we will explore attention based methods of modelling prosody and inducing a notion of "prominence attention" into the model.

Chapter 6

Sequence-to-Sequence Models for Long form Audio

So far we have shown results using traditional linguistic features with mainly frame-based models as well as explored a method of improving the linguistic feature labelling of phrase breaks and prominence using an iterative approach. However, most of these frame based methods assume we have a good front-end that exists for the language and has been trained to produce accurate labelling. However, as we saw from the results in the previous chapter, using hand-engineered features produces speech that is intelligible, but it does not necessarily produce speech which is natural sounding, especially for training data consisting of audiobooks with rich prosody. In addition, our efforts to use iterative approaches to improve upon this labelling also did not yield improvements. Thus, in this chapter we will replace the hand engineered linguistic features with a neural network and allow the neural network to learn good feature representations as well as ***where*** and ***what*** to focus on, to produce better prosody.

The recent success of sequence to sequence models with attention [Sotelo et al., 2017], [Taigman et al., 2018], [Wang et al., 2017], [Tachibana et al., 2018], [Arik et al., 2017] for generating good audio directly from characters or phones has been promising. However, many of these attention mechanisms do not generalize well to long utterances and in addition, they have not been designed to capture prominence as part of the attention mechanism. Thus, the goal in this chapter is to look at various methods of augmenting or modifying sequence-to-sequence attention based models for better prosody prediction and control. First, we compare existing attention based models for prosody prediction with different input feature representations. We then look at one method of augmenting these end-to-end models with utterance level prosody embeddings guided by metrical theory. Third we propose a novel multi-scale, multi-headed, multi-hop attention (3M attention) to capture a set of local and global prosody related attention weights. Finally, we look at using the error signal obtained as an extra set of *prominence weights*, since, it would be expected that a model might have larger errors where there are larger variations in prosody.

In the next section, we will first lay the foundation of common terminology to be used and then in the subsequent sections describe the model, the prosody embedding

mechanism and proposed extensions to the attention mechanism for long-form audio.

6.1 Attention Mechanisms

Various attention mechanisms have been developed. In this section, I will briefly describe some of these attention mechanisms and the common definition of some terms. Broadly, the attention mechanism can be defined as :

$$C(i) = \sum_{j=0}^{T_{enc}} F(Query(i), Key(j)).Value(j)$$

where, $C(i)$ is the context vector at time $i \in [0..T_{dec}]$. The function F is called the compatibility function and calculated using some function of the $Query$ and Key. It determines by how much to weight the $Values$.The $Query(i)$ is some function of the decoder state and the $Key(j)$ and $Value(j)$ are some functions of the encoder inputs or outputs or a combination of the two and $j \in [0..T_{enc}]$. Thus different attention mechanisms differ depending on what is used for the $Query, Key$ and $Value$ and the compatibility function. In section 6.1.1, we will briefly describe some of the most common mechanisms.

6.1.1 Terminology

In all of the equations we will use the following notation:
h_j is the encoder output for $j \in [0..T_{enc}]$,
s_i is the state at the last layer of the decoder, and
c_i the context vector for time-step i, for $i \in [0..T_{dec}]$,
$\alpha_{ij} = F(Query(i), Key(j))$ are the weights i.e., the normalized outputs of the compatibility function and
e_{ij} are the un-normalized weights before the application of the softmax, i.e,

$$\alpha_{ij} = \frac{exp(e_{ij})}{\sum_{k=0}^{T_{enc}} exp(e_{ik})}$$

In addition, if not mentioned, in most cases we will assume that the final context vector is,

$$c_i = \sum_{j=0}^{T_{enc}} \alpha_{ij} h_j$$

i.e., we assume the default $Value$ to be the encoder output.

- **Additive Attention:** Additive Attention also sometimes referred to as Bahdanau attention was first proposed in [Bahdanau et al., 2014]. It is called additive, because the compatibility function is additive in nature. Specifically, it is described as:

  $$e_{ij} = v_a^T \tanh(W_a s_{i-1} + U_a h_j),$$

  where W_a, U_a and v_a are all learnable parameters.

- **Dot-product Attention:** Instead of using the additive attention, one can combine the key and query using a dot product. This attention is described as:

$$e_{ij} = \phi(s_{i-1}).\psi(h_j),$$

where ψ and ϕ are some functions, and in many cases identity. Many times this dot product is scaled by the square-root of the dimensionality of the key vector to reduce variance and then it is called scaled dot-product attention.

- **Content based, Location based and Hybrid Attention:** In the speech recognition literature, [Ren et al., 2015],[Chan et al., 2016], they make a distinction in terms of whether one takes into account the current location information or not, by using a recurrence in the attention. Thus, broadly one could define the compatibility function in each case as:
  - Content based: This is the original version of attention. However in case of speech, we also want to take into account the current location since speech has many more similar repeating units in terms of phones.

$$\alpha_i = F(s_{i-1}, h)$$

  - Hybrid Attention: Thus by adding in location information, by also giving it information about the previous attention vector, we can tell it where exactly it is in the sequence. Thus, this attention can be described as,

$$\alpha_i = F(s_{i-1}, \alpha_{i-1}, h)$$

  - Location based Attention: If we ignore the keys and only use the previous attention vector and decoder state to predict the next attention vector, it becomes location based attention.

$$\alpha_i = F(s_{i-1}, \alpha_{i-1})$$

- **GMM-Attention:** This is another form of the location based attention, originally proposed by Graves [Graves, 2013] in the context of sequence modelling with bias. In this type of attention, the attention weights are estimated as a Gaussian Mixture Model (GMM), where the GMM parameters are some function of the Query vector. It is called location based attention, because for each Gaussian component one predicts the length of the window (the variance), the location of the window (mean) and the importance of that window (mixture weights). This type of attention was also used in [Sotelo et al., 2017],[Taigman et al., 2018] with some modifications from the original paper.
- **Self Attention:** Also called intra-attention involves attending to the same sequence. Thus in this case the key, query and value all come from the previous timesteps of the same sequence being predicted.

- **Multi-step or Multi-hop Attention:** Multi-step or Multi-hop attention was proposed in [Gehring et al., 2017]. It involves allowing each layer in the decoder to have its own attention mechanism. In some variations of this, the first layer gets the original query vector which is the decoder target and then each subsequent layer uses the output of the previous decoder's layer as the query vector.
- **Multi-headed Attention:** Multi-headed attention was introduced in [Vaswani et al., 2017], projects the same key, query and value vector into H different projections using a feed-forward layer, to yield H different context vectors. These different heads are expected to capture different aspects of the encoded signal.

In this chapter, we only be using soft attention in all cases, since using hard monotonic attention, *i.e.*, assigning the context vector, c_i at time i to some h_j is difficult because of sampling. One can use a soft attention to approximate hard alignment in expectation as is done in [Raffel et al., 2017], however, for all of our experiments we will only consider soft-attention.

6.1.2 Related Work

So far, most end-to-end TTS models have used a variety of attention mechanisms. DeepVoice3 [Ping et al., 2018] and the DC-TTS model [Tachibana et al., 2018] both are convolutional based architectures and borrow heavily from the [Gehring et al., 2017] and [Vaswani et al., 2017] architectures and thus use dot product attention. Tacotron [Wang et al., 2017] and its newer version Tacotron2 [Shen et al., 2018], VoiceLoop [Taigman et al., 2018] and Char2Wav [Sotelo et al., 2017] use location based attention. Tacotron and Tacotron2 uses the hybrid or recurrent additive attention which is location sensitive, while Char2Wav and Voiceloop both use Graves location based GMM attention [Graves, 2013]. Char2Wav has a recurrent encoder and decoder while Tacotron has a composite unit called the CBHG a unit consisting of **C**onvolution, **B**idirectional Layers of **H**ighway and **G**ated Recurrent Units and residual connections, while the Voice loop uses the notion of a buffer with three MLP's to encode attention, output and the buffer update vector. Tacotron 2 has a simplified architecture consisting of vanilla LSTM and convolution layers instead of the CBHG stack.

Thus, given the diversity of model architectures and attention mechanisms, we will concentrate on the DeepVoice3 and its related DC-TTS architecture and use these convolution based architectures to test various attention mechanisms. One common trend however that emerges from the literature on end-to-end sequence models for TTS is that windowing or using more local attention as opposed to using the entire sequence, is one mechanism for enforcing some form of monotonicity and is helpful in making the model converge faster and produce better quality output, in addition to making the training more stable and robust to attention failures on long sequences. In the next section, we will briefly describe the baseline model that we will be using and our proposed extension to this model with multi-hop, mutli-scale, multi-headed attention and error feeding.

6.2 Attention based End-to-End models for Prosody

6.2.1 Model Descriptions

- **DeepVoice3:** The DeepVoice3 [Ping et al., 2018] as described in Section 4.1.2, is a fully convolutional attention based model similar to the model described in [Gehring et al., 2017]. It consists of a fully convolutional encoder and decoder with dilated gated causal convolution layers, while the post-net is non-causal with dilated gated convolutions. It uses a dot-product attention along with position embeddings for text and Mel-spectral frames. The details of number of layers as well as other hyper-parameters can be found in Appendix A.
- **Dilated Convolution TTS Model:** The Dilated Convolution TTS model (DC-TTS) [Tachibana et al., 2018], uses an architecture similar to that of Tacotron [Wang et al., 2017], except that it is a fully convolutional based architecture without the RNN. They use a combination of highway layers and dilated convolution layers. They also do a time-upsampling via the convolution layers in the postnet. They use an extra binary divergence loss along with a guided attention loss. The binary divergence loss is used to stabilize the gradients for very deep networks and the guided attention loss helps make the attention almost monotonic and both of these losses help make the model converge faster.

In Table 6.1 we compare these two models for different text input representations on the Blizzard dataset. We see that the DeepVoice3 model seems to be better than the DC-

Table 6.1: *DTWMCD results on Blizzard dataset using various input text representations on two attention based convolution end-to-end models example wavs..*

Input-Type	Deep Voice	DC-TTS
Character	5.12 ± 2.69	5.29 ± 2.85
Phone	5.12 ± 2.69	5.28 ± 2.84
Phone with word breaks	5.10 ± 2.67	5.27 ± 2.83
Phone with syllable breaks	5.10 ± 2.67	5.28 ± 2.833

TTS model, and the phone with word breaks seems to be the best input representation in terms of DTWMCD. However, if one listens to the sample linked wavefiles, it seems like the models with lower MCD also tend to be the less prosodically rich models and it is not really clear whether a lower MCD actually means a better quality of synthesis in terms of prosody. However, in the absence of any other metric, we will use the model with the lowest MCD, the DeepVoice3 model with word breaks as the baseline for further experiments. In addition, we use the combined loss function used in the DC-TTS paper for faster convergence and is described in the next section.

6.2.2 Guided Attention and Loss Function

In our experiments we found that it was hard to make the model converge fast on long utterances and utterances with diverse prosodic variations. We found that the attention collapses in the middle of training and needed to be restarted each time. Thus, to control errors in attention, and be able to generalize to longer sequences, we use a windowed attention mechanism, that uses a very small window of attention vectors to enforce almost monotonic alignments. Thus in our default model, we use a window length of 4, 1 vector backward and 3 vectors forward, thus ensuring that at each time we get almost monotonic alignments and faster convergence. In addition, in [Tachibana et al., 2018], they found that adding a small amount of binary divergence loss also helped to stabilize gradients, so we apply that to our loss function as well.

Thus, the entire loss function is :

$$\mathcal{L}_{Total} = \mathcal{L}_{Mel} + \mathcal{L}_{Linear} + \mathcal{L}_{Attention}$$

where: $\mathcal{L}_X$, X stands for either Mel or Linear Loss and is defined as:

$$\mathcal{L}_X = \mathcal{L}_{L1} + \mathcal{L}_{Bdiv} + \mathcal{L}_{Done}$$

where, $\mathcal{L}_{L1}$ is the $L1$ loss, $\mathcal{L}_{Bdiv}$ is the binary divergence loss first proposed in [Tachibana et al., 2018] and defined as the cross entropy loss between true Mel spectral features $\mathcal{S}_{ft}$ and predicted Mel spectral features $\mathcal{Y}_{ft}$ and defined as:

$$\mathcal{L}_{Bdiv} = \mathbb{E}_{ft}[-\mathcal{S}_{ft} \log \mathcal{Y}_{ft} - (1 - \mathcal{S}_{ft}) \log(1 - \mathcal{Y}_{ft})]$$

The done loss is just a binary cross entropy loss per predicted frame whether it is the end of the sequence or not. The attention loss is defined as:

$$\mathcal{L}_{Attention} = \mathbb{E}_{nt}[\mathcal{A}_{nt} \odot] \mathcal{W}_{nt}$$

where $\mathcal{A}_{nt}$ is the attention matrix and $\mathcal{W}_{nt}$ are the guided attention weights to constrain it to be almost monotonic and is defined as:

$$\mathcal{W}_{nt} = 1 - \exp\left(- \frac{\left(\frac{n}{N} - \frac{t}{T} \right)^2}{2g^2} \right)$$

where, n is the current text position, N is length of input text sequence, t is the current frame position and T is the length of the Mel spectral sequence, g is a standard deviation and which determines how wide you would like to penalize the attention weights outside the window, and in our case is 0.2.

6.2.3 Multi-scale, Multi-hop, Multi-headed attention (3M)

Since we would like to capture prosody at different scales, a local scale for pitch accents as well as a larger scale for phrase breaks, we wondered if we could do this with

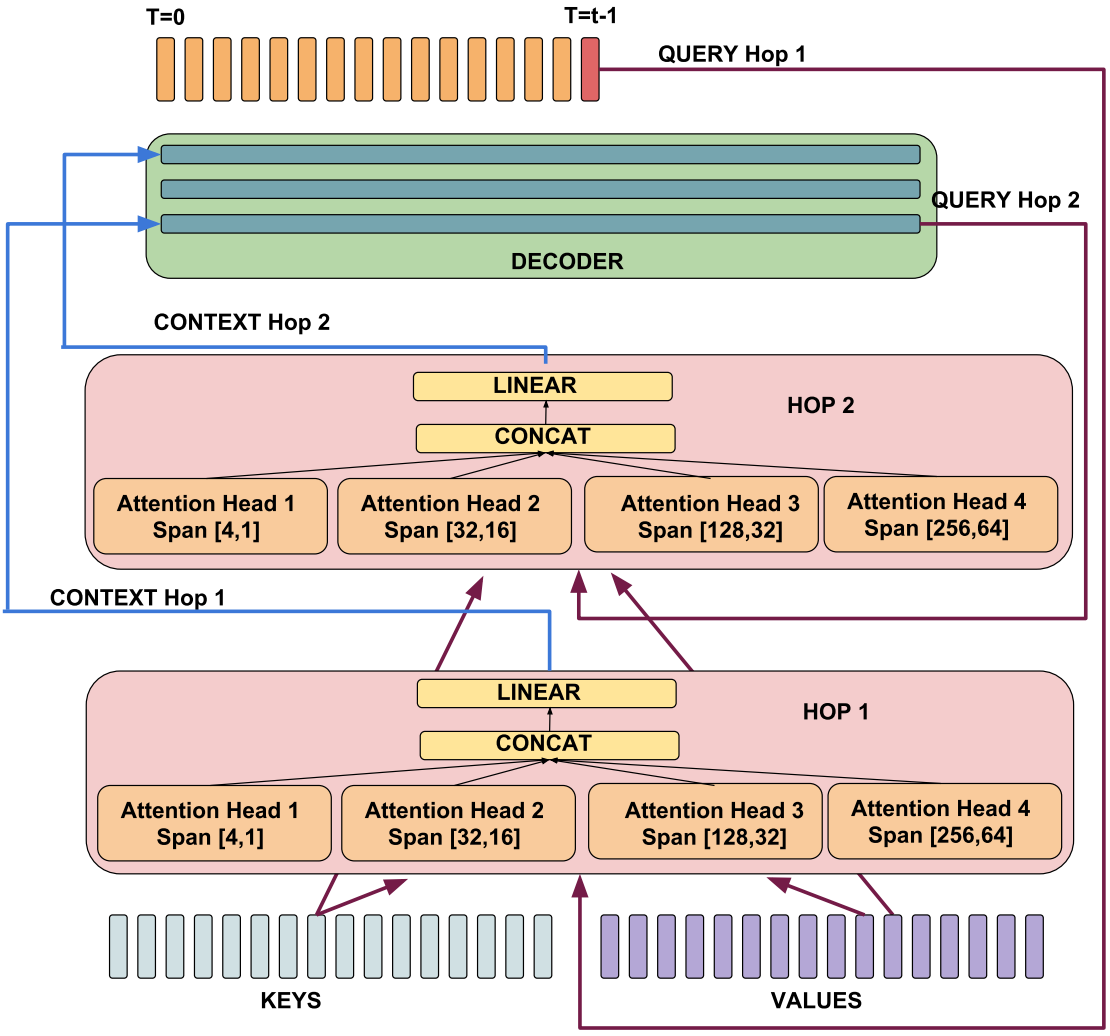


Figure 6.1: *Multi-scale, Multi-hop, Multi-headed Attention, showing a model with 4 heads, 2 hops and scales per head of 5, 48, 160 and 320 respectively.*

multi-headed attention, each attending to different lengths of the input. We call this multi-scale, multi-headed attention. And since we have this multi-scale, multi-headed attention at each decoder layer, with the output of the first decoder layer serving as the query for the next layer, we also have multiple hops. Thus, we call this attention multi-scale, multi-headed, multi-hop attention.

Results: We see that by adding multiple scales we do not see much improvement in prosody as compared to the baseline in terms of DTWMCD. One reason for this maybe that using the guided attention loss makes using multiple scales useless. So we also trained 3M attention models for 2 and 5 hops without guided attention. However, in terms of DTWMCD it still does not seem to make much of a difference. An interesting observation note however from the synthesized wavefiles is that even though the

Table 6.2: *DTWMCD results on Blizzard dataset using Multi-scale, Multi-head, Multi-hop (3M) Attention example wavs.*

No. Hops	No. Heads	Scales of each head [forward, backward]	DTWMCD
1-first-layer	1	$[3, 1]$	5.15 ± 2.72
1-last-layer	1	$[3, 1]$	5.12 ± 2.68
2 (Baseline)	1	$[3, 1]$	5.12 ± 2.70
2	4	$[4, 1][16, 4][32, 8][64, 16]$	5.15 ± 2.72
2	4	$[4, 1][32, 16], [128, 32], [256, 64]$	5.13 ± 2.70
3	4	$[4, 1][16, 4][32, 8][64, 16]$	5.15 ± 2.72
3	4	$[4, 1][32, 16], [128, 32], [256, 64]$	5.16 ± 2.70
5	4	$[4, 1][16, 4][32, 8][64, 16]$	5.14 ± 2.72
5	4	$[4, 1][32, 16], [128, 32], [256, 64]$	5.15 ± 2.71

DTWMCD for single hop, single head attention at first and last layer is comparable, the quality isn't as great. Especially when using attention directly at the last layer, we see that the quality of the synthesized wavefiles is the worst. The same can be said about the synthesized wavefiles with a single hop and multiple scales at the last layer. However, do notice that adding in multiple scales improves the phasiness slightly. Using shorter windows also has a problem in terms of making the model stop at the appropriate time. We see that using 2 hops with multiple scales gives us some improvements in synthesis quality, however, going beyond hops does not give us any benefits.

6.2.4 Prosody Control with Error Feeding

In the previous section we saw that using just two attention heads gives us good results and there are no gains in adding multiple attention heads over many scales. One reason for this might be the fact that the first attention heads ends up learning an almost monotonic attention and the gains with multiple attention heads over multiple other scales has no effect. Thus, in this section we look at another method of introducing *prominence attention* by essentially using the errors at the output of the decoder as an additional feature stream. Our hypothesis is that the errors are larger at the points in the output acoustic sequence where there are maximum deviations in prosody, *i.e.*, the errors are highest where the prosody is most different. Thus, we could think of the error stream as features of how much to importance to give each Mel frame in the output sequence at the decoder in order to improve its prosody. We call this error sequence the *prosody control sequence*, and it is calculated as a mean of the squared errors per frame. During test time, we can feed a sequence of floating point numbers to realize the prosody we want. It seems reasonable since greater deviations in prosody from the mean would cause higher errors if the models are good at predicting the mean statistics.

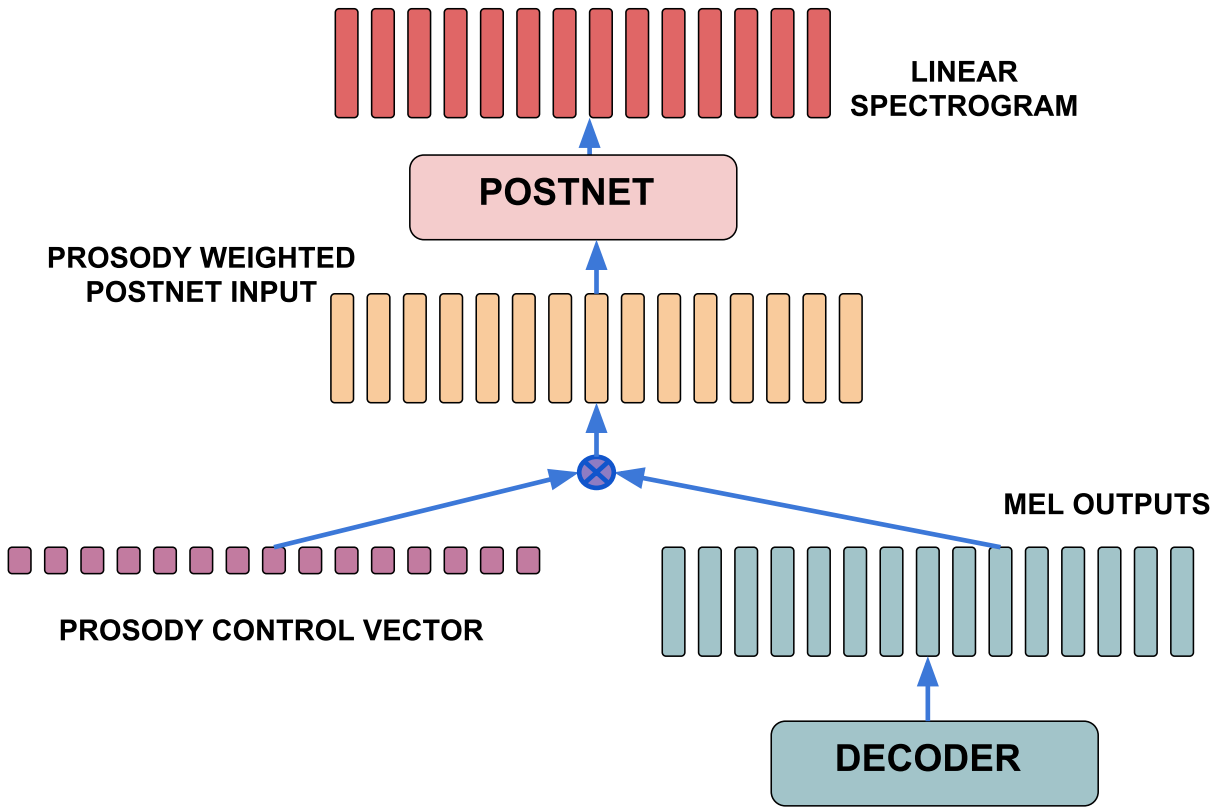


Figure 6.2: *Error Feeding Mechanism.*

Training: We can either give the Mel outputs as the input sequence to the postnet

Table 6.3: *DTWMCD results on Blizzard dataset using Error Feeding example wavs*

Output type	Added Decoder Outputs to Error	Constant Value	DTWMCD
Decoder Output	Yes	0	5.00 ± 2.59
Decoder Output	No	1	5.41 ± 2.92
Decoder State	Yes	1	5.21 ± 2.77
Decoder State	No	0.5	5.15 ± 2.71

or give it the predicted Mel Targets. In addition, to calculate the errors, we can either give the error stream directly or add the decoder targets to it and use those as the prominence weights to weight the input to the postnet. Table 6.3, lists all four combinations of these weighted inputs.

Results: We see that the error feeding mechanism of using the Mel outputs, with decoder outputs added to the error stream gives us a 0.1 DTWMCD improvement. Since during test time we need to feed it some control sequence, we gave it a vector of some constant value as indicated in the results table, in order to be able to calculate DTWMCD. In addition, we also experimented by controlling the prosody by giving it simple control vectors such as a sine wave of a particular frequency, an increasing and decreasing control vector, a square wave, etc, and in most cases, we could perceive differences in volume but not prosody according to the simple manipulations. Further exploration in transferring audio from another audio waveform to our target, by first decoupling F0 seems like a promising future direction to pursue.

6.2.5 Summary

Thus, in this section we explored two methods of making the neural network learn to focus on more prosodically motivated attention weights. In the first method we used a multi-scale, multi-hop, multi-headed attention mechanism, which assumes that the inherent prosodic structure we are trying to model is a hierarchical model of relative prominence. However, we find that the model does not do any better than the baseline which does not use a multi-scale architecture. In the second scenario, we use an error feeding mechanism where we treat the errors in the predicted Mel output sequence as attention weights. We find that this improves results a little and also produces waves with simple manipulations in volume during synthesis. It will be interesting to investigate this further while also adding some sort of additional attention mechanism at the output of the decoder and as input to the postnet.

6.3 Prosody Embeddings for Speech Synthesis

In the previous section we explored one method of capturing the prominence and prosody in an unsupervised fashion within a neural network pipeline via various attention mechanisms. In this section, we explore another method of representing the prosody and explicitly augmenting the model with these utterance-level prosody embeddings. Since it is difficult to label data with prominence values, and there do not exist many large datasets with this labelling, we will instead try to learn this prominence via a metrical structure. The metrical structure as described in 5.3.1 is a binary tree rooted at the phrase and going till the syllables, having each child marked as either weak (W) or strong (S). The goal in this section is to be able to learn this binary tree structure in a bottom-up fashion, starting with syllables in an unsupervised fashion in order to improve the downstream quality of speech synthesis. Furthermore, the trees learned by the prosody embedding might make the model more interpretable, *i.e.*, we might be able to manipulate different intonations by manipulating the metrical structure. In addition, the prosody embeddings can serve to help style transfer and other emotional TTS applications.

This embedding would be in addition to the attention mechanism, and as such would give us utterance level prosody embeddings. Previous work in learning utterance level

prosody embeddings in a neural network setting has been explored in [Skerry-Ryan et al., 2018] and [Watts et al., 2015]. In [Skerry-Ryan et al., 2018] and [Watts et al., 2015], they learn these utterance level prosody embeddings in an unsupervised manner. In [Skerry-Ryan et al., 2018], they learn this as a 1 of K style tokens which are then aggregated through a separate Style Attention and given to the decoder, while in [Watts et al., 2015], they encode each utterance as a 1 of K encoding and then are able to control the style by traversing this 2-D embedding space. With our architecture, we would like to go one step further by assuming a more theoretically grounded method of inducing prosodic structure via the metrical tree. In the next section we will briefly describe the Tree-LSTM structure and then explain our input feature representation.

6.3.1 Model Description

To obtain utterance level prosody embeddings guided by metrical theory, we use Tree-LSTM as an additional prosody encoder and use the embeddings from this model as extra global embeddings at each layer in the decoder. In this section we will briefly describe the Tree-LSTM and features used.

Tree-LSTM Composition Function

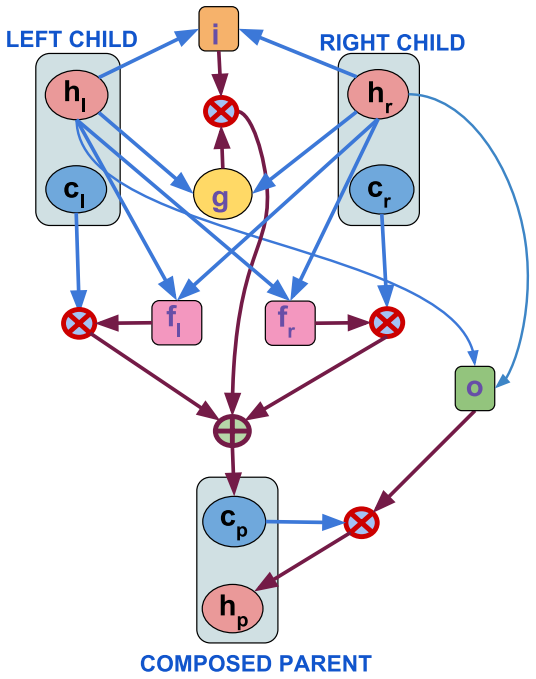


Figure 6.3: *Composition function of the Tree-LSTM illustrating the gating .*

RNN's are good at capturing sequential context. However, given that we want to capture a more hierarchical representation of relative prominence we need a structure

that can not only take into account the sequential context, but also have the ability to take into account hierarchical information and build up a representation by applying a function to some number of child nodes. One such architecture proposed in recent literature is the Tree-LSTM [Tai et al., 2015]. The tree-structure LSTMs incrementally build up a representation of an utterance by combining nodes pairwise at each time-step until only one representation is left. In our implementation of a Metrical Tree representation, we need to build up a binary tree given lexical stress values on syllables, and where the child nodes are ordered. Thus, we use the Binary tree LSTMs a special case of the N-ary Tree-LSTMs proposed in [Tai et al., 2015]. The binary Tree LSTM's composition function is defined as follows:

$$\begin{bmatrix} \mathbf{i} \\ \mathbf{f_l} \\ \mathbf{f_r} \\ \mathbf{o} \\ \mathbf{g} \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \sigma \\ \tanh \end{bmatrix} (\mathbf{W_{comp}} \begin{bmatrix} \mathbf{h_l} \\ \mathbf{h_r} \end{bmatrix} + \mathbf{b_{comp}})$$

$$\mathbf{c}_p = \mathbf{f}_l \odot \mathbf{c}_l + \mathbf{f}_r \odot \mathbf{c}_r + \mathbf{i} \odot \mathbf{g}$$

$$\mathbf{h}_p = \mathbf{o} \odot \tanh \mathbf{c}_p$$

Where, $\odot$ is the element wise product of elements, $\mathbf{c}_p$ and $\mathbf{h}_p$ represent the hidden and cell state of the composed parent. $\mathbf{i}$, $\mathbf{o}$, $\mathbf{g}$ are the LSTM cell input, output and gating functions and $\mathbf{f}_l$ and $\mathbf{f}_r$ and $\mathbf{c}_l$ and $\mathbf{c}_r$ are the forget and cell states of left and right child respectively. This implementation of the Tree-LSTM is similar to the SPINN model described in [Bowman et al., 2016], without the tracking LSTM. We instead use the implementation used in [Choi et al., 2018], with the RNN leaf transformation. Figure, 6.3, illustrates this composition function.

Unsupervised Tree-LSTM Grammar Induction

The SPINN model that was proposed in [Bowman et al., 2016] requires labelled trees for training the network. However, since in our particular case we do not have such training data, we need to look at methods that use this model in an unsupervised fashion. There have been some examples of the SPINN model being used in an unsupervised fashion. Examples include a version of the model proposed by Yogatama *et. al* in [Yogatama et al., 2017], that uses the REINFORCE [Williams, 1992] algorithm to find latent tree structures given only the text and optimizes it for a down stream task of sentiment classification. This work is further extended by Maillard *et.al* in [Maillard et al., 2017], where they first generate $O(N^2)$ nodes for the tree over N words and then select the best node combinations to join as a tree using a gating mechanism. Along similar lines, another unsupervised version of the SPINN model uses a categorical distribution to sample nodes to be composed at each step. Thus, instead of building partial trees, this model uses a categorical distribution to select only one node to be composed at each time-step.

In an analysis of the trees learned from these models, as was done in [Williams et al., 2018], they found that only the categorical distribution model using the ST Gumbel distribution yielded results that were better than a simple Tree RNN baseline and simple LSTM baseline. Moreover they found that the RL-SPINN model mainly yielded left branching trees for sentences greater than 7 words. Thus, for our application we chose to go with the ST-Gumbel model. The ST Gumbel model builds up a tree over N words by selecting one node to compose at each time using a learned scoring function. Thus it goes through $N-1$ compositions to obtain the embedding of the sentence. In this model, the node to be composed at each time-step is sampled using a straight-through Gumbel (ST-Gumbel) estimator to sample from a probability distribution over nodes to be composed. This scoring function is optimized with respect to a downstream task, in our case improving the combined loss as mentioned in Section 6.2.2.

6.3.2 Feature Representation

Input: We explored three input representations annotated with linguistic features derived using the Festival front-end at the syllable level, as well as the character, phone and phone based sequences with breaks at the word and syllable level. However, since the leaves of the metrical tree are syllables, we wanted to find a good representation of syllables that would work with this model. Thus we chose to experiment with three syllable sets. These were as follows:

- **Full-set:** The first we call the full-set which contains all the features dumped at the Syllable level that are annotated by Festival. A list of these features can be found in Appendix B.
- **Small-set:**The second set is a reduced subset of these features, which have been used previously in [Ronanki et al., 2014]. These included the lexical stress, accent, word initial or final, phrase initial or final as well as onset, coda and part-of-speech tag.
- **Tiny-set:** The final subset is a reduced subset without the part-of-speech tag as used in [Anumanchipalli et al., 2013], but including coda and onset features.

6.3.3 Results

Th results with these three feature sets as well as character and phone sequences is shown in Table, 6.4. We see that adding in an additional prosody embeddings guided by metrical structure does not make very much difference in the output prosody in terms of the DTWMCD measure.When we listen to the wavefiles, if you take for example the last wavefile which is the sentence, *"So, the elephants go on a log journey to find food and water"*. In this sentence, there are is small pause after "So" and the long is elongated. We see in most cases it gets the first pause right, while it is not able to get the elongation on long very well. The best one seems to be the one with syllable breaks, which seems to get the prosody right on all three sentences, however it ends up sounding a little phasey for the last sentence. From these results it seems that trying to improve the prosody further

Table 6.4: *DTWMCD results on Blizzard dataset using Tree-LSTM derived prosody embeddings example wavs*

Input-Type	DTWMCD
Char-Phone (Baseline)	5.12 ± 2.69
Character	5.13 ± 2.70
Phone	5.14 ± 2.71
Phone with word breaks	5.12 ± 2.70
Phone with syllable breaks	5.11 ± 2.70
Syllable Tiny-Set	5.11 ± 2.70
Syllable Small-Set	5.14 ± 2.72
Syllable Full-Set	5.12 ± 2.70

using the tree-lstm encodings does not yield much benefit. In the future we would like to try using only the Tree-LSTM as a sole encoder and use the prosody embeddings on an utterance directly for synthesis to see if it actually captures anything relevant. In addition, we would like to investigate adding an additional F0 multi-task loss to the Tree-LSTM encoder to guide it better and optimize it specifically for learning a better F0 contour.

Analysis

Fig. 6.4, plots a 2-dimensional, multi-dimensional scaling visualization for the tree embeddings learned using different inputs. Here we have shown all quoted utterances in the test set as blue and the narrated utterances as red. It is interesting to see that the first row, which has syllables or some knowledge of syllable units as input, provides some distinction between narrated sentences (shown in red) *vs.* quoted utterances, shown in blue. The first two figures, are using syllable based features, and even though they have overlapping clusters, one can see a clustering of blue and red points, showing us that the the tree-LSTM does learn some notion of prosody. The sub-figure (c) has phone inputs demarcated by syllable breaks, and it also shows some kind of clustering between quoted and narrated texts. While, the last three plots (d)-(f), show totally overlapping clusters. Thus, it seems like the prosody embeddings learned by the Tree-LSTM, differ depending on the input and seem to learn some notion of syllables with respect to the prosody of the utterances, since quoted utterances which are various character voices are prosodically different from the the narrated utterances.

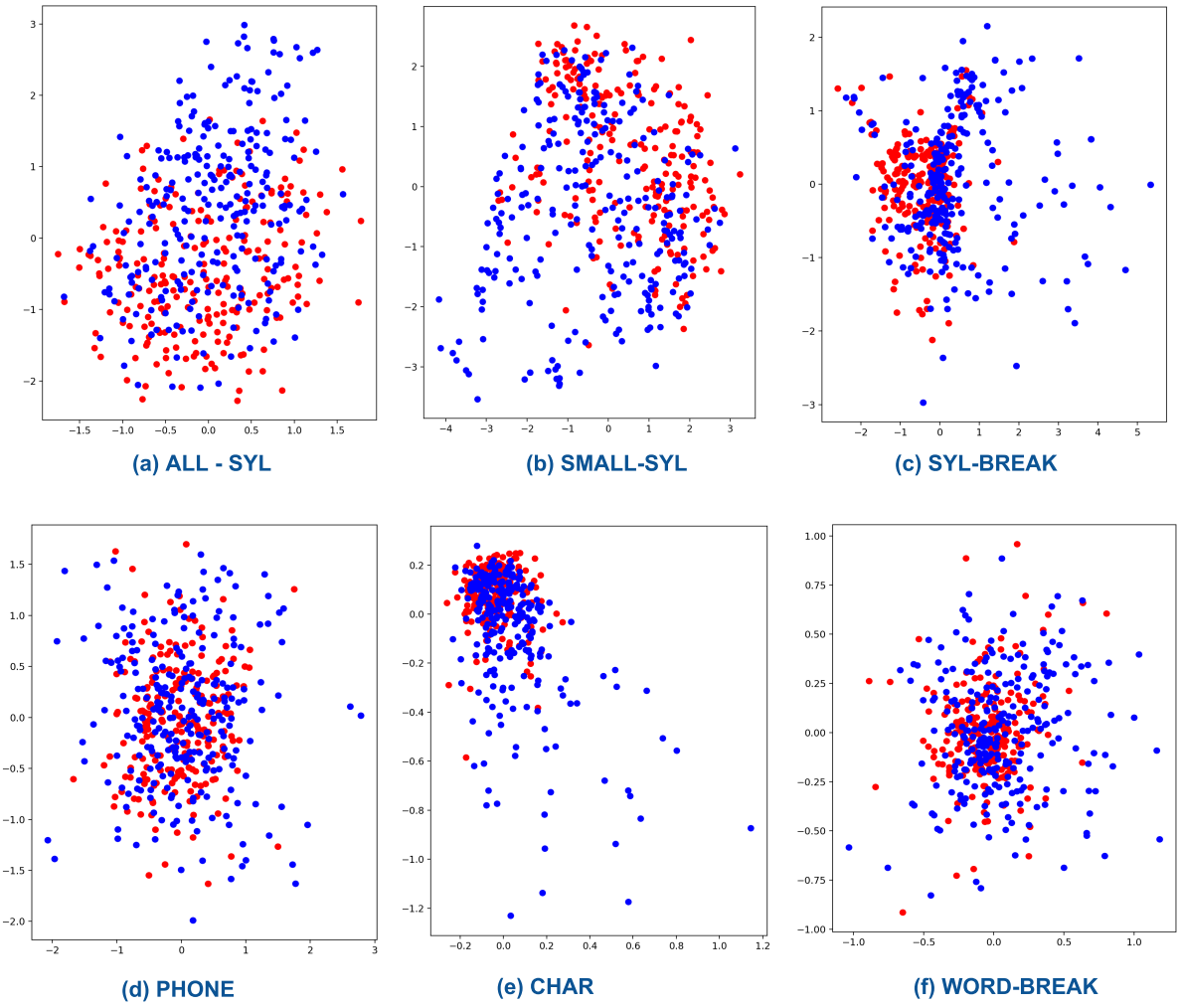


Figure 6.4: *Visualization of Prosody Embeddings learned with different Inputs.*

6.4 Summary and Discussion

In this chapter we looked at various ways of inducing more knowledge of prosody into the end-to end neural systems. First, we looked at two methods of learning a hierarchical set of *prominence weights*, one using multi-scale , multi-hop, multi-head attention and the other using error feeding. We find that even though DTWMCD does not capture the subtleties in the delivery of the utterances, on listening to the examples of 3M attention, we find that adding multiple scales improves performs slightly over a single head. In addition, we find that when using only a single hop, it is better to give it to the first layer, rather than the last layer, since the output from the last layer for both single and multiple heads sounds noisier and less smooth. In case of the error feeding mechanism, we find it gives us some benefit in terms of manipulating the output wavefile and is worth investigating further, especially in transferring acoustics from another sentence on a target sentence. We also look at learning utterance level embeddings of prosody, guided by metrical structure. We show the even though these embeddings capture some meaningful prosodic information for syllable level inputs, in terms of quoted *vs.* unquoted utterances, they do not improve DTWMCD. In the future we would like to use these learned prosody embedding for style transfer across utterances.

Chapter 7

Summary and Conclusions

7.1 Summary

In this thesis we have taken the first holistic approach to building speech synthesis systems from Found Data, with a goal of building such systems for low resource languages.

We approached this problem from three different aspects:

- **Data Selection:** Selecting good data for synthesis
- **Data Augmentation:** Integrating external resources to augment training data.
- **Models for Long form audio:** Building models for data having long form prosody structure.

In the next section we briefly describe our contributions for each of these three aspects and lay out future promising directions to pursue.

7.2 Contributions of this Thesis

7.2.1 Data selection

In this thesis we explored data selection strategies for found data with a variety of speaking styles. We showed experiments with an iterative strategy of selecting the best subset for synthesis, given a noisy corpus of data. For selection, we basically looked at two criteria:

1. **Selection of a seed dataset:** We looked at both only acoustic measures such as MCD to select a seed dataset, or only text based measures such as phone frequency to select a seed dataset. We found that optimizing for phone frequency gave us a slight advantage in improving the MCD.
2. **Selection of a suitable ranking metric to select good utterances:** We explored various metrics spanning spectral, text, correlation and duration based measures to rank data based on its suitability for building better speech synthesis systems. We showed results on an artificially degraded dataset containing mis-alignment errors

and channel noise. We found MCD, followed by duration to be the best metrics for ranking utterances, based on their suitability for synthesis.

In terms of model building, we also looked at the effect of re-clustering *vs.* re-alignment and found that only iteratively re-clustering, using noisy labels obtained by force-aligning the entire dataset was faster and gave us good enough results. We show these results using MCD as the metric on three corpora, one an artificially degraded corpus of clean speech, as well as two corpora of found data, one a single speaker corpus of public speeches and the other a multi-speaker corpus of telephone conversations.

Our main finding was that the MCD and duration calculated at the utterance level are good at finding suitable utterances for synthesis. We also showed that we can generally improve performance on noisy datasets by getting rid of the worst 10% of the data.

7.2.2 Data Augmentation

We explored data augmentation techniques for both transcribed and un-transcribed data.

- **Transcribed Data:** For the case of transcribed data we explored multi-lingual as well as multi-style training of end-to-end neural models.
  - (A) **Multi-Lingual models:** We explored attention based multi-speaker, multi-lingual systems for speech synthesis across many Indic languages. For these multilingual models we were interested in exploring mainly the following four criteria:
    - (i) **Factored Embeddings:** We compared using factored embeddings, where the global attributes such as gender, language and speaker where used as separate embeddings to the model, so as to enable it to share similar data across languages and speakers *vs.* an unfactored model, where each language-speaker pair is given a unique ID. From our results we find that factoring embeddings across global attributes does not improve performance and found that using a single model or an unfactored model is in fact better. We also find that there is not much difference between the performance of factored model *vs.* an unfactored model if there is a single speaker of a single language, further strengthening our case that training the model as an unfactored model allows it to over-fit to speaker and language characteristics better, which is advantageous to our case of speech synthesis.
    - (ii) **Scalability and Effect of External Speakers:** We also looked at how these models scale on a new language and speaker and what kind of external data helps. First, we show that by scaling it to many more languages and speakers, the performance generally improves both in terms of objective and subjective listening scores. In terms of how external data effects the model, we showed that adding multi-lingual external data helps improve synthesis results as opposed to starting from a model trained with English speech and these gains are prominent for smaller subsets of 100 and 250 utterances as shown using Gujarati female data on different subsets. We

also find significant gains in fine-tuning the model on the target data and show that augmenting with similar languages, helps, however, we find that adding external speakers of the same target language does not make any difference in the performance.

(iii) **Transferrability:** In an analysis of what these embeddings capture, we showed that we can generally transfer speaker characteristics across languages. However, we find that the embeddings for language and gender cannot be easily factored from the speaker and as such are not robustly transferable.

(iv) **Robustness to Noise:** We also show experiments with two multi-lingual corpora of noisy data. For the Babel Bengali data, which was originally collected for the purposes of keyword spotting, we found that neural unfactored attention models adapted from multi-lingual data improve significantly upon results obtained from CLUSTERGEN. Even though, one might not be able to use these models for high-quality synthesis, using these models for synthesis of short phrases as in for communication in a disaster scenario or using it for keyword recognition from synthesized utterances is possible. Using the synthesized speech for keyword recognition as opposed to the raw speech, might give more consistent results and is worth exploring further.

With respect to the much cleaner multi-speaker ASR Gujarati corpus, we find very interesting results. We show that even in the absence of speaker labels, the model ends up learning super-sentential context, and is reliably able to sample the same speaker when giving it in domain held-out test set. On an out of domain held-out test set, it samples random speakers, and maintains the speaker characteristics consistently over the utterance. Thus, the model learns to model global utterance level attributes even without speaker ID information, unlike the CLUSTERGEN model, which generates an average speaker voice, when trained with a multi-speaker dataset in single-speaker mode.

(B) **Multi-Style models**

(i) **Domain Adaptation:** For multi-speaker English data we were interested to see the effect of changing the composition of the training data so as to make it closer to the domain of the target speaker. We explore this in terms of accent, gender and pitch and found that changing the composition of the training data had little effect on the performance as measured by DTWMCD, while fine-tuning the data on the target speaker alone yielded performance improvements.

(ii) **Style Transfer:** We also explored multi-style training of attention based neural models on found data spanning various noise levels and speaking styles. We found that the style embedding ends up picking up channel characteristics as style. Especially in the case of noisy datasets such

as CallHome and TED corpora. We find that it picks up characteristics of telephone line for CallHome and reverberation in case of TED corpus instead of the actual speaking style. Thus transferring style from clean speech to other speaking styles does not really help transform it to sound more conversational and rather makes it noisy and not very understandable. On the other hand, transferring style from noisy speech to clean style improves intelligibility, and this is especially true when transferring from very noisy corpora such as CallHome. We see that in most cases, it is possible to synthesize data even with very noisy speech such as CallHome and from as little as 7 minutes of very noisy speech per speaker. In addition, in comparison to CLUSTERGEN based models, these models are significantly better in terms of quality, as shown by the listening tests as well as DTWMCD metrics.

- **Un-transcribed Data:** In the second part of Chapter 4, we consider a zero-resource setting, wherein we only have audio data and no other language resources. In this scenario, we show results using cross-lingual adaptation techniques in finding a good phonetic inventory to be able to build a decoder and obtain a transcript with which one can do synthesis. We consider two zero-resource scenario, one a clean multi-speaker setting with a Hindi corpus, and the other a noisy, limited data setting with Xitsonga. We find that our method of using cross-lingual phonetic decoding outperforms phonetic decoding for clean speech, however, it does not do very well on the very noisy corpus of Xitsonga.

7.2.3 Prosody Models for Long-form Audio

In this thesis, we also explored methods to better model the prosody using traditional frame-based models as well as neural attention models.

1. **Frame-based Prosody Models for Audiobook TTS:**
    - **RNN Models for Long-form Audio:** We explored various RNN based models for modelling long-form audio. Specifically we explored quasi-RNN, recurrent highway networks and clockwork RNNs, in an attempt to improve long-form prosody modelling. We found the recurrent highway network to be the best among all of the RNN based models we explored. From subjective listening tests we find that these models are good at intelligibility, however, they still fall short on naturalness.
    - **Iterative Prosody Labelling Strategy Inspired by Metrical Phonology.** In this thesis we also looked at improving the labelling strategy of stress and break values, with the goal of learning a Metrical Grammar and using the metrical structure as additional features in synthesis. Since, we do not have training data we explored an iterative method of learning a metrical grammar and using its features for synthesis. For these iterative methods, we explored various initialization and scoring functions. For the initialization methods, we found prosodic breaks and stress values predicted using duration, F0 and

acoustics to perform the best and it gave the lowest RMSE on all three audiobook datasets. For the scoring function, we explored various metrics spanning spectral, duration and F0 based measures and found that duration was the best. In addition, we found that optimizing prosody models with MCD is not the best strategy and we need to find alternate metrics that actually capture time-varying *"relative prominence"* at different scales in some way.

2. **Neural Attention models for Prosody Modelling**
   - **Prominence weights:** In this thesis, we looked at two methods of inducing a sense of *"prominence weights"*. One was the **multi-scale, multi-head, multi-hop (3M) attention**, and another was an **error feeding mechanism**. For the 3M attention, we find some improvements in increasing number of scales from 1 to 4 when using only 1 hop. In addition we find that if we directly use attention only on the last decoder layer, the performance is worse as compared to when we use it on the first decoder layer, with and without multiple scales. Increasing the hops from 1 to 5, we see that after 2 hops, we do not get much gain in improvement in quality of synthesis. In addition, we see that quality is slightly better without guided attention, when using 3M attention, however it is not significantly different.

     The error feeding mechanism, using decoder outputs gave promising initial results with a 0.1 DTWMCD improvement. We also found that using simple manipulations such as feeding a sine wave of certain frequency or a square wave, we could only control the volume of the final rendition of the waveform produced, but not control its prosody. However, we believe this is a promising direction for future research to allow one to control and manipulate the prosody during synthesis in end-to-end neural models.
   - **Prosody Embeddings:** In this thesis, we also explored one unsupervised method of learning prosody embeddings in an end-to-end neural framework guided by metrical theory, using Tree-LSTMs. We show that even though these prosody embeddings learn relevant prosodic information, in terms of clustering character voices *vs.* narration mode, for syllable level inputs, we see that they still do not improve the synthesis performance as measured by the DTWMCD. However, given that these prosody embeddings capture character voices, it would be interesting to see how they perform when used for style transfer as well as prosody control vectors.

7.3 Future Directions

- **Semi-supervised and generative models for multi-lingual synthesis:** In this thesis, we showed that it is possible to build multi-speaker, multi-lingual models with neural attention based models for synthesis. However, we found that the model gets confused when we have two very different languages from the same speaker, as was the case with English and Marathi. We believe that adding in a generative

decoder output layer might help mitigate this by learning a multi-modal distribution over acoustics seems like a promising next step. Thus, in the future we would like to use a generative decoder output layer to share data across speakers and languages. Exploring variational and other normalizing flow based models to learn a distribution over similar acoustics.

- **Joint Training of ASR and TTS Models:** One issue that we did not look at in this thesis is the degradation in end-to-end models with using ASR transcripts in cases of found data where only audio is present. In addition along similar lines as work done in [Tjandra et al., 2017] and [Taigman et al., 2018], we would like to explore methods where we can integrate the ASR into the end-to-end TTS model so as to train the ASR with the TTS objective of MCD.
- **Code-switching using end-to-end pipeline:** We find that the model shows promising results with code-switching. However, the code-switched examples that we have shown were in the same transcript as the target language. It would be interesting to explore training end-to-end speech synthesis models with code-mixed text as input, jointly optimizing the grapheme-to-phoneme (g2p) along with the TTS model.
- **Style Transfer and Speaker Characteristics:** In this thesis we found that the model does well on some speakers and not so well on others, even when using the same training utterances and recording conditions. This begs the question of whether there are certain speaker characteristics or speaking styles that is particularly suited for modelling with these models, since speakers like *RMS* and *AEW*, which perform well on traditional CLUSTERGEN based models, do not seem to perform as well with these models, while speakers like *AWB* and *LJM* sound much better than their CLUSTERGEN voices. Thus, exploring what speaking style and speaker characteristics are suitable for attention based neural models would be an interesting future direction to explore.
- **Prosody Control in Attention based Models:** We saw that using the error feeding strategy gave us some promising results in terms of controlling prosody in these attention based models. In the future, we would like to look at better ways of transferring style across utterances, and various other attention mechanisms that can be used specifically to learn a prosody control vectors from other utterances.
- **Prosody Style Transfer and prosody embeddings:** The Tree-LSTM derived prosody embeddings seem to learn some notion of prosody, as evidenced by their clustering of quoted *vs.* narrated text, for syllable level inputs. However, these embeddings do not improve downstream synthesis. Thus, it might be interesting to use these for further experiments with prosody control and prosody style transfer in attention based models. In addition, with respect to improving prosody of synthesized utterances, in the future we would like to take a look at better methods of capturing prosody guided by F0 and duration objective metrics. Specifically, we would like to explore hierarchical F0 and duration based generative models for prosody modelling.

- **Node Embeddings for Metrical Tree:** In this thesis, we only explored *"twiddling"* the phrase breaks and the stress values on syllables. In the future we would like to look at twiddling the inner structure at intermediate nodes and also look into methods that can better take into account the tree structure rather than just one sentence level encoding.
- **Better Metric for Prosody:** One main issue that we had in training prosody models is the lack of a clear objective that can give us indication of how well we are doing on prosody. Because rich prosody has large deviations in durations, F0 and acoustics, optimizing a mean statistic does not translate well to a good prosody metric. Looking at alternate human-in-loop or reinforcement learning techniques might be other future directions that are worth looking into.

This thesis has barely scratched the surface of using found data for speech synthesis. We have shown that it is possible to synthesize speech with about 10 mins of very noisy data. However, we find that such systems trained with this data also learn to mimic the noisy channel characteristics. What we would ideally like for these models to do is to be able to factorize the channel noise out allowing us to only use the clean speech parts. Some promising results in this direction have recently been shown with the Global Style Tokens [Wang et al., 2018], which have shown some promising results in being able to factorize out channel noise, speaker characteristics as well as the style from speech for found data involving TED talks and artificially degraded data. However, to be able to build a high quality speech synthesis system from found data what we would really need is to be able to mimic human language learning, where given few instances of audio in an unknown language one can build a limited domain speech synthesis system, which might be useful in a disaster relief scenario. What we would like is a system which when given few instances of noisy speech in an unknown language, can imitate those phrases as well as use similar words in a related domain while ignoring the channel and noise present in the data, to be able to synthesize high quality speech. To be able to do this we need to understand the processes involved in human learning and imitation of language and our ability as humans to reproduce and generalize sounds from only a few examples.

Appendix A

Neural Network Model Details

A.1 DeepVoice3 Model

The model architecture is described in detail in the paper [Ping et al., 2018]. For this model, these were the hyper-parameters we used:

- **General:**
  - global embedding dim: 16
  - freq hop size: 256
  - sample rate: 22050
  - downsample size: 4
  - dropout: 0.05
  - batch size: 16
  - optimizer: adam(beta1:0.5, beta2:0.9, eps:1e-06)
  - initial learning rate: 0.0005
- **Encoder:**
  - text embed dim : 256
  - encoder layers: 10
  - encoder channels: 512
  - dilation factors: 1, 3, 9, 27, 1, 3, 9, 27, 1, 3
  - kernel size: 3
- **Attention:**
  - attention window: (forward: 3, backward:1)
  - guided attention std deviation: 0.2
- **Decoder:**
  - mel dim: 80
  - pre-attention layers: 2

    - pre-attention dilation factors: 1, 3
    - decoder layers: 5
    - decoder channels: 256
    - dilation factors: 1, 3, 9, 27, 1
    - kernel size: 3
    - attention hops: 2, first and last decoder layer
- **Postnet:**
    - fft size: 1024
    - time-upsampling: 4
    - postnet channels: 256
    - dilation factors per block: 1, 3
    - kernel sizes per block: 1, 3

A.2 DC-TTS Model

The model architecture is described in detail in the paper [Tachibana et al., 2018]. For this model, these were the hyper-parameters we used:

- **General:**
    - global embedding dim: 16
    - freq hop size: 256
    - sample rate: 22050
    - downsample size: 4
    - dropout: 0.05
    - batch size: 16
    - optimizer: adam(beta1:0.5, beta2:0.9, eps:1e-06)
    - initial learning rate: 0.0005
- **Encoder:**
    - text embed dim : 128
    - encoder channels: 256
    - kernel size: 3
- **Attention:**
    - attention window: (forward: 3, backward:1)
    - guided attention std deviation: 0.2
- **Decoder:**
    - mel dim: 80

    - decoder channels: 256
    - kernel size: 3
- **Postnet:**
    - fft size: 1024
    - time-upsampling: 4
    - postnet channels: 256

Appendix B

Syllable Feature Sets

B.1 All-Set

All Features which are part of Festival till the Syllable level.

1. R:SylStructure.parent.R:Word.p.gpos
2. R:SylStructure.parent.R:Word.gpos
3. R:SylStructure.parent.R:Word.n.gpos
4. R:SylStructure.parent.R:Word.content_words_out
5. R:SylStructure.parent.R:Word.content_words_in
6. R:SylStructure.parent.R:Word.pbreak
7. R:SylStructure.parent.R:Word.blevel
8. R:SylStructure.parent.R:Word.word_break
9. R:SylStructure.parent.R:Word.word_duration
10. R:SylStructure.parent.R:Word.word_numsyls
11. R:SylStructure.parent.R:Word.word_start
12. R:SylStructure.parent.R:Word.word_end
13. p.accented
14. accented
15. n.accented
16. pp.stress
17. p.stress
18. stress
19. n.stress
20. nn.stress
21. asyl_in
22. asyl_out

23. lisp_get_rhyme_length
24. syllable_start
25. syllable_duration
26. syl_vowel_start
27. syl_startpitch
28. syl_pc_unvox
29. syl_out
30. syl_onsetsize
31. syl_onset_type
32. syl_numphones
33. syl_midpitch
34. syl_in
35. syl_endpitch
36. syl_codasize
37. syl_coda_type
38. syl_break
39. sub_phrases
40. ssyl_in
41. ssyl_out
42. position_type
43. pos_in_word
44. syl_vowel
45. lisp_cg_break
46. n.lisp_cg_break
47. n.n.lisp_cg_break
48. p.p.lisp_cg_break
49. p.lisp_cg_break
50. R:SylStructure.lisp_length_to_last_seg
51. R:SylStructure.parent.R:Phrase.parent.lisp_cg_find_phrase_number
52. R:SylStructure.parent.R:Phrase.parent.p.lisp_num_syls_in_phrase
53. R:SylStructure.parent.R:Phrase.parent.lisp_num_syls_in_phrase
54. R:SylStructure.parent.R:Phrase.parent.n.lisp_num_syls_in_phrase
55. R:SylStructure.daughter1.R:Segment.p.lisp_is_pau
56. R:SylStructure.daughtern.R:Segment.n.lisp_is_pau

B.2 Small-Set

All features including accentedness, stress, break level coda, onset, part-of-speech tag.

1. R:SylStructure.parent.R:Word.gpos
2. R:SylStructure.parent.R:Word.word_numsyls
3. R:SylStructure.parent.R:Word.pbreak
4. R:SylStructure.parent.R:Word.blevel
5. p.accented
6. accented
7. n.accented
8. pp.stress
9. p.stress
10. stress
11. n.stress
12. nn.stress
13. syllable_duration
14. syl_onsetsize
15. syl_onset_type
16. syl_codasize
17. syl_coda_type
18. syl_numphones
19. syl_in
20. syl_out
21. p.syl_break
22. syl_break
23. n.syl_break
24. p.syl_accent
25. syl_accent
26. n.syl_accent
27. sub_phrases
28. ssyl_in
29. ssyl_out
30. position_type
31. pos_in_word
32. syl_vowel

33. syl_pc_unvox
34. lisp_cg_break
35. n.lisp_cg_break
36. n.n.lisp_cg_break
37. p.p.lisp_cg_break
38. p.lisp_cg_break
39. R:SylStructure.daughter1.R:Segment.p.lisp_is_pau
40. R:SylStructure.daughtern.R:Segment.n.lisp_is_pau

B.3 Tiny-Set

Features include only stress, accentedness, coda and onset features.

1. p.accented
2. accented
3. n.accented
4. pp.stress
5. p.stress
6. stress
7. n.stress
8. nn.stress
9. syllable_duration
10. syl_onsetsize
11. syl_onset_type
12. syl_codasize
13. syl_coda_type
14. syl_numphones
15. syl_break
16. position_type
17. syl_vowel
18. syl_pc_unvox
19. R:SylStructure.daughter1.R:Segment.p.lisp_is_pau
20. R:SylStructure.daughtern.R:Segment.n.lisp_is_pau

Bibliography

Anumanchipalli, G. K. (2013). *Intra-Lingual and Cross-Lingual Prosody Modelling*. PhD thesis, Carnegie Mellon University, USA. 2.1.1, 5.3

Anumanchipalli, G. K., Oliveira, L. C., and Black, A. W. (2011). A statistical phrase/accent model for intonation modeling. In *INTERSPEECH 2011*. ISCA. 5.3

Anumanchipalli, G. K., Oliveira, L. C., and Black, A. W. (2013). Accent group modeling for improved prosody in statistical parametric speech synthesis. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2013*, pages 6890–6894. IEEE. 5.3, 6.3.2

Arik, S. O., Chrzanowski, M., Coates, A., Diamos, G., Gibiansky, A., Kang, Y., Li, X., Miller, J., Raiman, J., Sengupta, S., et al. (2017). Deep Voice: Real-time neural text-to-speech. *ICML 2017*. 6

Baby, A., Thomas, A. L., L, N. N., and Consortium, T. (2016). Resources for Indian languages. *CBBLR Workshop, International Conference on Text, Speech, and Dialogue - 2016*, 978-80-263-1084-6:37–43. 4.1.3

Badino, L., Canevari, C., Fadiga, L., and Metta, G. (2014). An auto-encoder based approach to unsupervised learning of subword units. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014*, pages 7634–7638. IEEE. 4.3.1

Badino, L., Clark, R. A., and Wester, M. (2012). Towards hierarchical prosodic prominence generation in TTS synthesis. In *INTERSPEECH 2012*. 5.3

Bahdanau, D., Cho, K., and Bengio, Y. (2014). Neural machine translation by jointly learning to align and translate. *ICLR 2015*. 6.1.1

Barnard, E., Davel, M. H., van Heerden, C., de Wet, F., and Badenhorst, J. (2014). The NCHLT speech corpus of the South African languages. *Proc. SLTU*, pages 194–200. 1

Bazaj, S. (2017). Festvox Indic Voices provided by Hear2Read. http://festvox.org/cmu_indic/. 4.1.3

Black, A. W. (2006). CLUSTERGEN: a statistical parametric synthesizer using trajectory modeling. In *INTERSPEECH 2006*. 2.1.1, 3.5.2, 4.3.2

Black, A. W., Bunnell, H. T., Dou, Y., Muthukumar, P. K., Metze, F., Perry, D., Polzehl, T., Prahallad, K., Steidl, S., and Vaughn, C. (2012). Articulatory features for expressive speech synthesis. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2012*, pages 4005–4008. 4.3

Black, A. W. and Kominek, J. (2009). Optimizing segment label boundaries for statistical speech synthesis. In *IEEE International Conference on Acoustics, Speech and Signal Processing, (ICASSP) 2009.*, pages 3785–3788. IEEE. 3.5.2

Black, A. W., Lenzo, K., and Pagel, V. (1998). Issues in building general letter to sound rules. *Third ESCA Workshop in Speech Synthesis*. 2.1.1

Black, A. W. and Lenzo, K. A. (2001). Optimal data selection for unit selection synthesis. In *4th ISCA Tutorial and Research Workshop (ITRW) on Speech Synthesis*. 1.2, 3.2

Black, A. W. and Lenzo, K. A. (2003). Building synthetic voices. *Language Technologies Institute, Carnegie Mellon University and Cepstral LLC*. 4.3.2

Black, A. W. and Muthukumar, P. K. (2015). Random forests for statistical speech synthesis. In *INTERSPEECH 2015*. 3.5.2

Black, A. W., Taylor, P., and Caley, R. (25th December 2014). The Festival speech synthesis system, system documentation, edition 2.4, for Festival Version 2.4.0. 2.1.1, 5.1, 5.2.1, 5.3.3

Black, A. W. and Tokuda, K. (2005). The Blizzard Challenge-2005: Evaluating corpus-based speech synthesis on common datasets. In *Ninth European Conference on Speech Communication and Technology*. 4.3.2, 5.2.1

Bollepalli, B., Black, A. W., and Prahallad, K. (2012). Modelling a noisy-channel for voice conversion using articulatory features. In *INTERSPEECH 2012*. 4.3

Bowman, S. R., Gauthier, J., Rastogi, A., Gupta, R., Manning, C. D., and Potts, C. (2016). A fast unified model for parsing and sentence understanding. *ACL 2016*. 6.3.1, 6.3.1

Bradbury, J., Merity, S., Xiong, C., and Socher, R. (2017). Quasi-recurrent neural networks. *ICLR 2017*. 5.2.2

Braunschweiler, N. and Buchholz, S. (2011). Automatic sentence selection from speech corpora including diverse speech for improved HMM-TTS synthesis quality. In *INTERSPEECH 2011*. 1.2

Breiman, L., Friedman, J., Stone, C. J., and Olshen, R. A. (1984). *Classification and Regression Trees, The Wadsworth Statistics and Probability Series, Wadsworth International Group, Belmont California (pp. 356)*. CRC press. 2.1.1

Caines, A., Bentz, C., Graham, C., Polzehl, T., and Buttery, P. (2015). Crowdsourcing a multi-lingual speech corpus: recording, transcription, and natural language processing. In *Language Resources and Evaluation (LREC) 2015*. 1

Chan, W., Jaitly, N., Le, Q., and Vinyals, O. (2016). Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2016*, pages 4960–4964. IEEE. 6.1.1

Chen, C.-P., Huang, Y.-C., Wu, C.-H., and Lee, K.-D. (2014). Polyglot speech synthesis based on cross-lingual frame selection using auditory and articulatory features. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 22(10):1558–1570. 4.1.1

Choi, J., Yoo, K. M., and Lee, S.-g. (2018). Unsupervised learning of task-specific tree structures with Tree-LSTMs. *Thirty-Second AAAI Conference on Artificial Intelligence (AAAI18)*. 6.3.1

Chollet, F. (2015). Keras. https://github.com/fchollet/keras. 5.2.5

Chung, J., Ahn, S., and Bengio, Y. (2017). Hierarchical multiscale recurrent neural networks. *ICLR 2017*. 5.2.3

Cooper, E., Chang, A., Levitan, Y., and Hirschberg, J. (2016a). Data selection and adaptation for naturalness in HMM-based speech synthesis. *INTERSPEECH 2016*, pages 357–361. 3.2

Cooper, E., Levitan, Y., and Hirschberg, J. (2016b). Data selection for naturalness in HMM-based speech synthesis. In *Speech Prosody*. INTERSPEECH 2016. 1.2, 3.2

De Vries, N. J., Davel, M. H., Badenhorst, J., Basson, W. D., De Wet, F., Barnard, E., and De Waal, A. (2014). A smartphone-based ASR data collection tool for under-resourced languages. *Speech communication*, 56:119–131. 4.3.2

Desai, S., Black, A. W., Yegnanarayana, B., and Prahallad, K. (2010). Spectral mapping using artificial neural networks for voice conversion. *IEEE Transactions on Audio, Speech, and Language Processing*, 18(5):954–964. 4.2.1

Domınguez, M. and Wanner, L. (2016). An automatic prosody tagger for spontaneous speech. *COLING*. (document), 5.3, 5.3.2, 5.3.4

Fan, Y., Qian, Y., Soong, F. K., and He, L. (2015). Multi-speaker modeling and speaker adaptation for DNN-based TTS synthesis. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2015*, pages 4475–4479. IEEE. 1.2

Fraga-Silva, T., Laurent, A., Gauvain, J.-L., Lamel, L., Le, V.-B., and Messaoudi, A. (2015). Improving data selection for low-resource STT and KWS. In *2015 IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 153–159. IEEE. 3.2

François, H. and Boëffard, O. (2001). Design of an optimal continuous speech database for text-to-speech synthesis considered as a set covering problem. In *INTERSPEECH*, pages 829–832. 3.2

Gales, M. J. (1998). Maximum likelihood linear transformations for HMM-based speech recognition. *Computer speech & language*, 12(2):75–98. 2.1.2

Gales, M. J. (2000). Cluster adaptive training of hidden Markov models. *IEEE Transactions on Speech and Audio Processing*, 8(4):417–428. 2.1.2

Gehring, J., Auli, M., Grangier, D., Yarats, D., and Dauphin, Y. N. (2017). Convolutional sequence to sequence learning. *ICML 2017*. 4.1.2, 5.2.6, 6.1.1, 6.1.2, 6.2.1

Gibson, M. (2010). Unsupervised cross-lingual speaker adaptation for HMM-based speech synthesis using two-pass decision tree construction. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2010*. 4.1.1

Graves, A. (2013). Generating sequences with recurrent neural networks. *arXiv preprint*

arXiv:1308.0850. 6.1.1, 6.1.2

Griffin, D. and Lim, J. (1984). Signal estimation from modified short-time fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 32(2):236–243. 4.1.2

Hakkani-Tür, D., Riccardi, G., and Gorin, A. (2002). Active learning for automatic speech recognition. In *IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), 2002*, volume 4, pages IV–3904. IEEE. 3.2

Harper, M. (2011). IARPA Babel program. http://www.iarpa.gov/Programs/ia/Babel/babel.html. 1, 4.1.3

He, J., Qian, Y., Soong, F. K., and Zhao, S. (2012). Turning a monolingual speaker into multilingual for a mixed-language TTS. In *Thirteenth Annual Conference of the International Speech Communication Association*. 4.2.1

Hermansky, H. (1998). Modulation spectrum in speech processing. In *Signal Analysis and Prediction*, pages 395–406. Springer. 3.5.3

Hsu, C.-C., Hwang, H.-T., Wu, Y.-C., Tsao, Y., and Wang, H.-M. (2016). Voice conversion from non-parallel corpora using variational auto-encoder. In *Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2016 Asia-Pacific*, pages 1–6. IEEE. 4.2.1

Hunt, A. J. and Black, A. W. (1996). Unit selection in a concatenative speech synthesis system using a large speech database. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP), 1996*, volume 1, pages 373–376. IEEE. 5.1

Imai, S., Kobayashi, T., Tokuda, K., Masuko, T., Koishida, K., Sako, S., and Zen, H. (2017). Speech signal processing toolkit (SPTK), version 3.11. 4.3.2

Imai, S., Sumita, K., and Furuichi, C. (1983). Mel log spectrum approximation (MLSA) filter for speech synthesis. *Electronics and Communications in Japan (Part I: Communications)*, 66(2):10–18. 2.1.1

Ito, K. (2017). The LJ speech dataset. https://keithito.com/LJ-Speech-Dataset/. 4.1.3, 4.2.2

Itoh, N., Sainath, T. N., Jiang, D. N., Zhou, J., and Ramabhadran, B. (2012). N-best entropy based data selection for acoustic modeling. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),2012*, pages 4133–4136. IEEE. 3.2

Jansen, A. and Church, K. (2011). Towards unsupervised training of speaker independent acoustic models. In *INTERSPEECH*. 4.3.1

Jansen, A., Thomas, S., and Hermansky, H. (2013). Weak top-down constraints for unsupervised acoustic model training. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2013*, pages 8091–8095. 4.3.1

Jansen, A. and Van Durme, B. (2011). Efficient spoken term discovery using randomized algorithms. In *Automatic Speech Recognition and Understanding (ASRU), 2011 IEEE Workshop on*, pages 401–406. IEEE. 4.3.1

Kaiser, J. F. (1990). On a simple algorithm to calculate the energy of a signal. In *International Conference on Acoustics, Speech, and Signal Processing, (ICASSP), 1990*, pages 381–384. IEEE. 3.5.3

Kemp, T. and Waibel, A. (1999). Unsupervised training of a speech recognizer: recent experiments. In *EuroSpeech*. 3.2

Kirchhoff, K. and Bilmes, J. (2014). Submodularity for data selection in statistical machine translation. In *EMNLP*. 3.2

Kirkpatrick, J., Pascanu, R., Rabinowitz, N., Veness, J., Desjardins, G., Rusu, A. A., Milan, K., Quan, J., Ramalho, T., Grabska-Barwinska, A., et al. (2017). Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526. 4.1.3

Kominek, J. and Black, A. W. (2004). The CMU Arctic speech databases. In *Fifth ISCA Workshop on Speech Synthesis*. 4.2.2, 4.2.2, 4.3.2, 5.1, 5.3, 5.3.3

Kominek, J., Schultz, T., and Black, A. W. (2008). Synthesizer voice quality of new languages calibrated with mean Mel cepstral distortion. In *SLTU*, pages 63–68. 1.1, 2.1.3, 3.3, 3.5.3, 3.6.3, 4.3.2

Koutnik, J., Greff, K., Gomez, F., and Schmidhuber, J. (2014). A Clockwork RNN. In *International Conference on Machine Learning (ICML)*, pages 1863–1871. 5.2.2

Ladd, D. R. (2008). *Intonational phonology*. Cambridge University Press. 5.3

Latorre, J., Iwano, K., and Furui, S. (2006). New approach to the polyglot speech generation by means of an HMM-based speaker adaptable synthesizer. *Speech Communication*, 48(10):1227–1242. 4.1, 4.1.1

Lee, C.-Y. and Glass, J. (2012). A nonparametric bayesian approach to acoustic model discovery. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 40–49. Association for Computational Linguistics. 4.3.1

Lenzo, K. A. (2017). *Improving Prosody Through Analysis by Synthesis*. PhD thesis, Carnegie Mellon University, Pittsburgh, USA. 5.3, 5.3.4

Li, B. and Zen, H. (2016). Multi-language multi-speaker acoustic modeling for LSTM-RNN based statistical parametric speech synthesis. *INTERSPEECH 2016*, pages 2468–2472. 1.2, 4.1

Li, Q. and Atlas, L. (2005). Properties for modulation spectral filtering. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP), 2005).*, volume 4, pages iv–521. IEEE. 2.1.1

Liberman, M. and Prince, A. (1977). On stress and linguistic rhythm. *Linguistic inquiry*, 8(2):249–336. 5.3

Liberman, M. Y. (1975). *The intonational system of English*. PhD thesis, Massachusetts Institute of Technology. (document), 5.3, 5.3.1, 5.4

Lin, H. and Bilmes, J. (2009). How to select a good training-data subset for transcription:

Submodular active selection for sequences. Technical report, DTIC Document. 3.2

Maillard, J., Clark, S., and Yogatama, D. (2017). Jointly learning sentence embeddings and syntax with unsupervised Tree-LSTMs. *arXiv preprint arXiv:1705.09189*. 6.3.1

Mehri, S., Kumar, K., Gulrajani, I., Kumar, R., Jain, S., Sotelo, J., Courville, A., and Bengio, Y. (2017). SampleRNN: An unconditional end-to-end neural audio generation model. *ICLR 2017*. 4.2.1

Metze, F. and Waibel, A. (2002). A flexible stream architecture for asr using articulatory features. In *INTERSPEECH*. 4.3

Mohammadi, S. H. and Kain, A. (2016). A voice conversion mapping function based on a stacked joint-autoencoder. In *INTERSPEECH*, pages 1647–1651. 4.2.1

Muthukumar, P. K. and Black, A. W. (2014). Automatic discovery of a phonetic inventory for unwritten languages for statistical speech synthesis. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014*, pages 2594–2598. IEEE. (document), 4.3, 4.3, 4.4

Nakashika, T., Takiguchi, T., and Minami, Y. (2016). Non-parallel training in voice conversion using an adaptive restricted Boltzmann machine. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(11):2032–2045. 4.2.1

Oord, A. v. d., Dieleman, S., Zen, H., Simonyan, K., Vinyals, O., Graves, A., Kalchbrenner, N., Senior, A., and Kavukcuoglu, K. (2016). WaveNet: A generative model for raw audio. *arXiv preprint arXiv:1609.03499*. 5.1

Ostendorf, M. and Veilleux, N. (1994). A hierarchical stochastic model for automatic prediction of prosodic boundary location. *Computational Linguistics*, 20(1):27–54. 5.3

Parlikar, A. (2013). *Style-Specific Phrasing in Speech Synthesis*. PhD thesis, Carnegie Mellon University, USA. 2.1.1

Pascanu, R., Mikolov, T., and Bengio, Y. (2013). On the difficulty of training recurrent neural networks. In *International Conference on Machine Learning*, pages 1310–1318. 5.2.2

Paul, D. B. and Baker, J. M. (1992). The design for the Wall Street Journal-based CSR corpus. In *Proceedings of the workshop on Speech and Natural Language*, pages 357–362. Association for Computational Linguistics. 4.3.2

Peng, X., Oura, K., Nankaku, Y., and Tokuda, K. (2010). Cross-lingual speaker adaptation for HMM-based speech synthesis considering differences between language-dependent average voices. In *IEEE 10th International Conference on Signal Processing*, pages 605–608. IEEE. 4.1.1

Pierrehumbert, J. B. (1980). *The phonology and phonetics of English intonation*. PhD thesis, Massachusetts Institute of Technology. 5.3

Ping, W., Peng, K., Gibiansky, A., Arik, S. O., Kannan, A., Narang, S., Raiman, J., and Miller, J. (2018). Deep Voice 3: 2000-speaker neural text-to-speech. *ICLR 2018*. 4.1, 4.1.2, 6.1.2, 6.2.1, A.1

Placeway, P., Chen, S., Eskenazi, M., Jain, U., Parikh, V., Raj, B., Ravishankar, M., Rosenfeld, R., Seymore, K., Siegler, M., et al. (1997). The 1996 Hub-4 Sphinx-3 System. In *Proc. DARPA Speech recognition workshop*, pages 85–89. 4.3.2

Prahallad, K. and Black, A. W. (2011). Segmentation of monologues in audio books for building synthetic voices. *Audio, Speech, and Language Processing, IEEE Transactions on*, 19(5):1444–1449. 1.2, 3.4.2

Qian, T., Hollingshead, K., Yoon, S.-y., Kim, K.-y., and Sproat, R. (2010). A Python toolkit for universal transliteration. *Language Resources and Evaluation (LREC)*. 4.1.2

Qian, Y., Xu, J., and Soong, F. K. (2011). A frame mapping based HMM approach to cross-lingual voice transformation. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2011*, pages 5120–5123. IEEE. 4.2.1

Raffel, C., Luong, T., Liu, P. J., Weiss, R. J., and Eck, D. (2017). Online and linear-time attention by enforcing monotonic alignments. *ICML 2017*. 6.1.1

Ren, S., He, K., Girshick, R., and Sun, J. (2015). Faster R-CNN: Towards real-time object detection with region proposal networks. In *Advances in neural information processing systems*, pages 91–99. 6.1.1

Ronanki, S., Watts, O., King, S., and Clark, R. (2014). Syllable based models for prosody modeling in HMM based speech synthesis. *Simple4AllReport*, 9(10):11. 6.3.2

Schatz, T., Peddinti, V., Bach, F., Jansen, A., Hermansky, H., and Dupoux, E. (2013). Evaluating speech features with the minimal-pair ABX task: Analysis of the classical MFC/PLP pipeline. In *INTERSPEECH 2013*, pages 1–5. 4.3.1

Schatz, T., Peddinti, V., Cao, X.-N., Bach, F., Hermansky, H., and Dupoux, E. (2014). Evaluating speech features with the minimal-pair ABX task (II): Resistance to noise. In *INTERSPEECH 2014*. 4.3.1

Shen, J., Pang, R., Weiss, R. J., Schuster, M., Jaitly, N., Yang, Z., Chen, Z., Zhang, Y., Wang, Y., Skerry-Ryan, R., et al. (2018). Natural TTS synthesis by conditioning WaveNet on Mel spectrogram predictions. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2018*. 6.1.2

Sitaram, S. (2015). *Pronunciation Modeling for Synthesis of Low Resource Languages*. PhD thesis, Carnegie Mellon University, USA. 2.1.1

Sitaram, S., Anumanchipalli, G. K., Chiu, J., Parlikar, A., and Black, A. W. (2013a). Text to speech in new languages without a standardized orthography. In *Proceedings of 8th Speech Synthesis Workshop, Barcelona*. 4.3

Sitaram, S., Palkar, S., Chen, Y.-N., Parlikar, A., and Black, A. W. (2013b). Bootstrapping text-to-speech for speech processing in languages without an orthography. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2013*, pages 7992–7996. IEEE. 4.3, 4.3.2

Skerry-Ryan, R., Battenberg, E., Xiao, Y., Wang, Y., Stanton, D., Shor, J., Weiss, R. J., Clark, R., and Saurous, R. A. (2018). Towards end-to-end prosody transfer for expressive speech synthesis with Tacotron. *arXiv preprint arXiv:1803.09047*. 6.3

Song, P., Zheng, W., and Zhao, L. (2013). Non-parallel training for voice conversion based on adaptation method. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2013*, pages 6905–6909. IEEE. 4.2.1

Sotelo, J., Mehri, S., Kumar, K., Santos, J. F., Kastner, K., Courville, A., and Bengio, Y. (2017). Char2Wav: End-to-end speech synthesis. *International Conference on Learning Representations (ICLR)*. 6, 6.1.1, 6.1.2

Srivastava, R. K., Greff, K., and Schmidhuber, J. (2015). Highway networks. *ICML 2015*. 5.2.2

Stan, A., Mamiya, Y., Yamagishi, J., Bell, P., Watts, O., Clark, R., and King, S. (2016). ALISA: An automatic lightly supervised speech segmentation and alignment tool. *Computer Speech & Language*, 35:116–133. 1.2

Stan, A., Watts, O., Mamiya, Y., Giurgiu, M., Clark, R. A., Yamagishi, J., and King, S. (2013). TUNDRA: a multilingual corpus of found data for TTS research created with light supervision. In *INTERSPEECH*, pages 2331–2335. 1.2

Swietojanski, P. and Renais, S. (2016). SAT-LHUC: Speaker adaptive training for learning hidden unit contributions. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),2016*, pages 5010–5014. IEEE. 1.2

Tachibana, H., Uenoyama, K., and Aihara, S. (2018). Efficiently trainable text-to-speech system based on deep convolutional networks with guided attention. *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP) 2018*. 4.1.2, 6, 6.1.2, 6.2.1, 6.2.2, A.2

Tai, K. S., Socher, R., and Manning, C. D. (2015). Improved semantic representations from tree-structured long short-term memory networks. *ACL*. 6.3.1

Taigman, Y., Wolf, L., Polyak, A., and Nachmani, E. (2018). Voice synthesis for in-the-wild speakers via a phonological loop. *ICLR 2018*. 4.1.3, 4.2.1, 6, 6.1.1, 6.1.2, 7.3

Takamichi, S., Toda, T., Black, A. W., Neubig, G., Sakti, S., and Nakamura, S. (2016). Postfilters to modify the modulation spectrum for statistical parametric speech synthesis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(4):755–767. 3.5.3

Taylor, P., Black, A. W., and Caley, R. (1998). The architecture of the Festival speech synthesis system. 2.1.1, 4.3.2

Taylor, P., Black, A. W., and Caley, R. (2001). Heterogeneous relation graphs as a formalism for representing linguistic information. *Speech Communication*, 33(1):153–174. 2.1.1

Team, T. T. D., Al-Rfou, R., Alain, G., Almahairi, A., Angermueller, C., Bahdanau, D., Ballas, N., Bastien, F., Bayer, J., Belikov, A., et al. (2016). Theano: A Python framework for fast computation of mathematical expressions. *arXiv preprint arXiv:1605.02688*. 5.2.5

Tjandra, A., Sakti, S., and Nakamura, S. (2017). Listening while speaking: Speech chain by deep learning. *2017 IEEE Automatic Speech Recognition and Understanding*

Workshop (ASRU), pages 301–308. 7.3

Toda, T., Black, A. W., and Tokuda, K. (2007). Voice conversion based on maximum-likelihood estimation of spectral parameter trajectory. *IEEE Transactions on Audio, Speech, and Language Processing*, 15(8):2222–2235. 4.2.1

Toda, T. and Tokuda, K. (2007). A speech parameter generation algorithm considering global variance for HMM-based speech synthesis. *IEICE TRANSACTIONS on Information and Systems*, 90(5):816–824. 2.1.1, 3.5.3

Tokuda, K., Yoshimura, T., Masuko, T., Kobayashi, T., and Kitamura, T. (2000). Speech parameter generation algorithms for HMM-based speech synthesis. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP) 2000*, volume 3, pages 1315–1318. IEEE. 2.1.1

Tur, G., Hakkani-Tur, D., and Schapire, R. E. (2005). Combining active and semi-supervised learning for spoken language understanding. *Speech Communication*, 45:171–186. 3.2

Van Santen, J. P. and Buchsbaum, A. L. (1997). Methods for optimal text selection. In *EuroSpeech*. 3.2

Varadarajan, B., Khudanpur, S., and Dupoux, E. (2008). Unsupervised learning of acoustic sub-word units. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Short Papers*, pages 165–168. Association for Computational Linguistics. 4.3.1

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010. 5.2.6, 6.1.1, 6.1.2

Veaux, C., Yamagishi, J., MacDonald, K., et al. (2017). CSTR VCTK corpus: English multi-speaker corpus for CSTR voice cloning toolkit. 4.1.3

Wan, V., Latorre, J., Yanagisawa, K., Braunschweiler, N., Chen, L., Gales, M. J., and Akamine, M. (2014a). Building HMM-TTS voices on diverse data. *IEEE Journal of Selected Topics in Signal Processing*, 8(2):296–306. 2.1.2

Wan, V., Latorre, J., Yanagisawa, K., Gales, M., and Stylianou, Y. (2014b). Cluster adaptive training of average voice models. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),2014*, pages 280–284. IEEE. 2.1.2

Wang, Y., Skerry-Ryan, R., Stanton, D., Wu, Y., Weiss, R. J., Jaitly, N., Yang, Z., Xiao, Y., Chen, Z., Bengio, S., et al. (2017). Tacotron: A fully end-to-end text-to-speech synthesis model. *INTERSPEECH*. 4.1, 6, 6.1.2, 6.2.1

Wang, Y., Stanton, D., Zhang, Y., Skerry-Ryan, R., Battenberg, E., Shor, J., Xiao, Y., Ren, F., Jia, Y., and Saurous, R. A. (2018). Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. *arXiv preprint arXiv:1803.09017*. 7.3

Watts, O., Stan, A., Clark, R., Mamiya, Y., Giurgiu, M., Yamagishi, J., and King, S. (2013). Unsupervised and lightly-supervised learning for rapid construction of TTS systems in

multiple languages from found data: evaluation and analysis. In *Proc. 8th ISCA Speech Synthesis Workshop*, pages 101–106. 1.2

Watts, O., Wu, Z., and King, S. (2015). Sentence-level control vectors for deep neural network speech synthesis. In *INTERSPEECH 2015*. 6.3

Wei, K., Liu, Y., Kirchhoff, K., Bartels, C., and Bilmes, J. (2014a). Submodular subset selection for large-scale speech training data. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP),2014*, pages 3311–3315. IEEE. 3.2

Wei, K., Liu, Y., Kirchhoff, K., and Bilmes, J. (2014b). Unsupervised submodular subset selection for speech data. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2014*, pages 4107–4111. IEEE. 3.2

Wester, M., Dines, J., Gibson, M., Liang, H., Wu, Y.-J., Saheer, L., King, S., Oura, K., Garner, P. N., Byrne, W., et al. (2010). Speaker adaptation and the evaluation of speaker similarity in the EMIME speech-to-speech translation project. *Proc. of 7th ISCA Speech Synthesis Workshop*. 1.2, 2.1.2

Williams, A., Drozdov, A., and Bowman, S. R. (2018). Do latent tree learning models identify meaningful structure in sentences? *Transactions of the ACL (TACL). 2018*. 6.3.1

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. In *Reinforcement Learning*, pages 5–32. Springer. 6.3.1

Wu, Y., Zhang, R., and Rudnicky, A. (2007). Data selection for speech recognition. In *Automatic Speech Recognition & Understanding, 2007. ASRU. IEEE Workshop on*, pages 562–565. IEEE. 3.2

Wu, Y.-J., King, S., and Tokuda, K. (2008). Cross-lingual speaker adaptation for HMM-based speech synthesis. In *Chinese Spoken Language Processing, 2008. ISCSLP'08. 6th International Symposium on*, pages 1–4. IEEE. 4.1.1

Wu, Y.-J., Nankaku, Y., and Tokuda, K. (2009). State mapping based method for cross-lingual speaker adaptation in HMM-based speech synthesis. In *INTERPSEECH*. 4.1.1

Yamagishi, J. (2006). Average-voice-based speech synthesis. *PhD Thesis, Tokyo Institute of Technology*. 1.2, 2.1.2, 2.1.2

Yamagishi, J. and Kobayashi, T. (2007). Average-voice-based speech synthesis using HSMM-based speaker adaptation and adaptive training. *IEICE TRANSACTIONS on Information and Systems*, 90(2):533–543. 2.1.2

Yamagishi, J., Kobayashi, T., Nakano, Y., Ogata, K., and Isogai, J. (2009). Analysis of speaker adaptation algorithms for HMM-based speech synthesis and a constrained SMAPLR adaptation algorithm. *Audio, Speech, and Language Processing, IEEE Transactions on*, 17(1):66–83. 1.2, 2.1.2, 2.1.2

Yamagishi, J., Kobayashi, T., Renals, S., King, S., Zen, H., Toda, T., and Tokuda, K. (2007a). Improved average-voice-based speech synthesis using gender-mixed modeling and a parameter generation algorithm considering GV. *Proc. 6th ISCA Workshop on Speech Synthesis*, pages 125–130. 2.1.2

Yamagishi, J., Kobayashi, T., Tachibana, M., Ogata, K., and Nakano, Y. (2007b). Model adaptation approach to speech synthesis with diverse voices and styles. In *IEEE International Conference on Acoustics, Speech and Signal Processing, (ICASSP) 2007.*, volume 4, pages IV–1233. IEEE. 2.1.2, 2.1.2

Yamagishi, J., Ling, Z., and King, S. (2008). Robustness of HMM-based speech synthesis. *INTERSPEECH*. 2.1.2

Yamagishi, J., Masuko, T., Tokuda, K., and Kobayashi, T. (2003a). A training method for average voice model based on shared decision tree context clustering and speaker adaptive training. In *IEEE International Conference on Acoustics, Speech, and Signal Processing, (ICASSP) 2003*, volume 1, pages I–716. IEEE. 2.1.2

Yamagishi, J., Tamura, M., Masuko, T., Tokuda, K., and Kobayashi, T. (2003b). A context clustering technique for average voice models. *IEICE TRANSACTIONS on Information and Systems*, 86(3):534–542. 2.1.2

Yamagishi, J., Usabaev, B., King, S., Watts, O., Dines, J., Tian, J., Guan, Y., Hu, R., Oura, K., Wu, Y.-J., et al. (2010). Thousands of voices for HMM-based speech synthesis–analysis and application of TTS systems built on various ASR corpora. *Audio, Speech, and Language Processing, IEEE Transactions on*, 18(5):984–1004. 1.2

Yamamoto, R. (December 2017). DeepVoice3 PyTorch Implementation. https://github.com/r9y9/deepvoice3_pytorch. 4.1.2

Yogatama, D., Blunsom, P., Dyer, C., Grefenstette, E., and Ling, W. (2017). Learning to compose words into sentences with reinforcement learning. *ICLR 2017*. 6.3.1

Yoshimura, T. (2002). *Simultaneous Modelling of Phonetic and Prosodic Parameters and Characteristic Conversion for HMM-Based Text-to-Speech Systems*. PhD thesis, Department of Electrical and Computer Engineering Nagoya Institute of Technology. 2.1.1

Zen, H. (2015). Acoustic modeling in statistical parametric speech synthesis - from HMM to LSTM-RNN. In *Proc. MLSLP*. Invited paper. 2.1.1

Zen, H., Braunschweiler, N., Buchholz, S., Gales, M. J., Knill, K., Krstulovic, S., and Latorre, J. (2012). Statistical parametric speech synthesis based on speaker and language factorization. *IEEE transactions on audio, speech, and language processing*, 20(6):1713–1724. 2.1.2

Zen, H., Senior, A., and Schuster, M. (2013). Statistical parametric speech synthesis using deep neural networks. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2013*, pages 7962–7966. IEEE. 5.1

Zen, H., Tokuda, K., and Black, A. W. (2009). Statistical parametric speech synthesis. *Speech Communication*, 51(11):1039–1064. 5.1

Zilly, J. G., Srivastava, R. K., Koutník, J., and Schmidhuber, J. (2017). Recurrent highway networks. *ICML 2017*. 5.2.2