Automatic Building of Synthetic Voices from Audio Books

Kishore Prahallad

CMU-LTI-10-xxx

July 26, 2010

School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213

Thesis Committee:
Alan W Black, Chair
Mosur Ravishankar
Tanja Schultz
Keiichi Tokuda, Nagoya Institute of Technology, Japan.

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

Keywords: Speech synthesis, audio books, voice conversion, speaker-specific phrasing

Abstract

Current state-of-the-art text-to-speech systems produce intelligible speech but lack the prosody of natural utterances. Building better models of prosody involves development of prosodically rich speech databases. However, development of such speech databases requires a large amount of effort and time. An alternative is to exploit story style monologues (long speech files) in audio books. These monologues already encapsulate rich prosody including varied intonation contours, pitch accents and phrasing patterns. Thus, audio books act as excellent candidates for building prosodic models and natural sounding synthetic voices. The processing of such audio books poses several challenges including segmentation of long speech files, detection of mispronunciations, extraction and evaluation of representations of prosody. In this thesis, we address the issues of segmentation of long speech files, capturing prosodic phrasing patterns of a speaker, and conversion of speaker characteristics. Techniques developed to address these issues include – text-driven and speech-driven methods for segmentation of long speech files; an unsupervised algorithm for learning speaker-specific phrasing patterns and a voice conversion method by modeling target speaker characteristics. The major conclusions of this thesis are –

- Audio books can be used for building synthetic voices. Segmentation of such long speech files can be accomplished without the need for a speech recognition system.
- The prosodic phrasing patterns are specific to a speaker. These can be learnt and incorporated to improve the quality of synthetic voices.
- Conversion of speaker characteristics can be achieved by modeling speaker-specific features of a target speaker.

Finally, the techniques developed in this thesis enable prosody research by leveraging a large number of audio books available in the public domain.

Executive summary

While current state-of-the-art text-to-speech (TTS) systems produce understandable speech, the prosody of synthesized utterances is not as good as naturally spoken utterances. Prosody of speech involves variation in intonation, duration, loudness and formant frequencies of speech sounds. In current TTS systems, prosodic models are built using speech databases such as CMU ARCTIC. These speech databases consist of isolated utterances which are short sentences or phrases such as *"He did not rush in."* and *"It was edged with ice."* . These sentences are selected to optimize the coverage of phones. Such utterances are not semantically related to each other, and elicit only one type of intonation, i.e., declarative. Other variants of intonation corresponding to paragraphs and utterances such as wh-questions (*what time is it?*), unfinished statements (*I wanted to* ..), yes/no questions (*Are they ready to go?*) and surprise (*What! The plane left already!?*), are typically not captured. A prosodically rich speech database includes intonation variations; pitch accents which make words perceptually prominent, as in, *I didn't shoot AT him, I shot PAST him*; and phrasing patterns - whereby certain words are grouped within the utterances - for naturalness and comprehension. Such databases aid in building better prosodic models and consequently natural sounding synthetic voices.

The process of building better prosodic models involves development/acquisition of prosodically rich speech databases. Development of such speech databases requires a large amount of effort and time. An alternative is to exploit story style monologues in audio books. These monologues already encapsulate rich prosody including varied intonation contours, pitch accents and phrasing patterns. Thus audio books act as excellent candidates for building prosodic models and natural sounding synthetic voices. This thesis aims to develop techniques which aid in building natural and stylistic voices by leveraging prosodically rich audio books. However, processing of audio books poses several challenges including segmentation of long speech files, detection of mispronunciations, extraction and evaluation of representations for prosody. In this thesis we address the issues of segmentation of

long speech files, capturing phrasing patterns specific to a speaker and conversion of speaker characteristics.

Segmentation of monologues: Monologues in audio books are long speech files, and segmentation of monologues is a non-trivial issue. Earlier works break long speech files into smaller segments using silence regions as breaking points. These smaller segments are given to an automatic speech recognition (ASR) system to produce hypothesized transcriptions. As the original text of utterances is also available, the search space of ASR is constrained using n-grams or finite state transducer based language model. In spite of search space being constrained, the hypothesized transcriptions are not always error-free; especially at the border of small segments where the constraints represented by language models are weak. Hence the original text is aligned with the hypothesized transcription to obtain, what are referred to as, islands of confidence. Between the islands of confidence, the Viterbi algorithm is employed to force-align the speech with the original text to obtain word/phone level boundaries. Apart from the practical difficulty in implementing this approach in the context of a TTS system, it strongly implies that a speech recognition system should be readily available before building a speech synthesis system. In this thesis, we propose an approach based on modifications to the Viterbi algorithm to process long speech files in parts. This enables segmentation of long speech files without a need for an ASR.

Speaker-specific phrasing: Phrasing is a phenomenon whereby speakers group certain words within the utterances. Automatic annotation of speech databases with prosodic phrase breaks aid in building better prosodic models. However, there is no agreement on describing the phrasing patterns in terms of acoustic features in the speech signal. The relationship between syntactic structure and prosodic phrase breaks is also not well understood. Moreover, prosodic phrasing may vary with speakers. In this thesis, we investigate whether prosodic phrase breaks are specific to a speaker, and if so how to annotate a speech database with speaker-specific phrase breaks. We demonstrate that prosodic phrase breaks are specific to a speaker, and propose an unsupervised algorithm to learn speaker-specific phrase breaks.

Conversion of speaker characteristics is another important issue from a listener's perspective. Conversion includes rendering a synthesized utterance in a voice preferred by the listener. This can be accomplished by learning a transformation function which converts a synthetic voice to a specified target speaker. To learn such a transformation, current voice conversion techniques rely on the existence of a parallel corpus, i.e., the same set of utterances recorded by both the source and target speakers. However, the collection of parallel data may not always be feasible.

For example, if the target speaker is a celebrity or speaks a different language, then he/she may not be available to record these parallel utterances. While there have been earlier works which avoid the need for parallel data, they still require speech data (though non-parallel) from source speakers *a priori* to build a conversion model. In this thesis, we address the issue of building voice conversion models by asking the question "can we capture speaker-specific characteristics of a target speaker from the speech signal (independent of any assumptions about a source speaker) and super-impose these characteristics on the speech signal of any arbitrary source speaker to perform voice conversion?". In this thesis, we propose a method to capture speaker-specific characteristics of a target speaker and avoid the need for speech data from a source speaker to train/adapt a voice conversion model.

The conclusions of this thesis are as follows:

- Audio books can be used for building synthetic voices. Segmentation of long speech files can be accomplished without the need for an ASR.
- Prosodic phrase breaks are specific to a speaker. Incorporation of speaker-specific phrase breaks improve the quality of synthetic voices.
- Artificial neural network based voice conversion performs as good as Gaussian mixture model based voice conversion. To build a voice conversion model, it is not necessary to have parallel or pseudo-parallel data. It can be achieved by modeling target speaker characteristics in the form of a nonlinear mapping function using artificial neural networks.
- Finally, the techniques developed in this thesis enable prosody research by leveraging a large number of audio books available in the public domain. We believe, this is an important milestone in prosody modeling and in building natural sounding synthetic voices.

Acknowledgements

Oh yeah. My advisor is cool!.

Contents

List of Figures

List of Tables

Chapter 1

Introduction to text-to-speech

Spoken language based search is one of the most common services being offered today to provide information about travel, health, finance and entertainment. For example, the services provided on telephone/cellphone by TELLME (1-800-555-tell) and GOOGLE (1-800-GOOG-411) fall under this category. Spoken language based search interfaces provide a natural and convenient mode for a majority of information exchange purposes. Such interfaces involve the following subsystems – a speech recognition system that converts speech into text, a spoken language understanding system that maps the words into actions and plans to initiate sequence of actions and a text-to-speech system that conveys information in spoken form [Huang et al., 2001]. The recent advances in these three subsystems are due to increase in storage and computation power of computers which has led to the use of data-driven statistical methods to build spoken language systems.

Of all the three subsystems the conversion of text into spoken form is deceptively nontrivial. A naive approach is to consider storing and concatenation of basic sounds (also referred to as phones) of a language to produce a speech waveform. But, natural speech consists of co-articulation i.e., effect of coupling two sound together, and prosody at syllable, word, sentence and discourse level, which cannot be synthesized by simple concatenation of phones. Another method often employed is to store a huge dictionary of the most common words. However, such a method may not synthesize millions of names and acronyms which are not in the dictionary and has to deal with generating appropriate intonation and duration for words in different context. Thus a text-to-speech approach using phones provides flexibility but cannot produce intelligible and natural speech, while a word level concatenation produces intelligible and natural speech but is not flexible. In order to balance

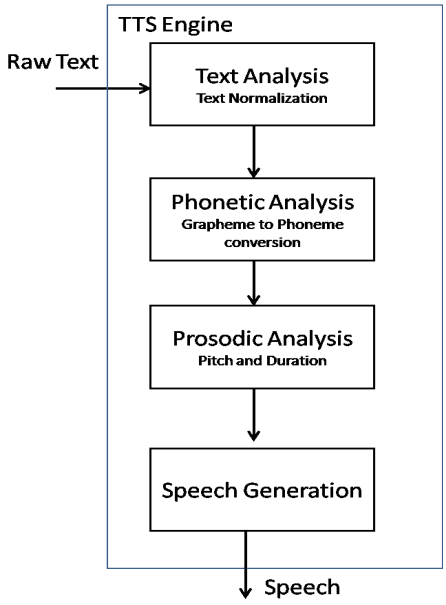


Figure 1.1: Architecture of Text to Speech System

between flexibility and intelligibility/naturalness, subword units such as diphones which capture essential coarticulation between adjacent phones are used as suitable units in a text-to-speech system.

1.1 Components of a text-to-speech

A typical architecture of a Text-to-Speech (TTS) system is as shown in Fig. 1.1. The components of a text-to-speech system could be broadly categorized as text processing and methods of speech generation.

1.1.1 Text processing

In the real world, the typical input to a text to speech system is text as available in electronic documents, news papers, blogs, emails etc. The text available in real world is anything but a sequence of words available in standard dictionary. The text contains several non-standard words such as numbers, abbreviations, homographs

and symbols built using punctuation characters such as exclamation '!', smileys ':-)' etc. The goal of text processing module is to process the input text, normalize the non-standard words, predict the prosodic pauses and generate the appropriate phone sequences for each of the words.

Normalization of non-standard words

The text in real world consists of words whose pronunciation is typically not found in dictionaries or lexicons such as "IBM", "CMU", and "MSN" etc. Such words are referred to as non-standard words (NSW). The various categories of NSW are: 1) numbers whose pronunciation changes depending on whether they refer to currency, time, telephone numbers, zip code etc. 2) abbreviations, contractions, acronyms such as ABC, US, approx., Ctrl-C, lb., 3) punctuations 3-4, +/-, and/or, 4) dates, time, units and URLs.

Many NSW's are homographs, i.e., words with same written form but different pronunciation. Some of the examples are: 1) IV which may be variously four (*Article IV*), the fourth (*Henry IV*), or I.V. (*IV drip*), 2) three or four digit numbers which could be dates and ordinary numbers (*in 2040, 2040 tons*). Machine learning models such as Classification and Regression Trees (CART) are used to predict the class of NSW which is typically followed by rules to generate appropriate expansion of a NSW into a standard form [Sproat et al., 2001].

Grapheme to phoneme conversion

Given the sequence of words, the next step is to generate a sequence of phones. For languages such as Spanish, Telugu, Kannada, where there is a good correspondence between what is written and what is spoken, a set of simple rules may often suffice. For languages such as English where the relationship between the orthography and pronunciation is complex, a standard pronunciation dictionary such as CMU-DICT is used. To handle unseen words, a grapheme-to-phoneme generator is built using machine learning techniques [Black et al., 1998].

Prosodic analysis

Prosodic analysis deals with modeling and generation of appropriate duration and intonation contours for the given text. This is inherently difficult since prosody is

absent in text. For example, the sentences – where are you going?; where are you GOING? and where are YOU going?, have same text-content but can be uttered with different intonation and duration to convey different meanings. To predict appropriate duration and intonation, the structure of input text needs to be analyzed. This can be performed by a variety of algorithms including simple rules, example-based techniques and machine learning algorithms. The generated duration and intonation contour can be used to manipulate the context-insensitive diphones in diphone based synthesis or to select an appropriate unit in unit selection voices [Black and Taylor, 1994].

1.1.2 Methods of speech generation

The methods of conversion of phone sequence to speech waveform could be categorized into parametric, concatenative and statistical parametric synthesis.

Parametric synthesis

Parameters such as formants, linear prediction coefficients are extracted from the speech signal of each phone unit. These parameters are modified during synthesis time to incorporate co-articulation and prosody of a natural speech signal. The required modifications are specified in terms of rules which are derived manually from the observations of speech data. These rules include duration, intonation, co-articulation and excitation function. Examples of the early parametric synthesis systems are Klatt's formant synthesis [Klatt, 1987] and MITTalk [Allen et al., 1987].

Concatenative synthesis

Derivation of rules in parametric synthesis is a laborious task. Also, the quality of synthesized speech using traditional parametric synthesis is found to be robotic. This has led to development of concatenative synthesis where the examples of speech units are stored and used during synthesis. The speech units used in concatenative synthesis are typically at diphone level so that the natural co-articulation is retained [Olive, 1977]. Duration and intonation are derived either manually or automatically from the data and are incorporated during synthesis time. Examples of diphone synthesizers are Festival diphone synthesis [Taylor et al., 1998] and MBROLA [Dutoit et al., 1996].

The possibility of storing more than one example of a diphone unit, due to increase in storage and computation capabilities, has led to development of unit selection synthesis [Hunt and Black, 1996]. Multiple examples of a unit along with the relevant linguistic and phonetic context are stored and used in the unit selection synthesis. The quality of unit selection synthesis is found to be more natural than diphone and parametric synthesis. However, unit selection synthesis lacks the consistency i.e., in terms of variations of the quality [Black and Taylor, 1997].

Statistical parametric synthesis

Statistical Parametric Synthesis (SPS) is one of the latest trends in TTS. The SPS methods produce speech from a set of parameters learned from the speech data. Unlike traditional parametric synthesis methods which require manual specification and hand-tuning of the parameters, the SPS methods use statistical machine learning models such as CART, HMMs, etc., to estimate the parameters of speech sounds and their dynamics. The SPS methods offer simplicity in storage by encoding the speech data in terms of a compact set of parameters, and also provide mechanisms for manipulation of prosody, voice conversion etc. The SPS methods are found to produce intelligible and consistent speech as compared to natural and often inconsistent speech by unit selection techniques [Black et al., 2007, Bennett and Black, 2006, Zen et al., 2007]. Please refer to Appendix C for details on different SPS techniques.

1.2 Reviewing the state-of-the-art

To review the current state-of-the-art in speech synthesis, we have used results from the Blizzard speech synthesis challenge. The purpose of Blizzard Challenge is to compare and contrast different speech synthesis techniques and systems on a bench marked database [Black and Tokuda, 2005]. Since 2005, several universities and systems have participated in this challenge. This has led to the congregation of several researchers on a common platform in Blizzard workshops to compare and contrast different synthesis techniques, with the goal to build naturally speaking synthesis systems. Blizzard 2010 is the current Blizzard Challenge, sixth in the series, in which participants built voices from a common dataset [Black, 2010].

In this thesis, we refer to the results of Blizzard 2007 challenge [Fraser and

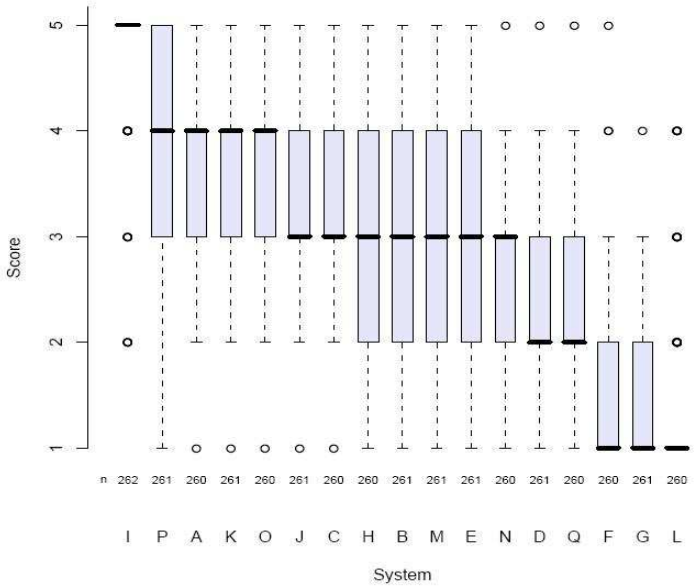


Figure 1.2: Box plots of similarity scores for TTS systems vs natural speech for Voice-A adapted from [Fraser and King, 2007].

King, 2007], where a large listening test was conducted which allows comparison of speech synthesis systems in terms of naturalness and intelligibility. Eight hours of speech data spoken by a single speaker was provided to each participant to build a speech synthesis system [Nil et al., 2007]. This speech data consisted of 3.7 hours of News genre, 3.6 hours of conversational genre and 0.8 hours of ARCTIC data [Kominek and Black, 2004b]. Each participant had submitted up to three voices Voice-A, Voice-B and Voice-C. The Voice-A was built using a full data set of about 8 hours speech; Voice-B was built using ARCTIC subset of about 1 hour and Voice-C was built using participant-selected subset of about 1 hour.

Each voice was evaluated in a listening test of 400 held-out utterances generated from the following genres: 1) Conversational - 100 2) News - 100 3) ARCTIC - 100 4) Modified Rhyme Test (MRT) - 50 and 5) Semantically Unpredictable Sentences (SUS) - 50. The evaluation was based on mean opinion scores, pairwise comparisons and type-in tasks. Fig. 1.2 shows the box plots displaying the scores between synthesis systems and the natural speech, where the letter *I* represents natural speech. From Fig. 1.2, it could be observed that in spite of having 8 hours of speech, the synthesis systems have scored lower in comparison with natural speech. It is evident from the Blizzard challenge 2007 that the current techniques for speech

synthesis have limitations in generating natural, consistent and stylistic synthetic voices [Fraser and King, 2007].

1.3 Thesis statement

While current state-of-the-art text-to-speech (TTS) systems produce understandable/intelligible speech, the prosody of synthesized utterances is not as good as naturally spoken utterances. Prosody of speech involves variation in intonation, duration, loudness and formant frequencies of speech sounds. In current TTS systems, prosodic models are built using speech databases such as CMU ARCTIC [Kominek and Black, 2004a]. These speech databases consist of isolated utterances which are short sentences or phrases such as *"He did not rush in."* and *"It was edged with ice."* . These sentences are selected to optimize the coverage of phones. Such utterances are not semantically related to each other, and elicit only one type of intonation, i.e., declarative. Other variants of intonation corresponding to paragraphs and utterances such as wh-questions (*what time is it?*), unfinished statements (*I wanted to ..*), yes/no questions (*Are they ready to go?*) and surprise (*What! The plane left already!?*), are typically not captured.

Speech databases rich in prosody helps in building better prosodic models. A prosodically rich speech database includes intonation variations; pitch accents which make words perceptually prominent, as in, *I didn't shoot AT him, I shot PAST him*; and phrasing patterns which divide an utterance into meaningful chunks for comprehension and naturalness. The process of building better prosodic models involves development/acquisition of a prosodically rich speech database, annotation of prosodic events such as pitch accents and phrasing patterns in the speech database, learning to predict these prosodic events from text and realizing the same in output speech.

Development of prosodically rich speech databases requires a large amount of effort and time. An alternative is to exploit story style monologues in audio books. These monologues already encapsulate rich prosody including varied intonation contours, pitch accents and phrasing patterns. Thus audio books act as excellent candidates for building prosodic models and a natural sounding synthetic voice. This thesis aims to develop techniques which aid in building natural and stylistic voices by leveraging prosodically rich audio books.

However, processing of audio books poses several challenges. A few of them are as follows.

- Segmentation of monologues: Monologues in audio books are long speech files. The issue in segmentation of large speech files is to align a speech signal (as large as 10 hours or more) with the corresponding text to break the speech signal into utterances corresponding to sentences in text and/or provide phone-level time stamps.
- Detection of mispronunciations: During the recordings, a speaker might delete or insert at syllable, word, sentence level and thus the speech signal does not match with the transcription. It is important to detect these mispronunciations using acoustic confidence measures so that the specific regions or the entire utterances can be ignored while building voices.
- Detection of pronunciation variants: Speakers may incorporate subtle variations at the sub-word during pronunciation of content words, proper nouns etc. and these pronunciation variants have to be detected and represented so that they could be produced back during synthesis.
- Features representing prosody: Another issue is the identification, extraction and evaluation of representations that characterize the prosodic variations at sub-word, word, sentence and paragraph level. These include prosodic phrase breaks and emphasis or prominence of specific words during the discourse of a story.
- Filtering: Often recordings may have multiple sources, thus filtering of multi-speakers data, music and speech and nullifying the noisy or channel effects may be needed.

1.3.1 Issues addressed in this thesis

In the scope of this thesis we address the issues of segmentation of long speech files, capturing of phrasing patterns specific to a speaker and conversion of speaker characteristics.

Segmentation of monologues: Monologues in audio books are long speech files, and segmentation of monologues is a non-trivial issue. Earlier works break long speech files into smaller segments using silence regions as breaking points. These smaller segments are given to an automatic speech recognition (ASR) system to produce hypothesized transcriptions. As the original text of utterances is also available, the search space of ASR is constrained using n-grams or finite state transducer based language model. In spite of search space being constrained, the

hypothesized transcriptions are not always error-free; especially at the border of small segments where the constraints represented by language models are weak. Hence the original text is aligned with the hypothesized transcription to obtain, what are referred to as, islands of confidence. Between the islands of confidence, the Viterbi algorithm is employed to force-align the speech with the original text to obtain word/phone level boundaries. Apart from practical difficulty in implementing this approach in the context of a TTS system, it strongly implies that a speech recognition system should be readily available before building a speech synthesis system. In this thesis, we propose an approach based on modifications to the Viterbi algorithm to process long speech files in parts. This enables segmentation of long speech files without an ASR.

Speaker-specific phrasing: Phrasing is a phenomenon whereby speakers group certain words within the utterances. Automatic annotation of speech databases with prosodic phrase breaks aid in building better prosodic models. However, there is no agreement on describing the phrasing patterns in terms of acoustic features in the speech signal. The relationship between syntactic structure and prosodic phrase breaks is also not well understood. Moreover, prosodic phrasing may vary with speakers. In this thesis, we investigate whether prosodic phrase breaks are specific to a speaker, and if so how to annotate a speech database with speaker-specific phrase breaks. We demonstrate that prosodic phrase breaks are specific to a speaker, and propose an unsupervised algorithm to learn speaker-specific phrase breaks.

Conversion of speaker characteristics is another important issue from a listener's perspective. Conversion includes rendering a synthesized utterance in a voice preferred by a listener. This can be accomplished by learning a transformation function which converts a synthetic voice to a specified target speaker. To learn such transformation, current voice conversion techniques rely on the existence of parallel corpus, i.e., the same set of utterances recorded by both the source and target speakers. However, the collection of parallel data may not always be feasible. For example, if the target speaker is a celebrity or speaks a different language, then he/she may not be available to record these parallel utterances. While there have been earlier works which avoid the need for parallel data, they still require speech data (though non-parallel) from source speakers *a priori* to build a voice conversion model. In this thesis, we address the issue of building voice conversion models by asking the question "can we capture speaker-specific characteristics of a target speaker from the speech signal (independent of any assumptions about a source speaker) and super-impose these characteristics on the speech signal of any arbitrary source speaker to perform voice conversion?". In this thesis, we propose a method to capture speaker-specific characteristics of a target speaker and avoid

the need for speech data from a source speaker to train/adapt a voice conversion model.

1.4 Organization of the thesis

Chapter 2 discusses the modifications to the Viterbi algorithm for segmentation of large speech files. Using these modifications, two different methods of segmenting a large speech file are proposed and are evaluated.

Chapter 3 discusses the application of modified Viterbi algorithm on large speech files found in audio books. The modifications to the Viterbi are implemented in a package referred to as INTERSLICE. A brief description of INTERSLICE is provided, and the experiments are conducted to demonstrate the usefulness of INTERSLICE in building several voices from audio books in the public domain. Synthetic voices are built in English as well as in Telugu.

Chapter 4 exploits the single speaker recordings of the audio books to build a speaker-specific prosodic phrasing module. An unsupervised algorithm is proposed to detect the prosodic phrase breaks in the speech signal. The usefulness of these automatically detected speaker-specific phrase breaks is demonstrated in the synthetic voices of English as well in Telugu.

Chapter 5 discusses the need for personalization of synthetic voice, and explains an Artificial Neural Network (ANN) based framework for voice morphing/conversion. Experiments are conducted to demonstrate that ANN based voice conversion performs as good as that of the state-of-the-art voice conversion based on Gaussian Mixture Models (GMM).

Chapter 6 proposes a method of capturing speaker-specific features using an ANN model, which allows to transform an arbitrary voice to a specified target speaker. Such model do not make use of any *a priori* knowledge of the speech data of the source speakers. Various demonstrations are provided how such a voice conversion model can help to personalize a synthetic voice.

Chapter 7 concludes the thesis by providing a summary, important conclusions and future work.

Chapter 2

Segmentation of monologues

Building synthetic voices involves segmentation of speech into phone level units. This can be accomplished by force-aligning an entire utterance with its text using the Viterbi algorithm. Such simple solution fails for utterances longer than a few minutes as memory requirements of the Viterbi algorithm increases with the length of utterances.

Monologues in audio books are long speech files, and segmentation of monologues is a non-trivial issue [Moreno and Alberti, 2009]. Earlier works have addressed this issue by breaking long speech files into smaller segments using silence regions as breaking points [Toth, 2004]. These smaller segments are then transcribed by an automatic speech recognition (ASR) system to produce hypothesized transcriptions. As the original text of utterances is also available, the search space of ASR is constrained using an n-gram or a finite state transducer based language model [Trancoso et al., 2006], [Moreno and Alberti, 2009]. In spite of search space being constrained, the hypothesized transcriptions are not always error-free; especially at the border of small segments where the constraints represented by a language model are weak [Stolcke and Shriberg, 1996], [Moreno and Alberti, 2009]. Hence the original text is aligned with the hypothesized transcription to obtain, what are referred to as, islands of confidence. Between the islands of confidence, the Viterbi algorithm is employed to force-align the speech with the original text to obtain word/phone level boundaries [Moreno and Alberti, 2009], [Moreno et al., 1998]. Apart from practical difficulty in implementing this approach in the context of a TTS system, it strongly implies that a speech recognition system should be readily available before building a speech synthesis system.

In this thesis, we propose an approach based on modifications to the Viterbi

algorithm to process long speech files in parts. Our approach differs significantly from [Trancoso et al., 2006], [Moreno et al., 1998] and [Moreno and Alberti, 2009], as we do not need a large vocabulary ASR or employ language models using n-grams or a finite state transducer to constrain the search space. Since the proposed approach is based on modifications to the Viterbi algorithm, it is suitable for languages (especially for low resource languages), where ASR systems are not readily available.

2.1 The Viterbi algorithm

Let $Y = \{\boldsymbol{y}(1), \boldsymbol{y}(2), \ldots, \boldsymbol{y}(T)\}$ be a sequence of observed feature vectors[1] extracted from an utterance of T frames. Let $S = \{1, \ldots, j, \ldots, N\}$ be a state sequence corresponding to the sequence of words in text of the utterance. A forced-alignment technique aligns the sequence of feature vectors (Y) with the given sequence of states (S) using a set of existing acoustic models[2]. The result is a sequence of states $\{x(1), x(2), \ldots, x(T)\}$, unobserved so far, for the observation sequence Y. The steps involved in obtaining this unobserved state sequence are as follows.

Let $p(\boldsymbol{y}(t)|x(t) = j)$ denote the emission probability of state j for a feature vector observed at time t and $1 \leq j \leq N$, where N is the total number of states. Let us define $\alpha_t(j) = p(x(t) = j, \boldsymbol{y}(1), \boldsymbol{y}(2), \ldots, \boldsymbol{y}(t))$. This is a joint probability of being in state j at time t and of having observed all the acoustic features up to and including time t. This joint probability could be computed frame-by-frame using the recursive equation

$$\alpha_t(j) = \sum_i \alpha_{t-1}(i) a_{i,j} p(\boldsymbol{y}(t)|x(t) = j) \tag{2.1}$$

where $a_{i,j} = p(x(t) = j|x(t-1) = i)$. Note that the Eq. (2.1) indicates sum of paths, and it transforms to the Viterbi algorithm if the summation is replaced with a $\max$ operation, as shown in Eq. (2.2).

$$\alpha_t(j) = \max_i \{\alpha_{t-1}(i) a_{i,j}\} p(\boldsymbol{y}(t)|x(t) = j). \tag{2.2}$$

The values of $a_{i,j}$ and $p(\boldsymbol{y}(t)|x(t) = j)$ are significantly less than 1. For large values

[1] Speech signal is divided into frames of 10 ms using a frame shift of 5 ms. Each frame of speech data is passed through a set of Mel-frequency filters to obtain 13 cepstral coefficients. See Appendix A for more details.

[2] The acoustic models used to perform segmentation of large audio files are built using about four hours of speech data collected from four CMU ARCTIC speakers (RMS, BDL, SLT and CLB). See Appendix B for more details.

of t, $\alpha_t(.)$ tends exponentially to zero, and its computation exceeds the precision range of any machine [Rabiner, 1989]. Hence $\alpha_t(.)$ is scaled with term $\frac{1}{\max_i\{\alpha_t(i)\}}$, at every time instant t. This normalization ensures that values of $\alpha_t(.)$ are between 0 and 1 at time t.

Given $\alpha_t(.)$, a backtracking algorithm is used to find the best alignment path. In order to backtrack, an addition variable ϕ is used to store the path as follows.

$$\phi_t(j) = \underset{i}{\text{argmax}}\{\alpha_{t-1}(i)a_{i,j}\}, \tag{2.3}$$

where $\phi_t(j)$ denotes a state at time $(t-1)$ which provides an optimal path to reach state j at time t. Given values of $\phi_t(.)$, a typical backtracking for forced-alignment is as follows:

$$x(T) = N \tag{2.4}$$

$$x(t) = \phi_{t+1}(x(t+1)),\ t = T-1, T-2, \ldots, 1. \tag{2.5}$$

It should be noted that we have assigned $x(T) = N$. This is a constraint in the standard implementation of forced-alignment which aligns the last frame $\boldsymbol{y}(t)$ to the final state N. An implied assumption in this constraint is that the value of $\alpha_T(N)$ is likely to be maximum among the values $\{\alpha_T(j)\}$ for $1 \leq j \leq N$ at time T. The forced-alignment algorithm implemented using Eq. (2.4) and Eq. (2.5) is henceforth referred to as FA-0.

In order to illustrate the usefulness of Eq. (2.4), let us consider the following example. A sequence of two syllables separated by a short pause, as in *"$ba < pau > sa$"*, is uttered and feature vectors are extracted from the speech signal. This sequence of feature vectors is forced-aligned with a sequence of HMM states corresponding to phones /b/, /aa/, /pau/, /s/ and /aa/. Fig. 2.1 displays the values in alpha matrix (HMM states against time measured in frames). These values are obtained using Eq. (2.2) and are normalized between 0 and 1 at each time frame. The dark band in Fig. 2.1 is referred to as beam and it shows how the pattern of values of α closer to 1 is diagonally spread across the matrix. From Fig. 2.1, we observe that at the last frame ($T = 201$), the last[3] HMM state ($N = 15$) has the highest value of α, thus justifying the use of Eq. (2.4) in standard backtracking.

2.2 Modifications to the Viterbi algorithm

The constraint $x(T) = N$ is useful when an entire utterance is force-aligned with its text. As noted earlier, it is not a viable approach for long speech files such as

[3]There are five phones in the sequence. Each phone has three emitting states. Hence the last HMM state $N = 15$.

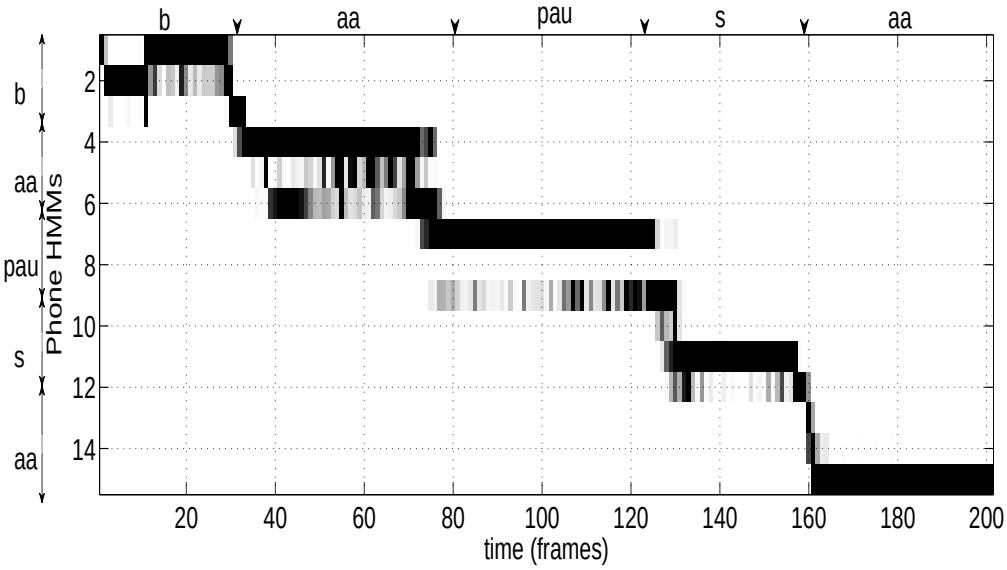


Figure 2.1: An alpha matrix obtained for the alignment of feature vectors corresponding to utterance of *"*$ba < pau > sa$*"* with the HMM state sequence corresponding to phones /b/, /aa/, /pau/, /s/ and /aa/. The markers along the axis of time indicate manually labeled phone boundaries.

monologues in audio books. This could be addressed by aligning the long speech file in parts. We propose two different methods for aligning a long speech file in parts (see Section 2.3 for more details). Each of these methods is explained below with examples. *The numerical values used in these examples are for descriptive purposes only.*

In method-I, the long speech file is sliced into chunks of 30 seconds. The first 30-second chunk is force-aligned with first 120 words in text. If we assume a speaking rate of three words per second, then 30 seconds of speech should roughly corresponds to first 90 words of 120 word sequence. The unknown variable here is the length of word sequence corresponding to the 30-second chunk of speech. This situation is similar to state sequence $S' \subset S$ emitting Y, where $S' = \{1, \ldots, j, \ldots, N'\}$ and $N' < N$. If the Viterbi algorithm is modified to handle $S' \subset S$ emitting Y, i.e., to identify the word sequence corresponding to first 30 seconds of speech, then we succeed in obtaining hidden state sequence for first 30 seconds of speech. This process could then be repeated for the next 30-second chunk of speech until the end of long speech file.

In method-II, the text corresponding to long speech file in sliced into chunks of words. The first 90-word chunk is force-aligned with first 60 seconds of speech. Assuming a speaking rate of three words per second, the 90-word sequence should

roughly correspond to first 30 seconds of speech signal. The unknown variable here is the length of speech signal corresponding to the 90-word sequence. This situation is similar to state sequence S emitting $Y' \subset Y$, where $Y' = \{\boldsymbol{y}(1), \boldsymbol{y}(2), \ldots, \boldsymbol{y}(T')\}$ and $T' < T$. If the Viterbi algorithm is modified to handle S emitting $Y' \subset Y$, i.e., to identify the speech portion corresponding to first 90 words in text, then we succeed in obtaining an alignment for first 90 words in text. This process could then be repeated for the next 90-word chunk until the end of the audio book.

In other words, the constraint $x(T) = N$ in the Viterbi algorithm needs to be modified for situations when 1) the sequence of feature vectors Y is an emission of a sequence of states $S' \subset S$ or 2) the state sequence S emits a sequence of feature vectors $Y' \subset Y$. The following sections describe the proposed modifications to the Viterbi algorithm required for handling these situations.

2.2.1 Emission by a shorter state sequence

Given that Y is an emission sequence for a corresponding sequence of states $S' \subset S$, then the backtracking algorithm can be modified as in Eq. (2.6).

$$x(T) = \underset{1 \leq j \leq N}{\operatorname{argmax}} \{\alpha_T(j)\} \tag{2.6}$$

$$x(t) = \phi_{t+1}(x(t+1)),\ t = T-1, T-2, \ldots, 1. \tag{2.7}$$

Equation (2.6) presents the modified constraint that the last frame $\boldsymbol{y}(T)$ could be aligned to a state which has the maximum value of α at time T. This modified constraint allows the backtracking process to pick a state sequence which is shorter than S. The forced-alignment algorithm implemented using Eq. (2.6) and Eq. (2.7) is henceforth referred to as FA-1.

In order to examine the suitability of Eq. (2.6), the feature vectors corresponding to utterance of *"$ba < pau >$"* are force-aligned with the HMM state sequence corresponding to phones /b/, /aa/, /pau/, /s/ and /aa/. Fig. 2.2(a) displays the alpha matrix of this alignment. It should be noted that the dark band in Fig. 2.2(a) - the beam of alpha matrix - is not diagonal. Moreover at the last frame ($T = 109$), the last state ($N = 15$) does not have the highest value of α. Thus Eq. (2.4) will fail to obtain a state sequence appropriate to the aligned speech signal. From Fig. 2.2(b), we can observe that the HMM state 9 has the highest alpha value at the last frame, and Eq. (2.6) can be used to pick HMM state 9 automatically as the starting state of backtracking. Thus the use of Eq. (2.6) and Eq. (2.7) provides a

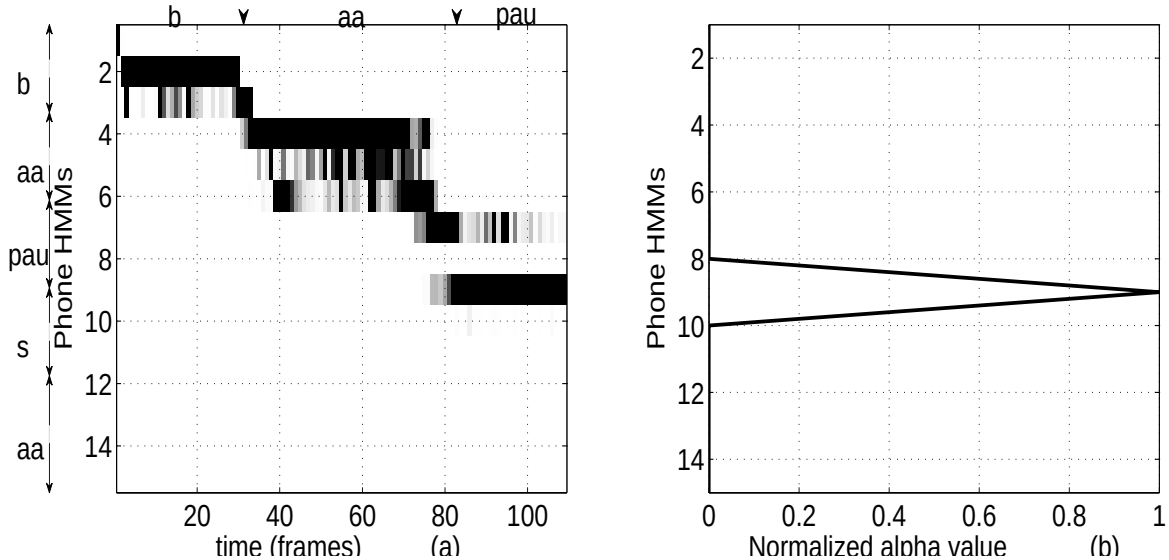


Figure 2.2: (a) An alpha matrix obtained for the alignment of feature vectors corresponding to utterance of *"ba < pau >"* with the HMM state sequence corresponding to phones /b/, /aa/, /pau/, /s/ and /aa/. The markers along the axis of time indicate manually labeled phone boundaries. (b) Alpha values of all states at the last frame ($T = 109$).

state sequence, which is shorter than the originally aligned state sequence, but has an appropriate match with the aligned speech signal.

2.2.2 Emission of a shorter observation sequence

When a given state sequence S emits a sequence $Y' \subset Y$, the backtracking algorithm can be modified as follows. Let $T' < T$ be the length of Y'. To obtain the value of T', the key is to observe the values of $\alpha_t(N)$ for all t. If $1 \leq t << T'$ then $\alpha_{T'}(N) < 1$, and as $t \rightarrow T'$ then $\alpha_t(N) \rightarrow 1$[4]. This property of state N could be exploited to determine the value of T'. Eq. (2.8) formally states the property of state N, and could be used to determine the value of T'.

$$\alpha_t(N) = \begin{cases} < 1 & 1 \leq t < T' \\ = 1 & t \geq T' \end{cases} \tag{2.8}$$

[4]From Eq. (2.2), it is trivial to observe that a state j achieves an alpha value of 1 at time t, only if it is highly likely to be observed at t. This is dictated by the terms $\max_i\{\alpha_{t-1}(i)a_{i,j}\}$ and $p(\boldsymbol{y}(t)|x(t) = j)$. The alpha value of state N being 1 at time T' implies that the state N is highly likely to be observed at T', and thus the length of observation sequence Y' is T'.

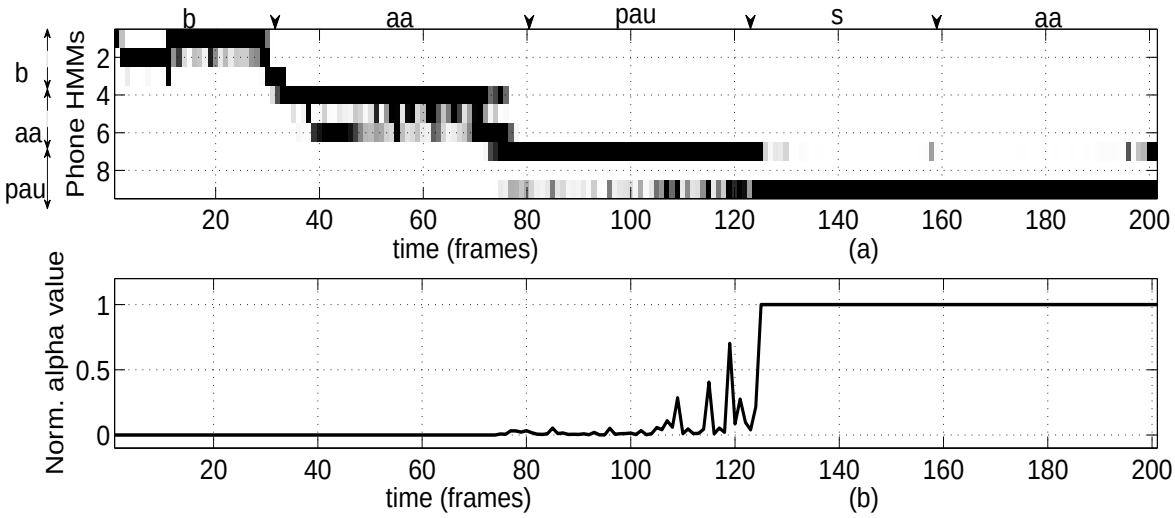


Figure 2.3: (a) An alpha matrix obtained for the alignment of feature vectors corresponding to utterance of "$ba < pau > sa$" with the HMM state sequence corresponding to phones /b/, /aa/ and /pau/. The markers along the axis of time indicate manually labeled phone boundaries. (b) Alpha values of the last state ($N = 9$) for all frames.

Given $T' < T$, the backtracking algorithm is modified as follows.

$$x(T') = N, \tag{2.9}$$

$$x(t) = \phi_{t+1}(x(t+1)),\ t = T' - 1, \ldots, 1. \tag{2.10}$$

Equation (2.9) presents the modified constraint that the last state N could be aligned to a feature vector at time $T' < T$, where T' denotes the length of Y' by satisfying the Eq. (2.8). The modified constraint in Eq. (2.9) allows the backtracking process to pick an observation sequence which is shorter than Y. The forced-alignment algorithm implemented using Eq. (2.9) and Eq. (2.10) is henceforth referred to as FA-2.

In order to examine the suitability of Eq. (2.9), the feature vectors corresponding to utterance of "$ba < pau > sa$" are force-aligned with the HMM state sequence corresponding to phones /b/, /aa/ and /pau/. Fig. 2.3(a) displays the alpha matrix of this alignment. From Fig. 2.3(b), it can be observed that the time instant at which the alpha value for the last state ($N = 9$) reaches 1 also denotes the length of shorter observation sequence ("$ba < pau >$"). Thus Eq. (2.9) and Eq. (2.10) can be used to pick a shorter observation sequence corresponding to the state sequence used for alignment.

2.3 Segmentation of long speech files

So far, we have discussed the modifications to the Viterbi algorithm to handle cases of $S' \subset S$ emitting Y and S emitting $Y' \subset Y$. In this section, these modifications are shown to be useful in processing long speech files. Two different methods to process long speech files are described below.

2.3.1 Segmentation using FA-1

In this method, the large speech file is sliced into chunks of d_b seconds. To begin with, the first chunk of speech is force-aligned with a sequence of words from the beginning of the text. The unknown variable here is the length of this sequence. To resolve this issue, we overestimate the length based on average speaking rate of three words per second [Picheny et al., 1986]. Thus a longer sequence of words is force-aligned with the chunk of speech. This leads to the case of $S' \subset S$ emitting Y, which could be handled by FA-1. The result of using FA-1 is the correct length of word sequence corresponding to the first chunk of speech. This process is repeated until the end of the long speech file. This method of segmenting a long speech file using FA-1 is henceforth referred to as SFA-1. The formal description of SFA-1 is as follows.

Let Φ denote an audio book and $\{w(1), \ldots, w(m), \ldots, w(M)\}$ denote the sequence of words present in Φ. Let $\{\boldsymbol{y}(1), \ldots, \boldsymbol{y}(t), \ldots, \boldsymbol{y}(T)\}$ be the sequence of T feature vectors extracted from Φ. Let n_f denote the number of frames in a chunk of d_b seconds of speech. The value of d_b is 30 seconds in this experiment, and the choice of this value is not critical. Let n_w denote the number of words in d_b seconds, estimated as $n_w = \eta * d_b * \lambda$, where $\eta = 3$ indicates a speaking rate of three words per second. The value of $\lambda = 1.5$ is chosen such that the estimate of n_w is *higher* than the actual number of words in d_b seconds of speech.

1. $m = 1$, $t = 1$.
2. Let $F = \{\boldsymbol{y}(t), \boldsymbol{y}(t+1) \ldots, \boldsymbol{y}(t+n_f)\}$ and let $W = \{w(m), w(m+1), \ldots, w(m+n_w)\}$. A sentence HMM representing W is constructed such that it introduces an *optional* silence model between two adjacent words. This optional silence HMM model helps to capture any pause regions inserted by the speaker between two adjacent words.
3. Force-align F with a sentence HMM of W using FA-1. Let $x(t), x(t+1), \ldots, x(t+n_f)$ be the state sequence obtained as a result of forced-alignment between F

and W. In FA-1, the observation vector $\boldsymbol{y}(t+n_f)$ is aligned to a state $x(t+n_f)$, which has the maximum alpha value at time $(t+n_f)$.

4. Note that the speech block F is an *ad hoc* block considered without any knowledge of pause/word/sentence boundary. The state $x(t+n_f)$ need not be an ending state of a word HMM and hence only an initial portion of state sequence is considered. Let δ be the minimum non-negative integer value ($\delta \geq 0$) such that $x(t+n_f-\delta)$ is an ending state of a word HMM in the vicinity of $x(t+n_f)$. Considering the state sequence $\{x(t), x(t+1), \ldots, x(t+n_f-\delta)\}$, the corresponding sequence of words $W' = \{w(m), w(m+1), \ldots, w(m+n'_w)\}$ is obtained, where $n'_w \leq n_w$. Starting from $w(m)$, a word $w(m+n''_w)$ is located such that there exists a pause of $150-200$ ms after the word $w(m+n''_w)$, where $n''_w < n'_w$. Let n''_f be the number of frames aligned with the word sequence $\{w(m), w(m+1), \ldots, w(m+n''_w)\}$.

5. $t = t+n''_f$, $m = m+n''_w$.

6. Repeat the steps 2-6 until the end of Φ.

2.3.2 Segmentation using FA-2

In this method, the text corresponding to large speech file is divided into paragraphs[5]. In an audio book, the text is naturally arranged in paragraphs. Each paragraph consists of one or more sentences, and usually deals with a single thought or topic or quotes a character's continuous words. Let Φ consists of a sequence of K paragraphs $\{u(1), \ldots, u(k), \ldots, u(K)\}$. The words in the first paragraph $u(1)$ are force-aligned with the first d_u seconds of speech data. As d_u is not known *a priori*, we overestimate its value. Thus a longer speech chunk is force-aligned with the words in $u(1)$. This leads to the situation of S emitting $Y' \subset Y$, which could be handled by FA-2. The result of FA-2 is the correct length of speech chunk corresponding to the words in the first paragraph $u(1)$. This process is repeated for the remaining paragraphs until the end of text. The method of segmenting a large audio file using FA-2 is henceforth referred to as SFA-2. The steps involved in SFA-2 are as follows.

1. $k = 1$, $t = 1$.

[5]The definition of a paragraph is not critical for this method. A simple grouping of words can also be used.

2. Let $U = [u(k), u(k+1)]$
3. A heuristic estimate of duration of U is defined as $d_u = n_p * d_p$, where n_p is the number of phones in utterance U and d_p denotes the duration of a phone. The value of d_p is 0.1 seconds. It is chosen such that the estimated value of d_u is *higher* than the actual duration of the utterance U. Let n_f denote the number of frames in d_u seconds and let $F = \{\boldsymbol{y}(t), \boldsymbol{y}(t+1) \ldots, \boldsymbol{y}(t+n_f)\}$ denote the sequence of feature vectors.
4. Force-align F with the sentence HMM representing U using FA-2. As a result of this forced-alignment, the shorter observation sequence $\{\boldsymbol{y}(t), \boldsymbol{y}(t+1) \ldots, \boldsymbol{y}(t+n'_f)\}$ emitted by U is obtained, where $n'_f < n_f$.
5. Given that U is force-aligned with a longer observation sequence, the ending portion of alignment may not be robust - for example, the silence HMM model at the end of U might observe a few observation vectors of next utterance $u(k+2)$, especially if $u(k+2)$ begins with a fricative sound. Hence the observation sequence $\{\boldsymbol{y}(t), \boldsymbol{y}(t+1) \ldots, \boldsymbol{y}(t+n''_f)\}$ corresponding to utterance $u(k)$ alone is considered, where $n''_f < n'_f$.
6. $t = t + n''_f$, $k = k + 1$.
7. Repeat steps 2-6 until $k = K$.
8. In order to obtain phone boundaries for the last utterance $u(K)$ perform forced-alignment of $u(K)$ with $\{\boldsymbol{y}(t), \boldsymbol{y}(t+1) \ldots, \boldsymbol{y}(T)\}$ using FA-0.

2.3.3 SFA-1 Vs SFA-2

While both SFA-1 and SFA-2 performs segmentation of long speech files, there are differences in the output of these methods. SFA-2 segments the long speech files into utterances corresponding to paragraphs in text. As discussed earlier, a paragraph could be one or more sentences expressing a single thought or character's continuous words. The definition of a paragraph is not critical here, but it is important to understand that utterances obtained from SFA-2 correspond to boundaries of grammatical units (sentences) and logical units (thoughts, character's turns etc.) as shown in Table 2.1. Such segmentation is useful for modeling prosody at sentence and paragraph level, especially in text-to-speech. In contrast, as shown in Table 2.1, SFA-1 segments the long speech file into chunks of 1-30 seconds. These chunks need not be complete sentences, hence many provide inaccurate representation

Table 2.1: Example utterances obtained from SFA-1 and SFA-2

Utterances obtained from SFA-1
1. I do not know what your, opinion may be. Mrs Weston, said Mr Knightley,
2. of this great intimacy, between Emma and Harriet Smith,
3. but I think it a bad thing,
4. A bad thing. Do
5. you really think it a bad thing,
6. why so. I think they will neither of them, do the other any good.
Utterances obtained from SFA-2
1. “I do not know what your opinion may be, Mrs. Weston,” said Mr. Knightley, “of this great intimacy between Emma and Harriet Smith, but I think it a bad thing.”
2. “A bad thing! Do you really think it a bad thing?–why so?”
3. “I think they will neither of them do the other any good.”

of sentence boundaries and the corresponding prosodic boundaries. Thus it is preferred to use SFA-2 for text-to-speech, as it provides paragraph length utterances.

2.4 Evaluation

To evaluate SFA-1 and SFA-2, we have used the speech databases of *RMS* from the *CMU ARCTIC* [Kominek and Black, 2004a] and *EMMA* from Librivox [LibriVox, 2010]. The *RMS* speech database consists of 1132 utterances corresponding to 1132 paragraphs in text. Here each paragraph contains only one sentence. For our experiments, 1132 utterances were concatenated to create an artificial large speech file, henceforth referred to as Φ_r. The duration of Φ_r is 66 minutes. It could be argued that an artificial long speech file may not represent a distribution of pauses in an authentic long speech file. To enable such comparison, around 45 minutes of speech data - corresponding to first three chapters of *EMMA* in Librivox - was hand labeled to mark the beginning and ending of sentences. This database is referred to as Φ_e. Segmentation of long speech files can be evaluated in the following ways.

- *Boundaries of utterances:* The utterance boundaries obtained automatically from long speech files can be compared with hand labeled utterance boundaries in EMMA (Φ_e) or known utterance boundaries in RMS (Φ_r). This evaluation methodology is referred to as $E1$.

- *Phone boundaries:* The phone boundaries obtained automatically from long speech files can be compared with phone boundaries obtained from FA-0 (forced-alignment of utterances with their paragraph-length text). This evaluation methodology is referred to as $E2$.

- *Mel-cepstral distortion (MCD):* We can use the utterances obtained from segmentation of long speech files in a voice building process such as CLUSTERGEN, and evaluate the quality of the resulting synthetic voice. CLUSTERGEN is a statistical parametric synthesis engine [Black, 2006]. As a part of the voice building process, CLUSTERGEN takes the utterances and the corresponding text, and derives HMM state level boundaries using FA-0. Hence, this is an another type of evaluation for utterance boundaries. Depending on whether the utterance boundaries have errors or not, we hope to obtain a good/poor quality voice. Typically, errors in utterance boundaries lead to misalignment of utterances with its corresponding text. This results in a poor quality synthetic voice.

  The quality of a synthetic voice can be measured using Mel-cepstral distortion as follows. The utterances given to the voice building process are divided into mutually exclusive sets - referred to as training set and held-out set. A TTS voice is built from the utterances of the training set. This TTS voice is used to synthesize utterances of the held-out set from their corresponding text. The MCEPs of the synthesized version of an utterance is compared with the MCEPs of the corresponding natural version, and a Mel-cepstral distortion (MCD) is computed as below.

$$\text{MCD} = (10/\ln(10)) * \sqrt{2 * \sum_{l=1}^{25} (c_l^s - c_l^o)^2} \tag{2.11}$$

  where c_l^s and c_l^o denotes the l^{th} coefficient of the synthesized and the original utterances, respectively. This method is referred to as $E3$.

It is important to note that SFA-2 segments the long speech files into utterances corresponding to paragraphs in text, and hence it could be evaluated for $E1$, $E2$ and $E3$. SFA-1 segments the long speech files into chunks of 1-30 seconds long. As discussed in Section 2.3.3, the chunks obtained from SFA-1 need not correspond to paragraphs in text. This makes it difficult to compare the chunk boundaries with the hand labeled boundaries of utterances in long speech files. Hence SFA-1 does not permit its chunks for $E1$ and $E2$ evaluations. However, evaluation of SFA-1 can be done using $E3$.

Table 2.2: Evaluation of SFA-2 on RMS and EMMA voice. $E1$ measures the difference between utterance boundaries obtained automatically and the hand labeled/known utterance boundaries in the long speech file. $E2$ measures the difference between phone boundaries obtained automatically from long speech files and the phone boundaries obtained from FA-0. All values are measured in milliseconds.

	$E1$ (μ, σ)	$E2$ (μ, σ)
Φ_r	(35, 21)	(35, 22)
Φ_e	(138, 88)	(20, 52)

Table 2.3: MCD scores of different voices from *EMMA* Φ_e and *RMS* Φ_r.

		FA-0	SFA-1	SFA-2
Φ_r	MCD	5.27	5.30	5.29
	# train	1019 (59 min)	1046 (59 min)	1019 (59 min)
	# test	113 (7 min)	116 (7 min)	113 (7 min)
Φ_e	MCD	4.54	4.48	4.51
	# train	89 (42 min)	479 (42 min)	89 (42 min)
	# test	9 (4 min)	53 (4 min)	9 (4 min)

2.4.1 Results

Table 2.2 shows $E1$ and $E2$ evaluation results of SFA-2 on the *RMS* and *EMMA* speech databases. $E1$ measures the difference between utterance boundaries obtained automatically and the hand labeled/known utterance boundaries in the long speech file. The mean value of this difference is less than 140 milliseconds - suggesting that utterance boundaries obtained automatically match well with the known/hand labeled boundaries of the utterances in the *RMS* and *EMMA* speech databases. The values of $E1$ for *EMMA* indicate that mean and variance of difference in boundary locations is high for naturally long speech files than *RMS*.

$E2$ measures the difference between phone boundaries of SFA-2 and FA-0. These values shown in Table 2.2 indicate a reasonable agreement of SFA-2 with FA-0 on phone boundaries.

Let V_r^0, V_r^1 and V_r^2 denote the TTS voices built from Φ_r using FA-0, SFA-1 and SFA-2 respectively. Let V_e^0, V_e^1 and V_e^2 denote the TTS voices built from Φ_e using

FA-0[6], SFA-1 and SFA-2 respectively. Table 2.3 shows the MCD values for the three voices built from *EMMA* and *RMS*. It can be observed that MCD scores of SFA-1 and SFA-2 are similar to FA-0 of their respective voices. The evaluation results of $E1$, $E2$ and $E3$ suggest that the utterances obtained from SFA-1/SFA-2 are useful in building synthetic voices.

2.5 Summary

In this chapter, we have proposed modifications to the Viterbi algorithm and showed that the proposed modifications could be employed to segment a long speech file. Thus it alleviates the need of a large vocabulary speech recognition system (and the corresponding algorithms) for segmenting a long speech file. More importantly, the methods SFA-1 and SFA-2 enable the forced-alignment algorithm to be used for low resource languages, when a large vocabulary speech recognition system is not readily available.

In SFA-1, a large audio file is processed in 30-second chunks of speech data. Hence, the text transcription obtained for each of these blocks from modified forced-alignment need not be complete. For example, "The girl faced him." and "Her eyes shining with sudden fear." are two utterances obtained using SFA-1. These two utterances belong to a single sentence "The girl faced him, her eyes shining with sudden fear". Thus the processing of a long speech file in terms of 30-second chunks of speech data creates artificial sentence boundaries. This could lead to misrepresentation of utterance boundaries as well as prosodic characteristics of an utterance. In SFA-2, the audio file is segmented into utterances corresponding to paragraphs in text. The audio segment obtained for each utterance as a result of modified forced-alignment retain paragraph level information and the corresponding prosodic characteristics. While both SFA-1 and SFA-2 could be used for segmentation of long speech files, SFA-2 may be more suited for text-to-speech due to its property of segmenting a long speech file into paragraph-length utterances.

[6]If the utterance boundaries are hand labeled/known, the long speech files can be segmented at the known boundaries. This enables to apply FA-0 on the resulting utterances.

Chapter 3

Building voices from audio books

In the previous chapter, we have proposed modifications to the Viterbi algorithm and developed methods for segmentation of long speech files. In this chapter, we apply these methods for segmentation of long speech files in several audio books collected from the public domain for different speakers. We show that utterances obtained from large speech files can be used to build synthetic voices.

3.1 Audio books in public domain

An audio book is a spoken form of a written/printed book. The spoken form of a printed book could be abridged or unabridged. Abridged audio books have some text deleted or added, thus provide a spoken summary or paraphrasing of a printed book. Unabridged audio books are verbatim readings of a printed book, where there is no mismatch between the text and speech. It is these unabridged audio books that are of interest to us in this thesis.

Audio books are available in digital media such as CDs and are also available for download from the Internet. Our interest is in audio books available in the public domain of the Internet. The concept of public domain means that anyone can use the audio books however they wish, without any restrictions. A detailed discussion on the public domain and the associated restrictions, intellectual property rights and copyright can be found at [Wikipedia, 2010], [LibriVox, 2009].

Portals such as librivox.org, loudlit.org and audio books.org provide audio books in the public domain on the Internet. Of all the sites known to us, librivox.org

seems to provide the largest collection of audio books. As of April 2010, Librivox provides audio books in about 40 languages, with a majority of them in English. The portal has around 10000 books in English, 726 books in German, 516 books in Chinese, and 388 books in French. At Librivox, volunteers record chapters of the books in the public domain, and make these speech files freely available to the world. Books in the public domain are chosen as they are not covered by copyright and can be used by anyone without restrictions. However, it should be noted that the copyright law varies across countries and hence the copyright status of audio books too. The English books in Librivox are mostly chosen from the Gutenberg project (www.gutenberg.org) which has a catalog of public domain e-books (i.e., where the text is available in electronic form) according to U.S.A. copyright laws. Any book published in U.S.A. before 1923 is considered as public domain in U.S.A.

The audio books are typically recorded through Audacity, a free audio editor and recorder available for GNU/Linux, BSD, Mac OS, and Windows operating systems. The speech files are recorded at 128 Kbps in MP3 format, and are converted into 64 Kbps MP3 and ogg vorbis format. These formats are chosen primarily due to the available tools with Internet archive (www.archive.org) which act as the hosting platform for Librivox. Thus the best quality that is available from audio books in Librivox is from 128 Kbps in MP3 - which is a lossy compression format. Hence we based on our work on the 128 Kbps MP3 files to avoid any further degradation that can occur due to format conversion. However, it would be preferable to have speech data stored in a lossless format.

An important issue to be considered while using the audio books from the public domain is the quality of the text and the speech data. In the case of Librivox, most of the text is obtained from the Gutenberg project which has an accuracy target of 99% for its text. Hence, there are bound to be some errors (atleast 1%) in the text. Also, the persons recording these audio books are volunteers, and it is likely that they introduce spoken errors such as repeated text, additional text, mispronunciation etc. Librivox claims that while there are no specific standards set for recording, the recordings are checked by moderators. However, in our experience in using the EMMA corpus from Librivox, we found that additional text exists at the beginning of each chapter - where the speaker mentions that it is a Librivox recording, and his/her name - and also at the end of chapter[1]. We also found deletions, i.e.,

[1]In this thesis, the additional text at the beginning and ending of each chapter was added manually. Automatic detection of insertions, deletions and substitutions of text is an important issue in processing the audio books. Such automatic detection is beyond the scope of this thesis.

the speaker did not read a paragraph in one of the chapters[2]. Apart from these aberrations, we found the text and speech data to be of good quality (see Section 3.2 for more details).

Another issue that often arises is the voice quality of the volunteers, i.e., whether they sound good, accented, pleasant etc. As discussed earlier, Librivox welcomes all volunteers and has no process to test a volunteer voice for pleasantness, articulation, etc. Thus it is possible to have a variety of voice sources in these audio books. There is also no guarantee that the entire audio book is recorded by a single speaker, which is one of the requirements in building a synthetic voice. In the context of building synthetic voices, several of the these issues have to be considered while selecting a voice and an audio book from the public domain recordings.

In spite of these issues, these audio books act as excellent candidates for building synthetic voices due to the following reasons.

- Audio books encapsulate rich prosody including intonation contours, pitch accents and phrasing patterns.
- A large amount of good quality speech data from a single speaker is made available.
- Most importantly, the audio books provide an effortless and costless way of obtaining large amounts of good quality speech data.
- These audio books may also be useful for building speech recognition systems, and to analyze the voice characteristics of a speaker which has applications in speaker recognition and voice conversion.

3.2 High vs poor quality audio books

It is important to note that the scope of the thesis is limited to deal with high quality audio books. The typical characteristics of a high quality audio book are low rate of disfluency, quiet recording environment and a normal rate of speech which are further discussed in detail.

[2]This was detected during a random manual check on beginning and ending of utterance boundaries obtained from automatic segmentation of monologues.

Table 3.1: Examples of disfluencies found in the audio book of Walden.

Written	Spoken	Type of Error
before we judge of him	before we judges him	subs., del.
"*Thou dost .. worries were*".	*Quote.Thou dost .. worries were. End-quote*.	ins.

3.2.1 Rate of disfluency

It is well known that a spoken conversation or spontaneous speech contains disfluencies in form of repetitions ("the the"), repairs ("any health cov - any health insurance"), filled pauses ("uh, um") and false starts ("is uh did John abandon you?"). The rate of such disfluencies could be as high as 5-10% in conversational speech [Shriberg, 1999].

The audio books fall into the category of read speech, and it is typically assumed that read speech has no disfluencies. However, in our experience we have found that audio books from Librivox do have disfluencies in the form of insertions, deletions and substitutions. Table 3.1 shows an example of insertion, deletion and substitution found in the audio book of *WALDEN*. However, the disfluency rate empirically found[3] in these audio books is lower than 0.1%, i.e., a disfluency rate of 1 in 1000 words, which is significantly lower than the disfluency rate of 5-10% found in the conversational speech.

It should be noted it is possible to have audio books with a higher rate of disfluency. A typical scenario include a story with foreign names and places. For example, the story, *Trojan War* has been translated to Telugu, and the native speaker of Telugu had great difficulty in uttering foreign names and places in the story. Such audio books contain many disfluencies due to the complex vocabulary of the script. As a result, these audio books have text which differ significantly from the spoken form due to disfluencies. We refer to such audio books as poor quality audio books, and these books pose significant challenges to forced-alignment as well as speech recognition algorithms.

3.2.2 Recording environment

The recording environment also plays a major role in defining the quality of an audio book. A high quality audio book would have recordings done in a quiet

[3]This was manually observed on two chapters of *WALDEN* and *EMMA* picked randomly.

environment with a good microphone. It is also assumed that the quite environment would not carry any reverberation, noisy sounds such as door knocks, page flips, cell phone rings etc.

Signal-to-Noise Ratio (SNR) is one of the measures typically used to indicate the quality of an audio. The Librivox books seem to maintain a fairly good quality recordings. The SNR values computed for a sample of four audio books in Librivox was found to be 30, 32, 32 and 18 dB. These values could be benchmarked against SNR values for TIMIT and CMU ARCTIC datasets which were found to be 41 and 40 dB respectively. It should be noted that the Librivox recordings are done in a quiet environment, without any special echo treatment done to the recording environment. The TIMIT and CMU ARCTIC datasets are studio quality recordings done using a noise cancellation microphone in a specially designed chamber where the door and the walls are specially treated for echo cancellation.

3.2.3 Rate of speech

Another important factor is the rate of speech which could be defined as the number of syllables per second. The average duration of a syllable is around 250-300 ms, and hence a rate of 3-4 syllables per second is typically considered as normal speaking rate. A faster speaking rate often manifests as sloppy speech, where the articulators of the speech production mechanism may not reach their target position completely in the process of uttering more number of syllables in a second. A slower speaking rate leads to elongation of duration of syllables, and hence may not sound natural. The average speaking rate computed for a sample of four audio books was found to vary from 3.5 to 4 syllables per second.

3.3 INTERSLICE

Building synthetic voices from audio books is a nontrivial issue. Audio books have monologues which are long speech files, and their segmentation is not easily supported by existing tools in FestVox. FestVox is a suite of tools which allows building synthetic voices in multiple languages [Black and Lenzo, 2009]. However, there is no support in FestVox for building synthetic voices using long speech files. In order to facilitate such a voice building framework within FestVox, we have built a set of modules to handle long speech files, and these set of modules are grouped in a package referred to as INTERSLICE. Fig. 3.1 depicts the features of

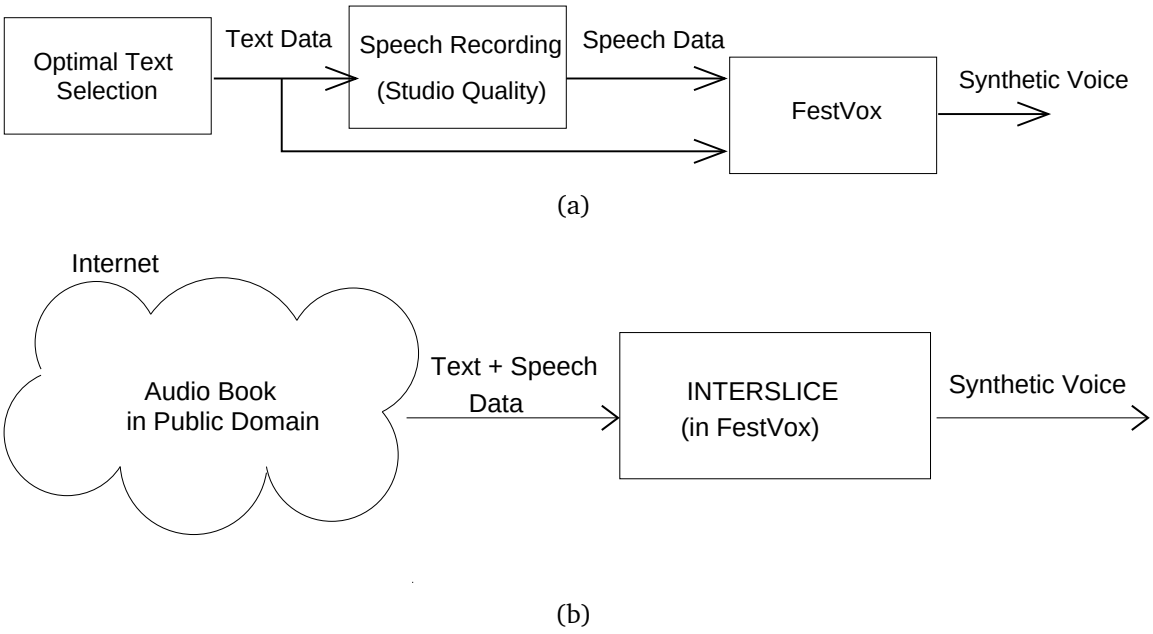


Figure 3.1: (a) Standard build process of TTS. (b) Build process of TTS using audio books and INTERSLICE.

INTERSLICE in the FestVox suite. To build a voice, the traditional steps involves optimal selection of sentences, and recording these sentences in a studio. Such a process is not only costly and laborious but ends up in single sentence recordings. Contrary to the traditional method of collecting such speech data, the audio books provide an alternative by avoiding the selection of sentences and the recording time. Given an audio book and the corresponding transcription, the INTERSLICE tool can be used to segment it and a synthetic voice could be built within the FestVox framework. Thus the use of audio books and the INTERSLICE package in FestVox provide a platform for rapid building of synthetic voices in multiple languages.

3.3.1 Supported languages and acoustic models

To segment long speech files, the INTERSLICE package supports both SFA-1 and SFA-2 methods (see Section 2.3.1 and Section 2.3.2 for details). Both methods require context-independent (CI) phone level HMM models, and also a mechanism of obtaining phone sequences from the given text. The phone level HMM models and grapheme-to-phoneme conversion are language specific resources. The INTERSLICE package currently supports English (US phoneset) and Telugu (one of the Indian

languages). However, the framework is generic enough to extend it to other languages by obtaining a grapheme-to-phoneme conversion and CI acoustic models.

The acoustic models that support English (US phone set) are obtained from four hours of speech data collected from four CMU ARCTIC speakers (see Appendix B). The acoustic models supporting Telugu language are obtained from CMU-NK speech database. This database consists of two hours of speech data recorded by a female speaker.

Pronunciation dictionary

For US English, the grapheme-to-phoneme conversion available in Festival is used to obtain the phone sequences for a given text. Telugu is a language which has almost one to one correspondence between what is written and what is spoken. For example, the word *"katha"* can be broken into sequence of phones *"/k/ /a/ /th/ /a/"*. Hence, the grapheme-to-phoneme conversion for Telugu is almost straightforward and is implemented using a set of rules.

3.4 Application of INTERSLICE for building voices

The process of building synthetic voices from audio books using INTERSLICE and FestVox is as follows.

- Audio books consist of monologues which are long speech files, and the corresponding text arranged in paragraphs. Each paragraph consists of one or more sentences, and typically deals with a single thought or topic or quotes a characters continuous words.

- Use INTERSLICE for segmentation of long speech files in audio books. It should be noted that INTERSLICE supports both SFA-1 and SFA-2 methods of segmentation (see Section 2.3 for details). SFA-2 segments the long speech files into utterances corresponding to paragraphs in text. SFA-1 segments the long speech file into chunks of 30 seconds. The text corresponding to these chunks may or may not be complete sentences. Chunks which are incomplete provide inaccurate representation of sentence boundaries and the corresponding prosody. Thus it is preferred to use SFA-2, as it provides utterances corresponding to paragraphs in text.

- The output of both the SFA methods is the beginning and ending boundaries of utterances as well as the phone boundaries in these utterances. The acoustic models used in INTERSLICE to obtain these phone boundaries are speaker-independent. It is known that speaker-dependent acoustic models are better than speaker-independent to obtain phone boundaries [Angelini et al., 1997]. Hence, only the utterance boundaries from the output of INTERSLICE is considered. This information is used to slice the long speech files into utterances corresponding to paragraphs in text.

- Given the utterances and the corresponding text, the CLUSTERGEN engine in FestVox builds a synthetic voice. An important step in building a CLUSTERGEN voice is to obtain phone and HMM state level boundaries in the utterances [Black, 2006]. This is accomplished by using flat-start initialization in Baum-Welch training of speaker-dependent HMMs [Prahallad et al., 2006]. A step-by-step process of building a CLUSTERGEN voice is described in Appendix C.

3.4.1 Voice from the audio book of EMMA

As an initial experiment, we collected the recordings of *EMMA*. These recordings are done by a female speaker. The name of this speaker in Librivox forum is Sherry, while the catalog name is Sherry Crowther. All the recordings of Sherry obtained from the audio book of *EMMA* were concatenated to form a large speech file, henceforth referred to as Φ_e, whose duration is 17.35 hours. We downloaded the associated text from the Project Gutenberg, and added text at the beginning and end of each chapter to match the introductions and closings made by the speaker. The text was arranged[4] into 2693 paragraphs.

Both SFA-1 and SFA-2 were applied on Φ_e, and CLUSTERGEN voices were built [Black, 2006]. Let V_e^1, V_e^2 denote the TTS voices built from Φ_e using SFA-1 and SFA-2 respectively. Table 3.2 shows the Mel-cepstral distortion (MCD) scores obtained on TTS voices of V_e^1 and V_e^2. MCD is an objective measure to calibrate quality of synthetic voices, and it is empirically observed that studio quality recordings such as CMU ARCTIC have MCD scores in the range of 4-7 [Black, 2006],[Kominek et al.,

[4]The e-text of audio books taken from the project Gutenberg has one or more line spacings between two paragraphs. This was conveniently exploited in this thesis as paragraph markers. However, often, due to text being archaic or style of the author to express a character's thoughts aloud, some paragraphs are extremely long - up to 100 sentences or more. Hence, a preprocessing was done on large paragraphs to break them into smaller ones - restricting the size to be around 250 words.

Table 3.2: MCD scores obtained on TTS voices of *EMMA* (V_e^1, V_e^2).

	MCD	# utts. (train)	# utts. (held-out)
V_e^1	5.09	13757 (15.57 hrs)	1528 (1.74 hrs)
V_e^2	5.04	2424 (15.67 hrs)	269 (1.67 hrs)

Table 3.3: DND listening tests on V_e^1 and V_e^2

	diff	no-diff
V_e^1 vs V_e^2	17/50	33/50

2008]. Thus the MCD scores obtained on V_e^1/V_e^2 indicate that the methods SFA-1 and SFA-2 could be applied for segmentation of large speech files such as *EMMA* corpus, whose resultant is useful for building synthetic voices.

Table 3.3 shows the results of listening tests conducted on V_e^1 and V_e^2. A set of five speakers (henceforth referred to as subjects) participated in the listening test. A set of 10 utterances were synthesized from these two voices. Each subject was asked to listen to an utterance synthesized by V_e^2 and compare it against the utterance of same text synthesized by V_e^1. The subject was asked whether there is a difference or no-difference in the pair of utterances. We henceforth refer to this listening test as DND (difference-no-difference) test. The results indicate that in 33 out of 50 utterances, the subjects did not perceive any difference between the voices V_e^1 and V_e^2. The subjects perceived a difference between V_e^1 and V_e^2 in 17 out of 50 utterances. A subset of these utterances was manually analyzed. It was found that there are variations in the durations of utterances from V_e^1 and V_e^2. These variations are in the order of 300-500 milliseconds for 30-second long utterances. It should be noted that CLUSTERGEN predicts duration of phones based on contextual features, and this prediction is learnt based on duration of phone boundaries observed in the training set. Given that the voices being examined here have different duration models, it is difficult to pinpoint reasons for variations in predicted durations of phones. Other than these minor variations in durations, we did not perceive any difference in the spectral quality of the voices.

3.4.2 More voices from audio books of Librivox

To demonstrate the usefulness of INTERSLICE on a larger number of audio books, we have selected three more audio books from Librivox. The details of these

Table 3.4: Details of the audio books used to build voices. Here forum name and catalog name refers to the speaker who has recorded the audio book

Title	Author	Forum Name	Catalog Name	Gender	# hours
Emma	Jane Austen	Sherry	Sherry Crowther	Female	17.35
Pride and Prejudice	Jane Austen	LibraryLady	Annie Coleman Rothenberg	Female	18
Walden	Henry David Thoreau	GordMackenzie	Gord Mackenzie	Male	18
Sense and Sensibility	Jane Austen	Kaffen	Mark F. Smith	Male	18

Table 3.5: MCD scores obtained on TTS voices for EMMA (V_e^2), Pride and Prejudice (V_p^2), Walden (V_w^2) and Sense and Sensibility (V_s^2). Here the upper script ^{2} indicates the use of SFA-2 method to segment the large speech files.

	Gender	MCD	# utts. (train)	# utts. (held-out)
V_e^2	F	5.04	2424 (15.67 hrs)	269 (1.67 hrs)
V_p^2	F	6.02	2218 (11.99 hrs)	246 (1.41 hrs)
V_w^2	M	4.96	1134 (12.84 hrs)	126 (1.46 hrs)
V_s^2	M	5.12	2087 (12.17 hrs)	232 (1.30 hrs)

audio books are shown in Table 3.4. Let Φ_p, Φ_w, Φ_s denote the audio books of "Pride and Prejudice", "Walden", "Sense and Sensibility" read by "LibraryLady", "GordMackenzie" and "Kaffen" respectively. On these audio books, INTERSLICE was used to segment large speech files using the SFA-2 method. Let V_p, V_w, V_s denote the CLUSTERGEN voices built from Φ_p, Φ_w and Φ_s respectively. Table 3.5 shows that the MCD values obtained for V_p, V_w and V_s are in the acceptable range of 5-7. This experiment demonstrates that the algorithms used in INTERSLICE are directly applicable to several audio books without any modifications.

3.4.3 Voice from a Telugu audio book

The framework of INTERSLICE can be extended to new languages fairly quickly. To add a new language, INTERSLICE requires a set of context-independent acoustic models and a pronunciation dictionary or letter-to-sound rules. For example, to build a voice from a Telugu audio book, we chose acoustic models built from the CMU-NK speech database corresponding to 2 hours of speech. The script in Telugu has good correspondence with the sounds, and hence a set of simple letter-to-sound

rules were used to obtain phone sequence. The audio book in Telugu consisted of multiple short stories from a popular children's magazine *Chandamama*. The book was read by a professional speaker in a studio, and the duration of this book is around 2 hours. Let this audio book be referred to as Φ_l. Using INTERSLICE, SFA-2 algorithm was applied to segment the long speech file Φ_l. The acoustic models built from CMU-NK voice were used in this process. A synthetic voice (V_l), was built using CLUSTERGEN framework, and the MCD score for this voice was found to be 5.76.

3.5 Summary

In this chapter, we described the process of building synthetic voices from audio books taken from public domain recordings. We explained the INTERSLICE package which has been developed specifically to deal with long speech files found in audio books. INTERSLICE is integrated within FestVox suite of tools, and is available for download along with FestVox. INTERSLICE provides both SFA-1 and SFA-2 methods to segment long speech files. Subsequently, tools such as CLUSTERGEN and CLUNITS available in Festvox can be used to build a statistical parametric voice or a unit selection voice respectively. It should be noted that INTERSLICE assumes that the audio books are of high quality, i.e., lower rate of disfluency, a normal speaking rate and a noiseless environment in recording the audio book. To demonstrate the usefulness of INTERSLICE, we have built four voices from four audio books (two female and two male) from Librivox and a Telugu voice from a Telugu audio book.

Chapter 4

Speaker-specific phrase breaks

In previous chapters, we have shown that audio books could be exploited to build synthetic voices. A major motivation to use audio books is to incorporate natural prosody in synthetic voices by building better prosodic models. Prosody of speech involves variation in intonation, duration, loudness and formant frequencies of speech sounds. Prosodic units (also referred to as prosodic phrases) are characterized by several acoustic cues including a coherent intonation contour. At the boundary between prosodic units, it is known that the pitch resets and the duration of a rhyme is longer [Taylor, 2009]. To incorporate such duration and intonation changes in synthetic speech, it is important to model phrasing patterns, i.e., the location of prosodic phrase breaks in utterances. In this chapter, we focus on building a prosodic phrase break annotator, and study its effect on the quality of synthetic speech.

4.1 Prosodic phrase breaks

Prosodic phrasing is a mechanism by which a speaker groups words within an utterance. Appropriate grouping helps in better comprehension and naturalness of an utterance [Frazier et al., 2006]. Prosodic phrase boundaries (also referred to as prosodic phrase breaks) are manifested in the speech signal in the form of pauses as well as relative changes in the intonation and duration of syllables. Tone and break Indices (ToBI) model of intonation [Silverman et al., 1992], analyzes prosody using two pitch targets, high and low. According to ToBI, prosodic phrases have pitch accents - where pitch target is relatively high - indicating the prominence of

certain syllables in a phrase. The pitch target at the boundary of prosodic phrases is treated separately from a pitch accent. In ToBI model, there are five levels of breaks found in utterances. The five levels are:

- '0' - tighter connection than for a default word boundary, e.g., the medial affricate in contractions of "did you" or a flap as in "got it"
- '1' - normal word break
- '2' - break marking a lower-level perceived grouping of words that does not have an intonational boundary marker
- '3' - intermediate phrase boundary, cued by some pre-boundary lengthening and a phrase tone
- '4' - intonational phrase boundary, cued by more pre-boundary lengthening and a full boundary tone.

The break indices 3 and 4 must contain at least one pitch accent, and correspond to prosodic phrase boundaries. Studies have also shown that acoustic cues - such as pre-pausal lengthening of rhyme, speaking rate, breaths, boundary tones and glottization - play a role in indicating the phrase breaks in a speech signal [Wightman et al., 1992], [Redi and Shattuck-Hufnagel, 2001], [Kim et al., 2006].

In order to illustrate the complex nature of acoustic cues that characterize prosodic phrase breaks, we conducted a listening experiment using a set of one or two paragraph length utterances. This set of five utterances were part of a story (Chapter 2 of *EMMA*) recorded by a female speaker in the Librivox database [LibriVox, 2010]. The story was recorded in a story telling fashion with pauses wherever required. The original recordings of these utterances are referred to as set-A. From these recordings, pauses within the utterances were removed manually. These pause-clipped utterances are referred to as set-B.

A set of 5 non-native speakers of English acted as listening subjects in this experiment. On the first day, the subjects were asked to listen to the utterances from set-B. They were given the text of these utterances - with punctuations removed and all letters converted to lower case, and were asked to provide a punctuation mark wherever they perceived a break in acoustic signal. On the second day, the same five subjects were asked to listen to the utterances from set-A, and were asked to mark a punctuation in text, wherever they perceived a break.

A sample utterance used in this experiment is: *" Sorrow came (75:5/5) – a gentle sorrow (370:5/5) – but not at all in the shape of any disagreeable consciousness*

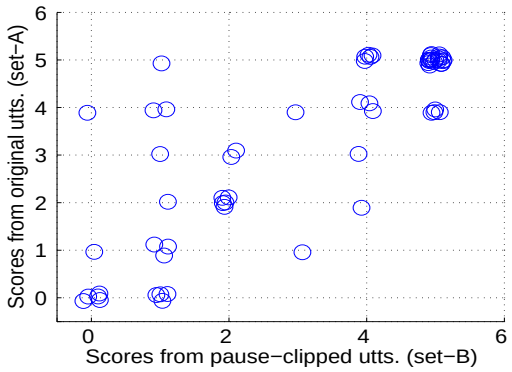


Figure 4.1: Scatter plot of scores obtained for utterances in Set-A and Set-B

(550:4/5). Miss Taylor married (640:5/5). It was Miss Taylor's loss which first brought grief (550:5/5). It was on the wedding-day of this beloved friend that Emma first sat in mournful thought of any continuance (1290:5/5).... " . Sets of three numbers printed at different locations in this example denote -

- duration of the pause (in milliseconds) as found in the original recordings,
- number of subjects perceiving a break at this location i while listening to this utterance from set-B, which is denoted by s_i^B, and
- number of subjects perceiving a break at this location while listening to this utterance from set-A, which is denoted by s_i^A.

The value of the pair (s_i^B, s_i^A) range from $(0, 0)$ to $(5, 5)$. In total there were 63 locations in all of the five utterances, where subjects perceived a break.

A scatter plot of the pair of values (s_i^A, s_i^B), where $0 \leq s_i^A \leq 5$, $0 \leq s_i^B \leq 5$, and $i = 1..63$ is shown in Fig. 4.1. The values of (s_i^A, s_i^B) are referred to as scores in Fig. 4.1. The scatter plot demonstrates a correlation of 0.82 between the values of s_i^B and s_i^A. Further analysis showed that -

- in 92% of the total 63 locations, at least one subject from set-B and one from set-A agreed on perceiving a break and
- in 33.3% of the locations, all the five subjects from set-A and five from set-B agreed on perceiving a break.

Table 4.1: Syllable level features extracted at phrase break

Break Features	Description
pause duration	Duration of the pause at the word boundary
vowel duration	Vowel duration in the syllable
f0_maxavg_diff	Diff. of max and avg f0
f0_range	Diff. of max and min f0
f0_avgmin_diff	Diff. of avg and min f0
f0_avgutt_diff	Diff. of syl avg and utterance avg f0
en_maxavg_diff	Diff. of max and avg energy
en_range	Diff. of max and min energy
en_avgmin_diff	Diff. of avg and min energy
en_avgutt_diff	Diff. of syl avg and utterance avg energy

The overall correlation of 0.82 between the values of s_i^B and s_i^A indicate that acoustic cues other than simple pause play a major role in indicating a phrase break in the speech signal.

This experiment shows that acoustic cues other than pauses play a role in indicating prosodic phrase breaks. However, the representation and parameterization of these complex acoustic cues is not well understood [Taylor, 2009]. Hence, many of these complex acoustic cues are often represented by simpler features such as average duration, F0 and energy values as shown in Table 4.1 [Ananthakrishnan and Narayanan, 2008].

4.1.1 Syntactic vs prosodic phrase breaks

A question that often arises is whether there is any relation between prosodic phrase breaks and the syntactic structure of an utterance. Consider the example, *"John and Mary were running very quickly"*. The major syntactic constituents of this sentence are a noun phrase (NP) and a verb phrase (VP) as shown below. The syntactic break for this sentence is between NP and VP.

(NP (John and Mary)) (VP (were running very quickly)))

For the same sentence, below are examples indicating three different prosodic phrase breaks. Here '|' denotes a prosodic phrase break.

John and Mary were running very quickly.

John and Mary | were running very quickly.

John and Mary were running | very quickly.

In the first example, there is no prosodic break. In the second, the prosodic break is before the phrase *"were running very quickly"* and coincides with the syntactic break. In the third, the prosodic break is before the phrase *"very quickly"*.

From these examples, it is easy to notice that there is some correspondence between the prosodic phrase breaks and syntactic phrase breaks of an utterance. While it is known that there is a relationship between syntax and prosody, but the relationship is not well understood or formally defined. There is no general theory which explains the correspondence between syntax and prosody [Bachenko and Fitzpatrick, 1990], [Taylor, 2009].

In the context of TTS, it is essential to predict prosodic phrase breaks in the text. Prosodic phrase breaks predicted from the text are used by different modules such as F0 generation, duration and insertion of pauses. Modeling prosodic phrase patterns involves building a prosodic phrase break annotator (PBA) and a prosodic phrase break predictor (PBP). A PBA model annotates text/speech data with the location of prosodic phrase breaks. Often human operators act as PBAs - the annotation is done by listening to speech data. This could also be achieved by machine learning techniques, which use acoustic cues to locate prosodic phrase breaks. A PBP model predicts prosodic phrase breaks in the given text based on either a set of rules or machine learning techniques. The output of PBA - text data annotated with prosodic phrase breaks - is used to train a PBP model. Features related to syntactic level or part-of-speech sequence are extracted from text, and a machine learning model is built to predict a break or not-a-break between words.

Current techniques model phrasing patterns have the following limitations –

- A human annotator is used to annotate text with a break symbol between words which are perceived as being phrase breaks. This process of hand annotation is laborious, time consuming and is not scalable to multiple languages.
- Typically, a PBP model is trained on a standard corpus. For example, in Festival, a default PBP model for English is trained on Boston University Radio News corpus data and employed to predict breaks for all English voices. Thus the same prosodic phrasing pattern is used for all voices ignoring speaker-specific phrasing patterns.
- A PBP model assumes availability of syntactic parsers and/or part-of-speech taggers. The availability of such linguistic resources may be difficult for

minority or resource poor languages. Such situations need solutions which extract a new set of features from the text. For example, these features could be based on frequency count of words. Typically, words with very high frequency count are functions words, and an unsupervised clustering of words can be done based on frequency counts. This leads to representation of words as a sequence of cluster numbers similar to part-of-speech sequence.

In the scope of this thesis, the objective is to build a PBA model using machine learning techniques which make use of acoustic cues to locate prosodic phrase breaks. Such techniques make annotation faster and cheaper. At the same time, the ability to model phrasing patterns in a given speech database could bring in speaker-specific phrasing patterns. As a part of this investigation, we would like to know whether prosodic phrase breaks are specific to a speaker, and if so, propose a mechanism for learning speaker-specific phrase breaks. Another equally important aspect dealt with in this thesis is to demonstrate the usefulness of these speaker-specific phrase breaks for a TTS system.

4.2 Are prosodic phrase breaks speaker-specific?

In order to examine whether prosodic phrase breaks are specific to a speaker, we conducted the following experiment using a short story from *Emma* (Volume 1, Chapter 1), spoken by four speakers (Sibella, Sherry, Moira and Elizabeth). This story consisted of 54 paragraphs and around 3000 words. For every word in the story, a binary feature was derived indicating whether there was a break or not after the word. The presence of a break was indicated by '1' and the absence of a break was indicated by '-1'.

Let $\boldsymbol{S} = [s_1, \ldots, s_w, \ldots, s_D]$ denote the sequence of binary features derived for the words in the story using syntactic phrase breaks. Here $w = 1, \ldots, D$, where D is the number of words in the story. These breaks were derived using the Stanford Parser [Klein et al., 2003], which parses the text into syntactic constituents such as noun phrase, verb phrase and adjectival phrase. The end of each constituent was considered as a phrase break.

Let $\boldsymbol{R} = [r_1, \ldots, r_w, \ldots, r_D]$ denote the sequence of features derived for the words in the story using prosodic phrase breaks. These breaks were derived based on the duration of pause after a word. A pause was considered as a prosodic phrase break, if its duration was greater than 150 ms. This threshold of 150 ms was derived

Table 4.2: Correlation between syntactic phrase breaks and prosodic phrase breaks of different speakers.

	Syntactic	Elizabeth	Moira	Sherry	Sibella
Syntactic	1	0.29	0.30	0.23	0.31
Elizabeth		1	0.66	0.61	0.72
Moira			1	0.58	0.69
Sherry				1	0.62
Sibella					1

based on our earlier work on phrasing models [Keri et al., 2007]. Since the story was spoken by four different speakers, we derived the binary feature sequences $\boldsymbol{R}^s$, $\boldsymbol{R}^h$, $\boldsymbol{R}^m$ and $\boldsymbol{R}^e$ representing the prosodic phrase break sequences for Sibella, Sherry, Moira and Elizabeth respectively. The correlation coefficient between two feature sequences $\boldsymbol{X}$ and $\boldsymbol{Y}$ is calculated using Eq. (4.1).

$$c(\boldsymbol{X},\boldsymbol{Y}) = \frac{\sum_{w=1}^{D}(x_w-\bar{x})(y_w-\bar{y})}{\sqrt{\sum_{w=1}^{D}(x_w-\bar{x})^2}\sqrt{\sum_{w=1}^{D}(y_w-\bar{y})^2}}, \tag{4.1}$$

where $\boldsymbol{X}$ and $\boldsymbol{Y}$ are one of $\{\boldsymbol{S}, \boldsymbol{R}^s, \boldsymbol{R}^h, \boldsymbol{R}^m, \boldsymbol{R}^e\}$. $\bar{x}$, $\bar{y}$ denote the mean values of $\boldsymbol{X}$ and $\boldsymbol{Y}$ respectively. It should be noted that $c(\boldsymbol{X},\boldsymbol{Y}) = c(\boldsymbol{Y},\boldsymbol{X})$.

Table 4.2 displays the correlation coefficients between the syntactic phrase breaks and prosodic phrase breaks. It also displays the correlation coefficients between the prosodic phrase breaks of any two speakers. From Table 4.2, we can observe that correlation coefficients between the syntactic phrase breaks and prosodic phrase breaks vary from 0.26 to 0.33. These lower values indicate that syntactic phrase breaks and prosodic phrase breaks differ significantly. From Table 4.2, it could also be observed that the correlation coefficients between the prosodic phrase breaks of any two speakers vary between $0.6 - 0.7$. These values indicate that the correlation coefficients between the prosodic phrase breaks of any two speakers is higher than the correlation coefficients between the prosodic phrase breaks and syntactic phrase breaks. At the same time, the correlation coefficients between the prosodic phrase breaks of any two speakers are lesser than 1, thus suggesting that prosodic phrase breaks are specific to a speaker.

Another question that would be of interest is to study the correlation between

Table 4.3: Correlation between syntactic phrase breaks, prosodic phrase breaks and the breaks derived from punctuation marks.

	Syntactic	Elizabeth	Moira	Sherry	Sibella
Punct.	0.34	0.66	0.64	0.54	0.72

the prosodic phrase breaks and punctuations in text. Audio books are stories/novels and the text of these books inherit punctuation marks such as comma, period and exclamation from the authors. It would be interesting to know whether the phrasing patterns of speakers correspond to punctuation marks in text. Let $\boldsymbol{P} = [p_1, \ldots, p_w, \ldots, p_D]$ denote the sequence of binary features derived for the words in the story using punctuation marks. Here, p_w takes a value of 1 or -1 depending on whether there is a punctuation mark after/before w or not. The correlation coefficients computed between $\boldsymbol{P}$ and each of the sequences in $\{\boldsymbol{S}, \boldsymbol{R}^s, \boldsymbol{R}^h, \boldsymbol{R}^m, \boldsymbol{R}^e\}$ are shown in Table 4.3.

First, the correlation coefficient between the punctuation marks and syntactic breaks in Table 4.3 is 0.34. This suggests the punctuation marks in text need not adhere to syntactic phrase breaks. Secondly, the prosodic phrase breaks are better correlated to punctuation marks than syntactic phrase breaks. This could imply that the speakers might have been influenced by the punctuation marks during the recording of speech data. However, the correlation coefficient values are much less than 1, and vary from speaker to speaker. This suggests that the punctuation marks may indicate possible chunks in the text, but the speakers seem to have their own choice of phrasing patterns. Tables 4.2-4.3 provide sufficient evidence that prosodic phrase breaks are specific to a speaker.

4.3 Learning speaker-specific phrase breaks

To learn speaker-specific phrase breaks, we propose an unsupervised learning algorithm. Fig. 4.2 provides a schematic diagram of the proposed algorithm consisting of two phases. In the first phase, we hypothesize the location of phrase breaks in the speech signal using pauses as acoustic cues. As these initial estimates are obtained based on knowledge that pauses are good indicators of phrase breaks (refer to Section 4.1), one could treat the hypothesized phrase break locations as labeled data. In the second phase, features such as duration, F0 and energy are extracted from these locations for building a machine learning model, which is trained to classify these acoustic features as belonging to a break or not-a-break.

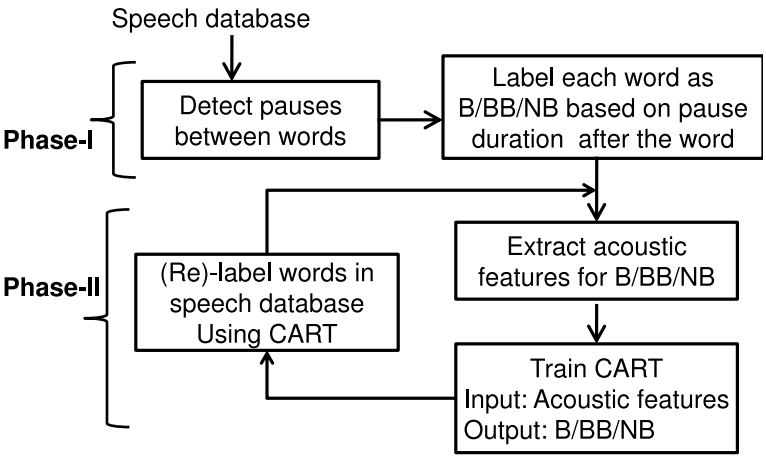


Figure 4.2: Flow chart of the proposed algorithm for learning speaker-specific phrase breaks.

We then attempt to bootstrap this model by re-labeling the data.

4.3.1 Phase 1: Using pauses as acoustic cues

In phase-1, we hypothesize the location of phrase breaks based on pauses in the speech signal. This phase is referred to as building a phrase break annotator (PBA-0), and the steps involved in building PBA-0 are as follows –

- Obtain word level boundaries and pause regions between the words in an utterance. This can be accomplished by force-aligning an utterance with its corresponding text. During the process of force-alignment, an optional silence HMM is introduced between each pair of words. If there exists a pause between any two words then it is automatically detected using the silence HMM model.
- Based on the duration (d) of the pause, two types of breaks (B / BB) are identified. Here B denotes a type of phrase break, when $50\ ms \geq d \leq 150\ ms$, and BB denotes another type of phrase break when $d > 150\ ms$.

4.3.2 Phase 2: Bootstrapping

In phase-2, we build a prosodic break annotator model (PBA-1), based on phrase breaks regions identified by PBA-0. The steps involved in building PBA-1 are as follows –

1. Extract duration, F0 and energy features from phrase break regions identified by PBA-0. At each phrase break, a set of 10 features related to duration, F0 and energy features is computed for the last syllable (ν). Similar features are computed for the two neighboring (one left and right) syllables of ν. The feature set computed for each syllable is shown in Table 4.1, and is based on the work in [Ananthakrishnan and Narayanan, 2008].
2. Build a classification and regression tree (CART) model, where the predictee is phrase break level (B / BB / NB) and the predictors are duration, F0 and energy features. Here NB denotes not-a-break. The features for NB are obtained by considering the syllables in a word which is immediately previous to a word identified as a break (B / BB).
3. Use the CART model to re-label the speech data and classify each word boundary as belonging to one of the classes: B / BB / NB. This step will provide a new set of training examples for B / BB / NB classes.
4. Update / retrain the CART model with the new set of training examples.
5. Repeat steps 3 and 4 for 1-2 iterations.

4.4 Evaluation of PBA models

To evaluate a PBA model, the location of predicted phrase breaks can be compared with manually identified phrase breaks, and the accuracy of a PBA model can be reported in terms of precision and recall.

However, such an evaluation criteria would limit the purpose of building a PBA model for languages and speech databases which may not have such hand labeling done. An alternate method of evaluation is to incorporate the prosodic phrase breaks predicted by a PBA model in a text-to-speech system, and perform subjective and objective evaluations to know whether prosodic phrasing aids in improving the quality of synthesized speech. To perform this evaluation, statistical

parametric synthesis (such as CLUSTERGEN [Black, 2006]) is a better platform than unit selection synthesis. CLUSTERGEN is a statistical parametric synthesizer which predicts duration and F0 for each phone in the input text. Spectral parameters are generated for each phone based on its duration. Speech is synthesized by exciting the spectral parameters with train of pulses or random noise. The mode of excitation depends on the predicted F0 values of phones. The effect of prosodic phrase breaks on spectral quality could be measured by using objective metrics such as Mel-cepstral distortion (explained below).

The process to incorporate and evaluate the effectiveness of prosodic phrase breaks in CLUSTERGEN is as follows:

- From PBA models, obtain the location of prosodic phrase breaks in the text of all utterances, using the three levels of phrase breaks ($NB/B/BB$) described previously.
- Divide this annotated text into training set (T-set) and held out test set (H-set).
- Use T-set to build a CLUSTERGEN voice (see Appendix C for details.). The build process of CLUSTERGEN is modified to use prosodic phrase breaks as one of the features for clustering the spectral parameters (see Appendix D for details.).
- Synthesize utterances from H-set. An objective evaluation is done by computing the spectral distortion between the original and synthesized utterances. However, due to variations in the durations of original and synthesized utterances, they are first aligned using dynamic programming and Mel-cepstral distortion (MCD) is computed between the aligned frames. The MCD measure between two Mel-cepstral vectors is defined as $MCD = (10/\ln 10) * \sqrt{2 * \sum_{i=1}^{25} (mc_i^t - mc_i^e)^2}$, where mc_i^t and mc_i^e denote the original and the synthesized Mel-Cepstra respectively. A lesser MCD value indicates a better quality of synthesis. MCD is calculated over all the Mel-cepstral coefficients, including the zeroth coefficient.

4.4.1 Results on hand labeled data

PBA models can be evaluated by comparing the location of predicted phrase breaks with manually identified phrase breaks. This evaluation can be done if a corresponding hand-labeled corpus is available. The Boston University Radio News corpus is

Table 4.4: Phrase breaks predicted from PBA-0 and PBA-1 are compared with the hand labeled phrase breaks. Precision, recall, F-measure indicates the accuracy of PBA models in predicting B/BB. True negative indicates the accuracy of PBA models in predicting NB.

	Precision	Recall	True Neg.	F-Measure	MCD
PBA-0	0.91	0.91	0.96	0.91	5.89
PBA-1	0.95	0.91	0.98	0.92	5.92
Baseline	-	-	-	-	6.06

one such corpus hand labeled with phrase breaks. It consists of broadcast radio news stories which include original broadcasts and radio laboratory simulations recorded from seven FM radio announcers [Ostendorf et al., 1995]. The database contains speech from three female (F1A, F2B and F3A) and four male speakers (M1B, M2B, M3B and M4B). Most of the recordings in this corpus are manually labeled with the orthographic transcription, phone alignments, part-of-speech tags and phrase break levels.

Our experiments are focused on the recordings of speaker *F2B* which has hand labeled phrase breaks. The *F2B* recordings contain 165 utterances corresponding to 80 minutes of speech data. The annotation of phrase breaks in Boston corpus follows ToBI style and defines five levels of breaks (refer to Sec. 4.1 for details). For our purposes we have considered break level 3 (referred to as B), and break level 4 (referred to as BB). Other break levels ($0-2$) are considered as NB, i.e., not-a-break.

Phrase breaks predicted from PBA-0 and PBA-1 are compared with the hand labeled phrase breaks. Table 4.4 shows the precision, recall and F-measure of PBA models in predicting B and BB. The high values of F-measure for both PBA-0 and PBA-1 indicate that pauses correlate well with B and BB type of breaks. A marginal improvement could be observed in the performance of PBA-1 over PBA-0. The value of true negative in Table 4.4 indicates the accuracy of PBA-0 and PBA-1 in predicting NB. These results indicate that PBA models could be used for automatic annotation of speaker-specific phrase breaks in a given speech database.

As discussed earlier, the evaluation of PBA models using hand labeled data is not always feasible. An alternative is to incorporate the phrase breaks in a TTS system and measure the objective and subjective improvements of the voice. CLUSTERGEN voices were built by using the phrase breaks predicted by PBA-0 and PBA-1. The predicted phrase breaks were used as one of the features for clustering the spectral parameters. Voices were built using 71 minutes of speech data. Evaluation of these

Table 4.5: Details of the audio books used in evaluation of PBA models including duration of the book, duration of utterances in training set (T-set) and testing set (H-set). The units of duration is hours.

Book	Duration	T-set	H-set
EMMA	17.33	15.67	1.66
WALDEN	13.57	12.72	1.45
PRIDE & PREJUDICE	13.45	11.99	1.46
IIIT-LEN	9.20	8.24	0.96

voices was carried out on 9 minutes of held-out test data. As shown in Table 4.4, the MCD scores obtained for voices using PBA-0 and PBA-1 were found to be lesser than the baseline voice. Here the baseline voice refers to the synthetic voice built using default settings of CLUSTERGEN. The default settings of CLUSTERGEN employ a statistical phrasing model built which assumes phrasing patterns to be same across different speakers [Taylor and Black, 1998]. MCD scores of PBA models being lesser than baseline voice suggest that use of speaker-specific prosodic phrasing improves the quality of synthetic voices.

4.4.2 Results on audio books

TTS based evaluation of PBA-0 and PBA-1 was carried out on audio books of *EMMA, WALDEN, PRIDE AND PREJUDICE* and on a Telugu audio book referred to as *IIIT-LEN*. Table 4.5 provides the details including duration of the book, duration of utterances in training set (T-set) and testing set (H-set).

As discussed in Section 4.3.1, PBA-0 and PBA-1 models were built for *EMMA, WALDEN, PRIDE AND PREJUDICE* and *IIIT-LEN* speech databases. Prosodic phrase breaks from PBA models, as described in Section 4.4, were incorporated to build CLUSTERGEN voices for *EMMA, WALDEN, PRIDE AND PREJUDICE* and *IIIT-LEN*. Performance of these voices evaluated on their respective H-sets using MCD is as shown in Table 4.6. It can observed that the MCD scores of PBA-0 / PBA-1 performs better than that of the *baseline* suggesting that the incorporation of speaker-specific phrase breaks improves the quality of synthetic speech. Here, *baseline* refers to CLUSTERGEN voices generated using default settings in CLUSTERGEN. Informal listening experiments conducted on PBA-0 / PBA-1 showed that the prosodic phrase breaks have improved the perception of the voice with respect to *baseline*.

Table 4.6: Objective evaluation of synthetic voices using PBA. MCD scores indicate spectral distortion of original and synthesized speech.

	MCD			
	EMMA	WALDEN	PRIDE & PREJUDICE	IIIT-LEN
Baseline	5.55	5.40	6.89	7.17
PBA-0	5.43	5.12	6.71	5.73
PBA-1	5.36	5.09	6.71	5.65

Table 4.7: Subjective evaluation of IIIT-LEN voice.

	Baseline	PBA-1	No-preference
Baseline vs PBA-1	5 / 60	26 / 60	29 / 60

4.4.3 Subjective evaluation

In addition to the objective evaluation, we also conducted a subjective evaluation of PBA-1. Native speakers of Telugu were asked to listen to 10 utterances synthesized from *baseline* and PBA-1, and state whether they preferred a particular voice or not. A total of 6 subjects participated in the listening test, resulting in 60 preference ratings. The listening test results are summarized in Table 4.7. Our results show that the TTS voice built using PBA-1 was preferred for 43% of utterances, while the *baseline* voice was preferred for only 8% of utterances.

4.5 Summary

To incorporate duration and intonation changes in synthetic speech, it is important to model phrasing patterns, i.e., the location of prosodic phrase breaks in utterances. In the current models, the prosodic phrasing patterns are assumed to be same across all voices. In this chapter, we showed that prosodic phrasing patterns are specific to a speaker. We proposed an unsupervised algorithm to learn speaker-specific phrasing patterns from a given speech database Experiments were conducted to study the usefulness of speaker-specific phrase breaks in a TTS system. The results indicate that speaker-specific phrase breaks aid in improving the spectral and perceived quality of synthesized utterances.

Chapter 5

Conversion of speaker characteristics

In previous chapters, we have exploited audio books for building synthetic voices and incorporated speaker-specific phrase breaks to improve the quality of synthesized utterances. However, one might want to listen to synthesized utterances in a voice of a target or a virtual speaker. This could be for several reasons including the celebrity status of a speaker or the listener's social/family relationship or merely a preference for the speaker. Voice conversion is a technique of rendering an utterance as if it was spoken by a target speaker. It enables new voices to be built quickly by alleviating the huge efforts involved in development of new voices. In this chapter, we focus on voice conversion techniques to render an utterance in the voice of a target speaker.

5.1 Need for speaker conversion

Building a synthetic voice in a traditional way involves an extensive and expensive process. The steps involved in this process are –

- Record about 8-10 hours of speech. It is preferred that the text being read has rich prosody including varied intonation contours, phrasing and prominence.
- Use INTERSLICE to segment large speech files in the recordings.
- Build a CLUSTERGEN or a unit selection voice, and perform required tunings. These tunings include extraction of better pitch marks, duration modeling etc.

These steps consume a lot of time and efforts and are often impractical. For example, suppose, a mother would like to have a story-telling computer with a TTS voice –

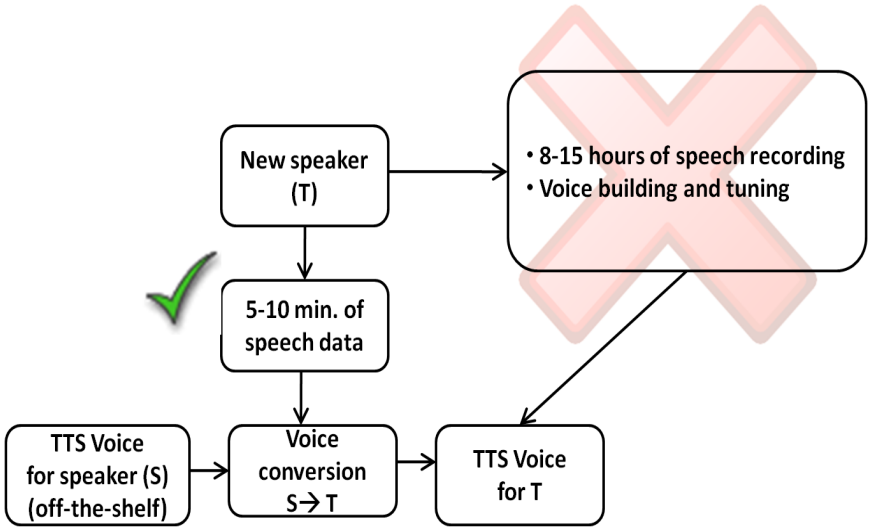


Figure 5.1: Building a new TTS voice using the conventional approach which is extensive and expensive vs the approach using voice conversion.

sounding like her. This TTS voice can be used for rendering stories to her children, when she is busy or away. To build such TTS voice, it is not a comfortable and easygoing process for the mother to sit in a studio and record for 8-15 hours. Apart from fatigue due to physical and cognitive load involved in recording long hours of speech, practical difficulties also exist in the form of illiteracy, mistakes in recording etc.

Rather, an easygoing process – shown in Fig. 5.2, would be to record only 5-10 minutes of speech data. It would be convenient to use an existing TTS voice, say from a speaker S and convert the characteristics of speaker S to sound like the mother (speaker T). The steps involved in this process are –

- Use an existing TTS voice from speaker S.
- Record 5-10 minutes of speech from speaker T.
- Build a speaker conversion module which converts the characteristics of speaker S to that of speaker T.
- Use the speaker conversion module ($S \rightarrow T$) as a postprocessing filter for the TTS voice of S as shown in Fig. 5.2.

This process alleviates the need for long hours of recordings, and is scalable to generate new voices with short amount of speech data from new speakers. Conversion

of speaker characteristics is often referred to as voice conversion.

5.2 Building a voice conversion system

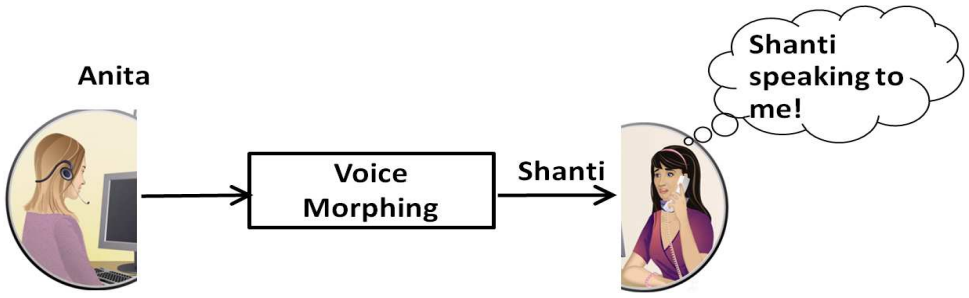
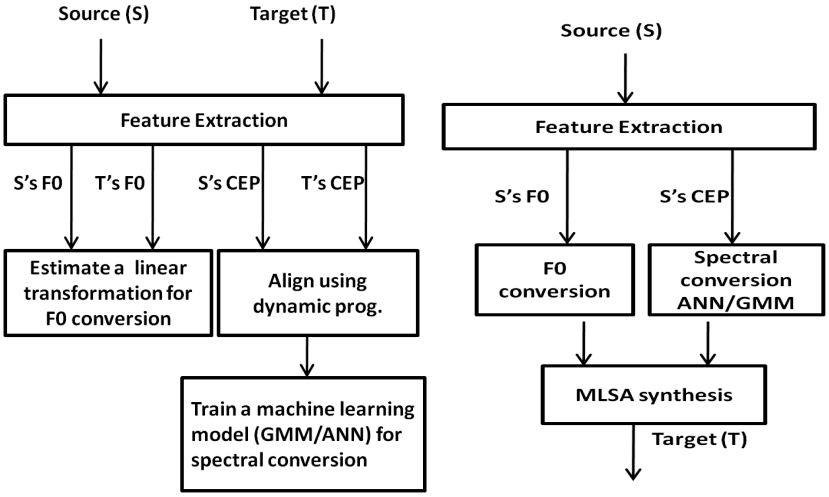


Figure 5.2: A lay-man understanding of a voice conversion system.

A typical voice conversion (VC) system morphs the utterance of a source speaker so that it is perceived as if spoken by a specified target speaker (see Fig. 5.2). Research studies have shown that characteristics of a speaker's individuality lies in vocal tract shape [Toda et al., 2007], excitation , [Rao, 2010], prosody such as intonation and duration [Toth and Black, 2008], and in expressive aspects affected by attitude and emotions. A complete voice conversion system should convert all types of speaker dependent characteristics of speech. However, due to limited understanding in parameterization of speaker's individuality [Kuwabara and Sagisaka, 1995], current state-of-the-art voice conversion systems are focused only on vocal tract shape represented by spectral features and excitation represented by fundamental frequency $F0$ [Toda et al., 2007]. The research question addressed is - "How to obtain an optimal mapping function which transforms the spectral hints of a source speaker to that of a target speaker?". To learn such transformation, current voice conversion techniques rely on the existence of parallel corpus, i.e., the same set of utterances recorded by both the source and target speakers [Toda et al., 2007].

Given the parallel data from a source and a target speaker, the training and conversion modules of a voice conversion system are as shown in Fig. 5.3. The steps involved in training a voice conversion model are as follows –

- The first step is to extract spectral and excitation parameters of the source and target speakers. In this work, the spectral parameters are represented



(a) Training (b) Conversion

Figure 5.3: Training and testing modules in voice conversion framework.

by Mel-cepstral coefficients (MCEPs) and excitation is represented by fundamental frequency (F_0). These representations are chosen so that they can be synthesized using Mel Log Spectral Approximation (MLSA) filter after transformation [Imai, 1983].

- Even though the source and target speakers have spoken the same utterance, they vary in duration. Hence the next step involves in aligning the parallel utterances using dynamic programming. This process results in providing a correspondence between sounds of source and target speakers.
- Given this correspondence, a Gaussian mixture model (GMM) or an artificial neural network (ANN) is trained to capture the mapping function.
- For conversion of F_0, a linear transformation is computed in log domain based on mean $F0$ and its standard deviation computed from the training data of the source and target speakers.

The steps in testing or conversion are as follows –

- Given a new utterance from the source speaker, MCEPs and F_0 are extracted from the speech signal.

- MCEPs are transformed onto the target speaker's acoustic space using GMM/ANN models. F_0 is transformed using a linear transformation.
- The transformed MCEPs along with F_0 are used as input to an MLSA filter to synthesize the transformed utterance. This filter generates the utterance from the transformed MCEPs and F0 values using pulse excitation or random noise excitation [Imai, 1983].

Each of these steps is explained in detail below.

5.2.1 Feature extraction

To extract features from a speech signal, an excitation-filter model of speech is applied. MCEP vectors are extracted as filter parameters and fundamental frequency (F_0) estimates are derived as excitation features for every 5 ms [Imai, 1983]. The length of an MCEP vector is 25 (i.e., 25 MCEP coefficients are extracted for every 5 ms).

5.2.2 Alignment of parallel utterances

As the durations of the parallel utterances typically differ, dynamic time warping (or dynamic programming) is used to align MCEP vectors of the source and the target speakers [Toda et al., 2007]. After alignment, let $\boldsymbol{x}_t$ and $\boldsymbol{y}_t$ denote the source and the target feature vectors at frame t respectively.

5.2.3 Spectral mapping using GMM

In GMM-based conversion [Toda et al., 2007], the learning procedure aims to fit a GMM model to the augmented source and target feature vectors. Formally, a GMM allows the probability distribution of a random variable $\boldsymbol{z}$ to be modeled as the sum of M Gaussian components, also referred to as mixtures. Its probability density function $p(\boldsymbol{z}_t)$ can be written as

$$p(\boldsymbol{z}_t) = \sum_{m=1}^{M} \alpha_m \, \mathcal{N}(\boldsymbol{z}_t; \boldsymbol{\mu}_m^{(z)}, \boldsymbol{\Sigma}_m^{(z)}) \ , \ \sum_{m=1}^{M} \alpha_m = 1, \ \alpha_m \geq 0 \tag{5.1}$$

where $\boldsymbol{z}_t$ is an augmented feature vector $[\boldsymbol{x}_t^T \ \boldsymbol{y}_t^T]^T$ The notation ^{T} denotes transposition of a vector. $\mathcal{N}(\boldsymbol{z}_t; \boldsymbol{\mu}_m^{(z)}, \boldsymbol{\Sigma}_m^{(z)})$ denotes the parameters of a Gaussian distribution and α_m denotes the prior probability with which the vector $\boldsymbol{z}_t$ belongs to the m^{th} component. $\boldsymbol{\Sigma}_m^{(z)}$ denotes the covariance matrix and $\boldsymbol{\mu}_m^{(z)}$ denotes the mean vector of the m^{th} component for the joint vectors. These parameters are represented as

$$\boldsymbol{\Sigma}_m^{(z)} = \begin{bmatrix} \boldsymbol{\Sigma}_m^{(xx)} & \boldsymbol{\Sigma}_m^{(xy)} \\ \boldsymbol{\Sigma}_m^{(yx)} & \boldsymbol{\Sigma}_m^{(yy)} \end{bmatrix}, \qquad \boldsymbol{\mu}_m^{(z)} = \begin{bmatrix} \boldsymbol{\mu}_m^{(x)} \\ \boldsymbol{\mu}_m^{(y)} \end{bmatrix}, \tag{5.2}$$

where $\boldsymbol{\mu}_m^{(x)}$ and $\boldsymbol{\mu}_m^{(y)}$ are the mean vectors of the m^{th} component for the source and the target feature vectors respectively. The matrices $\boldsymbol{\Sigma}_m^{(xx)}$ and $\boldsymbol{\Sigma}_m^{(yy)}$ are the covariance matrices, while $\boldsymbol{\Sigma}_m^{(xy)}$ and $\boldsymbol{\Sigma}_m^{(yx)}$ are the cross-covariance matrices, of the m^{th} component for the source and the target feature vectors respectively. The covariance matrices $\boldsymbol{\Sigma}_m^{(xx)}$, $\boldsymbol{\Sigma}_m^{(yy)}$, $\boldsymbol{\Sigma}_m^{(xy)}$ and $\boldsymbol{\Sigma}_m^{(yx)}$ are assumed to be diagonal in this work. The model parameters $(\alpha_m, \boldsymbol{\mu}_m^{(z)}, \boldsymbol{\Sigma}_m^{(z)})$ are estimated using Expectation Maximization (EM) algorithm.

The conversion process (also referred to as testing process) involves regression, i.e., given an input vector, $\boldsymbol{x}_t$, we need to predict $\boldsymbol{y}_t$ using GMMs, which is calculated as shown in the equation below.

$$H(\boldsymbol{x}_t) = E[\boldsymbol{y}_t|\boldsymbol{x}_t] \tag{5.3}$$

$$= \sum_{m=1}^{M} h_m(\boldsymbol{x}_t)[\boldsymbol{\mu}_m^{(y)} + \boldsymbol{\Sigma}_m^{(yx)}(\boldsymbol{\Sigma}_m^{(xx)})^{-1}(\boldsymbol{x}_t - \boldsymbol{\mu}_m^{(x)})], \tag{5.4}$$

where

$$h_m(\boldsymbol{x}_t) = \frac{\alpha_m \, \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu}_m^{(x)}, \boldsymbol{\Sigma}_m^{(xx)})}{\sum_{n=1}^{M} \alpha_n \, \mathcal{N}(\boldsymbol{x}_t; \boldsymbol{\mu}_n^{(x)}, \boldsymbol{\Sigma}_n^{(xx)})}, \tag{5.5}$$

is the posterior probability that a given input vector $\boldsymbol{x}_t$ belongs to the m^{th} component.

From Eq. (5.3), we can see that the GMM transformation deals with every feature vector independent of its previous and next frames. Thus it introduces local patterns in converted spectral trajectory which are different than that of the target's natural spectral trajectory. To obtain a better conversion of spectral trajectory, dynamic features such as delta and delta-delta MCEP coefficients are used in the mapping function. Further, a smoothing operation referred to as maximum

likelihood parameter generation (MLPG) is performed as a postprocessing technique. It has been shown in [Toda et al., 2007], that dynamic features and MLPG improves the performance of a GMM based voice conversion system. Toda *et. al.,* made another interesting observation on the variance (referred to as global variance) of transformed spectral parameters. GMM models under-estimate the natural variance of MCEPs, and thus the global variance (GV) of the transformed Mel-cepstra is smaller than that of the target ones. To circumvent this problem, a penalty term was introduced for the reduction of variance during mapping function. A more detailed description of these techniques can be found in [Toda et al., 2007].

In this work we have conducted GMM based VC experiments on the voice conversion setup built in FestVox distribution [Black and Lenzo, 2009]. This voice conversion setup is based on the work done in [Toda et al., 2007], and supports the conversion considering 1) MLPG based smoothing and 2) the global variance (GV) of spectral trajectory.

5.2.4 Spectral mapping using ANN

From Eq. (5.4), it can be observed that GMM based voice conversion is a linear regression model. Eq. (5.4) can be re-written as –

$$H(\boldsymbol{x}_t) \quad = \sum_{m=1}^{M} h_m(\boldsymbol{x}_t)[\boldsymbol{\mu}_m^{(y)} + \boldsymbol{W}_m \boldsymbol{x}_t^{'}], \tag{5.6}$$

where $\boldsymbol{W}_m = \boldsymbol{\Sigma}_m^{(yx)}(\boldsymbol{\Sigma}_m^{(xx)})^{-1}$ and $\boldsymbol{x}_t^{'} = \boldsymbol{x}_t - \boldsymbol{\mu}_m^{(x)}$. Eq. (5.6) shows that GMM based voice conversion is a weighted sum of linear regression models [Kain, 2001]. However, the relation between vocal tract shapes of two speakers is typically non-linear. Hence, nonlinear models such as artificial neural networks are explored to approximate the mapping function [Srinivas et al., 2010].

Artificial neural network (ANN) models consist of interconnected processing nodes, where each node represents the model of an artificial neuron, and the interconnection between two nodes has a weight associated with it [Yegnanarayana, 1999b]. ANN models with different topologies perform different pattern recognition tasks. For example, a feed-forward neural network can be designed to perform the task of pattern mapping, whereas a feedback network could be designed for the task of pattern association. A multi-layer feed forward neural network is used in this work to approximate the mapping function between the source and target vectors.

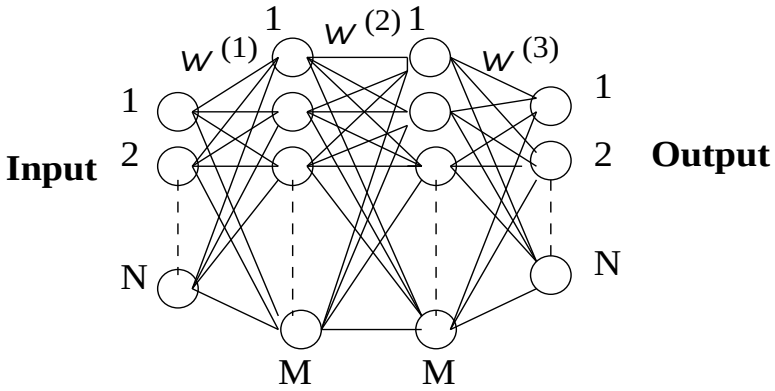


Figure 5.4: An architecture of a four layered ANN with N input and output nodes and M nodes in the hidden layers.

Figure 5.4 shows the architecture of a four layer ANN used to capture the transformation function for mapping the acoustic features of a source speaker onto the acoustic space of a target speaker. The ANN is trained to map the MCEPs of a source speaker to the MCEPs of a target speaker, i.e., if $G(\boldsymbol{x}_t)$ denotes the ANN mapping of a source vector $\boldsymbol{x}_t$, then the error of mapping is given by $\epsilon = \sum_t \|\boldsymbol{y}_t - G(\boldsymbol{x}_t)\|^2$. $G(\boldsymbol{x}_t)$ is defined as

$$G(\boldsymbol{x}_t) = \widetilde{g}(\boldsymbol{w}^{(3)} g(\boldsymbol{w}^{(2)} g(\boldsymbol{w}^{(1)} \boldsymbol{x}_t))), \tag{5.7}$$

where

$$\widetilde{g}(\vartheta) = \vartheta, g(\vartheta) = a \tanh(b\, \vartheta). \tag{5.8}$$

Eq. (5.7) shows the nonlinear mapping function of an ANN model. Here $\boldsymbol{w}^{(1)}, \boldsymbol{w}^{(2)}, \boldsymbol{w}^{(3)}$ represents the weight matrices of first, second and third hidden layers of the ANN model respectively. The values of the constants a and b used in $\tanh$ function are 1.7159 and $2/3$ respectively (see Appendix E for details).

A generalized back propagation learning is used to adjust the weights of the neural network so as to minimize ϵ, i.e., the mean squared error between the target and the transformed output values. Selection of initial weights, architecture of ANN, learning rate, momentum and number of iterations are some of the optimization parameters in training an ANN (see Appendix E for details). Once the training is complete, we get a weight matrix that represents the mapping function between the spectral features of a pair of source and target speakers. Such a weight matrix can be used to transform a feature vector from the source speaker to that of the target speaker.

5.2.5 Mapping of excitation features

Our focus is to get a better transformation of spectral features. Hence, we use the traditional approach of F_0 transformation as used in a GMM based transformation. A logarithm Gaussian normalized transformation [Liu et al., 2007] is used to transform the F_0 of a source speaker to the F_0 of a target speaker as indicated in the equation below.

$$\log({F_0}^t_{conv}) = \mu_{tgt} + \frac{\sigma_{tgt}}{\sigma_{src}}(\log({F_0}^t_{src}) - \mu_{src}), \tag{5.9}$$

where μ_{src} and σ_{src} are the mean and standard deviation of the fundamental frequency in logarithm domain computed on the training data of the source speaker, μ_{tgt} and σ_{tgt} are the mean and standard deviation of the fundamental frequency in logarithm domain computed on the training data of the target speaker. ${F_0}^t_{src}$ is the fundamental frequency of the source speaker at frame t in a test utterance and ${F_0}^t_{conv}$ is the corresponding converted fundamental frequency.

5.3 Evaluation criteria

Subjective evaluation

Subjective evaluation is based on collecting human opinions as they are directly related to human perception, which is used to judge the quality of transformed speech. The popular tests are ABX test, MOS test and similarity test.

- *ABX Test:* For the ABX test, we present the listeners with a GMM transformed utterance and an ANN transformed utterance to be compared against X, which will always be a natural utterance of the target speaker. To ensure that a listener does not become biased, we shuffle the position of ANN/GMM transformed utterances i.e., A and B, with X always constant at the end. The listeners would be asked to select either A or B, i.e., the one which they perceive to be closer to the target utterance.

- *MOS Test*: Mean Opinion Score (MOS) is another subjective evaluation where listeners evaluate the speech quality of the converted voices using a 5-point scale (5: excellent, 4: good, 3: fair, 2: poor, 1: bad).

- *Similarity Test:* In similarity test, we present the listeners with a transformed utterance and a corresponding natural utterance of the target speaker. The

listeners would be asked to provide a score indicating how similar the two utterances are in terms of speaker characteristics. The range of similarity test is also from 1 to 5, where a score of 5 indicates that both the recordings are from the same speaker and a score of 1 indicates that the two utterances are spoken by two different speakers.

Objective evaluation

Mel Cepstral Distortion (MCD) is an objective error measure known to have correlation with the subjective test results [Toda et al., 2007]. Thus MCD is used to measure the quality of voice transformation. MCD is related to filter characteristics and hence is an important measure to check the performance of mapping obtained by an ANN/GMM model. Computation of MCD has already been presented in Eq. (2.11).

5.4 Experiments and results

Database used for the experiments

Current voice conversion techniques need a parallel database [Toda et al., 2007], where the source and the target speakers record the same set of utterances. The experiments presented here is carried out on CMU ARCTIC database consisting of the same set of utterances recorded by seven speakers. Each speaker has recorded a set of 1132 phonetically balanced utterances [Kominek and Black, 2004a]. The ARCTIC database includes utterances of *SLT* (US Female), *CLB* (US Female), *BDL* (US Male), *RMS* (US Male), *JMK* (Canadian Male), *AWB* (Scottish Male), *KSP* (Indian Male). It should be noted that about 30-50 parallel utterances are needed to build a voice conversion model [Toda et al., 2007]. Thus, for each speaker we took around 40 utterances as training data (approximately 2 minutes) and a separate set of 59 utterances (approximately 3 minutes) as testing data.

Objective evaluation of a GMM based VC system

To build a GMM based VC system, we have considered two cases: 1) Transformation of *SLT* (US female) to *BDL* (US male) and 2) Transformation of *BDL* (US male) to *SLT* (US female). For both the experiments, the number of training utterances is 40

Table 5.1: Objective evaluation of a GMM based VC system for various training parameters where Set 1: *SLT* to *BDL* transformation; Set 2: *BDL* to *SLT* transformation

No. of mixtures	No. of params.	MCD					
		Without MLPG		With MLPG		With MLPG (& GV)	
		SLT-BDL	BDL-SLT	SLT-BDL	BDL-SLT	SLT-BDL	BDL-SLT
32	6176	6.367	6.102	6.152	5.823	6.547	6.072
64	12352	6.336	6.107	6.057	5.762	6.442	6.015
128	24704	6.348	6.068	6.017	5.682	6.389	5.907

(approximately 2 minutes) and the testing is done on the test set of 59 utterances (approximately 3 minutes). The number of MCEP vectors for 40 training utterances in *SLT* and *BDL* is 23,679 and 21,820 respectively.

Table 5.1 provides the MCD scores computed for *SLT*-to-*BDL* and *BDL*-to-*SLT* respectively for increasing number of Gaussians. It could be observed that the MCD scores decrease with the increase in the number of Gaussians, however, it should be noted that the increase in the number of Gaussians also increases the number of parameters in the GMM. With the use of diagonal covariance matrix, the number of parameters in the GMM with 64 and 128 Gaussian components is 12,352 and 24,704 respectively. We can also observe that the GMM based conversion with MLPG performs better than that of the GMM based system without MLPG. However, the GMM based VC system with MLPG produced lesser MCD scores than the GMM based VC system with MLPG and GV. While GV seemed to improve the quality of transformed speech based on human listening tests, it is not clear from [Toda et al., 2007] whether it also improves the score according to MCD computation. Considering the number of parameters used in the GMM model, we have used the GMM based VC system with 64 Gaussian components (with MLPG) for further comparison with an ANN based VC system.

Objective evaluation of an ANN based VC system

To build an ANN based VC system, we have considered two cases 1) *SLT*-to-*BDL* and 2) *BDL*-to-*SLT*. For both the experiments, the number of training utterances is 40 (approximately 2 minutes) and the testing is done on the test set of 59 utterances (approximately 3 minutes).

Table 5.2: MCD obtained on the test set for different architectures of an ANN model. (No. of iterations: 200, Learning Rate: 0.01, Momentum: 0.3) Set 1: *SLT* to *BDL*; Set 2: *BDL* to *SLT*

S.No	ANN architecture	No. of params.	MCD	
			Set 1	Set 2
1	25L 75N 25L	3850	6.147	5.652
2	**25L 50N 50N 25L**	**5125**	**6.048**	**5.504**
3	25L 75N 75N 25L	9550	6.147	5.571
4	25L 75N 4L 75N 25L	4529	6.238	5.658
5	25L 75N 10L 75N 25L	5435	6.154	5.527
6	25L 75N 20L 75N 25L	6945	6.151	5.517

Table 5.2 provide MCD scores for *SLT*-to-*BDL* and *BDL*-to-*SLT* respectively for different architectures of ANN. In this work, we have experimented with 3-layer, 4-layer and 5-layer ANNs. The architectures are provided with the number of nodes in each layer and the activation function used for that layer. For example, an architecture of 25L 75N 25L means that it is a 3-layer network with 25 input and output nodes and with 75 nodes in the hidden layer. Here, L represents "linear" activation function and N represents "tangential ($\tanh$)" activation function. From Table 5.2, we see that the four layered architecture 25L 50N 50N 25L (with 5125 parameters) provides better results when compared with other architectures. Hence, for all the remaining experiments reported in this section, the four layer architecture is used.

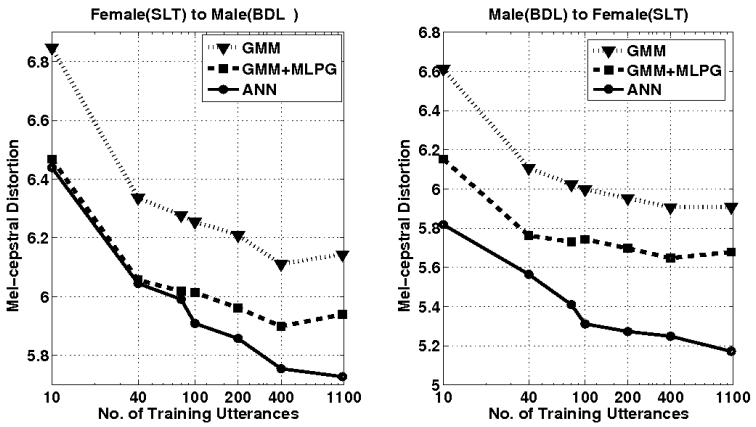


Figure 5.5: MCD scores for ANN, GMM+MLPG and GMM (without MLPG) based VC systems computed as a function of number of utterances used for training. The results for GMM based VC systems are obtained using 64 mixture components.

In order to determine the effect of number of parallel utterances used for training the voice conversion models, we performed experiments by varying the training data from 10 to 1073 parallel utterances. Please note that the number of test utterances was always 59. Figure 5.5 shows the MCD scores for ANN, GMM + MLPG and GMM (without MLPG) based VC systems computed as a function of number of utterances used for training. From Figure 5.5, we could observe that as the number of training utterances increase, the MCD values obtained by both GMM and ANN models decrease.

Subjective evaluation of GMM and ANN based VC systems

In this section, we provide subjective evaluations for ANN and GMM based voice conversion systems. For these tests, we have made use of voice conversion models built from 40 parallel utterances, as it was shown that this modest set produces good enough transformation quality in terms of objective measure. We conducted MOS, ABX and similarity tests to evaluate the performance of the ANN based transformation against the GMM based transformation. It has to be noted that all experiments with GMM use static and delta features but the experiments with ANN use only the static features. A total of 32 subjects were asked to participate in the four experiments listed below. Each subject was asked to listen to 10 utterances corresponding to one of the experiments. The results are presented in Fig. 5.6.

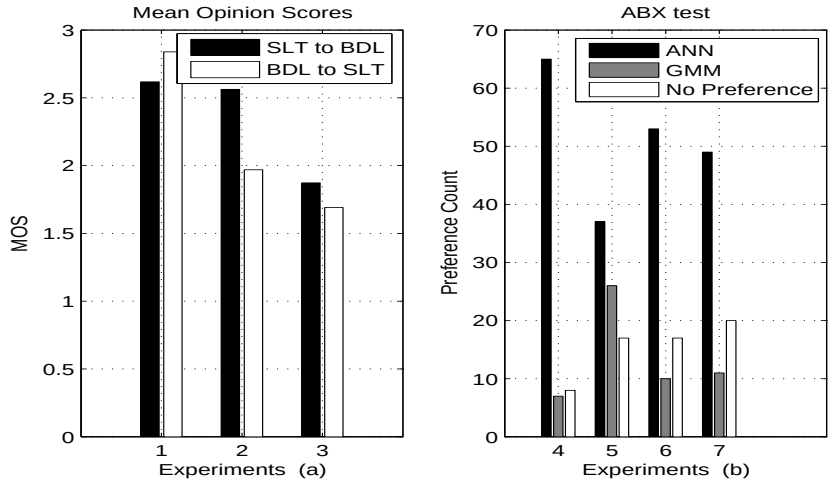


Figure 5.6: **(a)** - MOS scores for 1: ANN, 2: GMM+MLPG, 3: GMM. **(b)** ABX results for 4: ANN vs GMM+MLPG (M->F), 5: ANN vs GMM+MLPG (F->M), 6: ANN vs GMM (M->F), 7: ANN vs GMM (F->M)

Figure 5.6(a) provides the MOS scores of –

1) ANN,

2) GMM + MLPG and

3) GMM (without MLPG) based VC systems.

Figure 5.6(b) provides the results of ABX test of –

4) ANN vs (GMM + MLPG) for *BDL* to *SLT*,

5) ANN vs (GMM + MLPG) for *SLT* to *BDL*,

6) ANN vs GMM for *BDL* to *SLT* and

7) ANN vs GMM for *SLT* to *BDL*.

The MOS scores and ABX tests indicate that the ANN based VC system performs as good as that of the GMM based VC system. The MOS scores also indicate that the transformed output from the GMM based VC system with MLPG was perceived to be better than that of the GMM based VC system without MLPG.

Table 5.3: Average similarity scores between transformed utterances and the natural utterances of the target speakers.

Transformation Method	**Avg. Similarity Score**	
	SLT to *BDL*	*BDL* to *SLT*
ANN	2.93	3.02
GMM + MLPG	1.99	2.56

A similarity test is also performed between the output of the ANN/GMM based VC system and the target speaker's natural utterances. The results of this similarity test are provided in Table 5.3, which indicate that the ANN based VC system seems to perform as good or better as that of the GMM based VC system. The significance of difference between the ANN and the GMM+MLPG based VC systems for MOS and similarity scores was tested using hypothesis testing based on Student t-test, and the level of confidence indicating the difference was found to be greater than 95%.

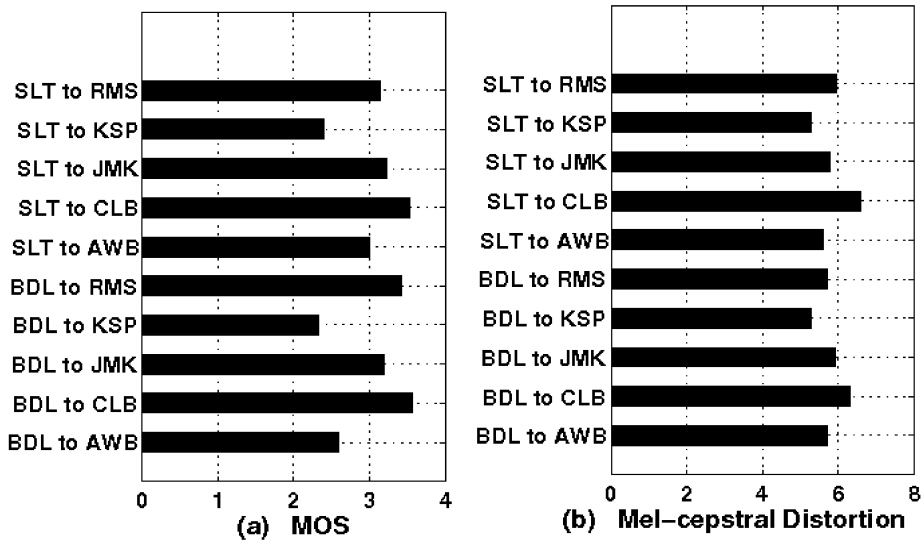


Figure 5.7: **(a)** MOS and **(b)** MCD scores for ANN based VC systems on 10 different pairs of speakers

In order to show that the ANN based transformation can be generalized over different databases, we have provided MOS and MCD scores for voice conversion performed for 10 different pairs of speakers as shown in Figure 5.7. While MCD values were obtained over the test set of 59 utterances, the MOS scores were obtained from 16 subjects, each performing the listening tests on 10 utterances. The MCD scores in Figure 5.7 are in the acceptable range of 5-8, indicating that ANN based VC conversion approach is applicable for different datasets.

A further analysis drawn from these results show that inter-gender voice transformation (ex: male to female) has an average MCD and a MOS score of 5.79 and 3.06 respectively while the intra-gender (ex: male to male) voice transformation has an average MCD and a MOS score of 5.86 and 3.0 respectively. Another result drawn from the above experiments indicates that the transformation performance between two speakers with the same accent is better than that when compared with performance on speakers with different accents. For example, the voice transformation from *SLT* (US accent) to *BDL* (US accent) obtained an MCD value of 5.59 and a MOS score of 3.17, while the voice transformation from *BDL* (US accent) to *AWB* (Scottish accent) obtained an MCD value of 6.04 and a MOS score of 2.8.

5.5 Discussion

In the area of spectral mapping, several approaches have been proposed since the first code book based spectral transformation was developed by Abe et. al. [Abe et al., 1988]. These techniques include artificial neural networks.

Narendranath et. al. [Narendranath et al., 1995] used ANNs to transform the formants of a source speaker to those of a target speaker. Results were shown that the formant contour of a target speaker could be obtained using an ANN model. A formant vocoder was used to synthesize the transformed speech. However, no objective or subjective evaluations were provided to show how good the transformed speech was. The use of radial basis function neural network for voice transformation was proposed in [Watanabe et al., 2002]. The work in [Rao, 2010] also uses ANNs for spectral and prosodic mapping, but relies on additional signal processing for automatic extraction of syllable like regions using pitch synchronous analysis. The method of voice conversion used in this thesis differs from these earlier approaches in the following ways –

- Earlier approaches used either a carefully prepared training data which involved manual selection of vowels and syllable regions [Narendranath et al., 1995] [Watanabe et al., 2002] or signal processing algorithms to locate syllable like regions [Rao, 2010]. The proposed approach in this work needs neither manual effort nor signal processing algorithms to locate syllable like regions. Our approach makes use of a set of utterances provided from a source and a target speaker and automatically extracts the relevant training data using dynamic programming to train a voice conversion model.
- In previous works, there have been no comparative studies to evaluate how an ANN based VC system performs in comparison with other approaches. In this work, we have compared ANN based approach to a widely used GMM based approach. In this comparative study, we have shown that ANN based voice conversion performs as good as that of a GMM based conversion. Subjective and objective measures are conducted to evaluate the usefulness of ANNs for voice conversion. The MCD scores of ANN based voice conversion are in the acceptable range of 5-8.

Our work on ANNs differ from GMMs in the following ways –

- GMM based conversion is a linear transformation. The number of parameters required for a 64-component GMM is 12,352 (see Table 5.1). In contrast,

ANN based transformation is nonlinear. The number of parameters in the four layer architecture of the ANN model is 5,125 (see Table 5.2). A reduction of 58% in the number of parameters indicates lesser number of floating point computations. This makes ANN more suitable for deployment in low power hand held devices.

- GMM based VC systems depend on MLPG for smoothing the trajectories. This smoothing is done as a postprocessing technique for the transformed utterance. Thus, GMMs require a context of at least 500-1000 milliseconds to perform MLPG based smoothing. In contrast, ANN based VC systems do not need any postprocessing technique. Both the subjective and objective evaluations show that ANN based VC transformation is better than GMM+MLPG. We believe this is a characteristic of the nonlinear mapping function of an ANN model, which seems to capture correlation between frames, implicitly during training. This makes it convenient to use ANN models in building on-line voice conversion systems (i.e., conversion on the fly without buffering any previous frames).

Finally, we should note that both ANN and GMM based VC techniques depend on collection of parallel data – the same set of utterances recorded by both the source and target speakers. Such collection may not always be feasible. In the next chapter, we primarily address this issue and propose methods which avoid parallel data for building voice conversion models.

5.6 Summary

In this chapter, we have discussed techniques for conversion of speaker characteristics, so that utterances sound as if spoken by a target speaker. We have developed an ANN based voice conversion model and evaluated it on several speakers of CMU ARCTIC datasets. We have also demonstrated that the performance of an ANN based voice conversion system is good as that of a GMM based conversion system. An important limitation of techniques presented in this chapter is requirement of parallel data for building voice conversion models.

Chapter 6

Modeling target speaker characteristics

So far we have discussed voice conversion (VC) approaches which rely on existence of parallel data, i.e, the source and the target speakers record the same set of utterances. Availability of such parallel data enables deriving a relationship between utterances of source and target speakers at a phone/frame level and build a VC system.

Availability of parallel data is not always feasible. For example, it is difficult to summon a celebrity or famous personality to record a parallel set of utterances. At the same time, if the languages spoken either the source and target speakers are different, then there is no possibility of recording a same set of utterances. However, clustering techniques could be used to derive a relationship between utterances of the source and target features at phone or sound level [Sundermann et al., 2003], [Sundermann et al., 2006]. For example – using the k-means clustering algorithm, the spectral vectors of the target speaker are segmented into M clusters. Similarly, the spectral vectors of the source speaker are segmented into N clusters. Now, by comparing the centroids of the source and target clusters, one can find phonetically equivalent source and target clusters. This enables in deriving pseudo-parallel data between the source and target speakers, and to train a voice conversion model. Another method is to train a voice conversion model on pre-existing parallel datasets. Speaker adaptation techniques are then used to adapt this voice conversion model to a particular pair of source and target speakers for which no parallel data is available [Mouchtaris et al., 2006].

While these methods avoid the need for parallel data, they still require speech

data (though non-parallel) from the source speakers *a priori* to build voice conversion models. This is a limitation to an application where an arbitrary user intends to transform his/her speech to a pre-defined target speaker without recording anything *a priori*.

6.1 Research question and challenges

Current voice conversion techniques focus on obtaining an optimal mapping function between the source and target speakers. The central research question here is – "How to obtain an optimal mapping function which transforms the acoustic hints of a source speaker to that of a target speaker?". This research question assumes that both the source and target speakers' data is available (either in parallel or in pseudo-parallel form), and the unsolved part of the riddle is just the optimal mapping function.

As discussed previously, the assumption of existence of parallel or pseudo-parallel data is not valid for many practical applications. Hence, we posed an alternative but relevant research question, which is – "How to capture speaker-specific characteristics of a target speaker from the speech signal (independent of any assumptions about a source speaker) and impose these characteristics on the speech signal of any arbitrary source speaker to perform voice conversion?". The problem of capturing speaker-specific characteristics can be attempted in one of the following ways –

- A new signal processing technique motivated by the speech production process could be developed to extract features specific to a speaker. This solution is hard, and needs breakthroughs in explicitly identifying the speaker-specific information in a speech signal.

- An autoassociative neural network (AANN) model can be trained to capture a speaker-specific distribution of feature vectors by performing an identity mapping of target speaker's acoustic space [Ikbal et al., 1999], [Yegnanarayana and Prahallad, 2002], [Joshi et al., 2008]. As shown in Fig. 6.1, the architecture of such an AANN model consist of five layers – an input layer, expansion layer, bottleneck layer, expansion layer and the output layer. During the process of training the AANN model to perform an identity mapping, it essentially captures a lower dimensional subspace (due to bottleneck layer) characterizing the target speaker. This property of bottleneck layer can be exploited for voice

conversion. To perform voice conversion, the feature vectors of an arbitrary source speaker can be projected onto the lower dimensional subspace using the bottleneck layer of the target speaker's AANN model. Given that, this lower dimensional subspace is specific to the target speaker [Yegnanarayana and Prahallad, 2002], the reconstruction from the subspace is expected to bear the identity of the target speaker.

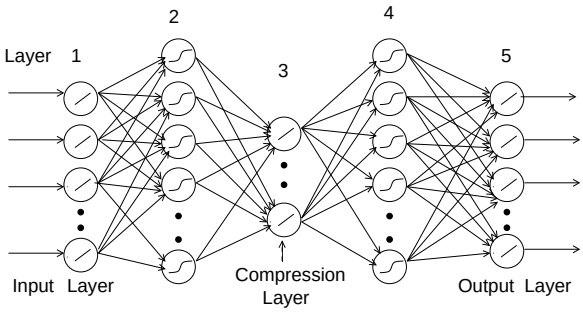


Figure 6.1: A Five layer AANN model

- The problem of capturing speaker-specific characteristics can be viewed as modeling a noisy-channel. Suppose, C is a canonical form of a speech signal – a generic and speaker-independent representation of the message in speech signal, passes through the speech production system of a target speaker to produce a surface form S. This surface form S carries the message as well as the identity of the speaker.

  One can interpret S as the output of a noisy-channel, for the input C. Here, the noisy-channel is the speech production system of the target speaker. The schematic diagram of this noisy-channel model is shown in Fig. 6.2.

Figure 6.2: Noisy channel model for capturing speaker-specific characteristics.

The mathematical formulation of this noisy-channel model is –

$$\underset{S}{\operatorname{argmax}}\, p(S/C) = \underset{S}{\operatorname{argmax}}\, \frac{p(C/S)p(S)}{p(C)} \tag{6.1}$$

$$= \underset{S}{\operatorname{argmax}}\, p(C/S)p(S), \tag{6.2}$$

as $p(C)$ is constant for all S. Here $p(C/S)$ could be interpreted as production model. $p(S)$ is the prior probability of S and it could be interpreted as the continuity constraints imposed on the production of S. It could be seen analogous to a language model of S.

6.2 Capturing speaker-specific characteristics

In this thesis, we use noisy-channel model approach for capturing speaker-specific characteristics of a target speaker. Here $p(S/C)$ is directly modeled as a mapping function between C and S using artificial neural networks. There have been similar efforts earlier to capture speaker-specific characteristics. Gong and Haton [Gong and Haton, 1992] proposed to capture a speaker-specific mapping function using a nonlinear vector interpolation model. Hermansky *et. al.,* and Misra *et. al.,* have provided a more rigorous experimentation and interpretation for capturing speaker-specific mappings using artificial neural networks [Hermansky and Malayath, 1998], [Misra et al., 2003]. Our work differ from these as follows –

- As described in Section 6.1, the problem of capturing speaker-specific characteristics is formulated as modeling a noisy-channel. Such interpretation is not provided in these earlier works. The formulation of noisy channel is explained in Eq. (6.1). As a result of this formulation, one can model $p(C/S)$ and $p(S)$ using Markov models to obtain $p(S/C)$[1].

- Gong *et. al.,*, Hermansky *et. al.,* and Misra *et. al.,* have applied the concept of capturing speaker-specific mappings for the task of speaker recognition. In this work, we apply it for the task of voice conversion.

The process of capturing speaker-specific characteristics and its application to voice conversion is explained below –

Suppose, we derive derive two different representations C and S from the speech signal with the following properties.

- $C \neq S$, and there may or may not be a redundancy between C and S.

[1] In the scope of this thesis, we chose to model $p(S/C)$ directly using artificial neural networks – this is coherent with the experiments in previous chapters. The implementation of $p(C/S)$ and $p(S)$ using Markov models can be a potential future work of this thesis.

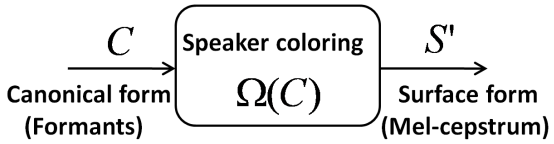


Figure 6.3: Capturing speaker-specific characteristics as a speaker-coloring function

- There exists a function $\Omega(.)$, such that an approximation S' of S can be obtained from C, $S' = \Omega(C)$.

Suppose, C is a canonical form of speech signal - a generic and speaker-independent form - approximately represented by first few formant frequencies and their bandwidths, and S is a surface form represented by Mel-cepstral coefficients (MCEPs). It can be argued that the C and S have redundant information of formant frequencies, and - by borrowing the knowledge from speaker recognition studies [Jin et al., 2007] - it is safe to assume that S has additional information of speaker characteristics. If there exists a function $\Omega(.)$ such that $S' = \Omega(C)$, where S' is an approximation of S - then $\Omega(C)$ can be considered as specific to a speaker. The function $\Omega(.)$ could be interpreted as speaker-coloring function. We treat the mapping function $\Omega(.)$ as capturing speaker-specific characteristics. It is this property of $\Omega(.)$, we exploit for the task of voice conversion. Fig. 6.3 depicts the concept of capturing speaker-specific characteristics as a speaker-coloring function. It should be noted that other representations of canonical form (C) include articulatory features and acoustic-phonetic features. However, due to simplicity and availability of existing tools for extracting formant features, we have chosen to experiment with formant frequencies in this thesis.

6.3 Application to voice conversion

Given the utterances from a target speaker T, the corresponding canonical form C_T of the speaker is represented by a number of formants, their bandwidths and delta features. One question is – how many formants to be used and whether the extraction of formants is reliable?. After experimenting with four and six formant frequencies, we found that the six formant frequencies are better suited for our purposes (see Section 6.4 and Table 6.1). The formant frequencies, bandwidths, fundamental frequency F_0 and probability of voicing are extracted using the ESPS toolkit [ESPS, 2009]. This is a widely used toolkit and provides reasonably good estimate of formant frequencies.

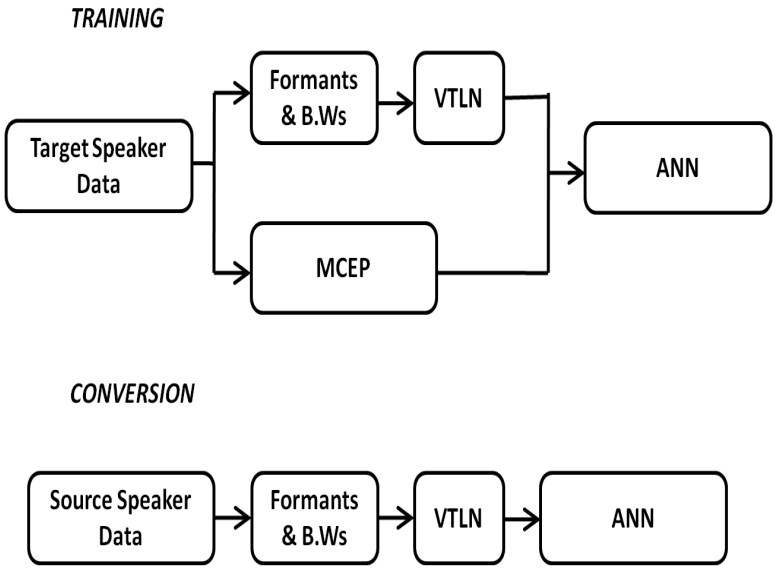


Figure 6.4: Flowchart of training and conversion modules of a VC system capturing speaker-specific characteristics. Notice that during training, only the target speaker's data is used.

To alleviate the effect of speaker characteristics, the formant features undergo a normalization technique such as vocal tract length normalization as explained in Section 6.3.1. The surface form S_T is represented by traditional MCEP features, as it would allow us to synthesize using the MLSA synthesis technique. The MLSA synthesis technique generates a speech waveform from the transformed MCEPs and F0 values using pulse excitation or random noise excitation [Imai, 1983]. An ANN model is trained to map C_T to S_T using backpropagation learning algorithm to minimize the Euclidean error $||S_T - S_T^{'}||$, where $S_T^{'} = \Omega(C_T)$. . Once the model is trained, it could be used to convert C_R to $S_T^{'}$ where C_R could be from any arbitrary speaker R. A schematic diagram of training and conversion modules is shown in Fig. 6.4. Notice that during training, only the target speaker's data is used.

6.3.1 Vocal tract length normalization

Vocal tract length normalization (VTLN) is a speaker normalization technique that tries to compensate for the effect of speaker-dependent vocal tract lengths by warping the frequency axis of the magnitude spectrum. Apart from use in speech recognition, VTLN has also been used in voice conversion [Sundermann et al., 2004], [Sundermann et al., 2003], [Sundermann et al., 2006].

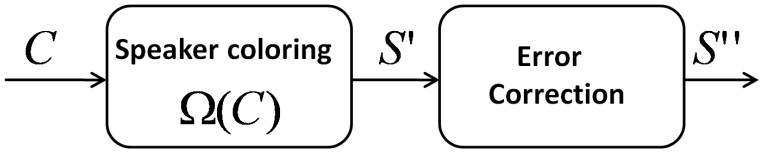


Figure 6.5: Integration of an error correction network with the speaker-coloring network.

Following the work in [Faria, 2003], we estimate the warp factors using pitch information and modify both formants and bandwidths. A piece-wise linear warping function as described in [Faria, 2003] is used in this work. The features representing C undergo a VTLN, to normalize the speaker effect.

6.3.2 Error correction network

We introduce a concept of an error correction network which is essentially an additional ANN network, used to map the predicted MCEPs to the target MCEPs so that the final output obtained features represent the target speaker in a better way. The integration of the error correction network with the speaker-coloring network is shown in Figure 6.5. Once $S_T^{'}$ is obtained, it is given as input to the second ANN model. Let $S_T^{''}$ denote the output of this second ANN model. It is trained to reduced the error $||S_T^{''} - S_T||$. Such a network acts as an error correction mechanism to correct any errors made by the first ANN model. It is observed that while the MCD values of $S_T^{'}$ and $S_T^{''}$ do not vary much, the speech synthesized from $S_T^{''}$ was found to be smoother than that of speech synthesized from $S_T^{'}$. To train the error correction network, we use 2-D features i.e., feature vectors from 3 left frames, and 3 right frames are added as context to the current frame. Thus the ANN model is trained with 175 dimensional vector (25 dimension MCEPs * (3+1+3)). The architecture of this error correction network is 175L 525N 525N 175L.

6.4 Experiments using parallel data

As an initial experiment, we used parallel data, where R = *BDL* and T = *SLT*. Features representing C_R were extracted from the *BDL* speaker and were mapped onto the S_T of *SLT*. This experimentation was done to obtain a benchmark performance for the experiments which map C_T to S_T (as explained in Section 6.4.1).

The features representing C undergo a VTLN (as discussed in Section 6.3.1),

Table 6.1: Results of source speaker (*SLT*-female) to target speaker (*BDL*-male) transformation with training on 40 utterances of source formants to target MCEPs on a parallel database. Here **F** represents Formants, **B** represents Bandwidths, Δ and $\Delta\Delta$ represents delta and delta-delta features computed on **ESPS** features respectively. **UVN** represents unit variance normalization.

S.No	Features	ANN architecture	MCD
1	4 F	4L 50N 12L 50N 25L	9.786
2	4 F + 4 B	8L 16N 4L 16N 25L	9.557
3	4 F + 4 B + UVN	8L 16N 4L 16N 25L	6.639
4	4 F + 4 B + Δ + $\Delta\Delta$ + UVN	24L 50N 50N 25L	6.352
5	F_0 + 4 F + 4 B + UVN	9L 18N 3L 18N 25L	6.713
6	F_0 + 4 F + 4 B + Δ + $\Delta\Delta$ + UVN	27L 50N 50N 25L	6.375
7	F_0 + Prob. of Voicing + 4 F + 4 B + Δ + $\Delta\Delta$ + UVN	30L 50N 50N 25L	6.105
8	**F_0 + Prob. of voicing + 6 F + 6 B + Δ + $\Delta\Delta$ + UVN**	**42L 75N 75N 25L**	**5.992**
9	**(F_0 + Prob. of voicing + 6 F + 6 B + Δ + $\Delta\Delta$ + UVN) + (3L3R MCEP to MCEP error correction)**	**(42L 75N 75N 25L) + (175L 525N 525N 175L)**	**5.615**

to alleviate the speaker effect. However, in this experiment, the mapping is done between *BDL*'s C_R to *SLT*'s S_T. The process of training such a voice conversion model is similar to the process explained in Section 5.4. In Section 5.4, the features of *BDL* speaker were represented by MCEPs, where as in this experiment, the formants and bandwidths are used. The results obtained in this section could also be compared with the results obtained in Section 5.4. Hence, VTLN was not performed on the features representing C_R in this experiment.

Training was done to map *BDL*-formants to *SLT*-MCEPs with 40 utterances. Testing was done on a set of 59 utterances. Table 6.1 shows the different representations of C_R and their effect on MCD values. These different representations include combination of different number of formants and their bandwidths, delta and acceleration coefficients of formants and bandwidths, pitch and probability of voicing. From the results provided in Table 6.1, we can observe that experiment 9 (which uses six formants, six bandwidths, probability of voicing, pitch along with their delta and acceleration coefficients) employing an error correction network provided better results in terms of MCD values. These results are comparable with the results of voice conversion with *BDL*-MCEPs to *SLT*-MCEPs mapping as found in Section 5.4.

6.4.1 Experiments using non-parallel data

In this experiment, we built an ANN model which maps C_T features of *SLT* onto S_T features of *SLT*. Here C_T extracted from *SLT* utterances is represented by six

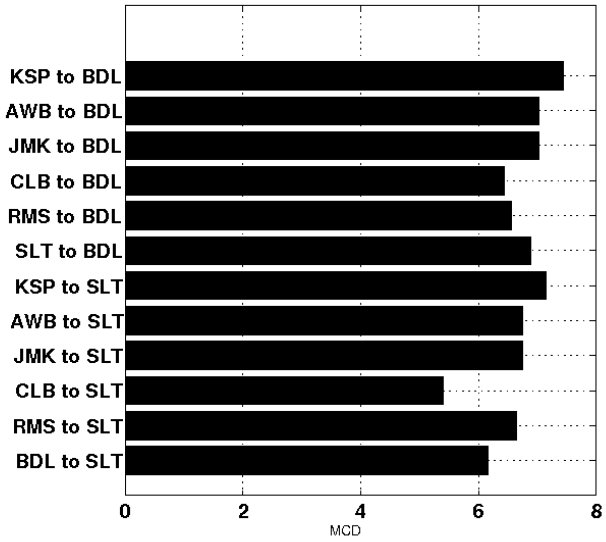


Figure 6.6: A plot of MCD scores obtained between multiple speaker pairs with *SLT* or *BDL* as the target speaker. The models are built from a training data of 24 minutes and tested on 59 utterances (approximately 3 min).

formants, six bandwidths, F_0, probability of voicing and their delta and acceleration coefficients as shown in feature set for experiment 9 in Table 6.1. The formants and bandwidths representing C_T undergo VTLN to normalize the speaker effects. S_T is represented by MCEPs extracted from *SLT* utterances. We use the concept of error correction network to improve the smoothness of the converted voice.

Figure 6.6 provides the results for mapping C_R (where R = *BDL*, *RMS*, *CLB*, *JMK* voices) onto the acoustic space of *SLT*. To perform this mapping the voice conversion model is built to map C_T to S_T (where T = *SLT*) is used. To perform VTLN, we have used the mean pitch value of *SLT*. Hence all the formants of source speaker are normalized with VTLN using mean of *SLT* F_0 and then are given to ANN to predict the 25 dimensional MCEPS. Similar results where the voice conversion model is built by capturing *BDL* speaker-specific features are also provided in Figure 6.6.

We also performed listening tests whose results are provided in Table 6.2 for MOS scores and similarity tests. For the listening tests, we chose 3 utterances randomly from each of the transformation pairs. Table 6.2 provides a combined output of all speakers transformed to the target speaker (*SLT* or *BDL*). There were 10 listeners who participated in the evaluations tests. The MOS scores and

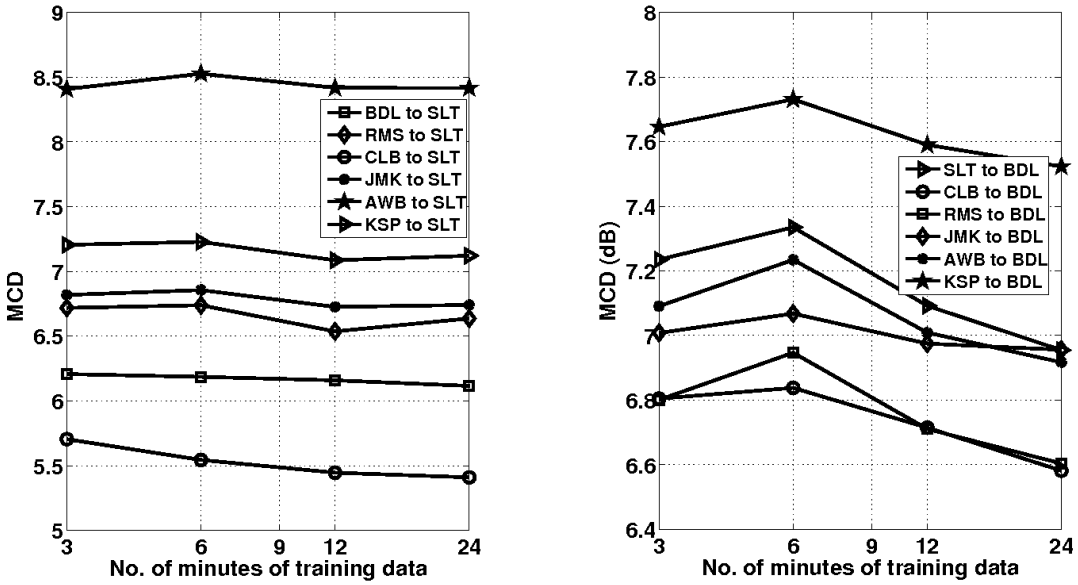


Figure 6.7: A plot of MCD v/s Data size for different speaker pairs, with *SLT* or *BDL* as the target speaker.

Table 6.2: Subjective evaluation of voice conversion models built by capturing speaker-specific characteristics

Target Speaker	MOS	Similarity tests
BDL	2.926	2.715
SLT	2.731	2.47

similarity test results are averaged over 10 listeners. The results shown in Figure 6.6 and Table 6.2 indicate that voice conversion models built by capturing speaker-specific characteristics using ANN models are useful. Figure 6.7 shows the effect of amount of training data in building the ANN models capturing speaker-specific characteristics. It could be observed that the MCD scores tend to decrease with the increase in the amount of training data.

To validate the proposed method on more number of speakers, we conducted experiments on other speakers from the ARCTIC set, such as *RMS*, *CLB*, *JMK*, *AWB* and *KSP*. The training for all these experiments was conducted on 6 minutes of speech data. However, the testing was done on the standard set of 59 utterances. The MCD scores provided in Table 6.3 are in the acceptable range of 5-8. This indicates that the methodology of training an ANN model to capture speaker-specific

characteristics for voice conversion could be generalized over different datasets. However, it should be noted that these MCD scores are higher in comparison with that of obtained from parallel data (see Fig. 5.7 in Section 5.4). This suggests that if parallel data is available, then it should be preferred to build a voice conversion model using parallel data.

Table 6.3: Performance of voice conversion models built by capturing speaker-specific features are provided with MCD scores. Entries in the first column represent source speakers and the entries in the first row represent target speakers. All the experiments are trained on 6 minutes of speech and tested on 59 utterances or approximately 3 minutes of speech.

Source \ Target	*RMS*	*CLB*	*AWB*	*KSP*
RMS	-	6.716	6.251	6.891
CLB	7.066	-	6.297	7.166
AWB	6.847	6.517	-	6.769
KSP	7.392	7.239	6.517	-
JMK	6.617	6.616	6.224	6.878
BDL	6.260	6.137	6.558	6.820
SLT	7.430	5.791	6.354	7.278

6.5 Application to cross-lingual voice conversion

Cross-lingual voice conversion is a task where the language of the source and the target speakers is different. In the case of a speech-to-speech translation system, a source speaker may not know the target language. Hence, to convey information in his/her voice in the target language, cross-lingual voice conversion assumes importance. The availability of parallel data is difficult for cross-lingual voice conversion. One solution is to perform a unit selection approach [Sundermann et al., 2004], [Sundermann et al., 2003], [Sundermann et al., 2006] to find units in the utterances of the target speaker that are close to the source speaker or use utterances recorded by a bi-lingual speaker [Mouchtaris et al., 2006]. Our solution to cross-lingual voice conversion is to employ the ANN model which captures speaker-specific characteristics. In this context, we performed an experiment to transform three female speakers (*NK, PRA, LV*) speaking Telugu, Hindi and Kannada respectively into a male voice speaking English (US male - *BDL*). Our goal here

Table 6.4: Subjective results of cross-lingual transformation. Utterances from *NK* speaking Telugu, *PRA* speaking Hindi and *LV* speaking Kannada are transformed to sound like *BDL*.

Source Speaker (Lang.)	Target Speaker (Lang.)	MOS	Similarity tests
NK (Telugu)	*BDL* (Telugu)	2.88	2.77
PRA (Hindi)	*BDL* (Hindi)	2.62	2.15
LV (Kannada)	*BDL* (Kannada)	2.77	2.22

is to transform *NK, PRA* and *LV* voices to *BDL* voice and hence the output will be as if *BDL* were speaking in Telugu, Hindi and Kannada respectively. We make use of *BDL* models built in Section 6.4.1 to capture speaker-specific characteristics. Ten utterances from *NK, PRA, LV* voices were transformed into *BDL* voice and we performed MOS test and similarity test to evaluate the performance of this transformation. Table 6.4 provides the MOS and similarity test results averaged over all listeners. There were 10 native listeners of Telugu, Hindi and Kannada who participated in the evaluations tests. The MOS scores in Table 6.4 indicate that the transformed voice was intelligible. The similarity tests indicate that cross-lingual transformation could be achieved using ANN models, and the output is intelligible and possesses the characteristics of *BDL* voice.

6.6 Summary

In this chapter, we have shown that it is possible to build a voice conversion model by capturing speaker-specific characteristics of a speaker. We have used an ANN model to capture the speaker-specific characteristics. Such a model does not require any speech data from source speakers and hence could be considered as independent of a source speaker. We have also shown that an ANN model capturing speaker-specific characteristics could be applied for cross-lingual voice conversion. A set of transformed utterances corresponding to results of this work is available for listening at `http://bit.ly/vctaslp`

In this chapter, we have used formant frequencies and their bandwidths to represent the canonical form of a speech signal. However, this representation may not be optimal. Other representations such as articulatory and acoustic-phonetic features need to be experimented. The application of ANN models capturing speaker-specific characteristics for cross-lingual voice conversion raises several research

issues. These include transformation of source sounds unknown to the target speaker. Studies have to be performed to compare the cross-lingual transformation of ANN models and (bi-lingual) human subjects.

Chapter 7

Concluding words

In this thesis, we have addressed the issues in development of voices from audio books; segmentation of monologues in audio books; modeling speaker-specific phrase patterns and conversion of speaker-characteristics in synthesized utterances. The major contributions of this thesis are as follows.

- **INTERSLICE:** To segment a long speech file, we have proposed modifications to the Viterbi algorithm. These modifications are implemented as a package referred to as INTERSLICE in FestVox - which is an open source tools for building synthetic voices. INTERSLICE segments long speech files without the need for a speech recognition system. Thus the proposed approach is also suitable for languages (especially low resource languages) with no availability of large vocabulary speech recognition.

- **An unsupervised algorithm for learning speaker-specific phrase breaks:** In this thesis, we have proposed an unsupervised algorithm to learn speaker-specific phrase breaks. The proposed algorithm consists of two phases. In phase-I, an hypothesis is made about the location of phrase breaks using pauses as acoustic cues. In phase-II, the hypothesized regions of phrase breaks are treated as labeled data. Features based on F0, duration and energy are extracted from these regions to build a classifier which labels each word with the class of break or not-a-break. This phrase break classifier is further bootstrapped with the rest of unlabeled data. By empirical evidence, we have shown that speaker-specific phrase breaks improves the quality of synthetic voices.

- **A method for modeling target speaker characteristics:** In this thesis, we have addressed the issue of personalization of synthetic voices by using voice conversion models. The question we address is - "can we capture speaker-specific characteristics of a target speaker from the speech signal (independent of any assumptions about a source speaker) and super-impose these characteristics on the speech signal of any arbitrary source speaker to perform voice conversion?". We have proposed a method to capture speaker-specific characteristics of a target speaker using an ANN model and avoid the need for speech data from a source speaker to train/adapt a voice conversion model.

7.1 Conclusions

- **Audio books in the public domain can be used for building synthetic voices.** It is known that story speech databases have rich prosody. Prior to this work, it was unclear whether the quality of audio books in public domain would be suitable for building synthetic voices. As a part of this thesis, we have observed that audio books - although in public domain, recorded and maintained by volunteers - offer excellent candidates for building synthetic voices. These books have a low disfluency rate, speech-to-noise ratio of around 30 dB and provide a large amount of speech data by a single speaker. We have shown that synthetic voices built from these audio books have Mel-cepstral distortion scores (an objective measure to evaluate the quality of synthetic voices) in the range of 4-7 dB. This is an acceptable range, and is similar to scores for voices built from high quality studio recordings.

- **Segmentation of long speech files can be accomplished without ASR.** To build voices from audio books, segmentation of monologues - long speech files - is an issue. Existing methods pose segmentation of long speech files as an automatic speech recognition (ASR) problem. A long speech file is broken into chunks, and each chunk is transcribed using an ASR with an adaptive and restrictive language model. In spite of search space being restricted, the transcription obtained from an ASR is not always error-free, especially at chunk boundaries. Hence, a post-processing stage is involved by aligning the original text with the obtained transcription. Apart from practical difficulty in implementing this approach (in the context of a TTS system), it strongly implies that a speech recognition system should be readily available before building a speech synthesis system. In this thesis, we have shown

that the segmentation of long speech files can be accomplished by suitable modifications to the Viterbi algorithm. These modifications allow to process a long speech file in parts, without any need for a speech recognition system.

- **Prosodic phrase breaks are specific to a speaker.** Prosodic phrase breaks are essential for comprehension and naturalness of utterances. Existing methods learn phrasing patterns on a standard corpus. For example, in Festival, a default phrasing model for English trained on the Boston University Radio corpus is employed to predict breaks for all English voices. Thus prosodic phrasing patterns are generalized across all voices while ignoring speaker-specific phrasing patterns. In this thesis, through empirical evidence we have shown that prosodic phrase breaks are specific to a speaker. This implies that speaker-specific phrasing patterns need to be modeled in TTS systems.

- **Speaker-specific phrase breaks improve the quality of synthetic voices.** In this thesis, we have shown that incorporation of speaker-specific phrase breaks improves the quality of synthetic voices. Studies have been conducted on multiple voices in English and Telugu. The experimental results show that speaker-specific phrase breaks improve the spectral quality in comparison with generic prosodic phrase breaks.

- **Artificial neural network (ANNs) based voice conversion performs as good as that of a Gaussian mixture model (GMM) based voice conversion.** Conversion of speaker-characteristics is one of several ways of personalizing a synthetic voice to listeners. Traditionally, Gaussian mixture models are used for voice conversion. In this thesis, we have shown that an ANN based voice conversion performs as well as or sometimes better than a GMM based voice conversion.

- **To build a voice conversion model, it is not necessary to have parallel or pseudo-parallel data.** To build a voice conversion model, current methods require a same set of utterances recorded by the source and target speakers. In case of non-availability of parallel data, pseudo-parallel data is derived by looking for similar sounds in the source and target speakers' recordings. However, such techniques do not allow a random unseen source speaker to convert his/her voice to a target speaker. In this thesis, we have shown that it is possible to capture speaker-specific characteristics of a target speaker independent of source speakers. Such method is shown to perform monolingual as well as cross-lingual voice conversion of any arbitrary speaker.

- **Enabling prosody research by leveraging audio books.** Finally, the techniques developed in this thesis enable prosody research by leveraging a large number of audio books available in the public domain. We believe, this is an important milestone in prosody modeling and in building natural sounding synthetic voices.

7.2 Future work

- **Modeling prosody in TTS**: Audio books encapsulate rich prosody including intonation contours, pitch accents and phrasing patterns. As a result of this work, we have a number of audio books segmented into paragraph length utterances. Several interesting research questions can be investigated on this outcome.
  - How to model pitch accents which make words perceptually prominent (as in, *I didn't shoot AT him, I shot PAST him*?).
  - How to model variants of intonation such as wh-questions (*what time is it?*), unfinished statements (*I wanted to* ..), yes/no questions (*Are they ready to go?*) and surprise (*What! The plane left already!?*) present in these audio books? This would also require some amount of text understanding to model such varied intonation. Another dimension on this issue is - whether the modeling algorithms should be domain and language dependent or independent?
  - What is the right level of unit to capture the pitch accents and intonation patterns? Is it at syllable, multi-syllable such as mora/foot, word or phrase level? It is also important to investigate long range dependencies such as sentence/paragraph coloring on F0 contours.
  - What is an appropriate representation of intonation contours? For example, TILT is a model of intonation that represents intonation as a sequence of continuously parameterized intonation events [Taylor, 2000]. The basic types of intonation events are pitch accents and boundary tones, and they are parameterized using rise/fall/connection (RFC) model. In the RFC model, the rise and fall parts are parameterized using amplitude, duration and tilt which expresses the overall shape of the event. Other possible representations include curve fitting, regression and sinusoidal modeling of intonation contours.

  - How to represent and parameterize intonation contours across sentences, and at a paragraph level? A paragraph is often defined as expression of a single thought or character's continuous words. Could intonation and duration be modeled at a character level in a story?
  - Another equally important aspect is to seek techniques for predicting appropriate intonation, duration and phrasing patterns from text. What is the right set of lexical and contextual features that is useful for predicting intonation, duration and phrasing patterns from text?
  - Evaluation of prosodic models is another challenging research topic. What are the objective and subjective measures to compare and evaluate two prosodic models? Traditional listening tests might be hardly useful. Hence, innovative ways of seeking listener's preference have to be sought. This could include character level prosody modeling in a story, and evaluation of a character by the listeners.

- **Language-Independent Models:** To build INTERSLICE, we have used speaker-independent HMMs. To extend INTERSLICE to a new language, we need to have a set of speaker-independent acoustic models in the new language. It is important to note that our major interest in INTERSLICE lies in obtaining beginning and ending of utterances in long speech files. Thus, it would be interesting to build language-independent acoustic models and use them in INTERSLICE. Globalphone set is a good example of language-independent acoustic models [Schultz and Waibel, 1998].

- **Detection of mispronunciation:** In this thesis, we have considered only high quality audio books, which have a low disfluency rates. However, this may not be the case for all audio books. During the recordings, a speaker might delete or insert at syllable, word, sentence level and thus the speech signal does not match with the transcription. It is important to detect these mispronunciations using acoustic confidence measures so that the specific regions or the entire utterances can be ignored while building voices.

- **Detection of pronunciation variants:** Speakers may incorporate subtle variations at the sub-word during pronunciation of content words, proper nouns etc. and these pronunciation variants have to be detected and represented so that they could be produced back during synthesis.

- **Filtering:** Often recordings may have multiple sources, thus filtering of multi-speakers data, music and speech and nullifying the noisy or channel effects may be needed.

- **Modeling prosody and excitation in voice conversion:** Current voice conversion techniques rely mostly on spectral transformation. Prosodic features such as intonation and duration patterns also play an important role in characterizing a speaker. Given parallel data between the source and target speakers, prosodic transformation is attempted on similar lines of spectral transformation [Toth and Black, 2008],[Rao, 2010]. However, the issue of capturing intonation and duration features of a speaker in the absence of parallel data is still an open question.

  Another important aspect is the excitation modeling in voice conversion. Current techniques represent excitation using F0 and a linear transformation is performed based on mean and variance of F0 of a target speaker. Studies in [Kain and Macon, 2001] have shown that excitation information plays a role in perceiving naturalness and speaker characteristics. Hence, it is important to investigate methods for better representation and transformation of excitation.

- **Better representation for speaker-specific mapping:** In this thesis, we have attempted to capture speaker-specific features by formulating it as a mapping from a lower dimensional space to a higher dimensional space. This approach avoids any need for source speaker's data *a priori*. The lower dimensional features are interpreted as speaker-independent message part of the signal, where as the higher dimensional features as speaker-dependent message part of the speech signal. Formant frequencies and bandwidths have been used to represent the lower dimensional features, where as traditional MCEPs are used to represent the higher dimensional feature. However, formant features may not be the best representation. Features such as articulatory parameters need to be investigated for representing speaker-independent message part of the signal.

Appendix A

Extraction of features from a speech signal

To extract the feature vectors from a speech signal, the characteristics of the speech signal are assumed to be stationary over a short duration of time (between 10-30 ms). The speech signal is pre-emphasized using a difference operator and is divided into frames of 10 ms using a frame shift of 5 ms. Each frame of speech data is passed through a Hamming window and then through a set of Mel-Frequency filters to obtain 13 cepstral coefficients. Thus each frame of speech data is represented by a vector of 13 coefficients Rabiner and Juang [1993].

Appendix B

Acoustic models

In this work, the acoustic models used in forced-alignment of large audio files are built using about four hours of speech data collected from four CMU ARCTIC speakers (RMS, BDL, SLT and CLB). These acoustic models are context-independent (CI) HMM models where each phone has three emitting states and two null states. The states in a phone HMM are connected in left-to-right fashion with out any skip arcs. The exception is *pau*, a silence HMM, where a skip arc is provided to optionally omit the middle emitting state. Each state is modeled by a two component Gaussian mixture model. Each Gaussian component is modeled by a 13-dimensional mean vector and a diagonal covariance matrix. The HMM models were initialized using a flat start and were trained using Baum-Welsh re-estimation algorithm.

Appendix C

CLUSTERGEN

Parametric synthesis techniques process the speech signal to derive parameters representing the speech signal and synthesize speech using these parameters. However, in traditional parametric synthesis methods, the parameters are derived manually and rules are prepared manually to incorporate co-articulation and prosody. Statistical parametric synthesis methods differ from the traditional parametric by using machine learning techniques to learn the parameters and any associated rules of co-articulation and prosody from the data.

One of the early works on statistical parametric synthesis is based on Hidden Markov Models (HMMs). The basic idea is to extract MCEP vectors features from the speech signal and build a set of context dependent models at sub-phonetic level often referred to as senones in speech recognition. During synthesis, a sequence of sub-phonetic states are obtained for the text to be synthesized. The distributions of these sub-phonetic states are used to derive a sequence of Mel-cepstral (MCEP) vectors maximizing the likelihood. If the mean vectors of the distributions are used then they would maximize the likelihood but produce step-wise discontinuities. Hence delta and delta-delta cepstral are used as constraints to obtain a smooth sequence of Mel-cepstral vectors. The approach to generate speech from Mel-cepstrals is similar to vocoder. The Mel-cepstrals are passed Mel Log Scale Spectral Approximator (MLSA) and are excited with white noise or pulse train to generate the speech signal [Zen et al., 2006].

In CLUSTERGEN [Black, 2006], decision trees are used to model the parameters. The parameters used are MCEP, duration and F0. A decision tree is built separately for each feature stream. The novelty of this approach lies in modeling the MCEP parameters at each frame level. Thus given the duration of each sub-phonetic model,

a decision tree would predict the most likely MFCC vector at each time interval of 5 ms. Since the prediction of each MCEP vector is done independent of its neighbors, a trajectory model can also be used to predict the a sequence of MFCC vectors.

The sequence of steps involved in building a CLUSTERGEN voice is as follows.

- Label the speech database using an HMM labeler. This labeler uses Baum-Welch algorithm to train context-independent HMM models from a flat-start. Each phone HMM model in this labeler has three states. The labels are generated at the state level.
- F0 and 25-dimensional MCEP vectors are extracted from the speech signal using a frame size of 5 ms and a frame shift of 5 ms. Given the phone labels, the F0 is interpolated through unvoiced regions. This effectively provides non-zero F0 values for all 5 ms frames that contain voiced or unvoiced speech.
- For each 25-dimensional MCEP vector, higher level features are extracted, including phonetic context, syllable structure, word position etc.
- Clustering of MCEP vectors is done using classification and regression tree (CART). A separate CART is built for every HMM state in a phone. The questions for this clustering are high level features and the predictee is an MCEP vector. As an impurity measure, the variance of the MCEPs at each node is the cluster is minimized.
- An additional tree is built to predict durations and F0 for each HMM state.
- During synthesis time, the phone string is generated from the text. Every phone is considered as a constituent of three HMM states. Thus for each HMM state, duration and F0 are predicted from respective CART trees, based on the phonetic context, syllabic structure, word position etc.
- Based on the predicted duration of each HMM state, a sequence of MCEPs is predicted using the CART tree.
- Speech is synthesized from the predicted MCEPs and F0 using the MLSA filter technique. This technique generates speech waveform from the predicted MCEPs and F0 using pulse excitation or random noise excitation.

Appendix D

Modifications to phrasing module

In "festvox/$<$ voice $>$_phrasing.scm", by default FestVox sets the parameter "Phrase_Method" to "prob_models" for English voices. "prob_models" is a probabilistic model trained to predict a break or not-a-break after a word [Taylor and Black, 1998]. This prediction is based on part-of-speech of the neighboring words and the previous word. It also combines an n-gram model of break and not-a-break using a Viterbi decoder to find an optimal phrasing for an utterance.

The use of probabilistic model does not give a control to insert breaks and not-a-break in an utterance. Hence, we opted to use "cart_tree" based phrasing method which is also supported in FestVox. This is a rule based method and inserts a break or not-a-break based on punctuation marks in the text. In order to use this method, we represented speaker-specific prosodic phrase breaks as special punctuation symbols in the text. Break symbols B was typically denoted by a comma ',' or a period '.', while BB was denoted by a semicolon ';'. The "cart_tree" module used in this work is as follows.

```
(set! cmu_us_phrase_cart_tree
  '
  ((lisp_token_end_punc in (";"))
  ((BB))
((lisp_token_end_punc in ("'" "." "?" "\"" ","))
   ((B))
((n.name is 0)
 ((BB))
 ((NB))))))
(set! phrase_cart_tree cmu_us_phrase_cart_tree)
(Parameter.set 'Phrase_Method 'cart_tree)
```

The following are the additional modifications to incorporate prosodic phrase

breaks in clustering process and duration modeling.

In “festival/clunits/all.desc”, add the following.

```
( R:SylStructure.parent.parent.pbreak
0
NB
BB
B
)
```

In “festival/clunits/mcep.desc”, add the following.

```
( R:mcep_link.parent.R:segstate.parent.R:SylStructure.parent.parent.pbreak
0
NB
BB
B
)
```

In “festival/dur/etc/statedur.feats”,
add “R:segstate.parent.R:SylStructure.parent.parent.pbreak”.
In “festival/dur/etc/dur.feats”, add “R:SylStructure.parent.parent.pbreak”.

Appendix E

Artificial neural network models

To train an artificial neural network (ANN) model, backpropagation learning algorithm is used in the pattern mode [Haykin, 1999] [Yegnanarayana, 1999a]. By incorporating some of the heuristics described in [Haykin, 1999] and [Hassoun, 1998], the actual algorithm used to train an ANN model used in this thesis is as follows:

NOTATION

- The indices i, j and k refer to the different units in the network.
- The iteration (time step) is denoted by n.
- The symbol $e_j(n)$ refers to the error at the output unit j for iteration n.
- The symbol $d_j(n)$ refers to the desired output unit j for iteration n.
- The symbol $y_j(n)$ refers to the actual output unit j for iteration n.
- The symbol $w_{jk}(n)$ denotes the synaptic weight connecting the output of the unit k to the input of unit j at iteration n. The correction applied to this weight at iteration n is denoted by $\Delta w_{jk}(n)$.
- The induced local field (i.e., weighted sum of all synaptic inputs plus bias) of unit j at iteration n is denoted by $v_j(n)$.
- The activation function describing the input-output functional relationship of the nonlinearity associated with unit j is denoted by $\varphi_j(.)$. For linear activation

of the unit j, $\varphi_j(v_j(n)) = v_j(n)$, whereas, for nonlinear activation of the unit $\varphi_j(v_j(n)) = a * tanh(b * v_j(n))$. The values of a, b are taken as 1.7159 and $2/3$ respectively (pg. 181 of Haykin [1999]).

- The bias applied to unit j is denoted by b_j; its effect is represented by a synapse of weight $w_{j0} = b_j$ connected to a fixed input equal to $+1$.
- The ith element of the input vector is denoted by $x_i(n)$.
- The learning rate parameter for each unit j is denoted by η_j, where $\eta_j = 0.04 * (1/\mathcal{F}_j)$. $\mathcal{F}_j$ denote the number of inputs (fan-in) for unit j (refer to pg. 211 of Hassoun [1998]). The scaling factor 0.04 is an empirically chosen value.

ALGORITHM

1 Initialize the weights w_{jk} connecting the unit j with the uniformly distributed random values taken from the set $[-3/\sqrt{(}\mathcal{F}_j), +3/\sqrt{(}\mathcal{F}_j)]$ (refer to pg. 211 of Hassoun [1998]).

2 Randomly choose a input vector $\boldsymbol{x}$

3 Propagate the signal forward through the network

4 Compute the *local gradients* δ

   – For an unit j at the output layer, $\delta_j = e_j(n)\varphi_j^{'}(v_j(n))$, where $\varphi_j^{'}(.)$ denotes the first derivate of $\varphi_j(.)$. Since the activation at the output units is typically linear $\delta_j = e_j$, where $e_j = d_j - y_j$.
   – For an unit j at the hidden layer, $\delta_j = \varphi_j^{'}(v_j(n)) \sum_k \delta_k(n) w_{kj}(n)$

5 Update the weights using $\Delta w_{ji}(n) = \eta_j \delta_j(n) y_i(n) + \alpha \Delta w_{ji}(n-1)$, where $\alpha = 0.3$ is the momentum factor.

6 Go to step 2 and repeat for the next input vector

This learning algorithm adjusts the weights of the network to minimize the mean square error obtained for each feature vector. If the adjustments of weights is done for all the feature vectors once, then the network is said to be trained for one epoch. The stopping criteria for training an ANN model is dependent on number of epochs (200-500) or on the validation error of the held-out data set.

Bibliography

M. Abe, S. Nakamura, K. Shikano, and H. Kuwabara. Voice conversion through vector quantization. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages 655–658, New York, USA, 1988. 66

J. Allen, M.S. Hunnicutt, D.H. Klatt, R.C. Armstrong, and D.B. Pisoni. *From text to speech: the MITalk system*. Cambridge University Press, New York, NY, USA, 1987. ISBN 0-521-30641-8. 4

S. Ananthakrishnan and S. S. Narayanan. Automatic prosodic event detection using acoustic, lexical and syntac tic evidence. *IEEE Transactions on Audio, Speech and Language*, 16(1):216–228, 2008. 40, 46

B. Angelini, C. Baralo, D. Falavigna, M. Omologo, and S. Sandri. Automatic diphone extraction for an italian text-to-speech synthesis system. In *Proceedings of EUROSPEECH*, pages 581–584, Rhodes, Greece, 1997. 32

J. Bachenko and E. Fitzpatrick. A computational grammar of discourse-neutral prosodic phrasing of English. *Computational Linguistics*, 16(3):155–170, 1990. 41

C.L. Bennett and A.W. Black. The Blizzard challenge 2006. In *Proc. of Blizzard Challenge 2006 Workshop*, Pittsburgh, USA, 2006. 5

A. W. Black. The blizzard challenge. `http://festvox.org/blizzard/`, 2010. 5

A. W. Black and K. Lenzo. Building voices in the festival speech synthesis system. `http://www.festvox.org`, December 2009. 29, 57

A. W. Black and K. Tokuda. The Blizzard Challenge - 2005: Evaluating corpus based speech synthesis on common datasets. In *Proceedings of INTERSPEECH*, pages 77–80, Lisbon, Portugal, 2005. 5

A.W. Black. CLUSTERGEN: A statistical parametric synthesizer using trajectory modeling. In *Proceedings of INTERSPEECH*, Pittsburgh, USA, 2006. 22, 32, 47, 93

A.W. Black and P. Taylor. Assigning intonation elements and prosodic phrasing for english speech synthesis from high level linguistic input. In *Proc. of ICSLP*, Yokohama, Japan, 1994. 4

A.W. Black and P. Taylor. Automatically clustering similar units for unit selection in speech synthesis. In *Proc. of Eurospeech*, pages 601–604, Rhodes, Greece, 1997. 5

A.W. Black, K. Lenzo, and V. Pagel. Issues in building general letter to sound rules. In *Proc. of 3rd ESCA Workshop on Speech Synthesis*, Jenolan Caves, Australia, 1998. 3

A.W. Black, H. Zen, and K. Tokuda. Statistical parametric speech synthesis. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, Honolulu, USA, 2007. 5

T. Dutoit, V. Pagel, N. Pierret, F. Bataille, and O.V. der Vrecken. The MBROLA project: Towards a set of high quality speech synthesizers free of use for non commercial purposes. In *Proc. ICSLP '96*, volume 3, pages 1393–1396, Philadelphia, PA, 1996. URL `citeseer.ist.psu.edu/506625.html`. 4

ESPS. ESPS source code from the esps/waves+ package. `http://www.speech.kth.se/software/`, 2009. 73

A. Faria. Pitch based vocal tract length normalization. in *ICSI Technical Report TR-03-001, Univ. of California, Berkeley*, Nov. 2003. 75

M. Fraser and S. King. The Blizzard Challenge 2007. In *Proc. of Blizzard Challenge 2007 Workshop*, Bonn, Germany, 2007. xvii, 5, 6, 7

L. Frazier, K. Carlson, and C. Jr. Clifton. Prosodic phrasing is central to language comprehension. *Trends in Cognitive Science*, 10(6):244–249, 2006. 37

Y. Gong and J-P. Haton. Non-linear vectorial interpolation for speaker recognition. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, volume 2, pages 173–176, 1992. 72

M. H. Hassoun. *Fundamentals of Artificial Neural networks*. Prentice-Hall of India, New Delhi, 1998. 97, 98

S. Haykin. *Neural networks: A comprehensive foundation*. Prentice-Hall Inc., New Jersey, 1999. 97, 98

H. Hermansky and N. Malayath. Speaker verification using speaker-specific mappings. In *Speaker Recognition and its Commercial and Forensic Applications*, France, 1998. 72

X.D. Huang, A. Acero, and H-W. Hon. *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*. Prentice Hall, 2001. 1

A. Hunt and A. W. Black. Unit selection in a concatenative speech synthesis system using a large speech database. *Proceedings of ICASSP-96*, 1:373–376, 1996. 5

M. S. Ikbal, H. Misra, and B. Yegnanarayana. Analysis of autoassociative mapping neural networks. In *IEEE Proceedings of the International Joint Conference on Neural Networks*, Washington, USA, 1999. 70

S. Imai. Cepstral analysis synthesis on the Mel frequency scale. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages 93–96, Boston, USA, 1983. 54, 55, 74

Q. Jin, T. Schultz, and A. Waibel. Far-field speaker recognition. *IEEE Trans. Audio, Speech and Language Processing*, 15(7):2023–2032, 2007. 73

S. Joshi, K. Prahallad, and B. Yegnanarayana. AANN-HMM models for speaker verification and speech recognition. In *IEEE Proceedings of the International Joint Conference on Neural Networks*, Hong Kong, China, 2008. 70

A. Kain. *High Resolution Voice Transformation*. PhD dissertation, OGI School of Science & Engineering, Oregon Health & Science University, 2001. 57

A. Kain and M. W. Macon. Design and evaluation of a voice conversion algorithm based on spectral envelop mapping and residual prediction. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages 813–816, Salt Lake City, USA, 2001. 88

V. Keri, S. C. Pammi, and K. Prahallad. Pause prediction from lexical and syntax information. In *International Conference on Natural Language Processing (ICON)*, Hyderabad, India, 2007. 43

H. Kim, T. Yoon, J. Cole, and M. Hasegawa-Johnson. Acoustic differentiation of L- and L-L% in switchboard and radio news speech. In *Proceedings of Speech Prosody*, Dresden, 2006. 38

D.H. Klatt. Review of text-to-speech conversion for english. *J. Acoust. Soc. Amer.*, 82:737–793, 1987. 4

D. Klein, D. Christopher, and D. Manning. Accurate unlexicalized parsing. In *Proceedings of the 41st Meeting of the Association for Computational Linguistics*, pages 423–430, 2003. 42

J. Kominek and A. W. Black. The CMU ARCTIC speech databases. In *Proc. 5th ISCA Speech Synthesis Workshop (SSW5)*, pages 223–224, Pittsburgh, USA, 2004a. 7, 21, 60

J. Kominek and A. W. Black. The CMU arctic speech databases. In *Proc. 5th ISCA Speech Synthesis Workshop*, Pittsburgh, PA, 2004b. 6

J. Kominek, T. Schultz, and A. Black. Synthesizer voice quality on new languages calibrated with Mel-cepstral distorion. In *International Workshop on Spoken Languages Technologies for Under-resourced languages (SLTU)*, Hanoi, Vietnam, 2008. 32

H. Kuwabara and Y. Sagisaka. Acoustic characteristics of speaker individuality: Control and conversion. *Speech Communication*, 16(2):165 – 173, 1995. 53

LibriVox. Copyright and public domain. `http://wiki.librivox.org`, December 2009. 25

LibriVox. Acoustic liberation of books in the public domain. `http://www.librivox.org`, July 2010. 21, 38

K. Liu, J. Zhang, and Y. Yan. High quality voice conversion through phoneme based linear mapping functions with STRAIGHT for Mandarin. In *Fourth Int. Conf. Fuzzy Systems Knowledge Discovery (FSKD 2007)*, pages 410–414, 2007. 59

H. Misra, S. Ikbal, and B. Yegnanarayana. Speaker-specific mapping for text-independent speaker recognition. *Speech Communication*, 39(3-4):301–310, 2003. 72

P. J. Moreno and C. Alberti. A factor automaton approach for the forced alignment of long speech recordings. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages 4869–4872, Taipei, Taiwan, 2009. 11, 12

P. J. Moreno, C. Joerg, J. M. van Thong, and O. Glickman. A recursive algorithm for the forced-alignment of very long audio segments. In *Proceedings of Int. Conf. Spoken Language Processing*, Sydney, Australia, 1998. 11, 12

A. Mouchtaris, J. V. Spiegal, and P. Mueller. Nonparallel training for voice conversion based on a parameter adaptation approach. *IEEE Trans. Audio, Speech and Language Processing*, 14(3):952–963, 2006. 69, 79

M. Narendranath, H. A. Murthy, S. Rajendran, and B. Yegnanarayana. Transformation of formants for voice conversion using artificial neural networks. *Speech Communication*, 16(2):207–216, 1995. 66

J. Nil, T. Hirai, H. Kawai, T. Toda, K. Tokuda, M. Tsuzaki, S. Sakai, R. Maia, and S. Nakamura. ATRECSS ATR english speech corpus for speech synthesis. In *Proc. of Blizzard Challenge 2007 Workshop*, Bonn, Germany, 2007. 6

J. P. Olive. Rule synthesis of speech from dyadic units. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages 568–570, Hartford, Connecticut, USA, 1977. 4

M. Ostendorf, P. J. Price, and S. Shattuck-Hufnagel. The boston university radio news corpus. In *Boston University Technical Report, No. ECS-95-001*, 1995. 48

M. A. Picheny, N. I. Durlach, and L. D. Braida. Speaking clearly for the hard of hearing II: Acoustic characteristics of clear and conversational speech. *Journal of Speech Hearing and Research*, 29:434–446, 1986. 18

K. Prahallad, A.W. Black, and R. Mosur. Sub-phonetic modeling for capturing pronunciation variation in conversational speech synthesis. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, Toulouse, France, 2006. 32

L. R. Rabiner. A tutorial on hidden Markov models and selected applications in speech recognition. *Proceedings of the IEEE*, 77:257–286, 1989. 13

L. R. Rabiner and B. H. Juang. *Fundamentals of Speech Recognition*. Prentice-Hall, 1993. 89

K. S. Rao. Voice conversion by mapping the speaker-specific features using pitch synchronous approach. *Computer Speech and Language*, 24(3):474–494, 2010. 53, 66, 88

L. Redi and S. Shattuck-Hufnagel. Variation in realization of glottalization in normal speakers. *Journal of Phonetics*, 29:407–429, 2001. 38

T. Schultz and A. Waibel. Language independent and language adaptive large vocabulary speech recognition. In *Proceedings of Int. Conf. Spoken Language Processing*, pages 1819–1822, Sydney, Australia, 1998. 87

E. Shriberg. Phonetic consequences of speech disfluency. In *Symposium on The Phonetics of Spontaneous Speech, Proceedings of International Congress of Phonetic Sciences*, volume 1, pages 619–622, San Francisco, USA, 1999. 28

K. Silverman, M. Beckman, J. Pitrelli, M. Ostendorf, C. Wightman, P. Price, J. Pierrehumbert, and J. Hirschberg. Tobi: a standard for labeling english prosody. In *Proceedings of Int. Conf. Spoken Language Processing*, pages 867–870, Banff, Alberta, Canada, 1992. 37

R. Sproat, A.W. Black, S. Chen, S. Kumar, M. Ostendorf, and C. Richards. Normalization of non-standard words. *Computer Speech and Language*, 15(3):287–333, 2001. 3

D. Srinivas, A. W. Black, B. Yegnanarayana, and K. Prahallad. Spectral mapping using artificial neural networks for voice conversion. *IEEE Trans. Audio, Speech and Language Processing*, 18(5):954–964, 2010. 57

A. Stolcke and E. Shriberg. Automatic linguistic segmentation of conversational speech. In *Proceedings of Int. Conf. Spoken Language Processing*, volume 2, pages 1005–1008, Philadelphia, PA, 1996. 11

D. Sundermann, H. Ney, and H. Hoge. VTLN based cross-language voice conversion. In *8th IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Virgin Islands, USA, Dec 2003. 69, 74, 79

D. Sundermann, A. Bonafonte, A. Hoge, and H. Ney. Voice conversion using exclusively unaligned training data. In *Proc. ACL/SEPLN 2004, 42nd Annual Meeting of the Association for Computational Linguistics / XX Congreso de la Sociedad Espanola para el Procesamiento del Lenguaje Natural*, Barcelona, Spain, July 2004. 74, 79

D. Sundermann, H. Hoge, A. Bonafonte, H. Ney, and S. Narayanan. Text-independent voice conversion based on unit selection. In *Proceedings of IEEE Int. Conf. Acoust., Speech, and Signal Processing*, pages 81–84, Toulouse, France, May 2006. 69, 74, 79

P. Taylor. *Text-to-Speech Synthesis*. Cambridge University Press, 2009. 37, 40, 41

P. Taylor and A. W. Black. Assigning phrase breaks from part-of-speech sequences. *Computer Speech and Language*, 12:99–117, 1998. 49, 95

P. Taylor, A.W. Black, and R. Caley. The architecture of the Festival speech synthesis system. In *3rd ESCA Workshop on Speech Synthesis*, pages 147–151, Jenolan Caves, Australia, 1998. 4

P. A. Taylor. Analysis and synthesis of intonation using the TILT model. *J. Acoust. Soc. Amer.*, 107(3):1697–1714, 2000. 86

T. Toda, A. W. Black, and K. Tokuda. Voice conversion based on maximum likelihood estimation of spectral parameter trajectory. *IEEE Trans. Audio, Speech and Language Processing*, 15(8):2222–2235, 2007. 53, 55, 57, 60, 61

A. R. Toth and A. W. Black. Incorporating durational modification in voice transformation. In *Proceedings of INTERSPEECH*, pages 1088–1091, Brisbance, Australia, 2008. 53, 88

A.R. Toth. Forced alignment for speech synthesis databases using duration and prosodic phrase breaks. In *Proceedings of 5th ISCA Speech Synthesis Workshop*, Pittsburgh, USA, 2004. 11

I. Trancoso, C. Duarte, A. Serralheiro, D. Caseiro, L. Carrico, and C. Viana. Spoken language technologies applied to digital talking books. In *Proceedings of INTERSPEECH*, Pittsburgh, USA, 2006. 11, 12

T. Watanabe, T. Murakami, M. Namba, T. Hoya, and Y. Ishida. Transformation of spectral envelope for voice conversion based on radial basis function networks. In *Proceedings of Int. Conf. Spoken Language Processing*, pages 285–288, Denver, USA, 2002. 66

C. W. Wightman, S. Shattuck-Hufnagel, M. Ostendorf, and P. J. Price. Segmental durations in the vicinity of prosodic phrase boundaries. *J. Acoust. Soc. Amer.*, 91 (3):1707–17, 1992. 38

Wikipedia. Public domain. `http://en.wikipedia.org/wiki/Public_domain`, April 2010. 25

B. Yegnanarayana. *Artificial Neural Networks*. Prentice-Hall of India, New Delhi, 1999a. 97

B. Yegnanarayana. *Artificial Neural Networks*. Prentice-Hall of India, 1999b. 57

B. Yegnanarayana and K. Prahallad. AANN: An alternative to GMM for pattern recognition. *Neural Networks*, 15(3):459–469, 2002. 70, 71

H. Zen, T. Toda, and K. Tokuda. The Nitech-NAIST HMM-based speech synthesis system for the blizzard challenge 2006. In *Proc. of Blizzard Challenge 2006 Workshop*, Pittsburgh, USA, 2006. 93

H. Zen, T. Nose, J. Yamagishi, S. Sako, T. Masuko, A.W. Black, and K. Tokuda. The hmm-based speech synthesis system version 2.0. In *Proc. of ISCA SSW6*, Bonn, Germany, 2007. 5