A HYBRID APPROACH TO SEGMENTATION OF SPEECH USING SIGNAL PROCESSING CUES AND HIDDEN MARKOV MODELS

A THESIS

submitted by

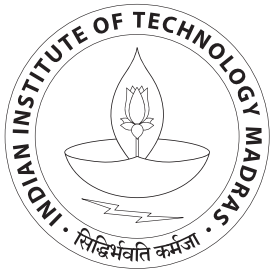
ASWIN SHANMUGAM S

for the award of the degree

of

MASTER OF SCIENCE

(by Research)



DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

INDIAN INSTITUTE OF TECHNOLOGY MADRAS

May 2016

THESIS CERTIFICATE

This is to certify that the thesis entitled **A Hybrid Approach to Segmentation of Speech Using Signal Processing Cues and Hidden Markov Models**, submitted by **Aswin Shanmugam S**, to the Indian Institute of Technology, Madras, for the award of the degree of **Master of Science (by Research)**, is a bona fide record of the research work carried out by him under my supervision. The contents of this thesis, in full or in parts, have not been submitted to any other Institute or University for the award of any degree or diploma.

Dr. Hema A. Murthy

Research Guide

Professor

Dept. of Computer Science and Engineering

IIT-Madras, 600 036

Place: Chennai

Date:

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my research advisor, Dr. Hema A. Murthy for taking me under her tutelage. Her constant encouragement and support helped me throughout my research work. I have been immensely benefited in multiple ways from the discussions and interactions I have had with her.

I am also thankful to my GTC committee members, Dr. S. Umesh, Dr. Deepak Khemani and Dr. Anurag Mittal for sparing their valuable time to evaluate the progress of my research work.

I would like to acknowledge the Department of Information Technology, Ministry of Communication and Technology, Government of India, for funding the project, "Development of Text-to-Speech synthesis for Indian Languages Phase II". It has been an absolute pleasure working with the TTS consortium.

I would like to thank my peers in *DON Lab* and *MS Lab* for creating a friendly and supportive ambience in the lab. Specifically, I thank Mrs. G. Kasthuri and my research collaborator Mr. Abhijit Pradhan. I would also like to thank Mr. Saketh for his suggestions on the organization of this thesis.

I take this opportunity to thank my parents, S. Subramanian and S. Annalakshmi for their limitless affection, support and encouragement.

ABSTRACT

Keywords: *hidden Markov model, flat start, embedded re-estimation, forced alignment, group delay, short-time energy, sub-band spectral flux, syllable, HTS*

The most popular method for automatic speech segmentation is embedded re-estimation of monophone hidden Markov models (HMMs) after flat start initialization, followed by forced alignment. This method may not yield accurate boundaries. To address this issue, short-time energy (STE) and sub-band spectral flux are used as acoustic cues to correct the boundaries. STE and sub-band spectral flux cannot be used directly for detecting boundaries because of local fluctuations, and thus group delay smoothing and root cepstral smoothing, respectively are used to smooth them.

The syllable is the fundamental unit of production. Any speech utterance is thus made up of a sequence of syllables. The acoustic energy between syllables is significantly lower than at the middle of a syllable. Boundary corrections are first made at the syllable level. Embedded re-estimation of monophone models is then restricted to the syllable level rather than the entire utterance. These models are more robust as re-estimation is performed on shorter segments of speech. Forced alignment is performed within the syllable to obtain phone level segmentation.

Essentially signal processing for detecting syllable boundaries and HMMs for acoustic modeling of phones work in tandem to obtain accurate segmentation at both phone and syllable levels. Considering phones and syllables as basic units, HMM based speech synthesis systems (HTS) are built with the proposed segmentation method. Listening tests indicate

that there is an improvement in the quality of synthesis compared to automatic HMM based segmentation.

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

ASR	Automatic Speech Recognition
DMOS	Degradation Mean Opinion Score
GD	Group Delay
HMM	Hidden Markov Model
HTS	HMM based Speech Synthesis System (‘H’ Triple ‘S’)
LTS	Letter-To-Sound
MLP	Multi Layer Perceptron
MLPG	Maximum Likelihood Parameter Generation
MLSA	Mel Log Spectral Approximation
PC	Pair Comparison
SBSF	Sub-Band Spectral Flux
SUS	Semantically Unpredictable Sentences
SVM	Support Vector Machine
TTS	Text-To-Speech
USS	Unit Selection Synthesis
WER	Word Error Rate

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

Overview of the Thesis

1.1 Introduction

The main aim of this thesis is to emphasize the importance of acoustic cues in segmentation of speech. An automatic hybrid segmentation algorithm which uses hidden Markov models (HMMs) and signal processing cues in tandem to produce accurate phone level segmentation is proposed. This segmentation algorithm can be used to segment the training corpus for a text-to-speech (TTS) system.

Segmentation of the speech corpus at syllable/phone level is an important phase in many speech processing applications, including training of TTS systems and automatic speech recognition (ASR) systems. Especially, speech synthesis requires very accurate and consistent segmentation [1, 2].

A TTS system takes text as an input and artificially produces corresponding human-like speech as output. The input text is first converted into a sequence of sub-word units like syllables/phones using letter to sound (LTS) rules. Waveforms are concatenated as in unit selection synthesis [3], or models are concatenated as in statistical parametric synthesis [4] to produce speech corresponding to that of the text. The training data for building a TTS system consists of a set of utterances spoken by a native speaker of the language along with the corresponding text transcriptions. It is essential to find the boundaries of sub-word units in these utterances to use them for synthesis. The process of determining syllable/phone

boundaries is called segmentation. Figure 1.1 shows a Hindi utterance segmented at the syllable and phone levels. Panel A shows the waveform. Panels B and C show the syllable and phone transcriptions respectively. Panel D shows the spectrogram. The thick solid vertical lines in Figure 1.1 correspond to syllable boundaries and the thin dotted lines represent phone boundaries.

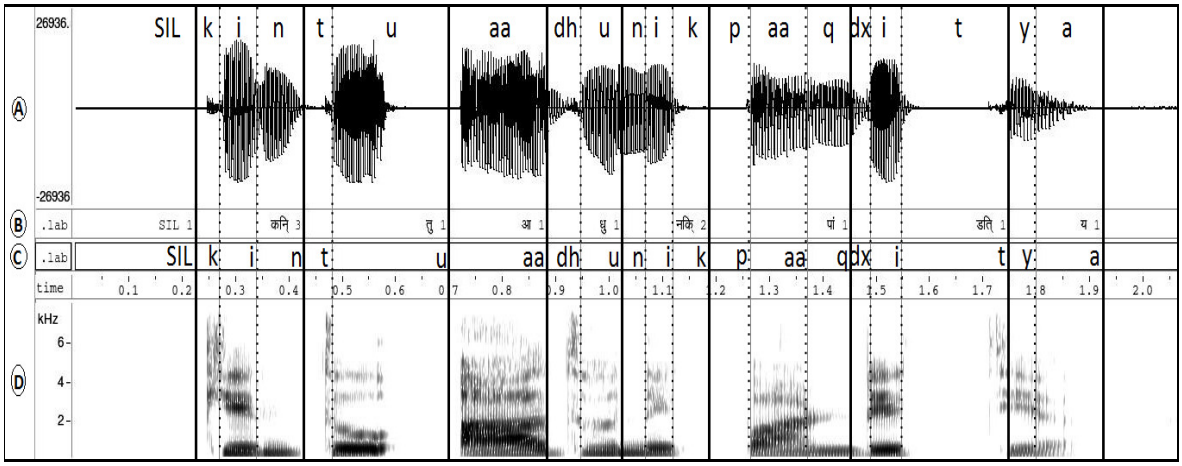


Fig. 1.1: A sample Hindi utterance segmented at the syllable and phone levels

It is very tedious to perform segmentation completely manually. Traditionally, HMMs [5, 6] are used to aid the segmentation process. Forced alignment [6], which exploits the knowledge of the phone sequence, is performed to obtain boundaries using HMMs. HMMs provide reasonably good boundaries but are not good enough to build high quality TTS systems [2, 7]. Thus, boundary correction algorithms are necessary.

Boundary correction techniques like [2, 8–10] require significant amounts of manually segmented data from the language to learn boundary models. There is a growing demand for building TTS systems for more languages. For low resource languages, it is hard to get manually labeled data from acoustic phonetic experts.

In this work, an attempt has been made to perform boundary correction using two acous-

tic cues - energy and spectral change. Signal processing techniques are used to extract boundary information from the acoustic cues and hence it is not completely data driven. In this work, an algorithm is proposed which combines HMMs with knowledge based domain specific signal processing to obtain accurate segmentation without the requirement of any manually segmented data.

1.2 Organization of the thesis

The rest of the thesis is organized as follows. Chapter 2 provides the background study of HMM based speech synthesis. It also discusses the current segmentation algorithms and their drawbacks. Chapter 3 discusses the importance of syllables in phone segmentation. It also presents an STE based algorithm to segment speech into syllable like units. In Chapter 3, a semi-automatic approach to phone segmentation is proposed. The approach proves the claim that acoustic cues can improve segmentation accuracy.

Chapter 4 explains the drawbacks of the proposed semi-automatic approach. It then proposes an automatic hybrid segmentation algorithm which uses STE for syllable boundary correction along with HMMs to obtain phone level segmentation. In Chapter 5, a technique is proposed, which uses spectral change as a cue to detect certain acoustic landmarks. It also presents a modification to the previously proposed hybrid segmentation algorithm. Chapter 6 presents the conclusions and the scope for future work.

1.3 Contribution of the thesis

The following are the main contributions of the thesis:

1. A semi-automatic approach to phone segmentation is proposed. It uses manually corrected syllable boundaries from an existing approach [11] and then automatically

derives phone boundaries.

2. An automatic hybrid segmentation algorithm is proposed. The algorithm uses HMMs to obtain initial boundary estimates and then uses signal processing techniques to correct some of the syllable boundaries based on certain rules. These corrected boundaries are fed back to HMM based segmentation to improve the location of other boundaries.

3. A variant of spectral flux named sub-band spectral flux (SBSF) is proposed as a boundary detector for sibilant fricatives and affricates. A technique based on root cepstral smoothing is proposed for smoothing SBSF. The peaks of the smoothed version of SBSF are shown to give boundaries of sibilant fricatives and affricates.

CHAPTER 2

Background and Related Work

Segmentation is an important component for both ASR and TTS. In ASR, large amounts of data is collected from various speakers and statistical models are used to segment the data. Unlike ASR, segmentation has to be very accurate for TTS. This is primarily because of the co-articulation that exists in speech and an unit used wrongly can result in significant deterioration in voice quality. The focus of this thesis is primarily in the context of speech synthesis. Experiments involving the proposed segmentation algorithm are primarily carried out in one paradigm of TTS, the HMM based speech synthesis framework. This chapter gives an overview of HMM based speech synthesis. Speech data of two Indian languages - Tamil and Hindi are used for experiments in this thesis and details specific to Indian language synthesis are given in this chapter. This chapter also discusses the existing segmentation algorithms for TTS and ASR.

2.1 Speech synthesis: an overview of different paradigms

Speech synthesis has evolved from formant based synthesis [12], through diphone based synthesis [13], to unit selection synthesis (USS) [3]. In formant based synthesis, the frequency location of formants, their corresponding amplitude and bandwidth are used as parameters and the resonators are connected either in cascade or parallel to approximate the vocal tract transfer function. For voiced sounds, the source is approximated by a train of

impulses and a turbulent noise source is used for unvoiced sounds. The synthesis output of this method lacks naturalness because the vocal tract function is highly approximated.

In diphone based synthesis, one example of each diphone (two adjacent half-phones) is contained in the database. During synthesis, the appropriate diphones are chosen and concatenated. The target prosody is also superimposed using techniques like pitch-synchronous overlap-add [13]. Although this method is more natural that formant based synthesis, the speech produced using a small set of controlled units is not human-like.

USS requires collection of a large database of utterances from a voice talent. Then, the database is segmented. During synthesis time, the target utterance is constructed by concatenating the optimal segments from the database. The units in the synthesis database is considered as a state transition network and the optimal segments are chosen with the help of two cost functions: *target cost* and *concatenation cost* [3]. The *target cost* is a penalty which depends on the difference between the context of the unit in the database and the target unit. The *concatenation cost* is a penalty dependent on the quality of concatenation of two consecutive units. After estimating the costs, a pruned Viterbi search is performed to choose and concatenate units.

Although USS produces high quality speech output, it has a few drawbacks. Most importantly, the footprint of USS is large. Also, it is not convenient in USS to control and change speech parameters for applications like emotional synthesis, multilingual systems and voice transformation.

HMM based speech synthesis systems (HTS) [14] are state-of-the-art in the area of speech synthesis and do not suffer from many of the drawbacks that other systems do. HTS extracts speech parameters from the waveforms and builds statistical models and uses them for synthesis. They are attractive owing to the fact that they are not only small in size, but they also provide much better control and scalability compared to the unit selection based

systems. The footprint of HTS systems are small because it doesn't store any pre-recorded speech waveforms directly. The working of the HTS framework is explained in detail in the subsequent section.

2.2 HMM based speech synthesis

The sub-word units are represented using HMMs in this framework, as they are generative models. Unlike ASR, where HMMs are used for classification, here HMMs are used to reconstruct speech by generating parameters. Traditionally, context dependent phones are used as the basic units of synthesis in HTS. Alternatively syllable can also be used as the basic units of synthesis [15], which is explained in Section 2.2.3.

2.2.1 The phone based HTS system

HTS requires labeled data as input. So, prior to the training phase, text should be converted into the corresponding phone sequence using LTS rules and then the waveforms should be segmented at the phone level.

In the training phase of HTS, spectral parameters (Mel generalized cepstral coefficients [16]), excitation parameters (log of the fundamental frequency) and their respective dynamic features (velocity, acceleration) are extracted from the speech data. These features are modeled using multi-stream [6] HMMs. Initially, context independent (monophone) HMM models are built using the generated labels as well as the features extracted. The duration of each state in a HMM is also modeled separately [14].

HTS uses three types of context: prosodic, linguistic and phonetic [17]. The phonetic context used in HTS by default is a pentaphone context, that is for every phone, two preceding and two succeeding phones are used as context. These contexts are added because the

acoustic realization of a phone depends on the context in which it appears. With these contexts, context dependent HMM models are built after being initialized with corresponding context independent HMM models. Tree based context clustering [18] based on a question set is performed to tie states. Tree based context clustering is performed to address two problems associated with the modeling and use of context dependent phones: (1) insufficient data to build all context dependent HMMs separately and (2) to handle unseen models during synthesis. An overview of the HTS system is illustrated in Figure 2.1[1].

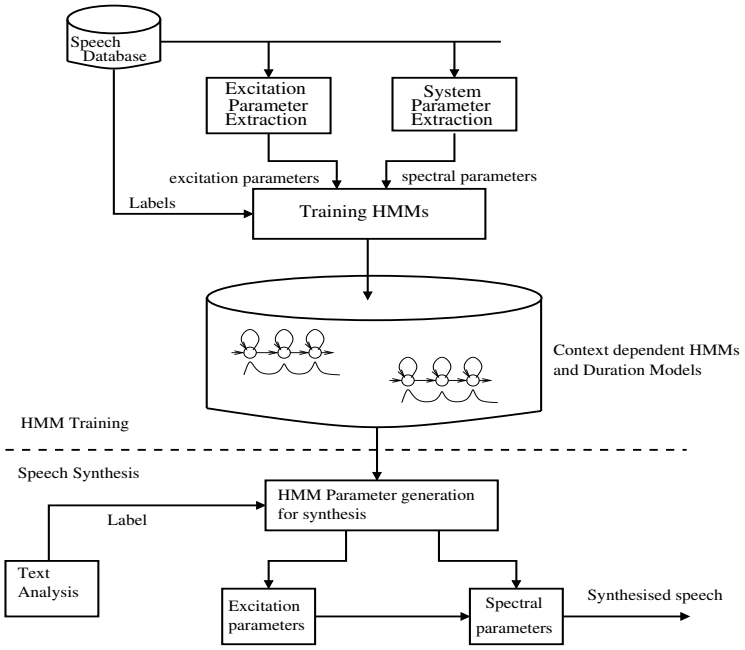


Fig. 2.1: Overview of HMM based speech synthesis

During synthesis, the required context-dependent HMMs are concatenated to obtain the sentence HMM. Appropriate models are chosen by traversing the decision tree [19] built during tree based context clustering. A parameter generation algorithm explained in [20] is applied to get the speech observation vector that maximizes the output probability. The speech waveform is synthesized from the generated spectral and excitation parameters using a source-filter model. If cepstral coefficients are used as features, the mel log spectral

[1]This figure has been redrawn from [17]

approximation (MLSA) [21] filter is used.

2.2.2 An implementation for Indian languages

To build a HTS system for a new language, three language specific components are required - a phone set, a question set and a set of LTS rules.

The common phone set and question set for Indian languages proposed in [22] is used in this thesis. Figure 2.2 shows the complete phone set for Tamil and Hindi. Acoustically similar phones in Hindi and Tamil are given a common label. The label for a vowel modifier is the same as that for a vowel. Labels 1-17 in Figure 2.2 are vowels and the rest are consonants.

The decision tree built during tree based context clustering is similar to a binary tree with a yes/no question at each node. A question set consisting of relevant linguistic and phonetic classifications is given as input to the clustering algorithm. The algorithm [18] chooses a question at each node of the tree such that there is maximum gain in the likelihood. The common question set [22] is a super set of questions across 13 Indian languages including Tamil and Hindi. Two example questions from the question set are given below:

- QS "L-Long_Vowel" $\{*\wedge aa-*,*\wedge aamq-*,*\wedge ii-*,*\wedge uu-*,*\wedge ee-*,*\wedge oo-*\}$ (Set of context dependent phones with left phonetic context as a long vowel)
- QS "R-Back_Stop" $\{*+k=*,*+kh=*,*+g=*,*+gh=*,*+kq=*\}$ (Set of context dependent phones with right phonetic context as a back stop consonant)

LTS rules provide a mapping between the orthographic representation (written form) and the acoustic representation (spoken form). For Indian languages, this mapping can be established by a finite set of rules and hence it need not be learnt from data. The input text is split into words. The LTS module takes a single word as argument. First, the word is

syllabified and then the syllables are converted to a phone sequence.

Sl.No.	Label	IPA	Hindi	Tamil
1	a	a	अ	அ
2	ax	ɔ	ऑ	-
3	aa	a:	आ	ஆ
4	i	ɪ, i	इ	இ
5	ii	i:	ई	ஈ
6	u	u, ʊ	उ	உ
7	uu	u:	ऊ	ஊ
8	rq	-	ऋ,ॠ	-
9	e	e	–	எ
10	ee	e:	ए	ஏ
11	ea	ɛ	ऍ	-
12	ei	ɛ:	ऐ	-
13	ai	aI	–	ஐ
14	o	o	ओ	ஒ
15	oo	o:	–	ஓ
16	au	aʊ	–	ஔ
17	ou	oʊ	औ	-
18	k	k	क	க
19	kh	kʰ	ख	-
20	g	g	ग	கV
21	gh	gʰ	घ	-
22	ng	ŋ	ङ	ங
23	c	ʧ	च	ச
24	ch	ʧʰ	छ	-
25	j	dʒ	ज	ஜ
26	jh	dʒʰ	झ	-
27	nj	ɲ	ञ	ஞ
28	tx	ʈ	ट	ட
29	txh	ʈʰ	ठ	-
30	dx	ɖ	ड	டV
31	dxh	ɖʰ	ढ	-
32	nx	ɳ	ण	ண
33	t	t̪	त	த

Sl.No.	Label	IPA	Hindi	Tamil
34	th	t̪ʰ	थ	-
35	d	d̪	द	தV
36	dh	d̪ʰ	ध	-
37	n	n	न,ऩ	ந,ன
38	nd		–	ந
39	p	p	प	ப
40	ph	pʰ	फ	-
41	b	b	ब	பV
42	bh	bʰ	भ	-
43	m	m	म	ம
44	y	j	य,य़	ய
45	r	r	र,ऱ	ர
46	l	l	ल	ல
47	lx	ɭ	–	ள
48	w	ʋ	व	வ
49	sh	ʃ	श	-
50	sx	ʂ	ष	ஷ
51	s	s	स	ஸ
52	h	ɦ	ह	ஹ
53	kq	q	क़	-
54	khq	x	ख़	-
55	gq	ɣ	ग़	-
56	z	z	ज़	-
57	jhq	ʒ	झ़	-
58	dxq	ɽ	ड़	-
59	dxhq	ɽʰ	ढ़	-
60	f	f	फ़	ஃப
61	rx	ɾ	–	ற
62	zh	ɻ	–	ழ
63	q		◌ं	-
64	hq		◌ः	-
65	mq		◌ँ	-

Fig. 2.2: Common label set for Hindi and Tamil

LTS rules for Tamil are fairly straightforward, as there is a one-to-one correspondence between the grapheme and phoneme representation. The only major issue in Tamil is that characters க (ka, ga), ட (txa, dxa), த (ta, da) and ப (pa, ba) can be associated with two different manners of articulation. The difference between the two, is the presence or absence of voicing in the stop consonant. For instance, the Tamil character க can be pronounced as both /ka/ (Unvoiced) and /ga/ (Voiced). Generally, they are unvoiced when they appear at the beginning of a word or as geminates. They are voiced when they appear in between the vowels or after the nasals ங, ஞ், ண், ன், ந், ம் (ng, nj, nx, n, nd, m) or after the consonants ய், ர், ல், ள், ழ் (y, r, l, lx, w, zh) [15]. For example, in the word அக்கா /akka/, the character occurs as a geminate and thus is unvoiced. In word அங்கம் /angam/, the character க occurs after the nasal ங and thus is voiced. There are exceptions to this rule in Tamil, primarily because of the presence of words that have been borrowed from other languages, especially Sanskrit. During syllabification, we add a separate tag for voiced stop consonants as per the rule. However, we ignore this tag in the phone transcription and use the same symbol as an unvoiced stop consonant from the label set. This is because context dependent models are built in conventional HTS and the context resolves the ambiguity. The reason for adding the tag during syllabification is explained in Section 2.2.3.

In Hindi, the major complication during syllabification is the problem of *schwa deletion* (vowel deletion). For some syllables ending with a unit of the CV type, the V part is replaced with a unit called *halant*. For example, the word चिनतन /cinatana/ without *schwa deletion* would be syllabified as चिन /cina/ and तन /tana/, but with *schwa deletion* it is syllabified as चिन् /cin/ and तन् /tan/. Converting syllables to phones is a straightforward mapping based on the phone set. For instance, syllable चिन् is phonified as “c i n”

2.2.3 Syllable based HTS

The phonetic context used in conventional phone based HTS is a pentaphone. A pentaphone is a large context. The consequence of this is that context dependent phone models will have to be built with very few examples. Nevertheless, a large context is required to accommodate the effects of articulation. So, state tying is done. Inter-syllable co-articulation, though present across syllables, is relatively less because syllables are the fundamental units of production. Syllable based HTS proposed in [15] uses only one positional context as opposed to close to 50 contexts in conventional phone based HTS [14].

A 5 state HMM is generally used for modeling phones. A syllable is defined as "C*VC*", where 'C' is a consonant and 'V' is a vowel. Hence, a syllable varies from a single phone (e.g. a vowel) to a sequence of phones. Thus a variable number of states are required to build syllable models. Given the transcription of a syllable into its constituent phones, the number of states for each syllable can be obtained as "$number\ of\ phones \times 5$". The prosody of a syllable differs significantly based on the syllable's position in a word. Based on experiments with syllable-based USS for Indian languages, positional context seems to improve USS performance [23]. Syllables are classified into three categories: *begin, middle,* and *end* based on their positions in a word and modeled separately.

Since it is not practical to model the complete set of syllables in a language, during synthesis we encounter a syllable that is not in the training database. An unseen syllable is mapped to a sequence of $\{(C*V) \bigcup C\}$ units. A fallback unit is a variable length unit like a syllable. Fallback unit models are built with the same positional context as syllables. An identical training process to that of syllables is adopted for fallback units. Figure 2.3 and Figure 2.4, illustrate the training process and testing process, respectively.

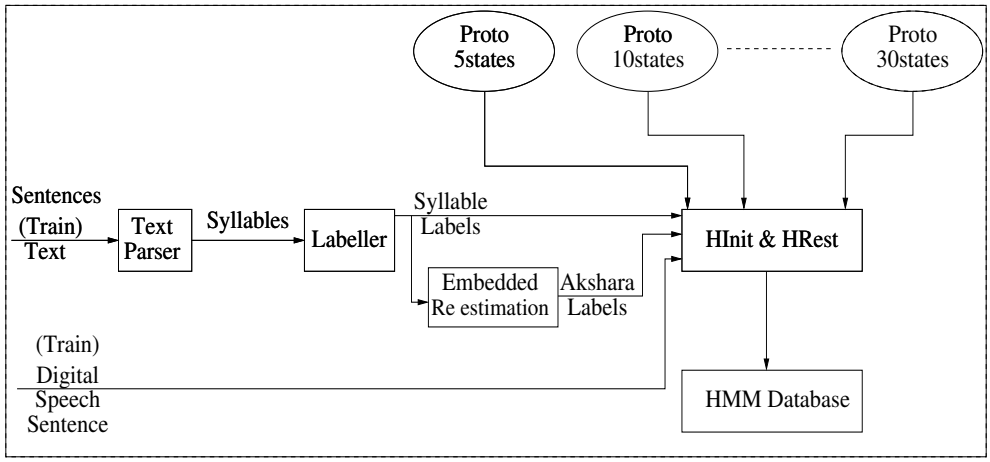


Fig. 2.3: Training phase of syllable based HTS

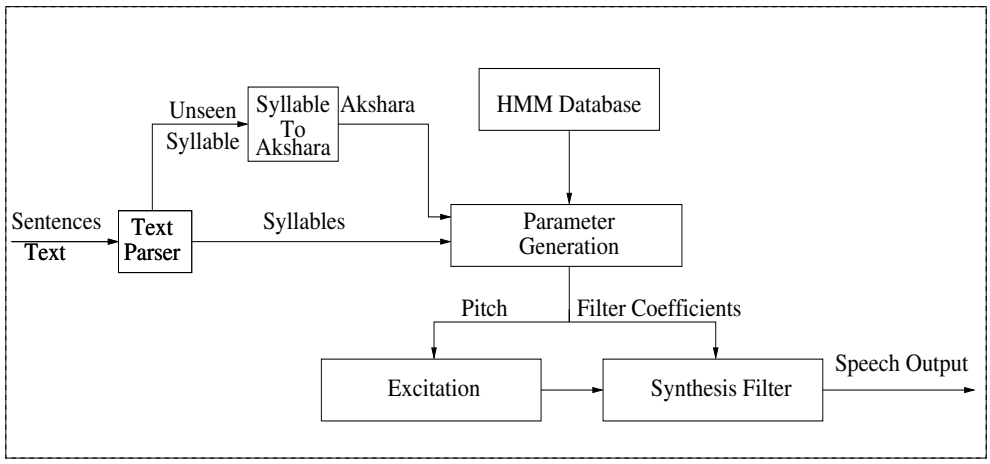


Fig. 2.4: Testing phase of syllable based HTS

Although, there is co-articulation between fallback units, it doesn't affect the overall synthesis quality significantly. This is because an unseen syllable seldom occurs during testing. Although, theoretically many syllables can be formed in a language, the phonotactics of the language makes most of the combinations invalid. The number of frequently occurring syllables in a language is not more than 300, and the distribution of the frequency of occurrence of these syllables follows a Zipf distribution [24] as can be seen from Figure 2.5.

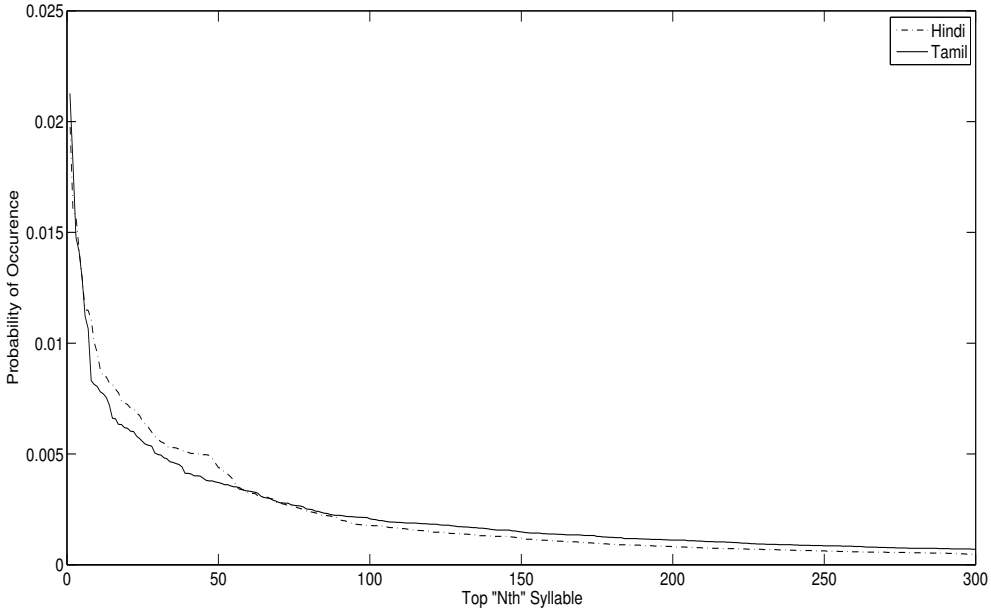


Fig. 2.5: Probability of occurrence of syllables

2.3 Importance of segmentation in speech synthesis

TTS systems rely on segmented speech corpora for training or construction purposes [7]. They require segmentation knowledge for constructing intonation, duration, and synthesis components [25]. As the consumer of speech synthesis systems are human ears, accurate and consistent segmentation needs to performed, to prevent sonic glitches in the synthesized output.

Like other paradigms of TTS, building a good quality HTS system for any language, also requires a well labeled database [1]. As explained in Section 2.2, context independent models are built in conventional HTS using the segmentation knowledge. For instance, if there is an average segmentation error of 10 ms in the detection of closure-burst boundaries (stop consonants), in some examples, the burst region might be missed altogether because these bursts are often less than 10 ms [7]. The purity of trained HMMs can be affected by

poor segmentation leading to sonic glitches in the synthesised output.

In syllable based HTS, segmentation is even more crucial because phonetic context is not added to the models. The location of boundaries have to be very accurate as syllable models are built with the segmentation information and used directly during synthesis.

2.4 Related work on segmentation

Conventionally HMMs have been used to aid the segmentation process [6]. This section will explain the process of using HMMs for segmentation. The segmentation output of HMMs require subsequent manual or automatic boundary corrections. Some existing boundary correction methods are discussed in this section.

2.4.1 Segmentation using HMMs

Speech can be considered to be a quasi-stationary over short periods of time. Features like Mel frequency cepstral coefficients (MFCCs) are extracted from the waveform. Features at the rate of X frames per second are extracted from the speech data. Frame length of Y ms ($Y > 1/X$) is used. This results in overlapped frames to accommodate for co-articulation. It is reasonable to assume speech signal as a realization of some message encoded as a sequence of one or more symbols [6]. HMMs are perhaps the best models that can represent sequential time-varying patterns.

HMMs have three parameters: $\lambda = (A, B, \pi)$, where, A is a matrix consisting of transition probabilities, B are the emission probabilities and π is the initial state distribution [26]. Baum-Welch re-estimation [5] is used to optimally estimate these parameters from the observation sequence (i.e. feature vectors extracted from the speech signal). In this thesis, the HTK toolkit is used [27] for implementing HMMs. In HTK package, π is incorporated

within A by using two extra non-emitting dummy states. These dummy states facilitate the construction of composite models.

Each phone in the language is modeled as an HMM. For high resource languages like English, HMM based ASR systems like [28] are used to perform segmentation by restricting its language model to the known input sequence. For low resource languages like Indian languages, there is no existing ASR system. So, HMM models have to be trained first. There are broadly two ways to train them: bootstrap approach (semi-automatic) and flat start approach (automatic). After training HMMs, forced alignment [6] using the transcription information can be performed to obtain segmentation.

Bootstrap labeling is performed in four steps: (1) A small subset of utterances are segmented manually at the phone level, (2) phone HMM models are built using the data that is manually labeled, (3) forced Viterbi alignment is performed on the rest of the data using the models built and (4) the models are refined iteratively until the segmentation is satisfactory. The problem with this approach is labeling the bootstrap data manually at the phone level. This segmentation should be carefully performed by experts who are familiar with acoustic phonetics. The flat start approach is explained in Section 2.4.2.

2.4.2 Baseline HMM based automatic segmentation

In the flat start approach, all monophone HMM models are initialized such that their state means and variances are equal to the global mean and variance. If some manually segmented data is available, isolated training of monophone HMMs can be performed. To train phone HMMs without any segmentation knowledge, embedded training is performed. During embedded training [27], transcription knowledge is used to construct a composite HMM for each utterance by concatenating phone HMMs. Embedded Baum-Welch re-estimation is performed on these utterance HMMs and all monophone HMM models are updated simul-

taneously. The baseline HMM based automatic segmentation can be summarized in three steps: (1) flat start initialization of phone HMMs, (2) embedded training and (3) forced Viterbi alignment.

2.4.3 Drawbacks of HMM based segmentation

A fundamental drawback of using HMMs for segmentation is that boundaries are not represented by them [29] as they do not use proximity to boundary positions as a criterion for optimality during training [2]. The boundaries are simply derived from the alignment of phone states with frames [29]. Acoustic landmarks at boundaries like abrupt spectral change [30] are not used to determine the location of a boundary. Results achieved using HMMs are good and quite robust, but they are not sufficient enough to build high quality synthesis systems [2]. So a combination of HMMs and boundary correction techniques are introduced to obtain accurate segmentation.

2.4.4 Supervised machine learning based correction methods

The common requirement of methods discussed here is the availability of around one hour of manually segmented data. Boundary models for various phone transitions are built using this data. HMMs are used to obtain initial segmentation and a boundary is searched in the vicinity using the trained models. In [8], a multi layer perceptron (MLP) was employed to refine the phone boundaries. To increase the accuracy of phoneme segmentation, several specialized MLPs were individually trained based on phonetic transition. The entire phone transition space was partitioned and a specialized MLP was allocated to each partition. In [9], support vector machines (SVMs) were used for modeling boundaries. In [2], the boundary between every pair of phones was modeled as a multi-state HMM. A special one-state HMM is used for detecting phoneme boundaries in [29].

The drawback of these approaches is the requirement of manually segmented data with the help of acoustic phonetic experts. There is hardly any manually labeled data available for Indian languages that has been carefully marked. The experiments for these approaches were performed only on English for which carefully labeled data is available. Nevertheless, these approaches are sensitive to manual labeling errors. There is a growing demand for building TTS systems for many low resource languages. Hence, it is essential to develop boundary correction techniques which are knowledge based.

2.4.5 Knowledge based correction methods

An iterative boundary correction technique is proposed in [1]. Synthesis quality measure is used as an inherent part of boundary optimization. The maximum likelihood parameter generation (MLPG) algorithm in the HTS framework is applied to generate parameters. The distance between predicted and actual frame is used as a metric to refine the boundary location. The boundary is shifted only by a single frame per iteration. The metric used in this approach does not represent a boundary. Moreover, the authors of [1] do not recommend it for refining flat start segmentation. The authors have assumed the starting point to be close to optimal. The approach is also computationally very expensive.

In [31], spectral boundary correction is performed using weighted slope metric [32]. The authors have used the weighted slope metric as a boundary detector to correct boundaries corresponding to all phone transitions. The boundaries were corrected by moving it to the nearby peak given by the metric. This approach has quite a few drawbacks. Not all phone transitions are accompanied by an abrupt spectral change (eg. semivowel to vowel). Hence, it might not be appropriate to use this metric to correct all phone boundaries. The authors have acknowledged that the detector used is likely to produce many spurious peaks and the decision on the location of phone boundary is somewhat risky without any human

intervention.

2.5 Motivation for the proposed method

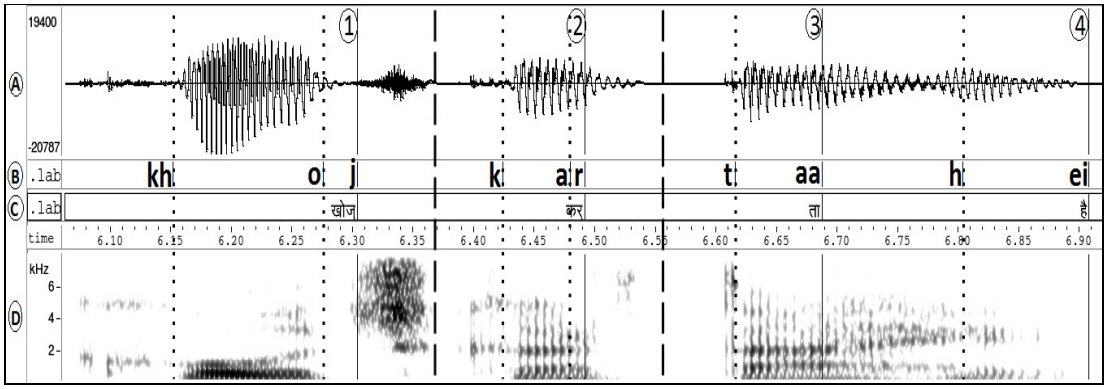


Fig. 2.6: Output of HMM based segmentation

There are no freely available speech recognition systems for Indian languages. Segmentation using a small amount of data using flat-start embedded training and forced Viterbi alignment is poor. Figure 2.6 shows only a portion of a Hindi utterance segmented automatically using the baseline HMM based automatic segmentation. Panel A shows the waveform. Panels B and C show the syllable and phone aligned transcription obtained using forced alignment respectively. Panel D shows the spectrogram. The dotted lines in Figure 2.6 are phone boundaries and the solid lines are syllable boundaries, obtained after forced alignment. The location of syllable boundaries 1 and 2 are significantly incorrect (the actual locations are shown using dashed lines). The actual location of boundary 1 is right after an affricate. As the energy of affricates is concentrated in the higher bands of the spectrum, the boundary is clearly visible in the spectrogram. The actual location of boundary 2 is right before an unvoiced stop consonant and this location can be extracted from the time domain signal as there is a significant dip in energy because of closure. In this thesis, an attempt has been made to use signal processing techniques to extract boundary information from

such acoustic cues and they are used in tandem with HMMs to improve the segmentation accuracy.

2.6 Summary

In this chapter, an overview of HTS was presented. Two variants of HTS - the conventional phone based HTS and syllable based HTS were discussed. Indian language specific components in HTS were discussed in detail. Then, the importance of segmentation in HTS was emphasized. The process of using HMMs for segmentation and the necessity of correcting the boundaries was detailed. This was followed by a discussion on the existing boundary correction techniques and their drawbacks.

CHAPTER 3

A Semi-Automatic Approach to Segmentation of Speech

3.1 Introduction

The syllable is the fundamental unit of speech production and cognition [33, 34]. It is, therefore, easier to determine syllable boundaries compared to phone boundaries. Phone transitions are not necessarily distinguishable owing to co-articulation in continuous speech. On the other hand, syllable boundaries are more or less distinct. The acoustic energy in the region between syllables is significantly lower than at the middle of a syllable.

Although the role of syllables in segmentation cannot be contested [35], appropriate acoustic cues are required to detect syllable boundaries. Syllable boundaries are characterized by low energy. The generic structure of a syllable consists of three parts: the onset, nucleus, and coda. The onset and coda can consist of consonants (onset and coda are optional in a syllable), while the nucleus is a vowel. The vowel region corresponds to a much higher energy region compared to a consonant region. The definition of a syllable in terms of the short-time energy (STE) function is suitable for almost all the languages, in the case of spontaneous speech. The STE can be used as a cue to determine syllable boundaries, but it cannot be applied directly owing to local fluctuations [36]. But it is shown in [11] that the STE function, when smoothed by performing group delay processing, can be used

to detect syllable boundaries. This chapter explains in detail the STE based group delay segmentation algorithm [11] and how it can be used to aid syllable level segmentation using a semi-automatic labeling tool [37].

With manual correction, group delay segmentation algorithm can be used to produce accurate syllable boundaries. In this chapter, an attempt is made to use the existing information of syllable boundaries, to obtain accurate phone boundaries. Although syllable boundaries are obtained with manual intervention, phone boundaries within syllables are obtained automatically using HMMs.

3.2 Group delay based segmentation of speech into syllable like units

Group delay based processing of STE can yield syllable boundaries as shown in [11, 38]. The basic idea of the algorithm is to make STE resemble the magnitude spectrum of an arbitrary real signal and to smooth it using group delay processing [39]. The algorithm for group delay segmentation [36] is given below:

1. The STE function $E[m]$ where $m = 0, 1, ..., M-1$ is calculated from the given speech utterance $x[i]$ as

$$E[m] = \sum_{i=m.f_s}^{(m.f_s)+L-1} (x[i])^2 \tag{3.1}$$

   where f_s is the frame shift and L is the window length. Let the minimum value of the STE function be E_{min}.

2. Compute the order N of fast Fourier transform (FFT) as given below

$$N = 2^{\lceil log(2M)/log(2) \rceil} \tag{3.2}$$

3. Compute $E^i[m]$ as $\frac{1}{E[m]^\gamma}$ after appending $(N/2-M)$ number of E_{min} to the sequence $E[m]$, where $\gamma = 0.01$[1]

4. Construct the symmetric part of the sequence $E^i[m]$ by lateral inversion. The resulting sequence is positive and symmetric. So it resembles the magnitude spectrum of an arbitrary real signal. Let's call this sequence $E[K]$.

5. The IDFT of $E[K]$ is computed. The resultant signal is the root cepstrum [40] and the causal portion of the same is a minimum phase signal [41]. Let's call this causal sequence $e_c[n]$.

6. A single sided Hanning window is applied on $e_c[n]$ and its minimum-phase group delay function $E_{gd}[K]$ is computed. The size of the window applied is

$$N_c = \frac{Length\ of\ STE}{Window\ scale\ factor\ (WSF)} \tag{3.3}$$

   The resolution of the group delay function is controlled by WSF. The smaller the WSF, the greater is the resolution and vice versa.

7. Peaks in the minimum phase group delay function $(E_{gd}[K])$ are detected. These locations correspond to low energy regions.

The various steps in the segmentation algorithm along with a sample output in each stage is illustrated by Figure 3.1. The algorithm does not take text transcription as an input, unlike HMM based forced alignment. This makes the procedure agnostic to text transcription. The extent of smoothing of the STE function, or in other words, the number of syllable boundaries, is determined by the size of the single sided Hanning window applied in the

[1] A small value of γ is chosen to reduce the dynamic range of STE

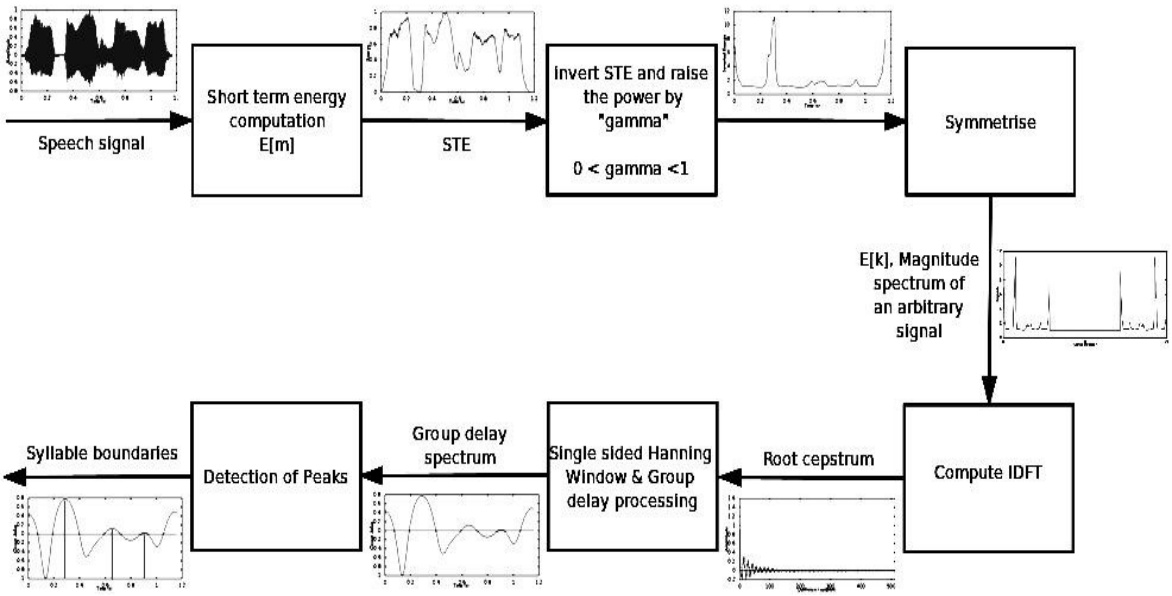


Fig. 3.1: Steps involved in group delay segmentation algorithm for obtaining syllable boundaries

sixth step of the algorithm. Note that the size of the window is not controlled directly in the algorithm. Instead, it is controlled using the parameter WSF, which is inversely proportional to the window size.

To reduce the computation time, an entire speech utterance is not given directly as an input to the group delay segmentation algorithm, instead, the utterance is divided into phrase like chunks by performing speech activity detection and then these chunks are given as the input.

3.3 Labeling tool

In the context of TTS, group delay segmentation algorithm cannot be used in isolation, as it does not consider transcription to be an input. The algorithm gives rise to two types of errors - *insertion* and *deletion errors*. When an additional syllable boundary is given in between

a syllable, it is an *insertion error*. When a syllable boundary is missed completely it is a *deletion error*. When the syllable sequence is directly mapped to the boundary sequence, even a single error can mess up the whole segmentation. If the boundary of a syllable is incorrect, the boundary of every syllable that follows it will also become incorrect. Also, the location of the boundary is not accurate in every place.

So, a labeling tool is developed in [37] to manually correct boundaries. Figure 3.2 shows the labeling tool. WSF is chosen depending on the syllable rate of the speaker. The output of the group delay segmentation algorithm is shown by the tool. The tool can be used to insert, delete and move boundaries for correct the segmentation.

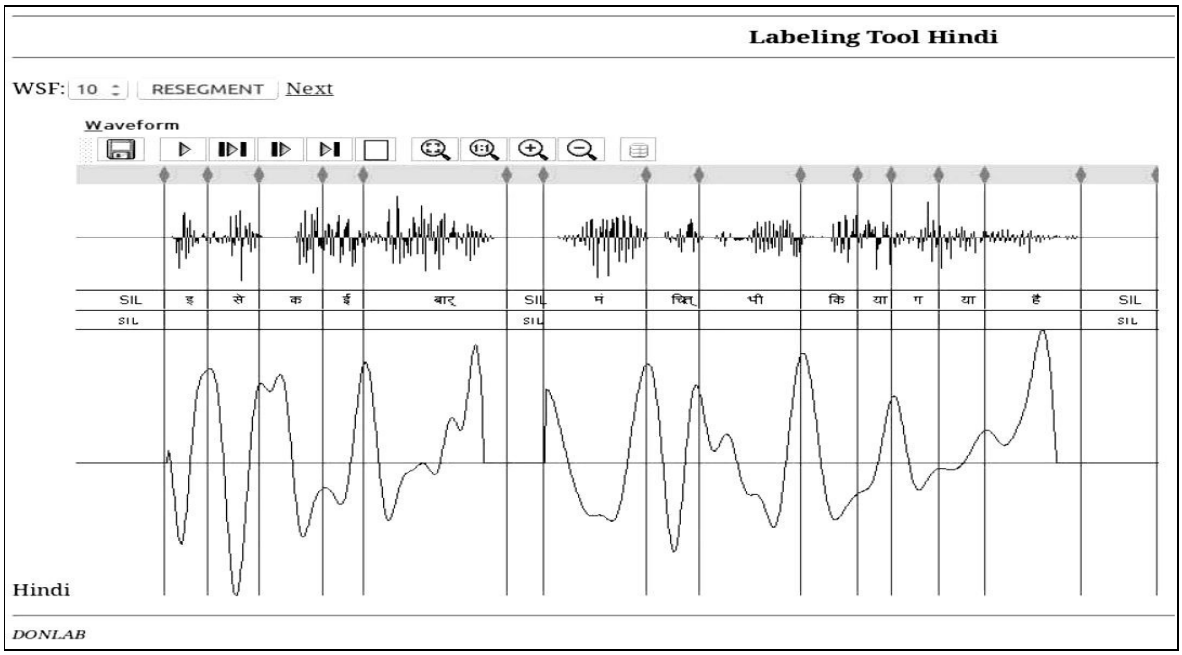


Fig. 3.2: Screenshot of semi-automatic labeling tool

3.4 Obtaining phone level segmentation

Syllable level boundaries were obtained using the labeling tool discussed in the previous section. The objective is to exploit this available information of syllable boundaries while

using HMMs for phone level segmentation. These syllable boundaries should act as anchor points, for the HMMs to find the phone boundaries.

As the syllable boundaries have been obtained already, each utterance in the speech data is split at the syllable level and stored separately. For example, if the utterance "text_001" has 'n' segments (syllables), the waveform is divided into 'n' files and stored separately as "text_001-1" to "text_001-n". Spectral parameters (Mel frequency cepstral coefficients) are extracted from these syllable waveforms. Baum-Welch embedded re-estimation [6] is performed on each of these syllables iteratively to build phone HMM models. Using these models, phone level alignment is performed on the syllable waveforms. By combining the individual phone level alignment of the syllables constituting an utterance, phone segmentation corresponding to the complete utterance is obtained.

Figure 3.3 summarises the above procedure with an example. When phone alignment is performed within syllable waveforms, the timestamps of the boundaries obtained are relative to the end time of the previous syllable in the utterance. To obtain the phone label file for the entire utterance, the end time of the previous syllable is added to all phone boundaries in the current syllable as shown in Table 3.1. For example, when phone alignment is performed on the syllable /baar/ (बार्), the end time of the previous syllable (0.728 second) in the utterance /ii/ (ई), is added to the boundaries of the phones /b/, /aa/, and /r/. After the phone alignment is performed within the syllable /baar/ (बार्), the end time of /b/ is 0.079 second. When the phone label file is formed, the end time of /ii/ (ई) is added and the end time of /b/ (ब्) becomes 0.807 second.

Table 3.1 shows a sequence of known syllable-boundaries for the phrase "इसे कई बार" (/isee kaii baar/) in an utterance and the boundaries of the phones that make up the syllable, which are obtained using the procedure explained in this section. The start time and the end time of a syllable (given in bold) shown in Table 3.1, remain unaltered in the phone label, as

forced alignment is performed intra-syllable.

Table 3.1: Phone labels from syllable labels

Syllable Label			**Phone Label**		
Beg	**End**	**Unit**	**Beg**	**End**	**Unit**
0.000	**0.234**	SIL	**0.000**	**0.234**	SIL
0.234	**0.363**	इ /i/	**0.234**	**0.363**	i
0.363	**0.516**	से /see/	**0.363** 0.398	0.398 **0.516**	s ee
0.516	**0.647**	क /ka/	**0.516** 0.576	0.576 **0.647**	k a
0.647	**0.728**	ई /ii/	**0.647**	**0.728**	ii
0.728	**1.113**	बार् /baar/	**0.728** 0.807 0.982	0.807 0.982 **1.113**	b aa r
1.113	**1.228**	SIL	**1.113**	**1.228**	SIL

The segmentation of an example phrase "उस रामानंद की खोज करता है" (/us raamaanaqd kii khoj kartaa hei/) is shown in Figure 3.4. Panel A shows the waveform. Panel B shows the segmentation at phone level given by the baseline HMM segmentation (i.e. flat start initialization, sentence level embedded re-estimation and sentence level forced alignment) explained in Chapter 2. Panel C is the segmentation given by the proposed semi-automatic method. Panel D shows the group delay function. The syllable boundaries obtained using the proposed method are highlighted using vertical lines. Except for the unvoiced fricative /s/ in the boundary of the syllable /us/, no other syllable boundary was moved during manual correction of labels using DONLabel. Three boundaries (peaks in the group delay function without vertical lines over it) were given within a syllable, that was also deleted during manual correction.

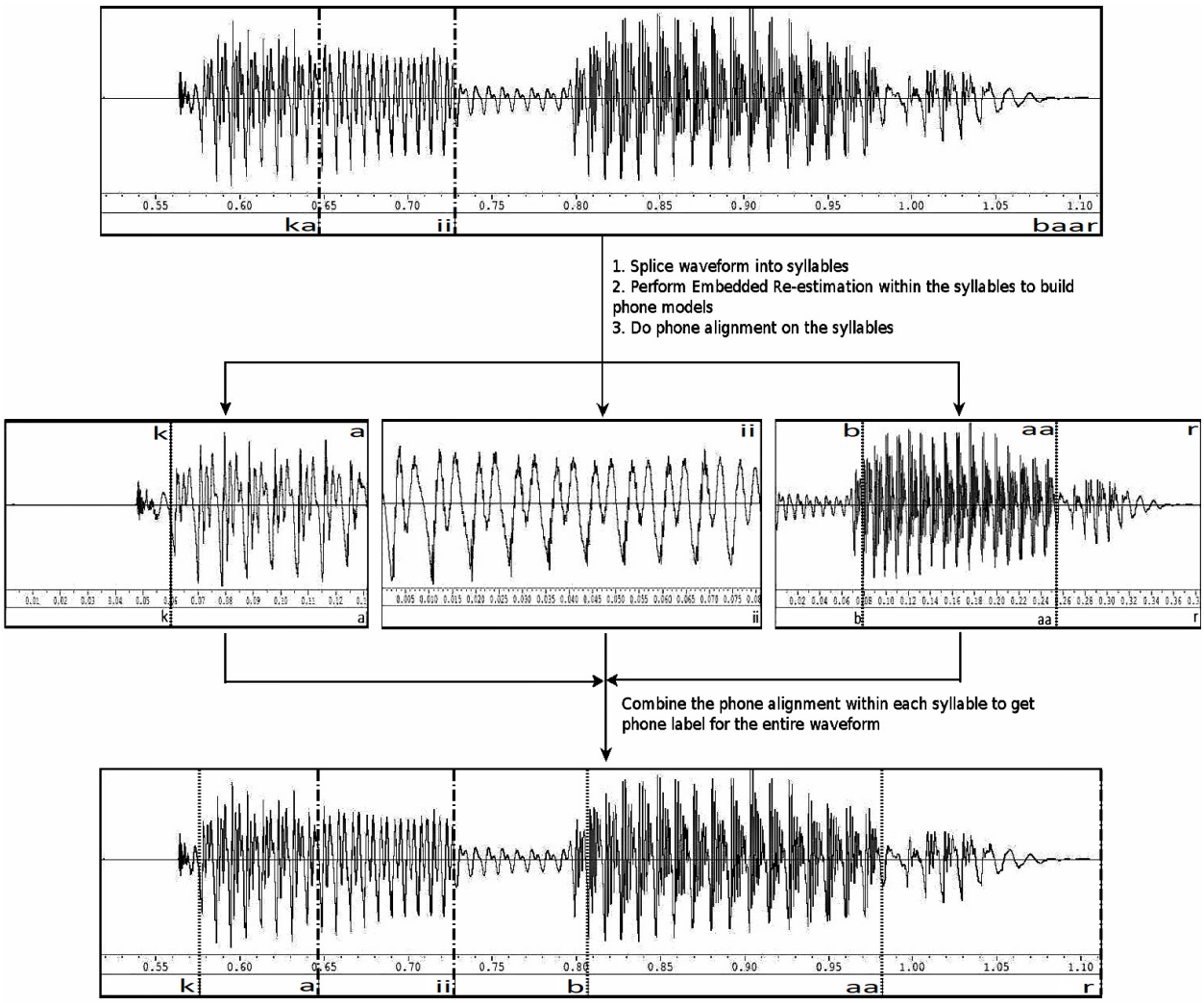


Fig. 3.3: Syllable enforced embedded re-estimation and alignment

The phone boundaries obtained using flat start segmentation in Figure 3.4, were observed to be inaccurate. For example, the syllable /kii/ is segmented only as /ii/ by flat start segmentation. Observing the waveform, it can be seen that the segment marked as /ii/ by flat start segmentation, also includes the stop consonant. The boundary of the phone /j/ in syllable /khoj/ being incorrect is also clearly visible in the waveform.

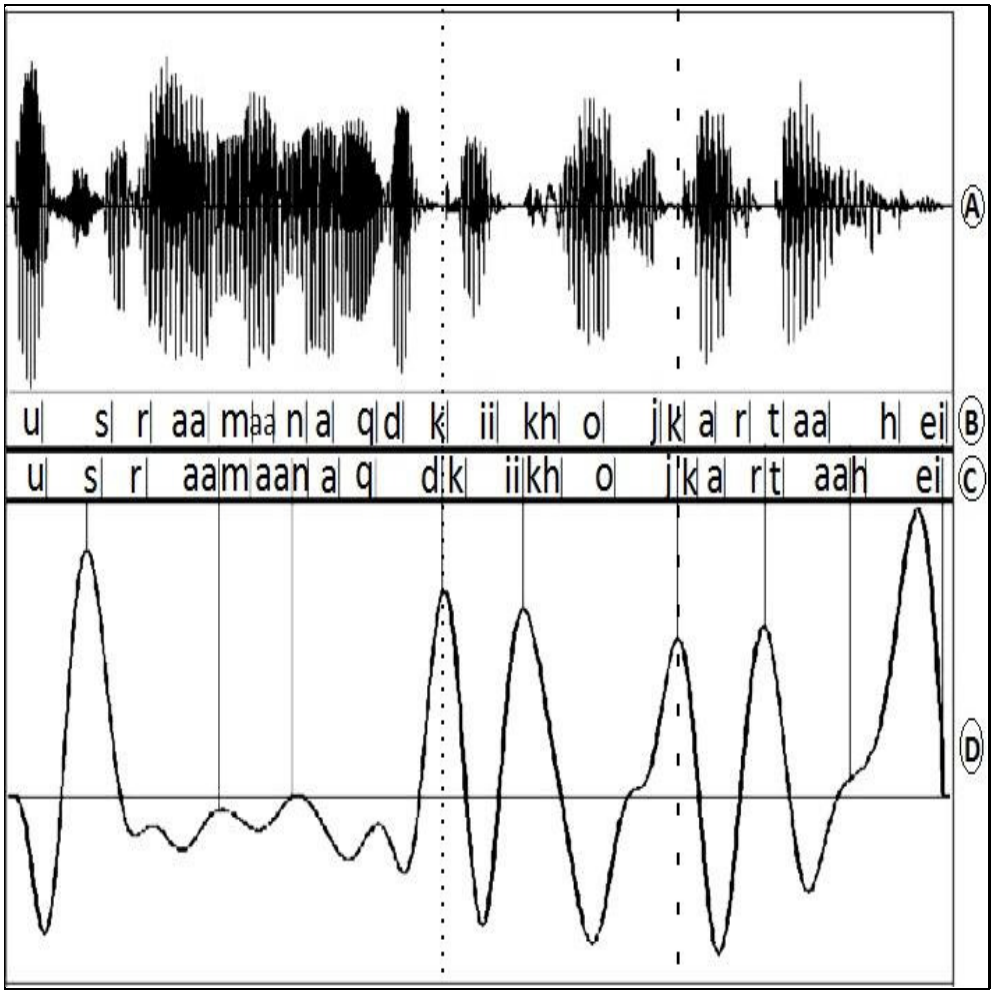


Fig. 3.4: Comparison of flat Start HMM and syllable enforced segmentation boundaries

3.5 Experiments and results

3.5.1 Experimental setup

1.35 hours of Hindi data (595 utterances) was used for training. All utterances have been recorded in a studio environment at 48000 Hz, 16 bits per sample, spoken by a single native Hindi male speaker. Segmentation is performed on this database with both the proposed semi-automatic approach and the baseline HMM based automatic flat start approach discussed in Chapter 2. Packages HTK-3.4.1 [6] and HTS-2.2 [14] were used. All 595 utter-

ances were labeled at the syllable level using DONLabel [37]. HMM models with five states and two mixture components per state were used to model each phone for both the proposed semi-automatic segmentation and baseline HMM segmentation.

3.5.2 Performance evaluation

Degradation Mean Opinion Score (DMOS) [22] was calculated for comparing naturalness and for intelligibility, semantically unpredictable sentences[2] were synthesized. Participants at the test were asked to listen to each sentence only once and transcribe the sentence. From the transcribed sentences, Word Error Rate (WER) was calculated.

Table 3.2: Comparison using DMOS and WER

System	**DMOS (Naturalness)**	**WER**
Semi-Automatic	2.98	3.19%
Flat Start HMM	2.89	5.04%

Pair comparison (PC) test [42] was also performed. Same sentences were synthesized with both the systems and the participants were asked to choose which system was better for each sentence during PC test. We denote the HTS system built with the proposed semi-automatic segmentation as "A" and the HTS system built with flat start HMM based segmentation as "B". During PC test, the order in which the sentences were played creates a bias. So, both "A-B" test and "B-A" test were performed with different sets of sentences. In "A-B" test, synthesized sentences of system "A" were played first and during "B-A" test,

[2]e.g. "आकाश मे सफ़ेद हाथी नाचता गाता जा रहा है"

synthesized sentences of system "B" were played first. Scores in column 1 and 2 of Table 3 are the percentages with which the first term of the pair is preferred. "A-B+B-A" in Table 3 denotes the order independent preference in percentage of system "A". It is calculated as

$$\text{"A-B+B-A"} = \frac{(\text{"A-B"} + (100 - \text{"B-A"}))}{2} \tag{3.4}$$

Table 3.3: Pair comparison tests, where "A" is the HTS system built with the proposed semi-automatic segmentation and "B" is the HTS system built with flat start HMM based segmentation

A-B	**B-A**	**A-B+B-A**
66.67	13.33	76.67

The results of pair comparison tests and other subjective evaluations clearly show that there is an improvement in synthesis quality when the proposed segmentation algorithm is used for building HTS systems.

The proposed semi-automatic segmentation was used in IIT Madras's submission [24] to Blizzard challenge 2014 for segmenting Hindi and Tamil data at the phone level. Conventional phone based HTS systems were built using the segmented data.

3.6 Summary

In this chapter, the importance of syllable boundaries in phone segmentation was discussed first. The group delay segmentation algorithm to detect regions of low energy using STE as a cue was explained in detail. It was shown how this algorithm can aid syllable level segmentation with minimal manual intervention. It was proposed that syllable boundary

information can be used for obtaining accurate phone boundaries by restricting parameter estimation of phone HMMs and subsequent alignment, at the syllable level. Comparison to baseline HMM based flat start automatic segmentation was performed by using the two approaches for segmenting the training data of a Hindi HTS system. A preference of 76.67% was attained.

CHAPTER 4

A Hybrid Approach to Segmentation of Speech

4.1 Introduction

The semi-automatic phone segmentation methodology proposed in Chapter 3 serves as a proof of concept that acoustic cues like STE can help in attaining better boundaries. However, the method has a few drawbacks. Importantly, the process is not completely automatic. Manual correction of syllable boundaries is a tedious task as it has to be performed on the entire database.

The approach is also very sensitive to manual errors. Inconsistent correction across humans is also a serious issue. Manual corrections are not consistent when insertion or movement of boundaries are involved. When a syllable is corrected erroneously, even subsequent use of HMMs won't make its location better.

Signal processing based syllable boundary detection and HMM based approach for phone segmentation seem to complement each other well. Nevertheless, the two are not tightly coupled, where the results of one are used to direct the performance of the other and vice-versa. HMMs are good at modeling the acoustic characteristics of a phone. HMM based forced alignment exploits the available transcription and thus does not miss actual boundaries by a large margin. On the flip side, HMMs are not good at modeling boundaries. On the other hand, group delay segmentation algorithm is good in detecting syllable boundary landmarks, but is agnostic to transcription and, therefore results in insertion and deletion

errors. In this chapter, an algorithm that combines these two contrasting approaches in a more efficient manner is proposed, to address the issues mentioned above.

4.2 Motivation

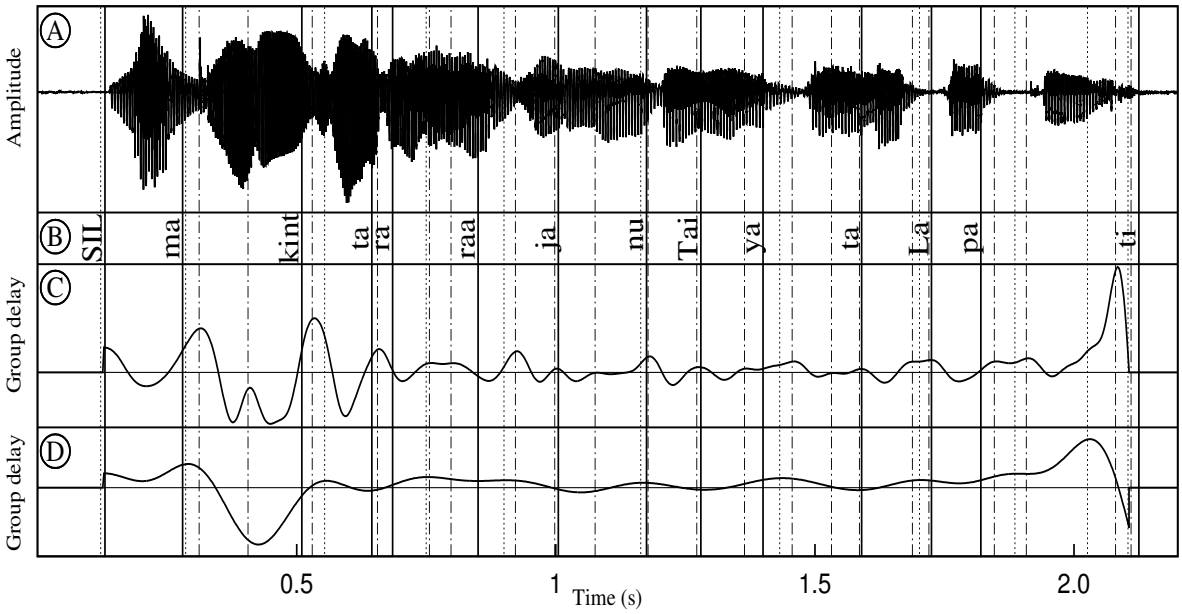


Fig. 4.1: Syllable boundaries given by HMM based segmentation (solid lines) and group delay based segmentation with WSF=10 (dashed lines) & WSF=30 (dotted lines)

Figure 4.1 shows the syllable level segmentation of the phrase "மகிந்தராஜனுடைய தளபதி" (makintararaajaˆnuTaiya taLapati[1]). The solid lines in Figure 4.1 show the syllable boundaries given by flat start initialized embedded training of monophone HMMs followed by forced Viterbi alignment (baseline HMM segmentation). Panel A shows the waveform. Panel B is the syllable aligned transcription obtained using HMMs. Panels C and D show two different group delay functions. The difference between the two functions is the change in the value of WSF. Observe that the boundaries given by the two are not identical. Nev-

[1]Tamil text written in ITRANS [43]

ertheless, although the lower value of WSF results in insertions and the higher value of WSF results in deletions, both match at some syllable boundaries. What is meant here is that, insertions lead to additional boundaries but the actual syllable boundaries are not misplaced. Similarly, deletions lead to some syllable boundaries get deleted, while other syllable boundaries are intact. This property of group delay based segmentation can be exploited. The function in the middle pane is obtained with 'WSF = 10' and the dashed lines are the corresponding boundaries. The other function is obtained with 'WSF = 30' and the dotted lines are the corresponding boundaries.

Although with WSF as '10', the number of spurious boundaries is large (resolution is high), nevertheless the correct boundaries are not misplaced. Such a group delay function should be first obtained for every utterance. This group delay function gives many low energy regions, which are possible candidates for being syllable boundaries. Boundaries given by HMMs are in the vicinity of the actual boundary. Boundary correction can be performed by moving the boundary to the closest low-energy region obtained from group delay segmentation. However, the location of boundaries obtained using group delay segmentation algorithm are not accurate in every case. So, correction rules are required to decide when boundary correction can be performed. Extensive experimentation with group delay segmentation shows that except for syllables starting or ending with a fricative, nasal, or starting with a semi-vowel, affricate, the boundaries are very accurate [36].

4.3 Hybrid approach

4.3.1 Correction rules

Extensive experimentation with group delay segmentation shows that except for syllables starting or ending with a fricative ('sx', 's', 'sh', 'z', 'f'), nasal ('n', 'nj', 'nx', 'nd', 'm,

'ng', 'mq'); or starting with a semi-vowel ('y', 'r', 'l', 'w', 'zh', 'lx', 'rx'), affricate ('c', 'ch', 'j', 'jh'), the boundaries are very accurate [36]. The correction rules depend only on the two phones present on either side of the boundary. Say, a decision has to be made as to whether a boundary between *syllable1* and *syllable2* should be corrected. Let *syllable1* be represented as "$P^* e_p$" and *syllable2* be represented as "$b_p P^*$", where P^* is a sequence of zero or more phones, b_p is the first phone of *syllable2* and e_p is the last phone of *syllable1*. The decision depends only on b_p and e_p. Syllable boundaries given by HMMs can be corrected if,

- (e_p **NOT in the set** {fricative, nasal, silence}) **AND** (b_p **NOT in the set** {fricative, affricate, nasal, semivowel, silence}).

4.3.2 The hybrid segmentation algorithm

Some segments given by HMM based segmentation suffer from gross durational errors, where the duration of one segment is significantly larger or smaller than the others. Hidden semi-Markov models (HSMM) [44] have explicit state duration probability distributions. All states of monophone HSMMs were initialized with global mean and global variance, followed by embedded reestimation. When alignment is performed using HSMMs, it was observed that although durational errors reduced significantly, the boundaries obtained by forced alignment using HSMMs were worse than those obtained with HMMs. HMM models are built with the segmentation obtained using HSMMs and then HMM based forced alignment is performed. This corrects durational errors in most of the places and the location of boundaries are also not very inaccurate.

If the syllable does not end with a nasal, fricative, silence or if it is not followed by a nasal, fricative, affricate, silence or a semi-vowel, the boundary of that syllable is moved to the nearby region of low energy given by the group delay function with high resolution as

explained in Section 4.2.

During embedded training [27], the transcription is used to construct a composite HMM for each utterance by concatenating phone HMMs. Embedded Baum-Welch re-estimation is performed and all monophone HMM models are updated simultaneously.

As segmentation at the syllable level is available, waveforms are split at the syllable level and embedded training is performed on these syllables similar to the semi-automatic approach explained in Chapter 3. Now, the composite HMMs obtained will be syllable HMMs and not utterance HMMs. Embedded Baum-Welch re-estimation is restricted to the syllable boundaries and monophone models are built [45]. These models are more robust as re-estimation is performed on shorter segments of speech. Using the new monophone models, forced alignment is performed on the entire utterance. The boundaries obtained using forced alignment are again compared with that of the group delay boundaries and corrected as before to obtain the final syllable level segmentation. Although, locations of low energies remain the same, this comparison is performed again because of the following reason. When HMM based forced alignment was performed initially, the models were not robust and hence the boundaries in some places can be misplaced by a large margin. During correction phase, the boundaries could have moved to the wrong peak. When alignment is performed again using more robust models, the boundaries given by HMMs will probably come closer to the desired peak.

To obtain segmentation at the phone level, embedded Baum-Welch re-estimation is performed at the syllable level again to refine the monophone models. With these models phone level alignment is performed on the syllable waveforms [45]. By combining the phone level alignment of the syllables constituting an utterance, phone segmentation corresponding to the entire utterance is obtained as shown in Figure 4.3. In Figure 4.3, Panel A shows the waveform and panel E shows the spectrogram. Panel D shows the group delay (GD) func-

tion. Syllable level alignment obtained using the proposed method is given by panel B and the phone level segmentation obtained after forced alignment within the syllables is shown by panel C. Figure 4.2 gives the flowchart for the proposed method.

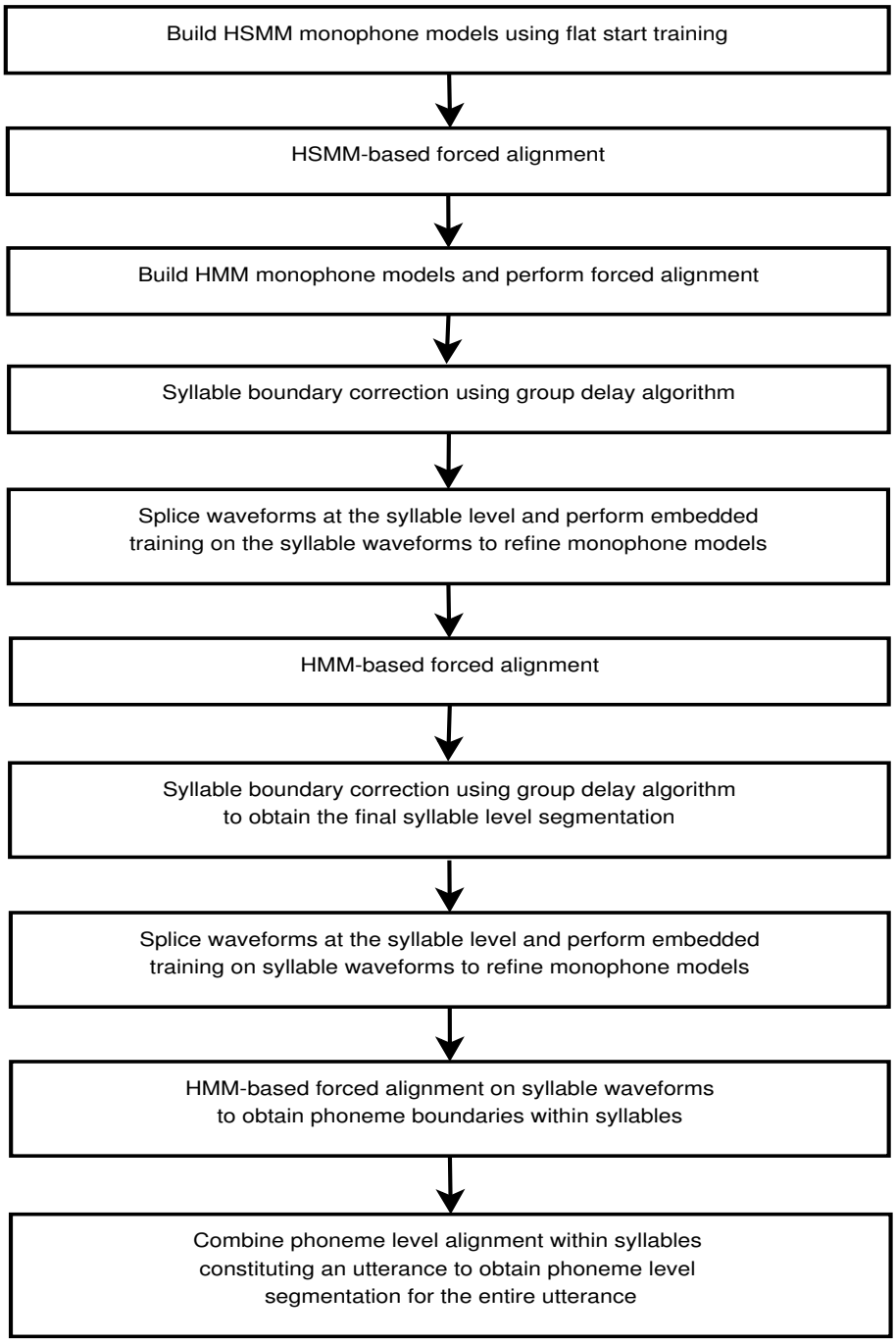


Fig. 4.2: Steps involved in the proposed hybrid method

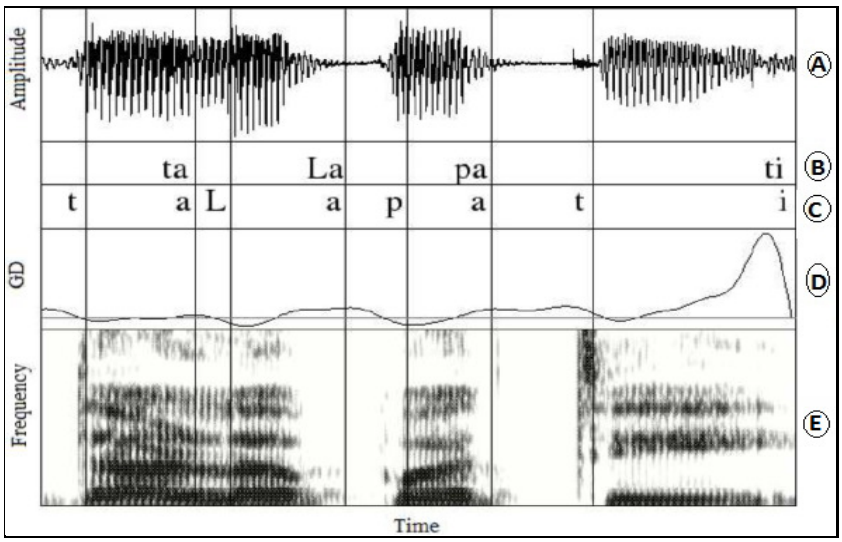


Fig. 4.3: Phone level segmentation after alignment within syllables

4.4 Experiments and results

4.4.1 Experimental setup

730 utterances of Tamil data spoken by a single native female speaker is used. For training HSMMs, mel cepstral coefficients and log of the fundamental frequency are used as features. All monophones are modelled as 5-state, 1-mixture HSMMs. For training HMMs, MFCCs are used and all monophones are modeled as 3-state, 2-mixture models. HTS-2.2 [14] and HTK-3.4.1 [27] are used. WSF is set as '10' in group delay segmentation algorithm.

4.4.2 Segmentation accuracy

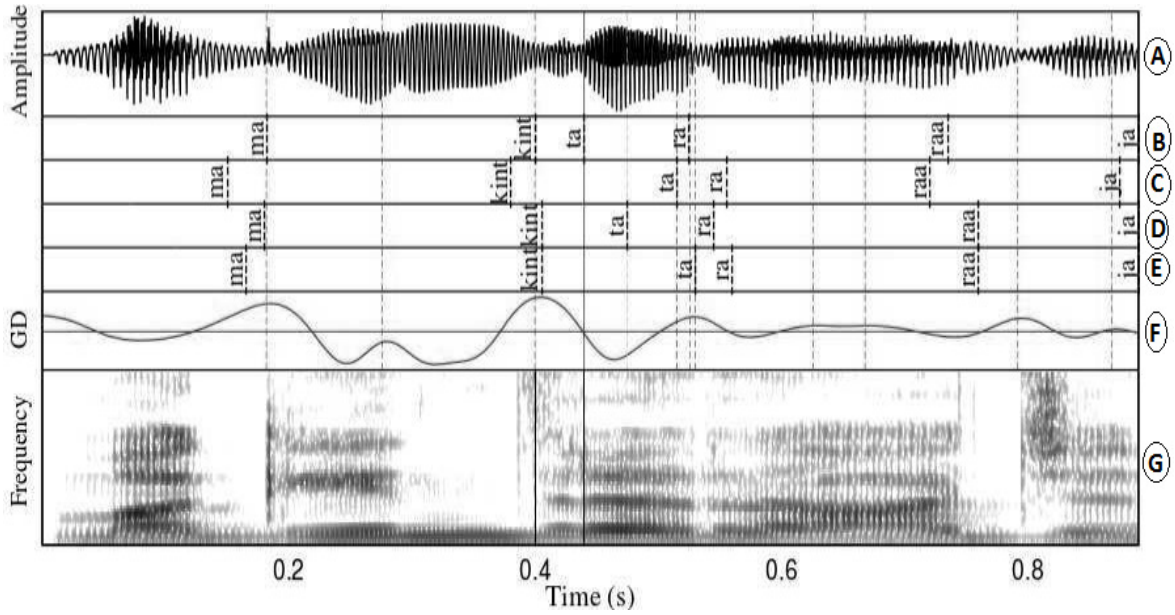


Fig. 4.4: Syllable level segmentation given by the proposed hybrid method compared to HMM, HSMM and HSMM followed by HMM

Panel A of Figure 4.4 shows the waveform. Panels B, C, D and E are the syllable aligned transcriptions given by the proposed method, HMMs, HSMMs and flat start training of HSMMs followed by HMMs respectively. Panel F shows the group delay function and panel G shows the spectrogram. The boundary of syllable "kint" is slightly misplaced when flat start training of HSMMs followed by HMMs are used. When boundary correction is performed in the proposed method, group delay peak in the proximity is taken and the location of the boundary is rectified.

Note that in Figure 4.4, the syllable "ta" marked in the spectrogram was identified correctly only by the proposed method (solid line). Both the statistical and the group delay approaches failed to find the boundary correctly. Group delay based approach did not detect

the boundary as the succeeding syllable “ra” starts with a semi-vowel. But in the proposed method, when the two approaches work in tandem, phone models are built in a robust manner, which leads to the correct alignment of the syllable. The inference from this example is that it is not necessary to correct every boundary using signal processing. Even correcting a subset of boundaries can help improve the overall segmentation. This example also shows that restricting parameter estimation of phone HMMs to a smaller segment gives more robust models.

Figure 4.5 shows another example. Panel A of Figure 4.5 shows the waveform. Panels B and C show the syllable aligned transcription given by the baseline HMM based segmentation and the proposed hybrid segmentation algorithm respectively. Panel E shows the spectrogram. The group delay function is shown by panel F along with the location of its peaks. The syllable segment “பட் (‘p’ ‘a’ ‘tx’)” given by baseline HMM segmentation is clearly incorrect. A significant portion of the preceding syllable “கை (‘tx’ ‘ai’)” and the succeeding syllable “ட (‘tx’ ‘a’)” is incorrectly included within the boundaries of syllable “பட்”. The same syllable segment “பட்” highlighted in Figure 4.5 is the output of hybrid segmentation. Here, the boundaries of all three syllables “பட்”, “கை” and “ட” are rectified because of STE based boundary correction.

4.4.3 Performance evaluation

With the segmentation given by the hybrid method and the conventional HMM based method, HTS systems [14] were built. Both phone based HTS and syllable based HTS [15] were built for Tamil. Subjective evaluations were performed on these TTS systems. First, semantically unpredictable sentences (SUS) were synthesized and participants of the test were made to transcribe them. Then, word error rate (WER) was calculated. The hybrid approach results in a lower WER as indicated in Table 4.1 .

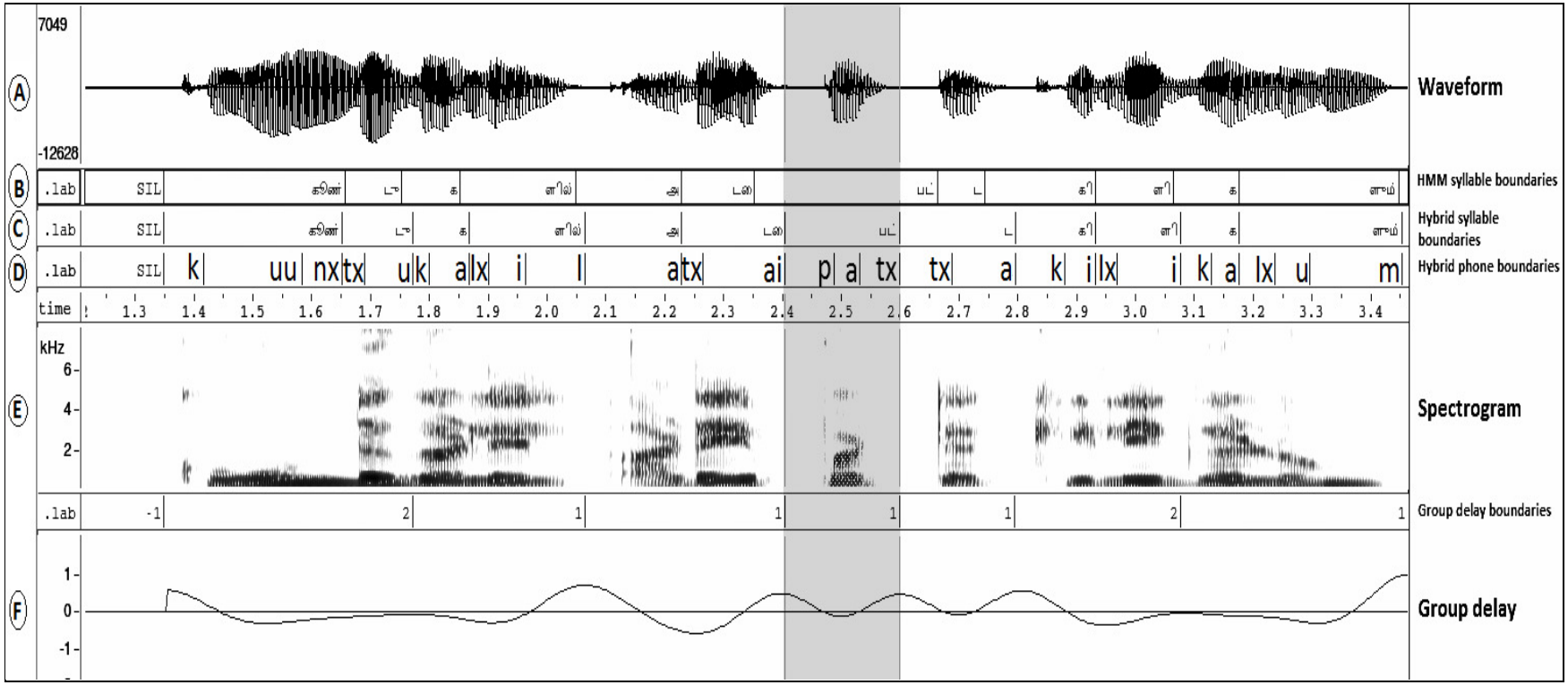


Fig. 4.5: Comparison of baseline HMM segmentation with hybrid segmentation

Table 4.1: Comparison using WER

System	**WER**
HTS - Syllable (HMM Segmentation)	11.11%
HTS - Syllable (Hybrid Segmentation)	7.07%
HTS - Phone (HMM Segmentation)	4.04%
HTS - Phone (Hybrid Segmentation)	1.01%

Pair comparison (PC) test [42] was also performed. For this comparison, the HTS system built with the hybrid segmentation method is referred to as "A" and the HTS system built with HMM based segmentation is referred to as "B". The results of PC test are shown in Table 4.2.

Table 4.2: Pair comparison tests

HTS - Syllable			**HTS - Phone**		
A-B	**B-A**	**A-B+B-A**	**A-B**	**B-A**	**A-B+B-A**
75	20	77.5	70	15	77.5

The proposed automatic hybrid segmentation algorithm was used in IIT Madras's submission [24] to Blizzard challenge 2014 for segmenting Gujarati, Telugu and Rajasthani datasets. Both syllable based USS systems and phone based HTS systems were trained using the segmented data.

4.5 Summary

In this chapter, the drawbacks of the semi-automatic segmentation algorithm proposed in Chapter 3 was discussed first. It was proposed that syllable boundary corrections can be performed automatically by overestimating the number of low energy regions using group delay segmentation algorithm. Correction rules were proposed to optimize boundaries obtained using HMMs, by moving certain boundaries to low energy regions in the vicinity. Then, a hybrid automatic segmentation algorithm was proposed. The proposed hybrid segmentation algorithm was compared to the baseline HMM segmentation by using the two approaches for segmenting the training data of a Tamil HTS system. Subjective evaluation using pair comparison tests shows 77.5% preference for the HTS system built with hybrid segmentation.

CHAPTER 5

Importance of Sub-Band Spectral Flux in Segmentation

5.1 Introduction

The main aim of this chapter is to propose spectral change also as a cue along with STE to perform syllable boundary correction in the hybrid segmentation algorithm. The algorithm proposed in Chapter 4 does not correct any syllable boundary consisting of fricatives, affricates and nasals. Boundaries of sibilant fricatives (‘sx’, ‘s’, ‘sh’, ‘z’) and affricates (‘c’, ‘ch’, ‘j’, ‘jh’) have significant and clear spectral change. This chapter proposes a method to detect these boundaries.

The boundary correction rules for hybrid segmentation is modified in this chapter to correct boundaries at fricatives, affricates and nasals. The acoustic realization of a phone seems to be different depending on the position in the syllable. The positional context is therefore added to phones before syllable level embedded re-estimation to model them separately.

5.2 Spectral change as a cue

Spectral change as a function of time that can give a cue to fricative/affricate boundaries is required to enable correction. Weighted slope metric [26] is a measure of both spectral band energy difference and spectral transitions. Weighted slope metric is directly used in

[31] to correct all phone boundaries. As discussed in Chapter 2, the authors of [31] have acknowledged that when the function is used directly, it gives a lot of spurious peaks.

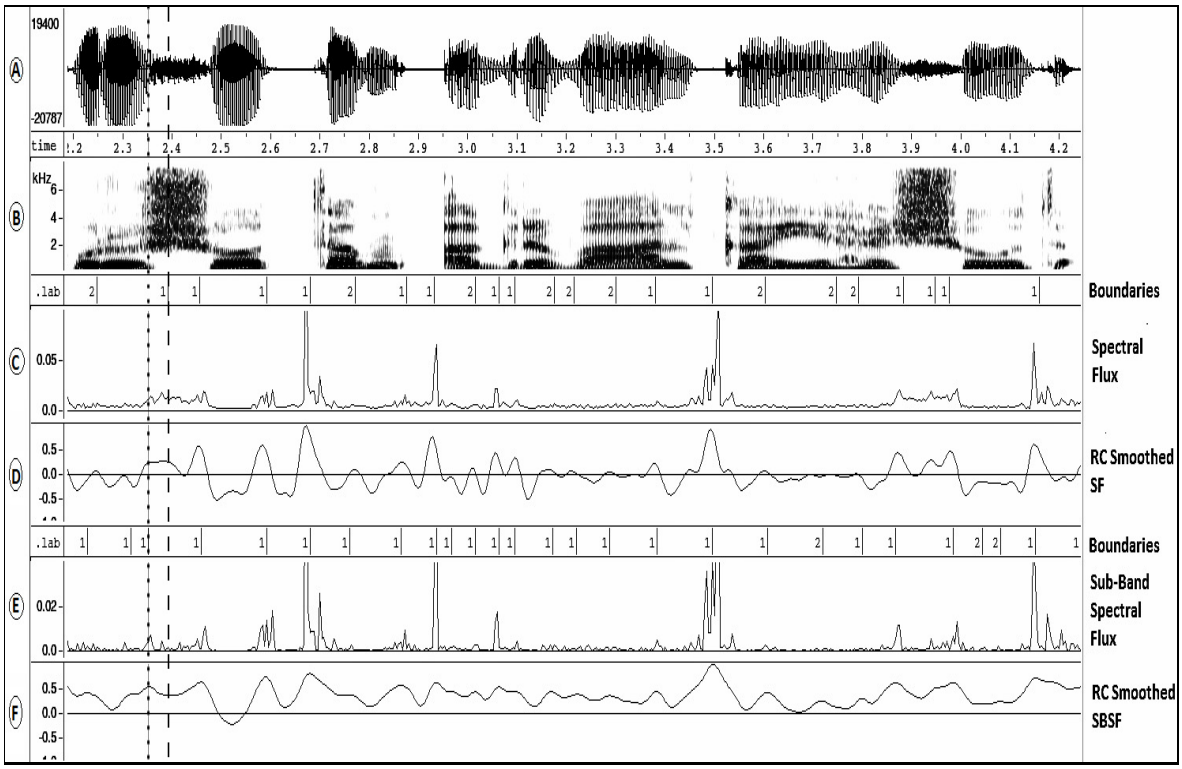


Fig. 5.1: Comparison of sub-band spectral flux with spectral flux

Spectral flux is a simpler function than weighted slope metric that gives a measure of spectral change. Spectral flux is the Euclidean distance between the normalized power spectrum of a frame and the normalized power spectrum of its previous frame. The power spectrum is normalized by dividing all its coefficients by its maximum coefficient.

In Figure 5.1, panels A and B show the waveform and the spectrogram respectively. Panel C shows the spectral flux. Spectral flux has many prominent peaks in certain sibilant fricative and affricate regions. The smoothed version of spectral flux (the algorithm for smoothing is explained in the next section) is shown by panel D along with its peaks. The

location of boundaries obtained using spectral flux is not accurate in certain places (like the boundary in Figure 5.1, shown using a dashed vertical line).

Phone boundaries can be characterized by energy changes in different bands of the spectrum [31]. This Section defines a new function named sub-band spectral flux (SBSF) . To compute SBSF, the normalized power spectrum is uniformly divided into four bands (assuming sampling rate as 16 KHz) and the energies within these bands are calculated. The squared difference between the band energies of a frame and the band energies of its previous frame gives SBSF. If 512 point FFT is used for calculating power spectrum, spectral flux is a distance between two vectors of 512 dimensions. On the other hand, SBSF is a distance between two 4 dimensional vectors. If $E[i]$ is the energy in the i^{th} band of the normalized power spectrum, the SBSF of frame n is calculated as

$$SBSF_n = \sum_{i=1}^{4} (E_n[i] - E_{n-1}[i])^2 \tag{5.1}$$

Panel E in Figure 5.1 shows SBSF. The smoothed version of SBSF is shown by panel F. The erroneous boundary given by spectral flux is detected accurately when SBSF is used (shown using a dotted vertical line).

A syllable can be defined in terms of STE as explained previously. The peaks in the reciprocal of the STE function correspond to syllable boundaries. On the other hand, when SBSF is used as a boundary detector, the peaks of SBSF correspond to phone boundaries when the particular phone transition is accompanied by a significant change in spectral characteristics. This property is not true for all phone transitions, eg. a transition from a vowel to a semi-vowel.

5.3 Boundary detection algorithm for sibilant fricatives and affricates

SBSF is a positive function. When it is made symmetric, it looks like a magnitude spectrum (inspired from group delay segmentation algorithm [11] explained in Chapter 3). So, techniques used to smooth magnitude spectrum can be applied for this signal too. In this algorithm, root cepstral [40] smoothing is used. The algorithm is illustrated as a flowchart in Figure 5.2. The algorithm is explained below.

1. Compute SBSF function $S[m]$ where $m = 1, 2, ..., M-1$ from the given speech utterance using Equation 5.1. Let the minimum non-zero value of the SBSF function be S_{min}. Let, $S[0] = S_{min}$.

2. Compute the order N of FFT as given below

$$N = 2^{\lceil log(2M)/log(2) \rceil} \tag{5.2}$$

3. Compute $S^1[m]$ as $S[m]^\gamma$ after appending $(N/2 - M)$ number of S_{min} to the sequence $S[m]$, where $\gamma = 0.001$[1]

4. Construct the symmetric part of the sequence $S^1[m]$ by lateral inversion. The resulting sequence is positive and symmetric. So, it resembles the magnitude spectrum of an arbitrary real signal. Let's call this sequence $S[K]$.

5. The IDFT of $S[K]$ is computed. The resultant signal is the root cepsrum [40] and the causal portion of the same is a minimum phase signal [41]. Let's call this causal sequence $s_c[n]$.

6. A single sided Hanning window is applied on $s_c[n]$ and its magnitude spectrum

[1] A small value of γ is chosen to reduce the dynamic range of SBSF

$S_{mag}[K]$ is computed. The size of the window applied is

$$N_c = \frac{Length\ of\ SBSF}{Window\ scale\ factor\ (WSF)} \tag{5.3}$$

7. Peaks in $S_{mag}[K]$ are detected and they are locations of significant spectral change.

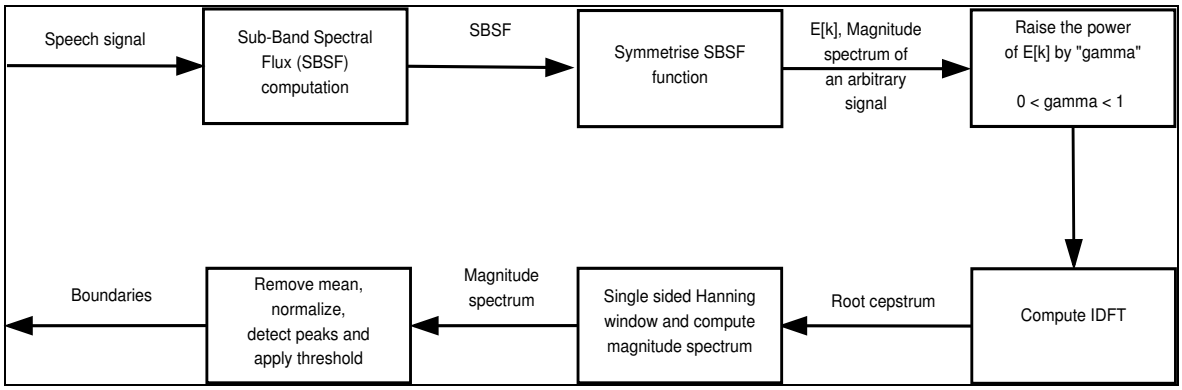
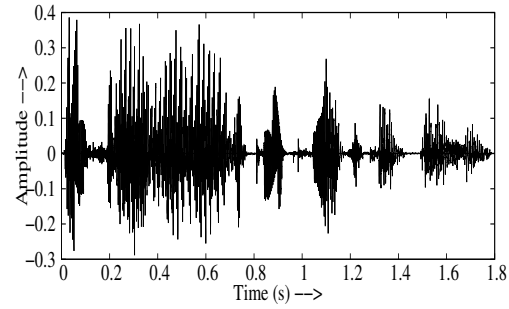
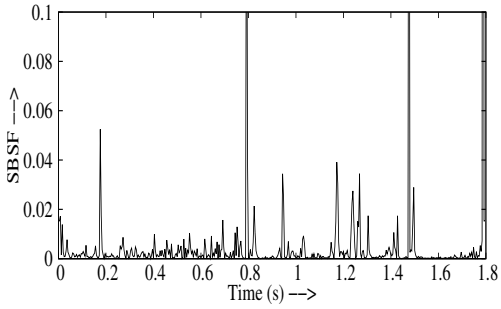


Fig. 5.2: Steps involved in SBSF based boundary detection algorithm

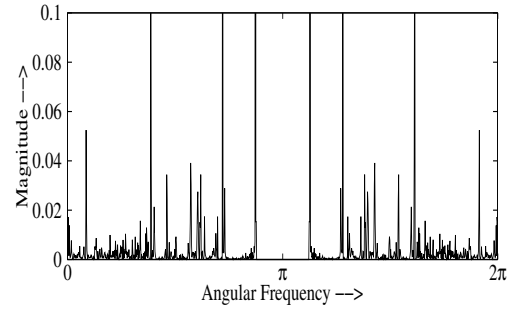
Figure 5.3 illustrates with an example, the outputs of various blocks in SBSF based boundary detection algorithm.



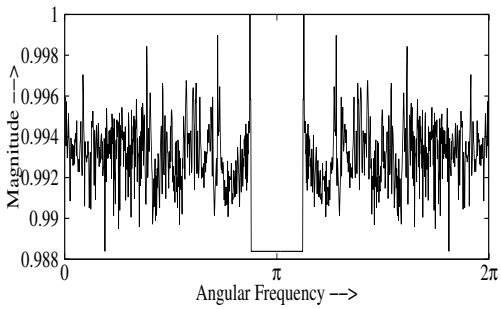
(a) Speech signal



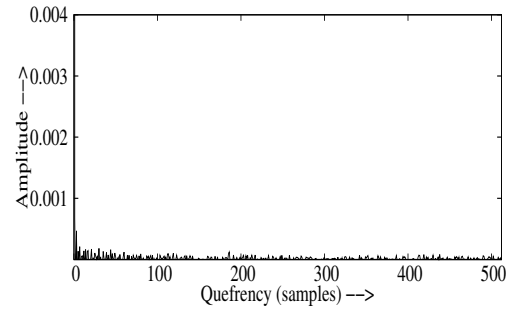
(b) Sub-band spectral flux



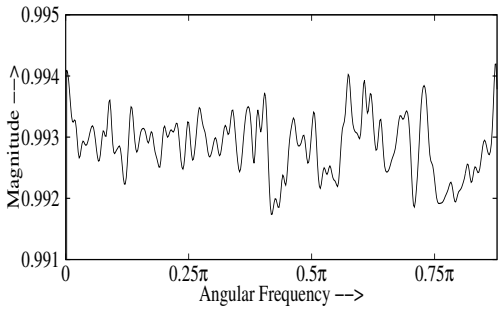
(c) SBSF symmetrized to resemble magnitude spectrum



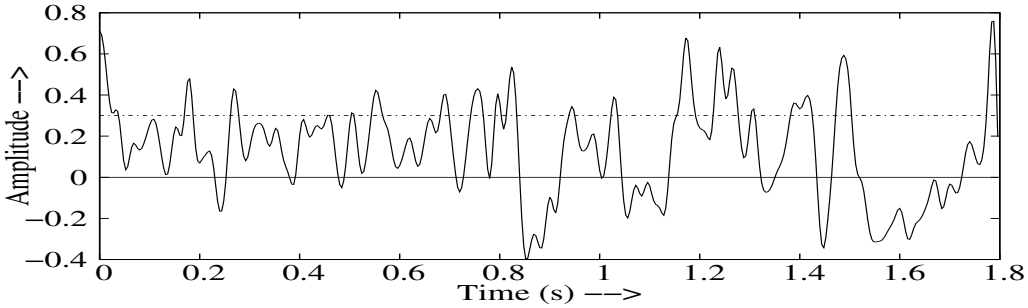
(d) Magnitude spectrum with reduced dynamic range



(e) Root cepstrum



(f) Root cepstral smoothed magnitude spectrum



(g) Smoothed version of SBSF with threshold. Peaks above the threshold are locations of significant spectral change

Fig. 5.3: Outputs of various blocks in SBSF based boundary detection algorithm

5.4 The modified hybrid segmentation approach

A syllable can be defined in terms of STE as explained in Chapter 2. When the inverted STE is used as a boundary detector, the physical interpretation of detected low energy locations are syllable boundaries. On the other hand, when SBSF is used as a boundary detector, it gives phone boundaries if the particular phone transition is accompanied by a significant change in spectral characteristics. This property is not true for all phone transitions, eg. a transition from a vowel to a semi-vowel. Although SBSF has the potential for correcting certain phone boundaries directly, the new hybrid segmentation algorithm uses SBSF only for syllable boundary correction.

Correction rules in hybrid segmentation have to be modified, primarily to incorporate additional cues given by SBSF. Another change is the use of thresholds. Previously, all peaks of the smoothed and inverted STE were used for correction and a peak was chosen based on the proximity to boundary given by HMM. The height of the peak is also important. After smoothing inverted STE and SBSF, the DC component is removed (mean is made zero) and the signal is normalized between -1 and +1. Only peaks above some threshold (peaks above the line "$y = threshold$") are considered as boundaries.

5.4.1 Modified correction rules

The main objective of the new correction rules is not to increase the number of corrections, but to perform more reliable corrections. In the previous algorithm, syllable boundaries at nasals, fricatives and affricates were not corrected using STE. This is remedied here. Another objective is to perform only corrections of high confidence.

Similar to the previous chapter, the correction rules depend only on the two phones adjacent to the boundary. Say, a decision has to be made as to whether a boundary between

syllable1 and *syllable2* is to be corrected. Let *syllable1* be represented as "P^*e_p" and *syllable2* be represented as "b_pP^*", where P^* is a sequence of zero or more phones, b_p is the first phone of *syllable2* and e_p is the last phone of *syllable1*. The decision depends only on b_p and e_p.

Stop consonants ('b', 'bh', 'd', 'dh', 'dx', 'dxh', 'dxhq', 'dxq', 'g', 'gh', 'gq', 'k', 'kh', 'khq', 'kq', 'p', 't', 'th', 'tx', 'txh') are non continuant sounds [32] with two phases namely, closure and burst. Stop consonants are produced by building up pressure behind a total constriction somewhere in the oral tract (i.e. closure) and then suddenly releasing the pressure (i.e. burst). When b_p is a stop consonant, irrespective of e_p, there will be a significant dip in energy because of closure. Although this is true for all stop consonants, only unvoiced stop consonant boundaries are corrected for the following reason. For voiced stops ('b', 'bh', 'd', 'dh', 'dx', 'dxh', 'dxhq', 'dxq', 'g', 'gh', 'gq'), there is a small amount of low frequency energy called *voice bar*, even in closure. In unvoiced stops ('k', 'kh', 'khq', 'kq', 'p', 't', 'th', 'tx', 'txh'), because of the absence of voicing during closure, there is a *stop gap*.

For sibilant fricatives and affricates, the energy is prominent only in the higher frequency bands and hence SBSF can be used. Similarly, for nasals, the energy is prominent only in the lower frequency bands, and hence, it should be possible to use SBSF for detecting their boundaries. Unlike fricatives and affricates, the influence of SBSF on nasals are not consistent. Thus, SBSF was used for correcting nasals only when b_p is a stop consonant. A hierarchy of rules is proposed below:

- **If** $b_p ==$ (unvoiced stop), use STE with $threshold_1$ for correction.
- **Else if** $e_p ==$ (unvoiced stop), use STE with $threshold_2$ for correction.
- **Else if** ($b_p ==$ (fricative **OR** affricate)) **XOR** ($e_p ==$ (fricative **OR** affricate)), use SBSF with $threshold_3$ for correction.

- **Else if** ($b_p ==$ (unvoiced stop)) **AND** ($e_p ==$ (nasal)), use SBSF with $threshold_3$ for correction.
- **Else** do not correct.

For experiments in this chapter, the values of thresholds were empirically chosen to be $threshold_1 = 0.5$, $threshold_2 = 0.2$ and $threshold_3 = 0.3$. Apart from these thresholds, a durational threshold is also used during correction. The new syllable level label file is initialized with the timestamps of the label file obtained using HMMs. Boundary corrections are performed in the left to right order with respect to the waveform. A boundary correction is made only if it results in the duration of the present syllable and the succeeding syllable to be greater than 100 ms.

5.4.2 The modified hybrid segmentation algorithm

Unlike the algorithm proposed in the previous chapter, the process is started with HMMs directly. HSMMs were no longer needed. As an additional cue and better correction rules were used, HSMMs did not make any difference. Baseline automatic HMM segmentation is performed initially. Then syllable boundary correction is performed using the new rules.

The acoustic realization of the first and last phones in a syllable is significantly different from the ones in the middle. So, these phones are modeled separately by adding the syllable positional context. For instance, phone transcription of syllable "दूक(d uu k)" becomes "beg-d uu k_end" after adding context information. Syllables with a vowel in isolation are modeled separately. Flat start initialization is performed for syllable context phone models.

Syllable level embedded training is performed to train these syllable context phone models. Forced alignment is performed using syllable context phone models and then boundary correction is performed again. Syllable level embedded re-estimation is repeated to refine the models and finally forced alignment is performed within syllables to obtain phone

boundaries. Figure 5.4 shows the sequence of steps involved in the new hybrid segmentation algorithm in the form of a flowchart.

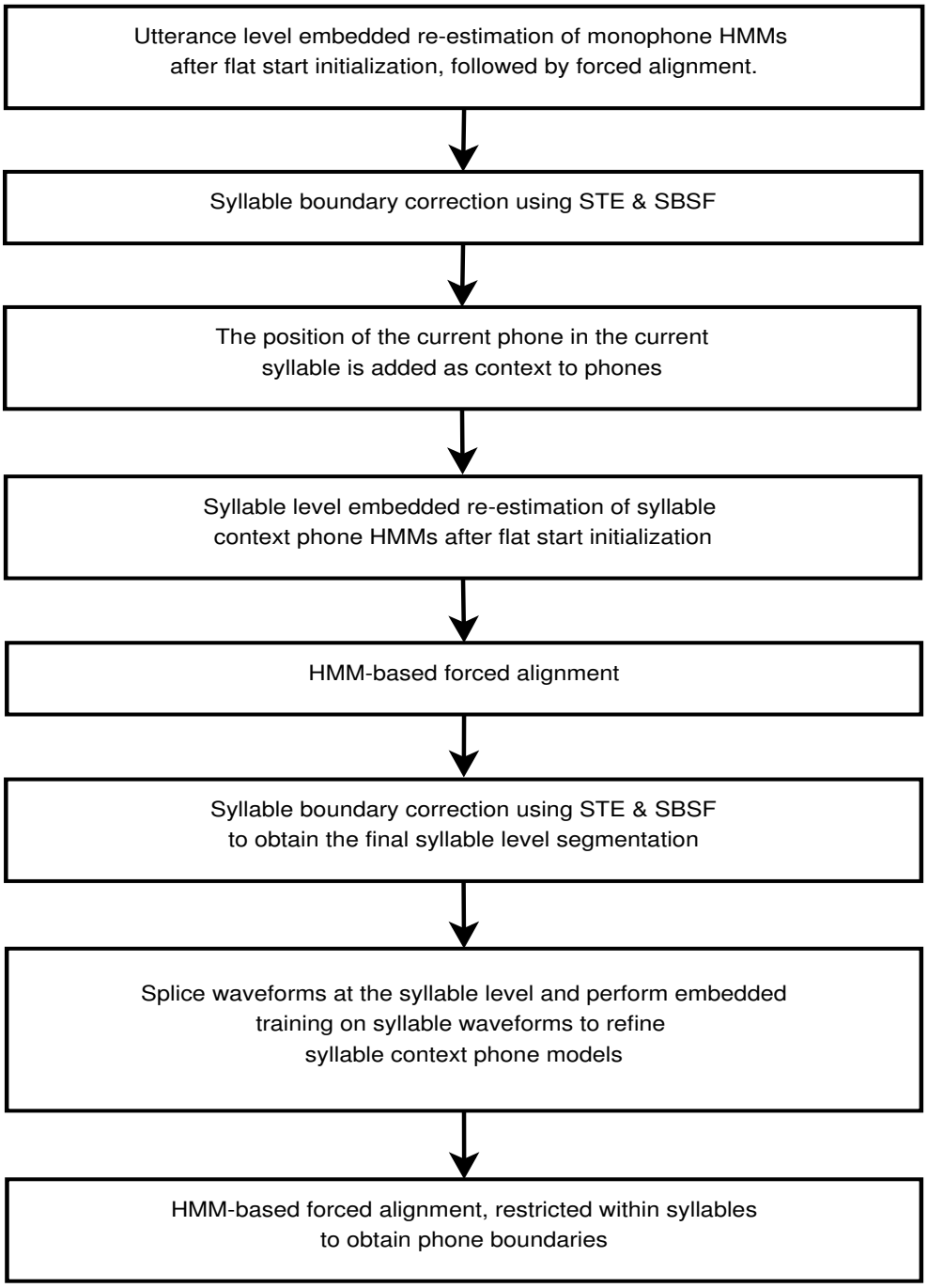


Fig. 5.4: Steps involved in the new hybrid segmentation algorithm

5.5 Experiments and results

5.5.1 Experimental setup

Three datasets were used for experiments:

1. Dataset 1 - 717 utterances (2.25 hrs) of Tamil, spoken by a single native female speaker
2. Dataset 2 - 871 utterances (2 hrs) of Hindi, spoken by a single native male speaker
3. Dataset 3 - 873 utterances (1.75 hrs) of Hindi, spoken by a single native female speaker

For training HMMs, MFCCs were used as features and all vowels were modeled as 5-state, 2-mixture models, and all consonants were modeled as 3-state, 2-mixture models. WSF was set as '6' in STE based group delay segmentation algorithm and it was set as '2' in SBSF based segmentation algorithm.

5.5.2 Discussions

Figure 5.5 illustrates the boundary refinement process with an example from Dataset 3. Panel A in Figure 5.5 shows the waveform. Panel B shows the syllable level segmentation given by the baseline HMM approach. Panels C to F show the output of the hybrid segmentation algorithm in various stages. Panel G shows the spectrogram. Panel H shows the smoothed inverted STE function and its peaks. The stippled line in the plot is the threshold at '0.2' ($threshold_2$) and the dashed line is the threshold at '0.5' ($threshold_1$). Panel I shows smoothed SBSF and its peaks. The dotted line in the plot is the threshold at '0.3' ($threshold_3$).

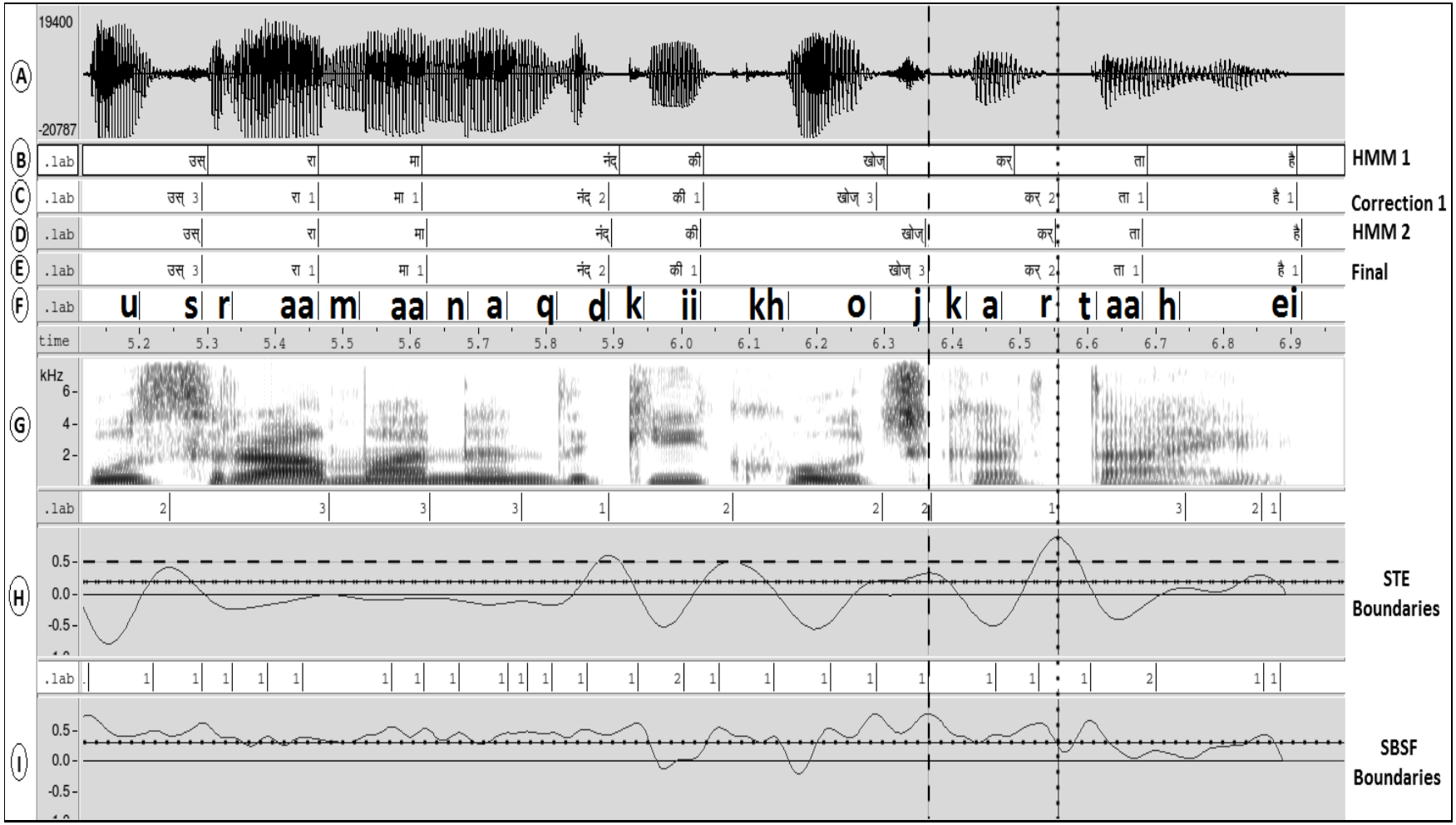
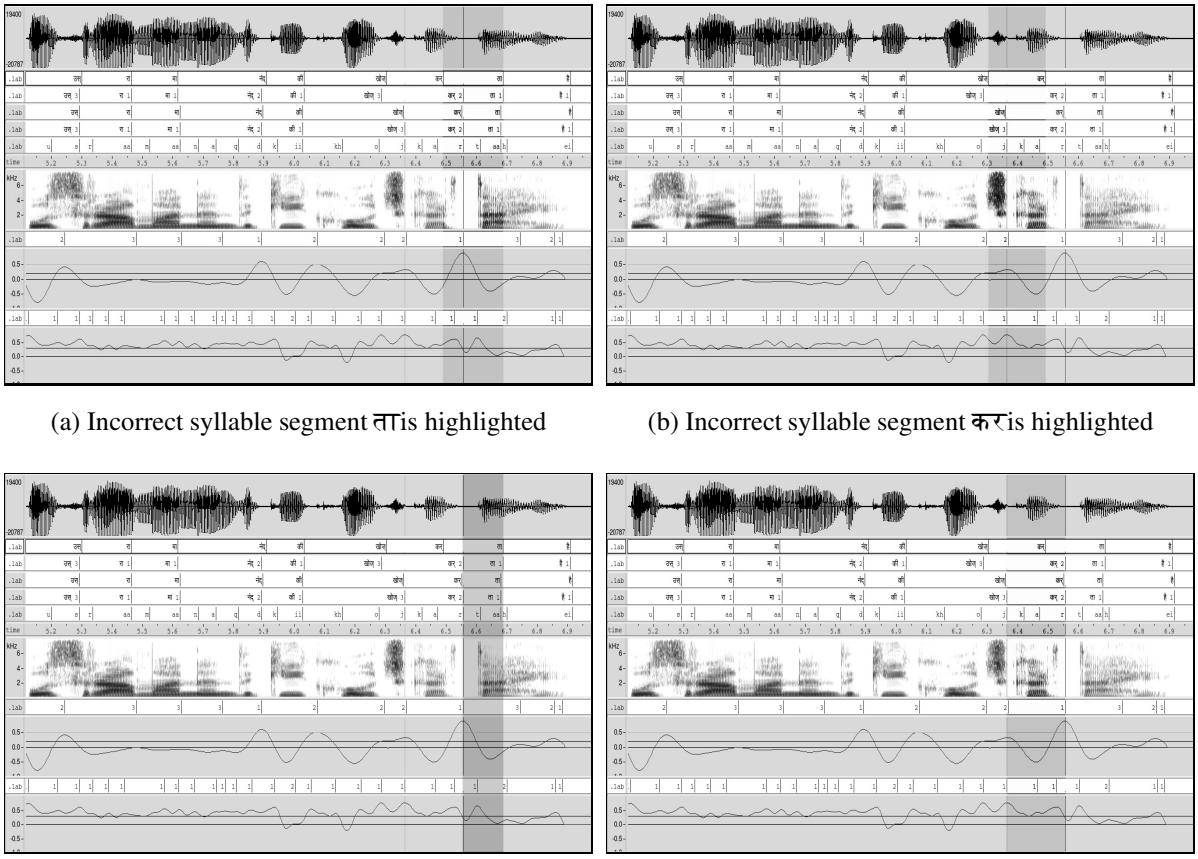


Fig. 5.5: Boundary refinement illustrated with an example

The location of two syllable boundaries are misplaced by a large margin in the output of the baseline HMM segmentation. The two boundaries are:

1. *Boundary 1* between syllables "कर('k' 'a' 'r')" and "ता('t' 'aa')". The incorrect syllable segment "ता" is highlighted in Figure 5.6(a).
2. *Boundary 2* between syllables "खोज('kh' 'o' 'j')" and "कर('k' 'a' 'r')". The incorrect syllable segment "कर" is highlighted in Figure 5.6(b).



(a) Incorrect syllable segment ताis highlighted

(b) Incorrect syllable segment करis highlighted

(c) The boundary between करand ताis corrected using STE. The corrected syllable segment ताis highlighted

(d) The boundary between खोजand करis corrected using SBSF. The corrected syllable segment करis highlighted

Fig. 5.6: Correction of incorrect boundaries in Figure 5.5, shown step-by-step

The second transcription panel shows the result after the first iteration of boundary correction. *Boundary 1* was corrected using STE as the succeeding syllable "ता('t' 'aa')" begins

with an unvoiced stop consonant. The corrected syllable segment "ता('t' 'aa')" is highlighted in Figure 5.6(c). *Boundary 2* was moved to an incorrect location using SBSF. Boundaries of SBSF correspond to phone transitions with significant spectral change. There was a boundary within syllable "खोज('kh' 'o' 'j')", before the realization of affricate 'j' and there was also a boundary at the end of the syllable (after the realization of affricate 'j'). As the initial boundary given by HMM was closer to the one within the syllable, it was moved to that location incorrectly.

The third transcription panel shows the output of the second iteration of HMM based forced alignment. *Boundary 2* has come closer to the actual boundary location. This is because of adding syllable positional context to phones and performing embedded re-estimation at the syllable level. The fourth transcription panel shows the final syllable label file, obtained after performing the second iteration of boundary correction. *Boundary 2* was moved to the correct location using SBSF. The corrected syllable segment "कर('k' 'a' 'r')" is highlighted in Figure 5.6(d).

In Dataset 3 (Hindi), '7213' syllable boundaries had b_p as an unvoiced stop consonant and out of that '5930' were corrected using STE. This justifies the use of threshold as 82.2% of possible boundaries were corrected with high confidence. Further, '172' of the uncorrected boundaries were corrected using SBSF because their e_p was a nasal. This adds another 2.4%. Dataset 3 also had '1129' syllable boundaries with only e_p as an unvoiced stop consonant and out of that '1010' were corrected using STE. Overall, 66.76% of the syllable boundaries were obtained using HMMs, 21.71% from STE and 11.53% from SBSF.

For Tamil, the phone transcription does not distinguish between voiced and unvoiced stop consonants. But, the algorithm intends to correct only boundaries with unvoiced stop consonants. In Dataset 1 (Tamil), '12684' syllable boundaries had b_p as a stop consonant and out of that '6254' were corrected using STE, '1191' were corrected using SBSF. The

boundaries corrected using STE mostly turn out to be the ones with unvoiced stop consonants as shown in Figure 5.7.

Panel A in Figure 5.7 shows the waveform. Panels B and C show the syllable and phone level segmentation obtained using HMMs. Panels D and E show the syllable and phone level segmentation obtained using hybrid segmentation. Panel F shows the spectrogram. Panel G shows the output of STE based group delay segmentation along with its boundaries. The boundaries labeled as '1' are above $threshold_1$. There are four boundaries with b_p as a stop consonant:

1. *Boundary 1* between syllables "னந்('n' 'a' 'nd')" and "தை('t' 'a')".
2. *Boundary 2* between syllables "மாய்('m' 'aa' 'y')" and "தூங்('t' 'uu' 'ng')".
3. *Boundary 3* between syllables "தூங்('t' 'uu' 'ng')" and "கி('k' 'i')".
4. *Boundary 4* between syllables "விட்('w' 'i' 'tx')" and "டாய்('tx' 'aa' 'y')".

The realization of the stop consonants only after *Boundary 1* and *Boundary 4* are unvoiced. These boundaries were corrected as the group delay peak crossed the threshold that was empirically set. For, *Boundary 2* and *Boundary 3*, there was no peak above the threshold in the vicinity and hence they were not corrected.

Figure 5.8 illustrates the importance of syllable level embedded re-estimation with an example. Panels B and C show the syllable and phone level segmentation obtained using HMMs. Panels D and E show the syllable and phone level segmentation obtained using hybrid segmentation. Panel F shows the spectrogram. Panel G shows the output of STE based group delay segmentation along with its boundaries. The location of two syllable boundaries are misplaced by a large margin in the output of HMM based segmentation. The two boundaries are:

1. *Boundary 1* between syllables "ன('n' 'a')" and "அக்('a' 'k')".

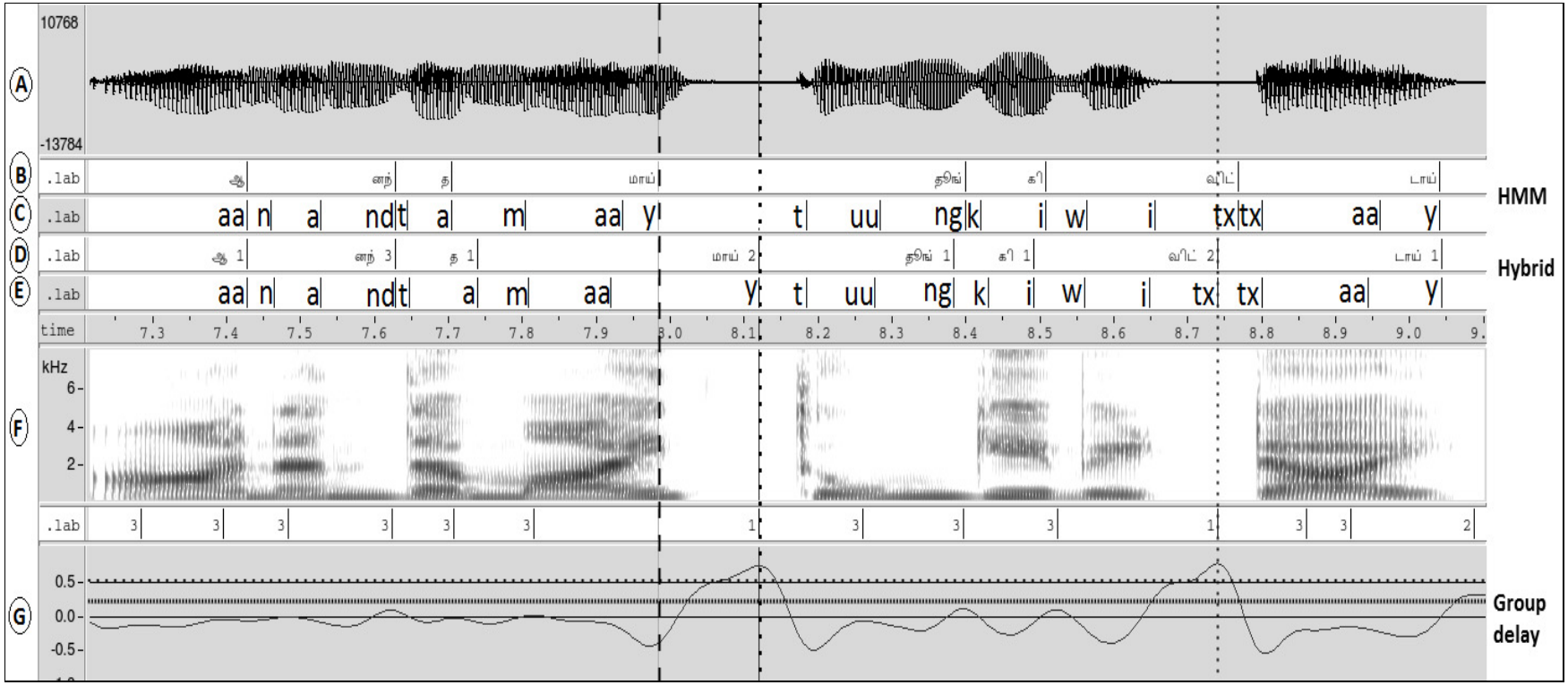


Fig. 5.7: STE based boundary refinement for unvoiced stop consonants in Tamil

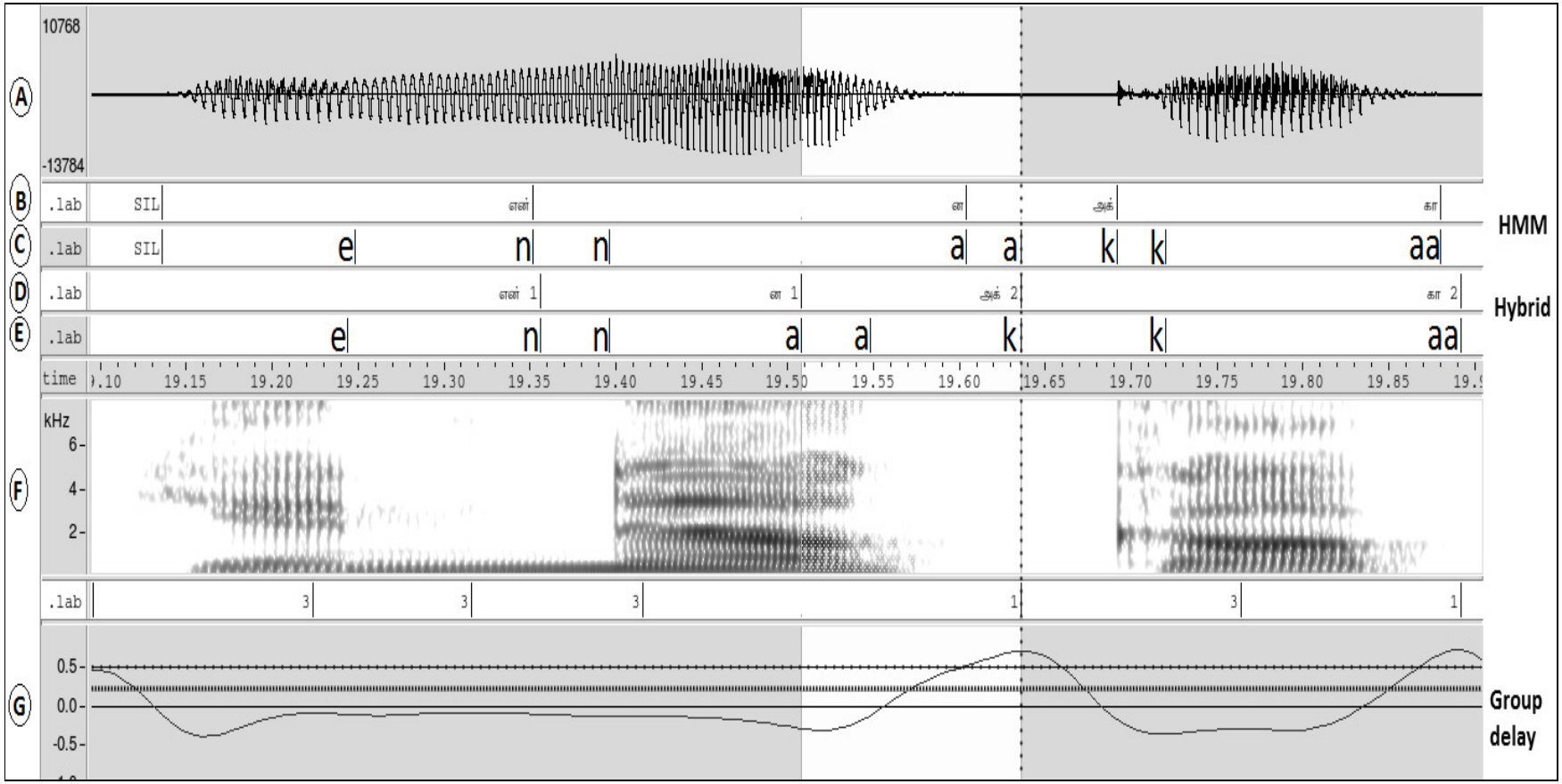


Fig. 5.8: Significance of syllable level re-estimation illustrated with an example

2. *Boundary 2* between syllables "அக்(‘a’ ‘k’)" and "கா(‘k’ ‘aa’)".

The syllable segment "ன(‘n’ ‘a’)" given by HMMs actually includes the region corresponding to both "ன(‘n’ ‘a’)" and "அக்(‘a’ ‘k’)". The syllable segment "அக்(‘a’ ‘k’)" is wrongly labeled in the closure region of the next segment. *Boundary 2* was corrected using STE and in subsequent forced alignment *boundary 1* got corrected because of syllable level embedded re-estimation.

5.5.3 Comparison with baseline segmentation

In the baseline HMM based segmentation, after flat start initialization ‘14’ iterations of embedded re-estimation was performed. Syllable positional context is added to phones (first and last phones in a syllable are treated separately) before syllable level re-estimation. The number of classes increases because of this. For instance, the number of context independent phones in dataset 3 (Hindi) is ‘57’ and it increases to ‘158’ after adding context. These syllable positional context phones are initialized with global mean, global variance and ‘7’ iterations of re-estimation is performed. After syllable boundary corrections are performed for the second time, another ‘7’ iterations of re-estimation is done. Totally, syllable level embedded re-estimation is performed for ‘14’ iterations like sentence level re-estimation. When the final phone models with the added syllabic position context of hybrid segmentation is used for sentence level forced alignment, the total acoustic likelihood increases compared to monophone models obtained from baseline HMM segmentation. This is true for all the three speakers used. Also, mostly even the average log probability per frame for various categories of phones like fricatives, vowels etc. increases as shown in Table 5.1, Table 5.2 and Table 5.3.

Table 5.1: Average log probability per frame for dataset 1 (Tamil, female speaker)

Method	Average log probability per frame						
	Nasals	Fricatives	Affricates	Semi-Vowels	Vowels	Stop Consonants	Overall
Hybrid	-69.67	-79.05	-79.98	-73.99	-67.23	-83.01	-72.99
HMM-FS	-70.86	-78.48	-80.40	-74.89	-68.01	-84.41	-73.93

Table 5.2: Average log probability per frame for dataset 2 (Hindi, male speaker)

Method	Average log probability per frame						
	Nasals	Fricatives	Affricates	Semi-Vowels	Vowels	Stop Consonants	Overall
Hybrid	-69.02	-77.74	-79.46	-74.16	-64.49	-82.05	-70.46
HMM-FS	-69.80	-78.24	-80.22	-74.60	-65.07	-82.82	-71.29

Table 5.3: Average log probability per frame for dataset 3 (Hindi, female speaker)

Method	Average log probability per frame						
	Nasals	Fricatives	Affricates	Semi-Vowels	Vowels	Stop Consonants	Overall
Hybrid	-75.06	-77.71	-80.70	-76.90	-69.31	-83.16	-74.85
HMM-FS	-74.91	-78.50	-81.00	-77.71	-69.82	-84.00	-74.99

5.5.4 Performance evaluation

With the segmentation given by the hybrid method and the baseline HMM based method, conventional phone based HTS systems [14] were built for dataset 1, dataset 2 and dataset 3. Additionally, STRAIGHT analysis-synthesis [46] method was used for dataset 2. Pair comparison (PC) tests [42] were performed. For these comparisons, the HTS system built

with the hybrid segmentation method is referred to as “A” and the HTS system built with baseline HMM based segmentation is referred to as “B”. The results of pair comparison tests show that there is an improvement in synthesis quality as shown in Table 5.4, Table 5.5 and Table 5.6.

Table 5.4: Pair comparison tests using phone based HTS for dataset 1 (Tamil, female speaker)

A-B	**B-A**	**A-B+B-A**
67.5	22.5	72.5

Table 5.5: Pair comparison tests using phone based HTS for dataset 2 (Hindi, male speaker)

A-B	**B-A**	**A-B+B-A**
72.2	22.2	75

Table 5.6: Pair comparison tests using phone based HTS for dataset 3 (Hindi, female speaker)

A-B	**B-A**	**A-B+B-A**
63	26	68.5

The syllable based HTS proposed in [15] used the semi-automatic labeling approach explained in Chapter 2. A syllable based HTS system for Tamil was built with dataset 1 and the proposed hybrid segmentation algorithm. It was compared to a similar system built

with semi-automatic segmentation. The results of pair comparison test show that hybrid segmentation gets 79.7% preference as shown in Table 5.7. The syllable based HTS system built with the hybrid segmentation method is referred to as "A" and the one built with semi-automatic segmentation [15] is referred to as "B".

Table 5.7: Pair comparison tests using syllable based HTS for dataset 1 (Tamil, female speaker)

A-B	**B-A**	**A-B+B-A**
78.2	18.8	79.7

With the segmentation given by the hybrid method and the baseline HMM based method, a syllable USS system [47] was built for dataset 1. Syllable based USS suffers from problems related to prosody. Correcting segmentation does not necessarily correct prosody. Hence, the improvement in USS was not as significant as HTS as shown in Table 5.8.

Table 5.8: Pair comparison tests using syllable based USS for dataset 1 (Tamil, female speaker)

A-B	**B-A**	**A-B+B-A**
66.7	42.9	61.9

Hybrid segmentation gives significant improvement in likelihood for stop consonants. This was reflected in the synthesized speech. Geminates (two stop consonants together) were always pronounced with more clarity when compared to HMM segmentation. Online phone based HTS synthesizers for Tamil, Hindi, Bengali, and Telugu built using the proposed segmentation algorithm can be found at "http://www.iitm.ac.in/donlab/hts/". Some

synthesized examples of Hindi and Tamil can be found at

"http://www.iitm.ac.in/donlab/hts/samples.php"

5.5.5 Experiments with other languages

The segmentation algorithm was used for segmenting three more languages:

1. Dataset 4 - 1631 utterances (3.5 hrs) of Telugu, spoken by a single native male speaker
2. Dataset 5 - 796 utterances (1.3 hrs) of Bengali, spoken by a single native male speaker
3. Dataset 6 - 1132 utterances (1.2 hrs) of Indian English, spoken by the same speaker as Dataset 1. The text used was same as that of ARCTIC databases [48].

The likelihood improvement similar to Hindi and Tamil were observed for Telugu, Bengali and Indian English as shown in Table 5.9, Table 5.10 and Table 5.11.

Table 5.9: Average log probability per frame for dataset 4 (Telugu, male speaker)

Method	**Average log probability per frame**						
	Nasals	**Fricatives**	**Affricates**	**Semi-Vowels**	**Vowels**	**Stops Consonants**	**Overall**
Hybrid	-76.00	-78.51	-80.78	-79.52	-76.07	-82.79	-78.00
HMM-FS	-76.23	-78.94	-80.31	-79.44	-76.13	-83.34	-78.13

Table 5.10: Average log probability per frame for dataset 5 (Bengali, male speaker)

Method	**Average log probability per frame**						
	Nasals	**Fricatives**	**Affricates**	**Semi-Vowels**	**Vowels**	**Stops Consonants**	**Overall**
Hybrid	-69.05	-76.93	-77.22	-78.97	-65.48	-80.21	-72.48
HMM-FS	-70.08	-77.21	-77.12	-79.43	-65.81	-80.46	-72.79

Table 5.11: Average log probability per frame for dataset 6 (Indian English, male speaker)

Method	Average log probability per frame						
	Nasals	**Fricatives**	**Affricates**	**Semi-Vowels**	**Vowels**	**Stops Consonants**	**Overall**
Hybrid	-67.92	-77.28	-78.47	-73.42	-60.06	-80.99	-69.64
HMM-FS	-68.54	-78.20	-78.94	-74.15	-61.10	-81.48	-70.16

For Indian English, the phone transcription was taken from CMU ARCTIC database [48] as it is. The NIST syllabification software [49, 50] was used to syllabify the phone sequence. Phonetic sequence corresponding to a phrase is given as input to the software to obtain the syllabic transcription. The syllabification obtained using this procedure was incorrect in a few places. The proposed hybrid segmentation algorithm is sensitive to syllabification rules. Hence, some boundaries were moved to incorrect locations for Indian English.

Figure 5.9 shows an example phrase from Dataset 6. The phrase shown is "A rifle shot beyond the ridge" and its corresponding phonetic sequence is /ah r ay f ax l sh ao t b ih y ao n d dh ah r ih jh/. The phonetic sequence was syllabified as "ah ray faxl shao tbih yaond dhah rihjh". Panel A in Figure 5.9 shows the waveform. Panels B and C show the syllable level segmentation given by the baseline HMM approach and hybrid approach respectively. Panel D shows the spectrogram. Panel E shows the smoothed inverted STE function and its peaks. Panel F shows smoothed SBSF and its peaks.

The syllable boundaries shown using dashed lines in Figure 5.9 were moved to more accurate locations using SBSF. The syllable "shao" was moved to an incorrect location (shown using dotted lines). This is because of incorrect syllabification. The actual syllabification in that place is "shaot bih". The stop consonant 't' was moved incorrectly to the next syllable during syllabification. This led to the improper alignment of that syllable.

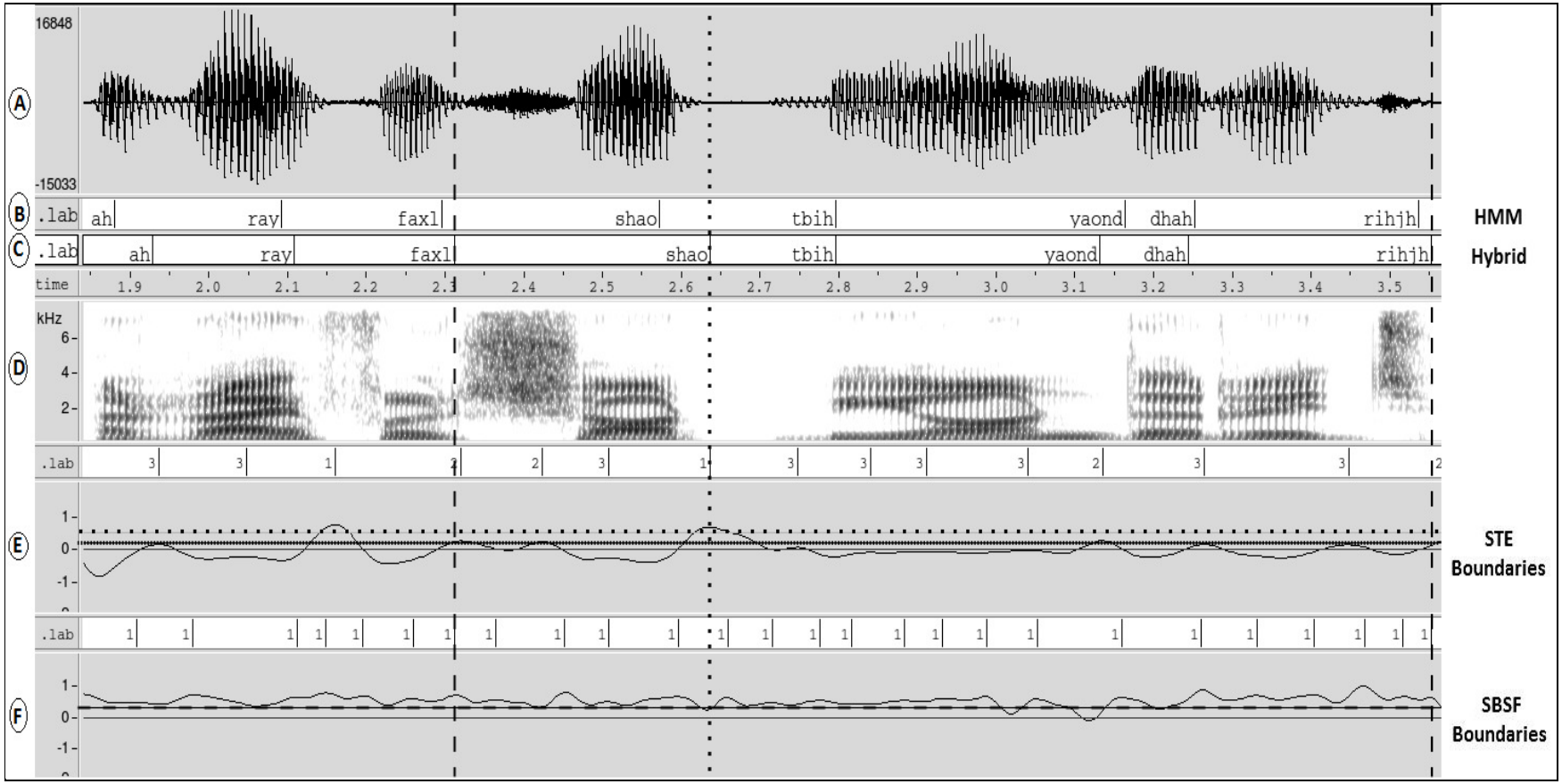


Fig. 5.9: Impact of incorrect syllabification on hybrid segmentation

5.6 Summary

In this chapter, the role of spectral change as a cue was discussed. The use of sub-band spectral flux (SBSF) to detect spectral changes was motivated. A boundary detection technique which uses root cepstral smoothing and SBSF was proposed. Next, improvements to hybrid segmentation algorithm were proposed. The new algorithm was experimented on Tamil and Hindi data. The results were compared to baseline HMM segmentation and the improvements were discussed in detail with a few examples. Pair comparison tests were performed and it shows that HTS systems built with hybrid segmentation gets more preference.

CHAPTER 6

Conclusions and Future Work

Speech synthesis requires accurate segmentation of data so that robust models can be built for the fundamental units. Although machine learning has been very successful for segmentation, it is not accurate enough to train high quality TTS systems. Machine learning techniques are robust on the average while signal processing techniques are accurate in the particular. In this work, an attempt has been made to synergize the benefits of knowledge-based domain specific signal processing and machine learning to obtain accurate phone level segmentation.

Criticism of the proposed work:

- Accurate syllabification rules are required to get the best out of the proposed segmentation algorithm. Boundary correction using incorrect syllabification can make the boundary location worse as explained in Section 5.5.5.

Future work:

Some of the possible extensions of this thesis work are described below:

1. One shortcoming of the proposed work is the lack of a mechanism to detect boundaries of all nasals. This can be explored.
2. The syllabification for Hindi and Tamil was based on a set of rules. Similar to performing boundary corrections based on proximity to signal processing cues, it can be

explored, if it can also be used to correct syllabification. For a new Indian language, syllabification rules of either Hindi or Tamil can be chosen based on closeness and a dictionary of word to syllable mappings can be created using these rules. Then the syllabification in the dictionary can be corrected by applying signal processing techniques on the available speech data of the language.

3. Ground truth labels were not available for Hindi and Tamil data used for experiments in this thesis. The segmentation algorithm was used for training HTS systems and it was evaluated by subjective tests on the synthesis quality. Objective approaches to either measure the segmentation accuracy directly or the quality of synthesis can be explored.

REFERENCES

[1] A. W. Black and J. Kominek, "Optimizing segment label boundaries for statistical speech synthesis," in *ICASSP*, pp. 3785 – 3788, IEEE, 2009.

[2] A. Sethy and S. S. Narayanan, "Refined speech segmentation for concatenative speech synthesis.," in *ICSLP*, pp. 149–152, ISCA, 2002.

[3] A. Hunt and A. Black, "Unit selection in a concatenative speech synthesis system using a large speech database," in *ICASSP*, pp. 373–376, IEEE, 1996.

[4] K. Tokuda and H. Zen, "Fundamentals and recent advances in HMM-based speech synthesis." `http://www.sp.nitech.ac.jp/~tokuda/tokuda_interspeech09_tutorial.pdf`.

[5] L. R. Rabiner, "A tutorial on hidden Markov models and selected applications in speech recognition," *Proceedings of the IEEE*, vol. 77, pp. 257–286, 1989.

[6] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, X. A. Liu, G. Moore, J. Odell, D. Ollason, D. Povey, V. Valtchev, and P. Woodland, "The HTK Book (for HTK Version 3.4)," *Cambridge University Engineering Department*, 2002.

[7] J. P. van Santen and R. W. Sproat, "High-accuracy automatic segmentation," in *Proc. of EUROSPEECH*, pp. 2809–2812, 1999.

[8] K.-S. Lee, "MLP-based phone boundary refining for a tts database," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, pp. 981–989, May 2006.

[9] H.-Y. Lo and H. min Wang, "Phonetic boundary refinement using support vector machine," in *ICASSP*, vol. 4, pp. 933–936, IEEE, April 2007.

[10] D. Toledano, "Neural network boundary refining for automatic speech segmentation," in *ICASSP*, vol. 6, pp. 3438–3441, IEEE, 2000.

[11] V. Kamakshi Prasad, T. Nagarajan, and H. A. Murthy, "Automatic segmentation of continuous speech using minimum phase group delay functions," *Speech Communication*, vol. 42, no. 3, pp. 429–446, 2004.

[12] D. H. Klatt, "Software for a cascade/parallel formant synthesizer," *the Journal of the Acoustical Society of America*, vol. 67, no. 3, pp. 971–995, 1980.

[13] E. Moulines and F. Charpentier, "Pitch-synchronous waveform processing techniques for text-to-speech synthesis using diphones," *Speech communication*, vol. 9, no. 5, pp. 453–467, 1990.

[14] "HMM-based speech synthesis system (HTS)." `http://hts.sp.nitech.ac.jp/`.

[15] A. Pradhan, S. Aswin Shanmugam, A. Prakash, V. Kamakoti, and H. A. Murthy, "A syllable based statistical text to speech system," in *EUSIPCO*, 2013. `http://www.eurasip.org/Proceedings/Eusipco/Eusipco2013/papers/1569738809.pdf`.

[16] K. Tokuda, T. Kobayashi, T. Masuko, and S. Imai, "Mel-generalized cepstral analysis – a unified approach to speech spectral estimation," in *Proc. ICSLP-94*, pp. 1043–1046, 1994.

[17] H. Zen, K. Tokuda, and A. Black, "Statistical parametric speech synthesis," *Speech Communication*, vol. 51, pp. 1039–1064, 2009.

[18] J. J. Odell, P. C. Woodland, and S. J. Young, "Tree-based state clustering for large vocabulary speech recognition," in *International Symposium on Speech, Image Processing and Neural Networks*, vol. 2, pp. 690–693, 1994.

[19] J. J. Odell, "*The Use of Context in Large Vocabulary Speech Recognition*." PhD dissertation, Cambridge University, 1996.

[20] K. Tokuda, T. Yoshimura, T. Masuko, and T. Kitamura, "Speech parameter generation algorithms for HMM-based speech synthesis," in *ICASSP*, pp. 1315–1318, IEEE, 2000.

[21] S. Imai, K. Sumita, and C. Furuichi, "Mel log spectrum approximation (MLSA) filter for speech synthesis," *Electronics and Communications in Japan (Part I: Communications)*, vol. 66, no. 2, pp. 10–18, 1983.

[22] B. Ramani, S. Lilly Christina, G. Anushiya Rachel, V. Sherlin Solomi, M. K. Nandwana, A. Prakash, S. Aswin Shanmugam, R. Krishnan, S. Kishore, K. Samudravijaya, P. Vijayalakshmi, T. Nagarajan, and H. A. Murthy, "A common attribute based unified HTS framework for speech synthesis in indian languages," in *SSW8*, pp. 291–296, 2013.

[23] V. M.V., A. Bellur, B. Narayan K., D. Thakare M., A. Susan, S. N.M., and H. A. Murthy, "Using polysyllabic units for text to speech synthesis in Indian languages," in *National Conference on Communication(NCC)*, 2010. `http://ieeexplore.ieee.org/&arnumber=5430193`.

[24] R. Krishnan, S. A. Shanmugam, A. Prakash, K. Sekaran, and H. A. Murthy, "IIT Madras's submission to the blizzard challenge 2014," in *Blizzard Challenge Workshop*, 2014. `http://www.festvox.org/blizzard/bc2014/IIT_Madras_Blizzard.pdf`.

[25] R. W. Sproat, ed., *Multilingual Text-to-Speech Synthesis*. Norwell, MA, USA: Kluwer Academic Publishers, 1997.

[26] L. Rabiner and B.-H. Juang, "*Fundamentals of speech recognition*," 1993.

[27] S. Young and P. Woodland, "HTK: Speech recognition toolkit." `http://htk.eng.cam.ac.uk/`.

[28] X. Huang, F. Alleva, H. Hon, K. Hwang, M. Lee, and R. Rosenfeld, "The SPHINX-II speech recognition system: an overview," *Computer Speech and Language*, vol. 7(2), pp. 137–148, 1992.

[29] J. Yuan, N. Ryant, M. Liberman, A. Stolcke, V. Mitra, and W. Wang, "Automatic phonetic segmentation using boundary models.," in *INTERSPEECH*, pp. 2306–2310, ISCA, 2013.

[30] K. N. Stevens, "Toward a model for lexical access based on acoustic landmarks and distinctive features," *The Journal of the Acoustical Society of America*, vol. 111, no. 4, pp. 1872–1891, 2002.

[31] Y. jun Kim and A. Conkie, "Automatic segmentation combining an HMM-based approach and spectral boundary correction," in *ICSLP*, pp. 145–148, 2002.

[32] L. R. Rabiner and R. W. Schafer, "*Theory and applications of digital speech processing*," 2010.

[33] O. Fujimura, "Syllable as a unit of speech recognition," in *ICASSP*, vol. 23, pp. 82–87, IEEE, February, 1975.

[34] S. Greenberg, "Speaking in shorthand- a syllable-centric perspective for understanding pronunciation variation," *Speech Communication*, vol. 29, pp. 159–176, 1999.

[35] J. Mehler, J. Y. Dommergues, U. Frauenfelder, and J. Segui, "The syllable's role in speech segmentation," *Journal of Verbal Learning and Verbal Behavior*, vol. 20, pp. 298–305, 1981.

[36] T. Nagarajan and H. A. Murthy, "Subband-based group delay segmentation of spontaneous speech into syllable-like units," *EURASIP Journal of Applied Signal Processing*, vol. 17, pp. 2614–2625, 2004.

[37] P. Deivapalan, M. Jha, R. Guttikonda, and H. A. Murthy, "DONLabel: An automatic labeling tool for indian languages," in *National Conference on Communication (NCC)*, pp. 263–266, February 2008.

[38] V. K. Prasad, *Segmentation and Recognition of Continuous Speech*. PhD dissertation, Department of Computer Science and Engg., Indian Institute of Technology Madras, Chennai, India, May 2002.

[39] H. A. Murthy and B. Yegnanarayana, "Group delay functions and its application to speech processing," *Sadhana*, vol. 36, pp. 745–782, November 2011.

[40] J. Lim, "Spectral root homomorphic deconvolution system," *IEEE Transactions on Acoustics, Speech and Signal Processing*, vol. 27, pp. 223–233, Jun 1979.

[41] T. Nagarajan, V. Kamakshi Prasad, and H. Murthy, "Minimum phase signal derived from root cepstrum," *Electronics Letters*, vol. 39, pp. 941–942, Jun 2003.

[42] P. Salza, E. Foti, L. Nebbia, and M. Oreglia, "MOS and pair comparison combined methods for quality evaluation of text to speech systems," *Acta Acustica*, vol. 82, pp. 650–656, 1996.

[43] A. Chopde, "ITRANS." `http://www.aczoom.com/itrans/`.

[44] H. Zen, K. Tokuda, T. Masuko, T. Kobayashi, and T. Kitamura, "Hidden semi-Markov model based speech synthesis," in *Proc. of ICSLP*, pp. 1393–1396, ISCA, 2004.

[45] S. Aswin Shanmugam and H. A. Murthy, "Group delay based phone segmentation for HTS," in *National Conference on Communications 2014 (NCC-2014)*, 2014. `http://ieeexplore.ieee.org/&arnumber=6811273`.

[46] H. Kawahara, I. Masuda-Katsuse, and A. De Cheveigne, "Restructuring speech representations using a pitch-adaptive time–frequency smoothing and an instantaneous-frequency-based f0 extraction: Possible role of a repetitive structure in sounds," *Speech communication*, vol. 27, no. 3, pp. 187–207, 1999.

[47] A. Black, P. Taylor, and R. Caley, "The festival speech synthesis system." `http://festvox.org/festival/`, 1998.

[48] J. Kominek and A. W. Black, "CMU ARCTIC databases for speech synthesis." `http://festvox.org/cmu_arctic/`.

[49] B. Fisher, "tsylb2-1.1 - syllabification software." `http://www.nist.gov/speech/tools`.

[50] D. Kahn, "*Syllable-based generalizations in English phonology*." PhD dissertation, Dept. of Foreign Literatures and Linguistics, Massachusetts Institute of Technology, 1976.

LIST OF PUBLICATIONS

1. S Aswin Shanmugam and Hema Murthy, **"A Hybrid Approach to Segmentation of Speech Using Group Delay Processing and HMM Based Embedded Reestimation,"** in Proc. of *Fifteenth Annual Conference of the International Speech Communication Association (INTERSPEECH 2014)*, pp. 1648–1652, Singapore, Sep' 14.

   http://www.isca-speech.org/archive/interspeech_2014/i14_1648.html

2. S. Aswin Shanmugam and Hema A. Murthy, **"Group Delay Based Phone Segmentation for HTS,"** in Proc. of *Twentieth National Conference on Communications (NCC 2014)*, Kanpur, India, Feb' 14.

   http://ieeexplore.ieee.org/&arnumber=6811273

3. A. Pradhan, A. Prakash, S. Aswin Shanmugam, G. R. Kasthuri, R. Krishnan and H. A. Murthy, **"Building speech synthesis systems for Indian languages,"** in Proc. of *Twenty First National Conference on Communications (NCC 2015)*, Mumbai, India, Feb' 15.

   http://ieeexplore.ieee.org/&arnumber=7084931

4. Raghava Krishnan, S Aswin Shanmugam, Anusha Prakash, Kasthuri Sekaran and Hema A Murthy, **"IIT Madras's Submission to the Blizzard Challenge 2014,"** *Blizzard Challenge 2014*, Singapore, Sep' 14.

   http://www.festvox.org/blizzard/bc2014/IIT_Madras_Blizzard.pdf

5. Abhijit Pradhan, Aswin Shanmugam S, Anusha Prakash, Kamakoti Veezhinathan and Hema Murthy, **"A Syllable Based Statistical Text to Speech System,"** in Proc. of *21st European Signal Processing Conference (EUSIPCO 2013)*, Marrakech, Morocco, Sep' 13.

   http://www.eurasip.org/Proceedings/Eusipco/Eusipco2013/papers/1569738809.pdf

6. B Ramani, SL Christina, GA Rachel, VS Solomi, MK Nandwana, A Prakash, Aswin Shanmugam S, R Krishnan, SP Kishore, K Samudravijaya, P Vijayalakshmi, T Nagarajan and Hema A Murthy, **"A Common Attribute based Unified HTS framework for Speech Synthesis in Indian Languages,"** in Proc. of *8th ISCA Workshop on Speech Synthesis (SSW8)*, pp. 291–296, Barcelona, Spain, Aug' 13.
http://ssw8.talp.cat/download/ssw8_proceedings.pdf

GENERAL TEST COMMITTEE

Chair	Dr. Deepak Khemani Department of Computer Science and Engineering
Research Advisor	Dr. Hema A. Murthy Department of Computer Science and Engineering
Other Members	Dr. Anurag Mittal Department of Computer Science and Engineering
	Dr. S. Umesh Department of Electrical Engineering

CURRICULUM VITAE

Name	Aswin Shanmugam S
Date of birth	19 March, 1991
Permanent address	15A/35 Srinivasa Nagar Main Road, Chitlapakkam Chennai - 600064, Tamil Nadu, INDIA.
E-Mail	sas91@outlook.com

Education

M.S.	Indian Institute of Technology Madras, Chennai (2012 - present)
B.Tech.	SSN College of Engineering, Anna University (2008 - 2012)