Unsupervised Speech Signal to Symbol Transformation for Zero Resource Speech Processing

Shekhar Nayak

A Thesis Submitted to

Indian Institute of Technology Hyderabad

In Partial Fulfillment of the Requirements for

The Degree of Doctor of Philosophy

Department of Electrical Engineering

May 2019

Declaration

I declare that this written submission represents my ideas in my own words, and where ideas or words of others have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be a cause for disciplinary action by the Institute and can also evoke penal action from the sources that have thus not been properly cited, or from whom proper permission has not been taken when needed.

_____

(Signature)

Shekhar Nayak

_____

(Name)

EE13P1008

_____

(Roll No.)

Approval Sheet

This Thesis entitled **Unsupervised Speech Signal to Symbol Transformation for Zero Resource Applications** by **Shekhar Nayak** is approved for the degree of Doctor of Philosophy from IIT Hyderabad.

(Dr. K. Sri Rama Murty) Adviser
Dept. of EE, IITH

Acknowledgements

This thesis work has been possible due to the untiring efforts and undiminishing interest of my guide, Dr. K. Sri Rama Murty. I am extremely grateful to him for accepting me as his student. Long discussions, iterative reviews of drafts and his constant pursuit for excellence has created deep impact on me. I am thankful to him for the excellent resources including high performance computing facilities at Speech Information Processing Laboratory (SIP Lab) without which this work would not be possible. The environment fostered by him in the lab is extremely collaborative and conducive for research. He opened several avenues to enhance my learning including internship, volunteering at top conferences, attending workshops. I would be always indebted to him for all the personal care taken by him throughout my tenure.

I would like to thank my doctoral committee members, Dr. Sumohana Channappayya, Dr. C. Krishna Mohan and Dr. Amit Acharya for their constructive feedback on this work. I would also like to thank Dr. Sunil Sivadas for providing me internship opportunity at Institute for Infocomm Research (I2R), Singapore which was great learning experience for me in the earlier stage of PhD. I am thankful to IIT Hyderabad authorities for providing all required facilities at lab and hostel which made it possible to conduct this research. I am grateful to the Ministry of Human Resource Development, Government of India for funding me throughout the PhD tenure.

I am thankful to all the past and present members of SIP lab including Senthil Kumar Mani, Jitendra Kumar Dhiman, Kallola Rout, Raghavendra Reddy, Karthika Vijayan, Shaik Mohammad Rafi, Swati Jindal, K. Bramhendra, A. Sivaganesh, R. Gowri Prasad, C. Shiva Kumar and V. Venkatesh for being there with me for all kind of support. The association with each one of them has enriched me personally and professionally. Special thanks to Saurabhchand Bhati for contributing immensely to this thesis by being involved in most of the works in different capacities. I would like to thank the interns Shashank Dhar, G. Ramesh and S. Sreekanth who were very dynamic and helpful.

This thesis is incomplete without the mention of my dear friends Nagabhushan, Shri Ram Vaishya, S. Veeramani and Rishabh Verma who have been with me from the beginning of the PhD journey. Their friendship and support has helped me to comfortably complete this journey. I would also like to thank my friends Sameeulla, Amarlingam, Bharath, Nagendra, Venkat Reddy, Narasimha for their constant support at different stages in the PhD. I am thankful to my labmates Sameeulla, Nagabhushan, Yoghitha, Sathya, Parimala, Appina Balasubramanyam for creating a wonderful environment in lab which felt like home instead of workplace.

I am greatly indebted to my professors Prof. S.D. Joshi and Prof. Arun Kumar from IIT Delhi who instilled the passion for research in me through their wonderful courses and insightful discussions. I am always thankful to my teachers Dr. Piyush Lotia and Mr. Kishore Kashyap who have constantly motivated me to pursue research.

I am unable to express in words the tremendous love and support of my parents Mr. Vibhuti Ranjan Nayak and Mrs. Archana Nayak, my aunt Ms. Rajani Nayak and dear brother Mr. Akash Nayak. Finally, I would like to thank everyone who have directly or indirectly contributed to this work.

Shekhar Nayak

Dedication

To My Family, Teachers, and Friends

Abstract

Zero resource speech processing refers to techniques which do not require manually transcribed speech data. The inspiration for zero resource is drawn from language acquisition in infants which is completely self-driven. Infants learn different abstraction levels i.e. phones, words and some syntactic aspects of the language they are exposed to, without any supervision or feedback. This motivated the research in speech community towards the development of completely unsupervised speech algorithms which can discover subword/word units from speech signal alone. The applications include spoken term discovery, language identification, keyword spotting etc. Zero resource techniques can be effective in solving problems associated with the development of speech systems for low resource languages.

Low resource languages have low amount of transcribed data and/or low number of native speakers. Several languages of the world have become endangered languages with almost negligible resources. The lack of transcribed data for low resource languages has inspired many directions to address this problem such as data augmentation, cross-lingual and multilingual techniques with limited success. In this thesis, we explore better feature representations for low resource speech recognition and later build unsupervised algorithms for zero resource speech processing which could lead to directions to effective solutions to the low resource problem.

Traditional speech recognition systems employed magnitude based features for building acoustic models. Phase of the speech signals is generally ignored as human ear was considered traditionally to be indifferent to phase. Recent perceptual studies have shown the importance of phase in human speech recognition. Motivated by this fact and in order to leverage the maximum information from limited transcribed data available in low resource settings, we propose to extract features from the analytic phase of speech signals for speech recognition. In order to avoid phase wrapping problem, instantaneous frequency is extracted from the speech signal without explicit phase computation. Different instantaneous frequency estimation methods are studied for providing effective features for speech recognition. Magnitude and phase based features are used to train separate phone recognition systems. Combining magnitude and phase based systems improves speech recognition in low resource settings and noisy conditions.

Inspired by the recent zero resource phenomenon in speech community, the problem of scarcity of transcribed data is addressed at more fundamental level by producing artificial or virtual transcriptions only from speech signals. Motivated from infant learning, zero resource speech processing aims at discovering acoustic word units from

speech signal alone without using any manual transcriptions or linguistic knowledge. We propose an unsupervised speech signal to symbol transformation approach to get virtual phones/labels from given speech signals. Syllable-like units obtained from multiple evidences for vowel endpoint detection from speech signals are presented as alternate units to virtual phones for signal to symbol transformation.

Several speech applications are presented which employ these virtual phones or syllable-like units for automatically transcribing the speech data in zero resource settings. Spoken term discovery and speaking rate estimation are achieved in zero resource settings using the proposed methods. A completely unsupervised language identification approach is proposed and is shown to perform close to the supervised approach. Further, a virtual phone recognition/synthesis approach based on signal to symbol transformation is proposed for ultra low bitrate coding. Future directions are provided to improve the low resource speech processing by employing automatic labeling to obtain performance closer to the supervised techniques.

Keywords: Acoustic segment modeling, Low resource, Speech recognition, Speech segmentation, Virtual phones, Zero resource.

Contents

List of Figures

List of Tables

List of Abbreviations

MFCC	Mel-frequency cepstral coefficients
IF	Instantaneous frequency
IFCC	Instantaneous frequency cosine coefficients
ASR	Automatic speech recognition
PER	Phone error rate
WER	Word error rate
DNN	Deep neural network
HMM	Hidden Markov model
GMM	Gaussian mixture model
ASM	Acoustic segment modeling
LMS	Least mean squares
TVAR	Time-varying auto-regressive
IF-ZC	IF estimation using zero-crossing method
IF-LMS	IF estimation using LMS algorithm
IF-TVAR	IF estimation using TVAR modeling
IF-FT	IF estimation using Fourier transforms
CD-US	Critically damped system with unit sample
UD-RN	Under damped system with random noise
UD-TI	Under damped system with train of impulses
AM	Amplitude modulated
FM	Frequency modulated
LID	Language identification
ES-KMeans	Embedded segmental K-means
PES-KMeans	Phoneme based embedded segmental K-means
DTW	Dynamic time warping
SRE	Speaking rate estimation

Chapter 1

Introduction

Zero resource refers to unsupervised techniques that rely only on the speech signal without the need for manual transcriptions or specific linguistic knowledge from human experts [1]. For under-resourced languages, there is a dearth of linguistic experts to produce manual labels for speech processing tasks. Therefore, zero resource speech processing has emerged as an active area of research in the speech community.

Studies on language learning in infants suggest that they have an inherent ability to discriminate between the phones of different languages and self-learn the phonetic structures in a language to which they are exposed to in the first year of their life [2]. The phonetic level acquisition is followed by learning of words, their meanings and subsequently various linguistic aspects of that language, i.e., lexical, syntactic, semantic, etc. This learning in infants happens without being taught or supervised in any way, even before they begin to speak [3]. This is a strong indication that it is possible to learn the linguistic structure completely in an unsupervised way directly from the speech signal itself. Such unsupervised techniques could improve the performance of low resource speech recognition.

1.1 Low resource speech recognition

Low resource languages lack in manually labeled speech data required for speech processing tasks. These languages include endangered languages with very low number of native speakers and the languages with large number of speakers but low amount of transcribed speech data. Menominee is a native American, tribal language spoken in Wisconsin, US and has less than ten speakers alive at present. Ethnologue[1] statistics state that out of total 7,111 languages of the world, 1,635 are spoken by less than

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

1,000 native speakers with 314 of them extinct.

There are approximately 1500 Indian languages out of which 30 languages are spoken by at least one million speakers but most of them are low resource in terms of availability of transcribed data [4]. Even if the speakers are available, the manual speech annotation task is laborious, expensive and time consuming. To transcribe one minute of speech, it requires approximately 30 minutes to two hours for a linguist depending on the skill level and the level of difficulty of the concerned data [5]. Automatic speech recognition (ASR) systems commonly use deep neural networks (DNN) based acoustic models [6] which require significantly higher amount of labeled data for training. Therefore, to develop techniques that are language independent and generalizable to any amount of training data is an important research problem for low resource ASR.

Effect of training data on DNN based ASR for Librispeech corpus is shown in Fig 1.1a. It is clearly evident that higher amount of training data leads to lower word error rates (WER), specially in case of difficult test data which includes challenging speakers who deliver higher WERs (test-other). The WER improvements with increase in training data are not significant for clean speech data (test-clean).

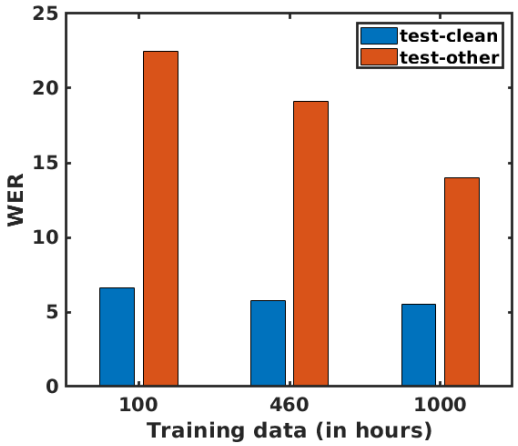
This lack of labeled data for low resource databases/languages tends to dominate the performance of ASR even with the state-of-the-art deep architectures or acoustic models. This is demonstrated in Fig 1.1b where different deep architectures are used for TIMIT with approximately 3 hours of training data and tested on TIMIT core test set. The architectures depicted in the figure are DNN, time-delay neural network (TDNN), bi-directional long short-term memory (BLSTM) and convolutional neural network (CNN). The results[2] demonstrate very close phone error rates (PER) for different architectures. Any significant improvement can not be observed from these acoustic models for TIMIT which is a low resource database.

Different strategies have been employed for low resource speech recognition to address the problem of labeled training data scarcity. Some of the important and popular approaches are discussed below -

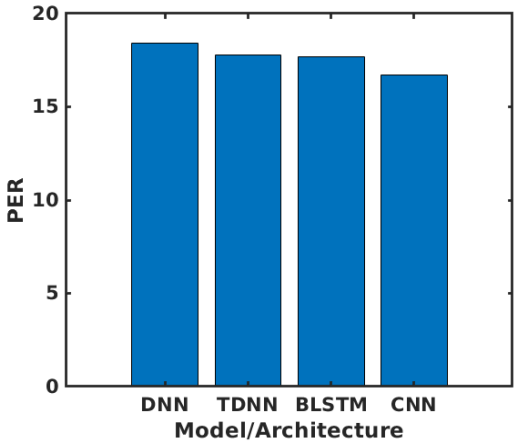
1.1.1 Data augmentation

Data augmentation techniques such as vocal tract length perturbation [10], speech rate distortion and frequency-axis random distortion [11] have been used to artificially generate additional training data to effectively train acoustic models for low resource speech recognition. Vocal tract length perturbation is applied on low resource training

[2]The results are compiled from different sources [6–9]



(a) Effect of training data on DNN based ASR for Librispeech corpus



(b) Performance of different architectures for TIMIT phone recogntion task

data to change the length of speakers' vocal tract by varying factors for generating artificial training utterances. Speech rate distortion involves altering the speech rate for data augmentation. Frequency-axis random distortion involves calculation of a distortion factor in small time-frequency region to realize a random distortion.

1.1.2 Multilingual speech recognition

Multilingual training of acoustic models [12] is another solution towards handling scarcity of training data. Shared architectures with common hidden layers for training across languages lead to small relative improvements in recognition accuracy [13, 14]. This approach heavily depends upon availability of labeled data from other languages and the improvements could be dependent on the closeness of the languages which are used for joint training. Cross-lingual recognition systems are shown to perform worser than the monolingual systems and multilingual systems [15].

1.1.3 Feature engineering

Generally, different magnitude based features like mel-frequency cepstral coefficients (MFCC), perceptual linear prediction (PLP) etc. and even the raw speech signal have been used for training DNNs and other architectures for speech recognition [16]. Different architectures, cross-lingual and multi-lingual systems, data augmentation strategies for low resource speech recognition have been extensively explored in the literature. This motivated us to discover better feature representations which could improve the performance of existing DNN based speech recognition. For this purpose, we investigate several instantaneous frequency (IF) based feature representa-

tions derived from analytic phase of speech signals. Generally, the phase information is ignored in ASR as the common features discussed above are derived solely from the magnitude spectrum. The combination of magnitude and phase based features was explored for improving the performance in low resource and noise robust speech recognition.

Despite the above strategies for low resource and with deep architectures, the performance of speech recognition for low resource languages is not at par with high resource languages. This motivates us to look into zero resource speech processing which could help us to create virtual transcriptions from speech signals to fill the scarcity of manually transcribed speech data.

1.2 Zero resource speech processing

The fundamental problem of lack of labeled data could be addressed more effectively, if labels could be generated artificially in an automatically just from given unlabeled speech waveforms. We develop a technique to produce virtual labels from given speech signals alone. This technique is termed as unsupervised speech signal to symbol transformation. We investigate the performance of these virtual labels on applications such as spoken term discovery, language identification and speaking rate estimation. An insight into future directions is provided as to how such virtual labels could be used for effectively solving the problem of low resource ASR.

We propose to automatically transcribe the unlabeled speech data for training systems efficiently in zero resource conditions. A three-step acoustic segment modeling (ASM) approach is proposed to label the data in an unsupervised manner. In the first step, we segment the speech signal into phone-like units, resulting in a large number of varying length segments. The second step involves clustering the varying-length segments into a finite number of clusters so that each segment can be labeled with a cluster index. The unsupervised transcriptions, thus obtained, can be considered as a sequence of virtual phone labels. In the third step, a DNN classifier is trained to map the feature vectors extracted from the signal to its corresponding virtual phone label. The transformation of the acoustic features from signal to virtual phone sequences is referred to as signal to symbol transformation. This transformation could be into phone-like units or syllable-like units depending on the segmentation level. The choice of the units is important and is application-specific. Different applications are discussed using these phone-like units or syllable-like units.

1.3 Organization of the thesis

The broad flow of ideas in this thesis is as follows. Improved feature representations for low resource speech recognition derived from instantaneous frequency of speech signals are proposed. Different IF estimation methods are investigated for deriving the acoustic features. A signal to symbol transformation approach is proposed towards transcription of speech signals in zero resource manner. A novel kernel-Gram segmentation and a segment labeling method are proposed for signal to symbol transformation. The virtual phones obtained using this approach are evaluated for spoken term discovery. Further, segmentation speech signals into larger units i.e., syllable-like units is discussed and spoken term discovery evaluations are done using these units. Finally, zero resource applications - language identification, speaking rate estimation and low bitrate coding are discussed. The thesis is organized as follows:

Chapter 2 presents a review on importance of phase in low resource speech recognition and zero resource speech processing. A brief review of existing literature on the applications to the proposed methods is also presented in this chapter.

Chapter 3 presents instantaneous frequency features for low resource ASR. Different IF estimation methods are discussed and are evaluated on speech-like synthetic signals. Instantaneous frequency cosine coefficients (IFCC) features extracted from analytic phase of speech signals are proposed for speech recognition. System combination of magnitude based MFCC and phase based IFCC features are shown to outperform these features individually for low resource and noise robust speech recognition.

Chapter 4 describes unsupervised speech signal to symbol transformation for zero resource speech processing. The steps involved in this approach are discussed namely, proposed kernel-Gram segmentation, a novel segment labeling approach based on graph clustering and graph growing; and unsupervised acoustic modeling. This approach labels given speech signals in terms of consistent virtual phones. Spoken term discovery using virtual phones is discussed as application.

Chapter 5 introduces syllable-like units as alternate to virtual phones as potential base units for signal to symbol transformation. Multiple evidences from source features, spectral features and Bessel features are combined to detect the vowel end points from speech signals. These vowel end points are used as anchor points to detect the boundaries of syllable-like units. The choice of syllable-like units for zero resource applications is described. A theta-rate oscillator based approach is discussed for estimating boundaries of syllable-like units. Further, spoken term discovery using syllable-like units is described.

Chapter 6 describes zero resource approach towards speaking rate estimation.

The syllable-like units obtained from multiple evidences based approach and theta oscillator based approach are used in unsupervised settings for estimating the speaking rate on TIMIT and Switchboard database.

An unsupervised phonotactic approach to language identification is proposed using these virtual phones. This approach termed as unsupervised parallel phone recognition is evaluated for language identification on eight Indian languages and compared with other relevant unsupervised and supervised approaches.

Virtual phone recognition/synthesis for ultra low bit rate coding is described as further application. Virtual phone recognition is discussed using signal to symbol transformation to convert speech signals to sequence of virtual phones. These virtual phones are encoded at very low bitrates. The speech signal is re-synthesized in target speaker's voice using neural network based synthesis approach.

Chapter 7 concludes the thesis and major contributions are highlighted. The possible future directions emanating from this work are discussed.

Chapter 2

Overview of low/zero resource speech processing

In recent times, zero resource speech processing has emerged as an active area of research in the speech community. Zero resource refers to unsupervised techniques that rely only on the speech signal without the need for manual transcriptions or specific linguistic knowledge from human experts [17]. The lack of transcribed data for low resource languages highlights the need for development of zero resource techniques which can be generalized to any language/speakers.

To demonstrate the issues associated with low resource languages, we consider the low resource speech recognition problem. Most common features for ASR are magnitude based and ignore the phase information. In case of low resource languages, it is important to make effective use of limited resources. Therefore, we explore the analytic phase of speech signals for low resource ASR.

2.1 Significance of analytic phase in low resource speech recognition

State-of-the-art automatic speech recognition systems are based on the features extracted from either the magnitude spectrum of the speech signal in the Fourier transform domain [6] or amplitude envelope in the analytic signal domain [18]. In either case, the phase components are usually ignored during the feature extraction process, as the human auditory system is believed to be phase-deaf [19]. However, in the recent past, quite a few perceptual studies have highlighted the importance of phase in human speech perception [20], [21]. Zeng *et al.*, demonstrated that the frequency modulated component of the speech signal significantly enhances the human speech

recognition in noise, as well as speaker and tone recognition [22] when used along with the amplitude modulated component. It has also been shown that uncertainty in phase results in higher word error rates [23] or lower intelligibility [24] for human listeners. Use of dynamic frequency modulated (FM) systems improved speech recognition in noise for cochlear implant recipients [20]. These works clearly demonstrate the indispensable nature of phase information in human perception.

A prominent reason for less exploration of analytic phase in speech recognition is the phase wrapping problem. It is difficult to discriminate between phase values differing by integer multiples of 2π. Therefore, it is computationally expensive to extract features from the phase component as compared to the magnitude component. Since, there has been no dearth of computational resources in recent times, this has lead to resurgence of phase based methods in speech community [25], [26].

Computation of analytic phase for a narrowband signal suffers from the phase wrapping problem, i.e., the phase values differing by integer multiples of 2π are indistinguishable. Feature extraction from the phase component requires higher computational complexity compared to its magnitude counterpart due to this phase wrapping problem. There is a renewed interest in the analysis of phase spectrum of speech signals [25], [26] due to ever increasing computational resources.

The problem of phase unwrapping can be avoided by computing its time-derivative, referred to as instantaneous frequency (IF) [27]. Another approach to avoid this problem is by computing the negative of frequency domain derivative of phase, referred to as group delay. Short-time phase spectral features derived from the group-delay spectrum were shown to improve the performance of speech recognition system [28, 29]. The issues associated with these features are that the group delay function is spiky in nature for speech signals and also the group delay features are shown to be sensitive to additive noise [30]. Despite the use of phase based features in these works, recent state-of-the-art speech recognition systems are only built using magnitude based features. Therefore, we explore different methods for IF estimation which could lead us to effective feature representation for ASR.

A detailed review on IF estimation techniques can be found in [31]. Several previous works have incorporated instantaneous frequency based features along with magnitude based features to improve ASR performance. Mean instantaneous amplitude and mean IF estimated using Energy Separation Algorithm along with the frequency modulation percentages in speech resonances were used as features for speech recognition [32]. Short time amplitude weighted IF and bandwidth based features were used as ASR front-end [33]. Features derived from envelope of the sub-band filter outputs of Gammatone filterbank were used for improving recognition rates in clean

and noisy conditions over MFCC features on a Chinese Mandarin digits corpus [34].

These works indicate the importance of instantaneous frequency derived from analytic phase of speech signals for ASR. In case of low resource, it is important to not discard the information from phase completely and effectively utilize it along with magnitude based features for improving the recognition accuracy. As the phase is important for speech perception in noisy conditions, we discuss the related works in the context of noise robust speech recognition.

There have been several previous approaches to incorporate different IF based features along with magnitude features to enhance ASR performance in noisy conditions. The output of an array of band-pass filters was decomposed into analytic and anti-analytic components to derive average IF and average log envelope features for clean training to recognize noisy speech [35]. Temporal AM and FM features were used for speech recognition in noisy environments [36]. Sub-band instantaneous frequency features with wavelet sub-band features were used for phoneme recognition tasks on TIMIT database under noisy conditions [37].

This provides a brief introduction to significance of analytic phase and IF based features for low resource ASR. We investigate several IF estimation methods for extracting effective phase based feature representations. This leads to IF cosine coefficients (IFCC) features which are evaluated for phoneme recognition on TIMIT database.

In order to minimize the requirement of training data for low resource and to achieve generalization across languages, we review zero resource speech processing methods in the next section.

2.2 Zero resource speech processing

To produce virtual labels for zero resource speech processing, acoustic segment modeling techniques are proposed in the literature.

2.2.1 Acoustic segment modeling

Acoustic segment modeling is a data-driven approach aimed at unsupervised discovery and modeling of acoustically similar subword units from the speech signal. A typical ASM system consists of three major steps, viz., speech segmentation, segment labeling and segment modeling [38]. In the segmentation step, the speech signal is divided into acoustically homogeneous segments of varying-length. In the segment labeling step, the varying-length segments are grouped into clusters using a suitable similarity

measure and each cluster index can be viewed as a virtual phone label. In the segment modeling step, unsupervised acoustic models are trained using all the acoustic segments sharing the same label. The acoustic models, thus obtained, can be used to decode any new utterance into a sequence of labels. The unsupervised ASM can be used in several applications including, but not limited to, language identification (LID) [38], query-by-example spoken term detection [39], speech summarization [40], topic identification [41, 42], etc.

ASM does not require supervision from manual transcriptions and hence it suffers from several critical issues at each of the three steps in its implementation. Moreover, these three steps are tightly coupled, and errors in one step severely affect the performance of the subsequent steps. For example errors in the speech segmentation step leads to poor clustering in the segment labeling step, which in turn affects the segment modeling. Hence, unsupervised segmentation of speech signal into phoneme-like units forms the crucial first step in the ASM framework.

Segmentation into phone-like units

Even though several methods have been proposed, over the past five decades, for speech segmentation [43–46], it is still an active area of research among the speech community due to the scope in improving the performance. An empirical study found that maximum spectral transition positions and phone boundaries are correlated significantly [47]. Estevan et al. used maximum margin clustering over a sliding window of frames to hypothesize the segment boundaries [48]. In this method, the frames in each sliding window are grouped into two clusters and the Euclidean distance between the cluster indices of successive sliding windows is used to hypothesize the boundaries. The length of the sliding window plays a critical role in the performance of this method - shorter window leads to false alarms while longer window leads to missed detection. Qiao et al. proposed a bottom-up agglomerative algorithm for unsupervised optimal phoneme segmentation [45]. In this approach, every frame in the speech signal is initially hypothesized as a potential boundary, i.e., they form single frame segments. An iterative algorithm is then employed to merge successive segments, based on an objective criterion until a predetermined number of segments are obtained. This approach requires the knowledge of the expected number of segments in the speech signal for termination and hence cannot be employed in a strictly unsupervised scenario. Linear discriminant analysis was used for extracting robust features from the input data. Dynamic programming was then used for deriving optimal segment boundaries [49]. A time-domain approach based on average level crossing rate of the speech waveform was proposed for segmentation [50]. The sam-

ples of the speech signal are normalized to the range $[-1, 1]$, and are assigned to predefined levels in this range. The valleys in the average level crossing rate, across all the levels, are hypothesized as the segment boundaries. The optimal levels are decided using manual transcriptions of the TIMIT database [51]. A comprehensive analysis of the hypothesized phone boundaries in unsupervised speech segmentation can be found in [52]. Most of the approaches require information about the number of segments in a given utterance. However, in an unsupervised setting, where there is no information available about the data, it is difficult to get this information apriori.

As segmentation is the first building step in ASM, it should have reliable performance. It should have a higher recall rate (the number of correctly detected phoneme boundaries) and a low false alarm (the number of boundaries that are detected by the algorithm but are not present) [47]. In unsupervised setting, no information about the dataset is used, so the algorithm should automatically detect the number of segments. A better segmentation algorithm can improve the ASM performance significantly, keeping the subsequent steps fixed. We propose a novel kernel-Gram based segmentation approach for speech segmentation.

Segmentation into syllable-like units

Early research showed that syllable is the fundamental unit for the perception of speech in infants [53]. Syllables have also been proved to be effective as a basis for the segmentation of speech signals [54]. Syllable-like units have been used as fundamental units for applications such as speech recognition [55], speaker verification [56], language identification [57] etc.

An oscillator based approach (SylSeg) for segmentation into syllable-like units was proposed based on theta-rate oscillations matching with the syllabic rate [58]. Embedded segmental k-means (ES-KMeans) was proposed for jointly segmenting and clustering starting with random initialization for segments and clusters [59]. This approach assumes that initial segmentation is not known in zero resource conditions and hence the standard k-means can not be used for clustering. BES-GMM and ES-KMeans both used SylSeg approach to restrict the permitted word boundaries.

ES-KMeans approach was shown to perform better than SylSeg and BES-GMM for STD task on Zero Resource 2015 Challenge. Phoneme based ES-KMeans (PES-KMeans) approach improved ES-KMeans by using the phonetic segmentation for better initialization and compact acoustic embeddings for clustering [60]. This work focuses on employing multiple evidences to detect the vowel boundaries for discovering the syllable-like units. These include excitation source information [56], source, spectral peaks and modulation spectrum energies [61], and Bessel features [62]. Different

syllable-like units can be obtained using these vowel boundaries.

Once the speech is segmented into varying-length segments of phone-like units or syllable-like units, these segments should be clustered to label the similar units with the same cluster index or label.

Segment labeling

The next step in the ASM pipeline is segment labeling, which should consistently assign unique labels to acoustically similar segments. Segmentation renders varying length segments which should be represented by a fixed-dimensional representation and then clustered into units which are similar acoustically. A vector quantization (VQ) based approach was used to cluster the segments obtained from a maximum likelihood based segmentation procedure [63]. Gaussian mixture models (GMM) have also been used for labeling the acoustic segments. GMM posteriors for different sound units are expected to occupy orthogonal subspaces, leading to better inter-phone discrimination. Since the GMM parameters are estimated using a large amount of speech data collected from several speakers, the GMM posterior features exhibit better speaker independence compared to the raw spectral features [64]. A major disadvantage of GMM based approach is it assumes the features are statistically independent, and completely ignore the sequence in which they evolve. As a result, there could be too many symbol/feature switches even within a single acoustic segment. To address this issue, segmental GMM was employed, which uses a polynomial fit to capture the time-varying trajectory of frame level features [65]. The polynomial coefficients, along with the length of the segment were used as the features during the clustering step. The problem of finding acoustic units was formulated as discovering self-organizing units (SOUs) [66]. Segmental GMM was used for initializing the Hidden Markov Model (HMM), then the sequences were labeled and model parameters were optimized iteratively. State-of-the-art ASM methods use spectral clustering based methods for labeling the segments [67]. The segments are represented by the average of the GMM posteriors of the frames within the segments ignoring the timing information.

Dynamic time warping (DTW) based methods have been used for segmenting and clustering the speech data [68]. The frame level similarity matrix is constructed using segmental DTW for entire utterances, and similar sub-sequences show up as regions of high similarity along off-diagonal in the frame level self-similarity matrix. A threshold on minimum similarity is used for reducing the number of possible matches. The discovered segments are clustered using the connected component algorithm [69]. These algorithms focus on locating isolated patterns and do not cover entire speech

data.

We propose a segment labeling approach based on graph clustering of variable-length segments obtained from segmentation step. A graph growing approach is proposed for online labeling of segments.

Acoustic modeling

The next step in ASM is segment modeling for which HMMs are most commonly used. Iterative modeling [38, 63, 70] recursively optimizes the acoustic models and segment labels to converge to label sequences and models the ones that best match the observations. During the training process, the segment boundaries and their labels are iteratively refined to improve segmentation/labeling accuracy. Iterative modeling [71] depends on initial labeling and converges towards local optima. The posterior feature vectors extracted from these trained acoustic models are used for further processing.

A non-parametric Bayesian approach was used to jointly solve the segmentation, labeling and modeling problem [72]. A Dirichlet process was used to determine the number of speech units, and HMM was used to model these units in the above-mentioned approach. HMMs have also been used for segmenting and labeling the acoustic segments using transcribed data for forced alignment [73]. An unsupervised acoustic unit discovery approach using non-parametric Bayesian phone-loop model was proposed for topic identification of spoken audio documents [42].

We use HMMs or DNNs for acoustic modeling in this work. The entire ASM approach is termed as signal to symbol transformation. This provides virtual transcriptions in terms of virtual labels given only speech signals without any linguistic information. The applications of ASM are discussed in the next section.

2.3 Applications of zero resource speech processing

The existing literature on the applications to acoustic segment modeling is discussed briefly in this section. Some of these applications are traditionally approached in supervised manner.

2.3.1 Language identification

Language identification (LID) is the task of detecting the language(s) involved in a given spoken utterance. LID plays a central role in multilingual speech recognition

systems. One of the most common approaches to multilingual speech recognition is to first detect the language being spoken and then employ a monolingual speech recognizer for that language [74]. LID also plays an important role in mining information from recorded speech archives [75]. Several comprehensive surveys on LID can be found in literature [76–78].

Depending on the source of language-specific information, LID systems can be broadly classified into acoustic LID systems and phonotactic LID systems [77, 79]. Acoustic LID systems rely on implicit language-specific information captured from acoustic and prosodic features, like nasality, tonality, intonation, stress etc., extracted from the speech signal. Acoustic LID systems are built using discriminative models trained on acoustic features [80], and they do not require manually transcribed speech data.

On the other hand, phonotactic systems rely on explicit language-specific information describing the permissible combinations of phonemes in a language to form meaningful words. As phonotactic LID systems exploit the co-occurrence statistics of phones [76], they require phonetically labeled data. Hence, even though phonotactic LID systems deliver performance comparable to the acoustic LID systems, their applicability is limited by the availability of manually transcribed speech data.

Phonotactic approaches to LID are based on acoustic modeling of phonemes followed by n-gram language modeling to discriminate between the languages [81–84]. In [85], the spoken utterance is converted to a discrete sequence of phonemes, and latent semantic analysis is applied on the decoded phonemes to detect the language-specific information.

Tong *et al.* proposed a target-language oriented phone selection strategy to incorporate discriminative phonotactic features in LID systems [86]. The log likelihood ratios of phonemes was used as an alternate way to incorporate phonotactic features in LID systems [87,88]. There are several approaches to phonotactic LID systems [89], of which parallel phone recognition followed by language modeling (PPRLM) is a prominent technique [90]. Many popular approaches for LID are supervised and require sufficient amount of labeled data for training [91].

LID systems trained with ASM labels provide am unsupervised alternative for identifying the language in low resource conditions. ASM as a front end for LID has been used in previous works to circumvent the requirement of transcribed data [38, 92]. Li et al. used universal set of phonemes or augmented phoneme inventory constructed as a super-set of phonemes from several languages to build universal ASM [38]. All the target languages were tokenized using universal ASM and a vector space modeling based classifier was used to identify the language. Therefore, the

ASM labels inherently provide a basis for LID in an unsupervised way.

An unsupervised phonotactic approach for LID is proposed using the signal to symbol transformation. This method is named as unsupervised parallel phone recognition (UPPR).

2.3.2 Spoken term discovery

Studies on language learning in infants suggest that they have an inherent ability to discriminate between the phones of different languages and self-learn the phonetic structures in a language to which they are exposed to in the first year of their life [2]. The phonetic level acquisition is followed by learning of words, their meanings and subsequently various linguistic aspects of that language, i.e., lexical, syntactic, semantic, etc. This learning in infants happens without being taught or supervised in any way, even before they begin to speak [3]. This indicates that it is possible to learn certain aspects of the linguistic structure in an unsupervised way directly from the speech signal itself. Based on this motivation, spoken term discovery (STD) is aimed at discovering acoustic word units directly from the speech signal without any transcription [17].

Given a speech signal, the STD system should produce the segments of speech with time-stamps along with labels corresponding to each segment which define the category of each segment. The STD task involves three steps, namely, matching, clustering and parsing. Matching refers to finding pairs of segments of speech which are similar. This involves speech segmentation followed by template matching. Clustering refers to assigning all the matching pairs unique cluster labels and thus building a corresponding lexicon. Parsing refers to segmenting any given speech signal and assigning the segments labels using the lexicon of cluster labels.

An unsupervised technique used Dirichlet process mixture model and Hidden Markov model (HMM) to simultaneously segment the speech and learn subword units [93]. Variational Bayes inference was used for automatic unit discovery [94]. Unsupervised acoustic modeling was proposed by partitioning Gaussian mixture model posteriorgrams using Siamese networks [95].

Features with reduced dimensions using unsupervised linear discriminant analysis were used for Dirichlet process Gaussian mixture model clustering for acoustic segment modeling [96]. Gaussian universal background model posteriorgrams were matched using dynamic time warping to improve STD performance [97]. Acoustic models were trained with very small amount of repeating word level annotations for ABX pair discrimination [98]. Different autoencoder architectures were proposed for ABX task and STD task [99,100]. Multi-task learning bottleneck features which were

speaker invariant were proposed for ABX task [101].

STD task evaluations are done using the virtual labels of both categories - phone-like and syllable-like units obtained from the proposed signal to symbol transformation.

2.3.3 Speaking rate estimation

The impact of speaking rate has been studied broadly on ASR. It has been shown that the accuracy of speech recognition decreases as the speaking rate increases. This has been attributed to the increased variation in pronunciations [102]. Incomplete articulation in fast speech leads to acoustic mismatch [103]. In the case of slow speaking rate, factors which affect ASR performance are hyper articulation and intra-syllabic pauses [104]. Fast speaking rate leads to more substitution and deletion errors whereas slow speaking rate leads to more insertion errors. Therefore, speaking rate dependent decoding and speaker adaptation techniques have been proposed to improve the accuracy of ASR [105]. Speaking rate has also been used as a speaker-specific feature for voice conversion [106]. Several techniques have been proposed in the literature for speaking rate estimation (SRE). The techniques for SRE can be broadly classified into acoustic and linguistic methods.

Acoustic methods: These methods estimate the speaking rate directly from the raw speech waveform. The energy rate or *enrate* was proposed for SRE by calculating the first spectral moment of the energy envelope of speech over short-time windows [107]. *enrate* was combined with two different peak picking estimators from the wideband energy envelope of speech and are averaged to arrive at a multiple rate estimator or *mrate*. A method to detect vowels based on smoothed modified loudness was proposed for SRE [108]. A GMM based online SRE approach was proposed in [109]. Wang et. al. have proposed to use subband and temporal correlations to detect syllables for SRE [110]. In the Praat script for SRE, the intensity peaks supported by intensity dips on either side are hypothesized as potential syllable nuclei [111]. In [112], convex cost functions were proposed to estimate temporal density function from time-frequency representation for SRE. A recurrent neural network based approach was proposed for online SRE [113]. A dictionary learning approach using non-negative matrix factorization was proposed for robust SRE [114].

Lexical methods: Lexical methods define speaking rate in terms of phone rate or word rate. Phone rate for an utterance is defined as the ratio of total number of phones to the total duration of phones, essentially, phones per second. The phones are counted after performing phone recognition [107]. A broad class phone recognizer was used for SRE [115]. If accurate phone level transcriptions are not available

but correct word level transcriptions are available, then forced alignment can be performed in order to get phone durations [116]. If both accurate phone level or word level transcriptions are not available, then the phonetic segmentation will not be good and will lead to incorrect information about the number of phones and their corresponding durations. Also, if the availability of data with the orthographic transcriptions is limited, then the ASR model training will not be effective, and it, in turn, will affect the SRE. Therefore, in recent times there is an increased interest in the speech community towards zero resource approaches which do not require any labeled data for training or any explicit linguistic knowledge [17].

The lexical methods directly use manual transcriptions to train the recognizers. The acoustic methods do not use manual labels but make use pause and noise labels in the manual transcriptions to split the utterances into spurts. A zero resource approach towards speaking rate estimation is proposed in this thesis using syllable-like units.

2.3.4 Text-to-speech (TTS) without text

A special session in Interspeech 2019 is organized as ZeroSpeech 2019 : TTS without text [117]. The challenge aims at building speech synthesizers without any text or label information. The goal is to discover subword units using spoken term discovery and transcribing the given speech signals using the lexicon of discovered units. The speech synthesis system can use these virtual labels as the only information for synthesizing speech in target speaker's voice.

This could be viewed as a low bitrate coding problem as the speech signals are coded as virtual labels which could be represented at much lower bitrates compared to speech signals. Another way to perceive this problem could be as unsupervised voice conversion where parallel transcribed data is not available in both source and target speaker's voices.

Building Text-to-speech (TTS) synthesis systems generally require speech and text and/or linguistic information [118]. A text free synthesis approach for building such system was proposed by obtaining automatic transcriptions at phone level from the speech signals [119]. A deep learning based approach for speech chain from speech to text and back to speech [120]. Recently, techniques have been developed without the use of parallel data for voice conversion [121–124].

It is important to note that training usual voice conversion systems requires parallel speech data from source and target speakers' voices [121]. Recently, techniques have been developed without the use of parallel data for voice conversion [121–124].

A submission has been made to this challenge and the associated special session. The proposed system is based on signal to symbol transformation for spoken term

discovery and neural network based speech synthesis.

Chapter 3

Instantaneous frequency features for low resource ASR

There have been several studies, in the recent past, pointing to the importance of analytic phase of the speech signal in human perception, especially in noisy conditions [20]. However, phase information is still not used in state-of-the-art speech recognition systems. Utilizing phase information may help specially in low resource conditions where already there is a lack of transcribed data. In this work, we illustrate the importance of analytic phase of the speech signal for automatic speech recognition. As the computation of analytic phase suffers from inevitable phase wrapping problem, we extract features from its time derivative, referred to as instantaneous frequency (IF) which can be estimated without explicit phase computation. We highlight the issues involved in IF extraction from speech-like signals, and then propose suitable modifications for IF extraction from speech signals. Different IF estimation methods are discussed below and are utilized to estimate known IF of speech-like synthetic signals.

3.1 Overview of IF estimation methods

The complex analytic signal $x_a(t)$ corresponding to the real signal $x(t)$ can be expressed as

$$x_a(t) = x(t) + jx_h(t), \tag{3.1}$$

where $x_h(t)$ denotes the Hilbert transform of $x(t)$ [125]. The analytic signal in (3.1) can be expressed in polar form as

$$x_a(t) = a(t)e^{j\phi(t)} \tag{3.2}$$

where $a(t)$ and $\phi(t)$ denote the amplitude envelope and analytic phase of the signal. The time derivative of the unwrapped analytic phase $\phi(t)$ is referred to as instantaneous frequency (IF) [125], and is given by

$$\Psi(t) = \frac{1}{2\pi}\frac{d\phi(t)}{dt} \tag{3.3}$$

The instantaneous frequency of discrete-time narrowband signal $x[n]$ can be computed using differencing operation on the unwrapped phase, which is not easy to obtain. Phase unwrapping methods are generally considered to be adhoc and not accurate [30]. Several methods are proposed in the literature to estimate IF avoiding the unwrapping of analytic phase [31].

3.1.1 IF estimation using zero-crossing method (IF-ZC)

The simplest way to estimate the IF of a signal is to count the number of zero-crossings over a small segment of the signal. The instantaneous frequency can be estimated from the average number of zero-crossings [31] in a short window of length $(2N+1)$ as

$$z[n] = \frac{\pi}{2(2N+1)} \sum_{m=-N}^{N} |\operatorname{sign}(x[m]) - \operatorname{sign}(x[m-1])| \tag{3.4}$$

where sign(.) denotes the signum function. The efficiency of this method critically depends on the size of the window [126]. Too large a window violates the local property of IF, and hence the estimate ceases to be IF. On the other hand, a smaller window size leads to noisy IF estimates.

3.1.2 IF estimation using LMS algorithm (IF-LMS)

The narrowband signal $x[n]$ can be expressed as an output of a time-varying all-pole filter as

$$\begin{aligned} x[n] &= \sum_{k=1}^{P} a_n[k]x[n-k] + u[n] \\ &= \mathbf{a}_n\mathbf{x}_n^T + u[n] \end{aligned} \tag{3.5}$$

where $\mathbf{x}_n = [x[n-1], x[n-2], \ldots, x[n-P]]^T$ and $\mathbf{a}_n = [a_n[1], a_n[2], \ldots, a_n[P]]^T$ denotes the time-varying predictor coefficients. $u[n]$ is a zero-mean white Gaussian noise process and P is the order of prediction. The IF of the narrowband signal is given by the dominant frequency component in the frequency response of the estimated

adaptive filter, and is given by

$$\Psi[n] = \frac{1}{2\pi} \arg\max_{\omega} \frac{1}{1 - \sum_{k=1}^{P} a_n[k] e^{-j\omega k}} \tag{3.6}$$

where the coefficients of the adaptive filter $\mathbf{a_n}$ can be estimated by least mean squares (LMS) algorithm [127], by minimizing the square of the instantaneous error $e[n] = x[n] - \mathbf{a}_n \mathbf{x}_n^T$. The filter coefficients can be sequentially updated using gradient descent, and are given by

$$\mathbf{a}_{n+1} = \mathbf{a}_n + \mu \mathbf{x}_n e[n] \tag{3.7}$$

where μ is the step size parameter which controls the rate of convergence [127]. The performance of LMS algorithm in estimating IF is severely affected by the choice of step size μ. A very small value of μ cannot track fast varying changes in the IF, where as, a large μ results in noisy IF estimates.

3.1.3 IF estimation using TVAR modeling (IF-TVAR)

In a time-varying auto-regressive (TVAR) model, the time-varying predictor coefficients $a_n[k]$ in (3.5) are expressed in terms of fixed basis functions $f_l[n]$, like Legendre polynomials or Bessel functions, as

$$a_n[k] = \sum_{l=0}^{Q} \alpha_{kl} f_l[n], 1 \leq k \leq P \tag{3.8}$$

where Q is the number of basis functions, and α_{kl} are the weights in linear combination which need to be estimated from $x[n]$. The algorithm for the estimation of α_{kl} can be found in [128]. Once the weights α_{kl} are estimated, the predictor coefficients can be computed from (3.8), which in turn can be used to estimate the IF from (3.6). The performance of this method depends on the choice of basis functions $f_l[n]$ and the number of basis functions Q. A smaller value of Q fails to track fast variations in IF, where as, a higher value of Q results in model overfitting to noise in the data.

3.1.4 IF estimation using Fourier transforms (IF-FT)

The IF can be computed by differentiating the logarithm of (3.2) with respect to t and equating the imaginary parts as

$$\Psi(t) = \frac{1}{2\pi}\phi'(t) = \frac{1}{2\pi}\Im\left\{\frac{x_a'(t)}{x_a(t)}\right\} \tag{3.9}$$

where $\Im\{.\}$ denotes the imaginary part and $x_a'(t)$ is the derivative of the analytic signal $x_a(t)$ which can be computed using the differentiation property of Fourier transform. The expression for IF in (3.9) can be implemented in the discrete-domain as [129]

$$\Psi[n] = \frac{1}{N} \frac{\Im\left\{x_a'[n] x_a[n]\right\}}{|x_a[n]|^2} \tag{3.10}$$

where N, $|x_a[n]|^2$ and $x_a'[n]$ denote the length, the amplitude envelope and the derivative of the analytic signal, respectively. The derivative $x_a'[n]$ can be computed using the differentiation property of Fourier transform as

$$x_a'[n] = j\mathcal{F}^{-1}\left\{kX_a[k]\right\} \tag{3.11}$$

where $X_a[k]$ is the discrete Fourier transform of $x_a[n]$ and $\mathcal{F}^{-1}$ denotes the inverse discrete Fourier transform. Though this method does not involve any hyper parameters, it works well only for synthetic narrowband signals arising in communication systems. It does not work well on speech-like signals, as illustrated next.

3.1.5 Evaluation of IF estimation methods

In order to study the accuracy of IF estimation methods, we have generated synthetic signals with known instantaneous frequency. It is done by simulating a time-varying all-pole system with a pair of complex conjugate poles at $r[n]e^{\pm j\theta[n]}$, whose input $u[n]$ and output $x[n]$ are related by

$$x[n] = 2r[n]\cos(\theta[n])x[n-1] - r^2[n]x[n-2] + u[n] \tag{3.12}$$

In such a system $r[n]$ and $\theta[n]$ control the instantaneous bandwidth and frequency of the output signal $x[n]$. Three kinds of narrowband signals are generated by modifying the characteristics of the system and source as follows:

(i) **Critically damped system with unit sample (CD-US):**
For this case, $r[n] = 1.0,\ \forall n$ and $u[n] = \delta[n]$. Since the poles of the system are lying on the unit circle, it is an unstable system and the unit sample response sustains for infinite period. Frequency modulated signals fall into this category. Fig. 3.1(b) shows a narrowband signal produced by such a system whose instantaneous frequency variation $\theta[n]$ is shown in Fig. 3.1(a).

(ii) **Under damped system with random noise (UD-RN):**
For this case, $r[n]$ is drawn randomly from a uniform distribution between $[0.95, 0.99]$. Notice that the poles must lie close to the unit circle, as we are interested in generating

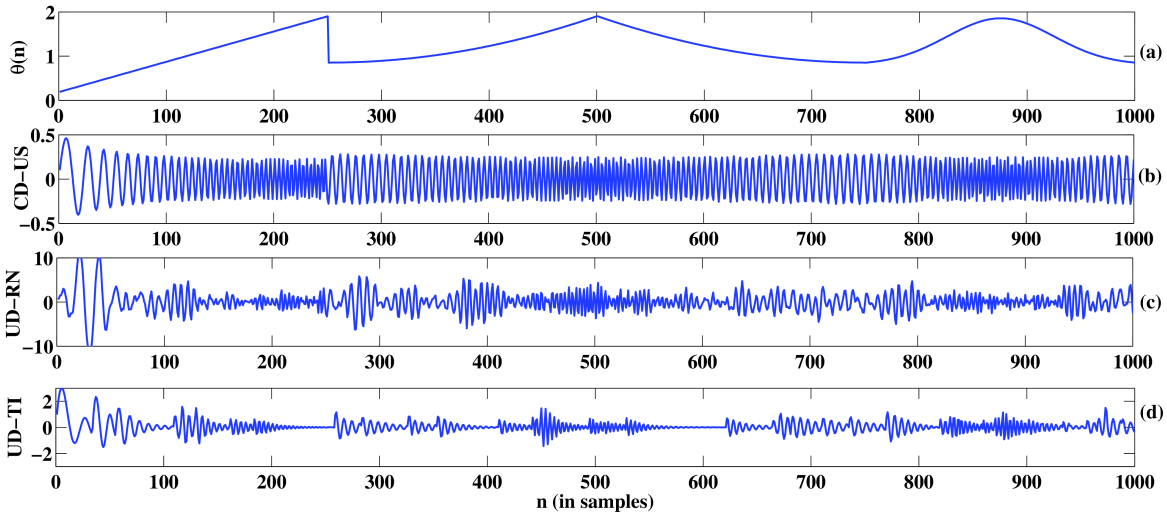


Figure 3.1: (a) IF of the system. System output for - (b) unit impulse (c) random noise (d) train of impulses.

narrowband signal. As the system is under damped, its unit sample response decays down to zero, and hence it needs to be excited continuously to generate a sustained output. In this case, the input $u[n]$ is drawn from a zero mean white Gaussian process. Fig. 3.1(c) shows a narrowband signal produced by exciting an under damped system with white Gaussian noise. Such a system generates narrowband signals with the characteristics of unvoiced/whispered speech sounds.

(iii) **Under damped system with train of impulses (UD-TI):**
For this case, the system characteristics remain exactly same as (ii), and the excitation signal $u[n]$ is modified to a train of quasi periodic impulses to simulate the characteristics of voiced speech. Fig. 3.1(d) shows a narrowband signal produced by exciting an under damped system with a quasi periodic sequence of impulses.

The instantaneous frequencies of the systems that generated the narrowband signals in Fig. 3.1(b), Fig. 3.1(c), and Fig. 3.1(d) are exactly same. However, the characteristics of the output signal look different because of the differences in their inputs and bandwidths. Our goal is to estimate the IF of the system, shown in Fig. 3.1(a), from the output signals shown in Fig. 3.1(b), Fig. 3.1(c), and Fig. 3.1(d). Fig. 3.2 shows the IF estimated by different methods, described above, in all the three scenarios. It can be concluded, from Fig. 3.2(a), that the performance of all the methods is equally good on CD-US signals, except at the abrupt change in IF at 250^{th} sample. This observation is in agreement with the mean squared error (MSE) between the actual and the estimated IFs shown in Table 6.3.

The better performance of the methods on CD-US signals may be attributed to the fact that during their production $r[n] = 1.0$ and $u[n] = \delta[n]$ are kept constant,

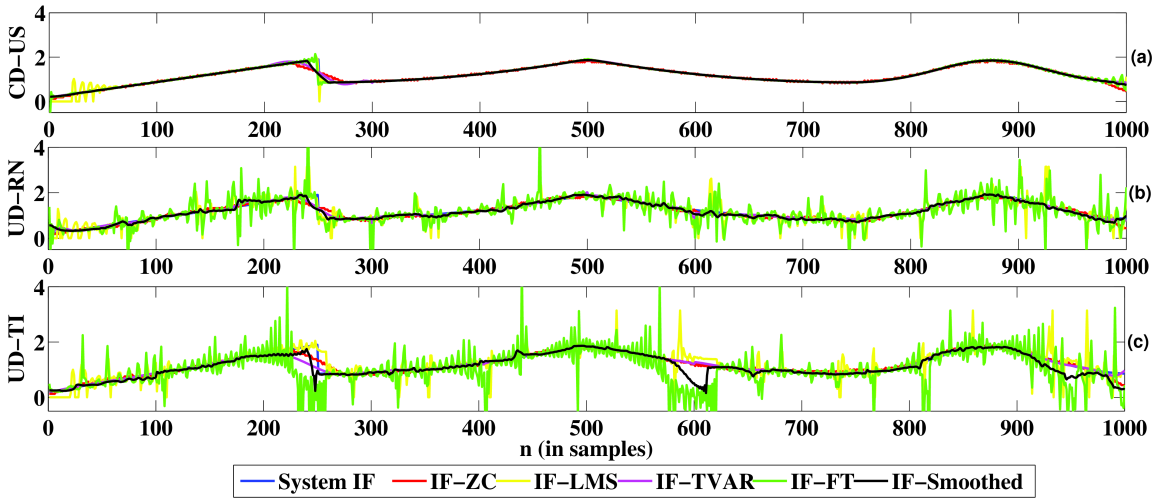


Figure 3.2: True and estimated IF for Synthetic signal for the three systems. System excited with - (a) unit impulse. (b) random noise. (c) train of impulses.

and only $\theta[n]$ is varying with time. On the other hand, during the production of UD-RN and UD-TI signals, all the three quantities $r[n], \theta[n]$, and $u[n]$ are varying with time, and hence there is a drastic drop in the performance of all the methods on these signals. Especially IF-FT method suffered most, compared to the other methods.

3.1.6 IF estimation for speech like signals

The characteristics of the voiced speech signal resemble UD-TI signals. The performance of the methods on UD-TI signals needs to be improved, in order to reliably estimate IF of speech signals. The performance of IF-FT method is extremely poor on UD-TI signals for the following reasons: (i) Since IF computation in (3.10) involves division by squared amplitude envelope of the NB component, the IF exhibits large fluctuations when the amplitude is closer to zero. These fluctuations in the region, from $n = 550$ to $n = 610$ in Fig. 3.2(c), are due to low energy of NB component in that region, as shown in Fig. 3.1(d). (ii) The fluctuations of IF can be due to the impulse-like nature of excitation source as well. During signal generation, the impulse responses of the time-varying system initiated at successive impulses in $u[n]$ are superposed to produce $x[n]$, which is manifested as phase discontinuity, and results in large peaks in the IF. The quasi-periodic peaks in the regions of IF shown in Fig. 3.2 correspond to impulse locations in $u[n]$. In order to minimize the effect of these two artifacts in IF computation, we smooth the numerator and denominator of (3.10), individually, as they are responsible for fluctuations in different regions. It is observed that the smoothing improves the IF estimates to a great extent as shown

Table 3.1: MSE (%) between true and estimated IF from different methods for various excitation signals with a center frequency of 0.5 Hz.

Features	**CD-US**	**UD-RN**	**UD-TI**
IF-ZC	0.0065	0.0120	0.0085
IF-LMS	0.0043	0.1154	0.0795
IF-TVAR	0.0037	0.0114	0.0239
IF-FT	0.0053	0.3455	0.7411
IF-Smoothed	0.0044	0.0092	0.0306

in Table 6.3.

3.2 Feature extraction from IF of speech signals

The IF computed on speech signal $s[n]$ is not meaningful, as it is a wideband signal. Hence the speech signal is passed through a bank of 40 linearly spaced, Gaussian shaped narrowband filters, each having a bandwidth of 400 Hz, centered at $(i-1)*200$ Hz, for $i = 1, 2, \ldots 40$. This operation results in 40 narrowband components $s_i[n],\ i = 1, 2, \ldots, 40$ extracted from the speech signal. IF $\Psi_i[n],\ i = 1, 2, \ldots 40$ is estimated from each of these 40 narrowband components. The IFs extracted from different NB components of the speech signal are plotted in the form of a pyknogram [130] in Fig. 3.4(b). Spectrogram of the speech signal is also provided in Fig. 3.4(a) for comparison. The pyknogram of the speech signal clearly shows the formant transitions which are extremely important for the identification of consonant sounds [130].

The center frequency of the filter is subtracted from the corresponding IF, and average IF is computed for every frame of 25 ms shifted by 10 ms, resulting in a 40-dimensional IF feature vector for every frame. Discrete cosine transform (DCT) is applied on the IF feature vector, in order to pack the information in a smaller number of coefficients. The first 13 coefficients are retained in the DCT domain to represent analytic phase specific information in the 25 ms frame, and are referred to as IF cosine coefficients (IFCCs). The IFCC features are appended with their first and second order time-derivatives to capture the temporal variations, resulting in a 39-dimensional feature vector. Figure 3.3 shows the process of IFCC features extraction. Detailed algorithm for feature extraction can be found in [131].

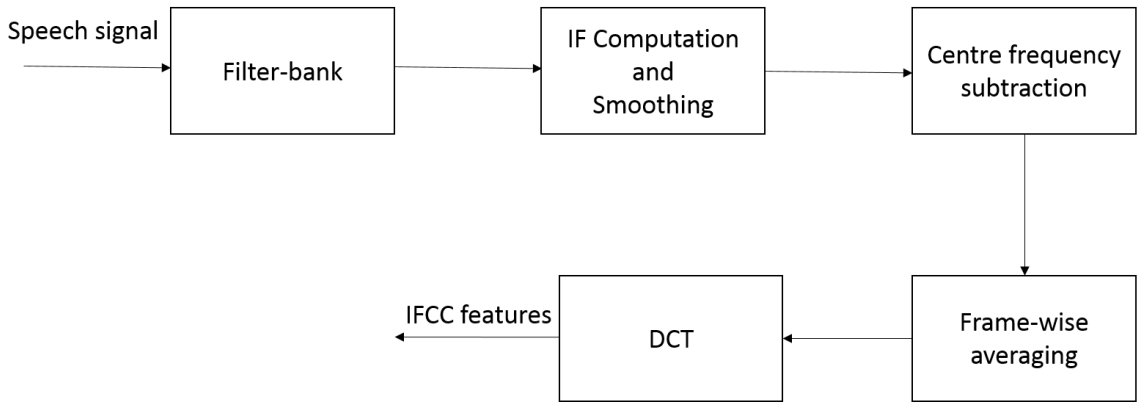


Figure 3.3: IFCC feature extraction process flow

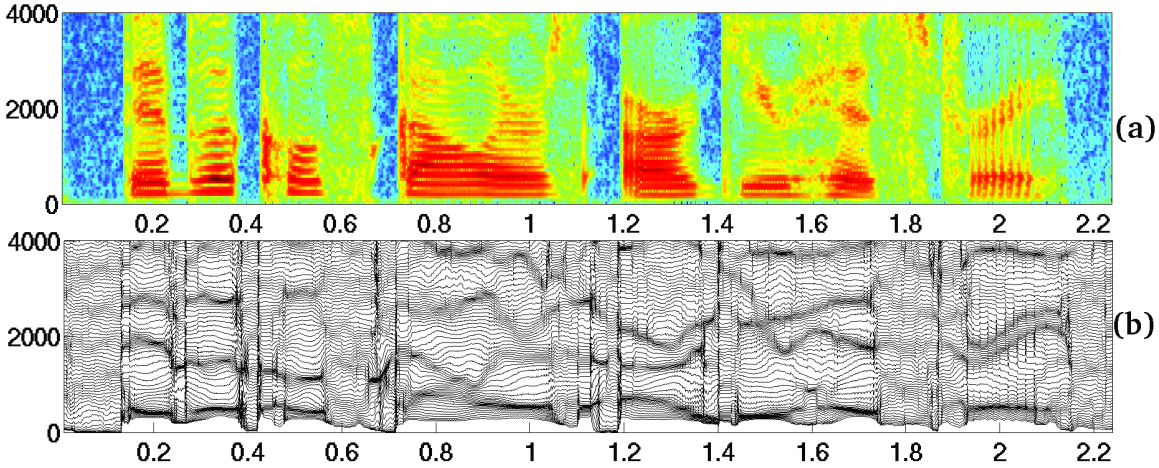


Figure 3.4: (a) Spectrogram and (b) Pyknogram of IF-Smoothed for a TIMIT sentence, sx42.wav.

3.3 Significance of IF in speech recognition

In order to demonstrate the significance of IF in speech processing, speech recognition experiments were conducted on TIMIT database [132]. The description of the data used for training, development (dev) and testing is same as in [133]. A DNN classifier is used to build acoustic models on IFCC features. The target labels for the DNN classifier are generated from the GMM-HMM baseline system trained on MFCC features. The MFCC features are extracted from frames of 25 ms, shifted by 10 ms, using 23 triangular filter banks, placed linearly in the mel-frequency domain. A total of 1951 tied triphone states, obtained from the baseline GMM-HMM system, are used as targets for the DNN systems.

Table 3.2: Phone error rate (%) for MFCC and different IF features based systems on TIMIT core test set

Feature	PER for Dev	PER for Test
IFCC-ZC	24.4	26.3
IFCC-LMS	23.9	26.4
IFCC-TVAR	21.8	24.0
IFCC-FT	23.9	26.2
IFCC-Smoothed	20.1	21.8
MFCC	17.1	18.4
MFCC+IFCC-Smoothed	15.8	16.8

3.3.1 DNN training using IFCC features

The magnitude spectrum captures the second order relations among the samples in time domain, where as the phase spectrum captures the higher order relations among the samples [131]. A DNN with nonlinear hidden layer is an ideal candidate to model the higher order statistical information captured in the IFCC features, which is derived from the phase of the speech signal. In this work, 6-layer DNN, with 1024 neurons in each layer, is used to map the IFCC features to the targets. The input to the DNN is 11 frames of 39 dimensional features concatenated together. The weights of the network are initialized using greedy layer wise pre-training [134]. During the testing, the posterior probabilities of the tied triphone states are converted into likelihoods by dividing them with the relative frequencies of the triphone states. The likelihoods obtained from the DNN are used as emission probabilities in HMM framework to decode the best possible phoneme sequence using Viterbi algorithm [135].

The performance of the speech recognition system on core TIMIT test set, evaluated in terms of phoneme error rate (PER), is given in Table 6.4 for different IFCC features. The performance of the DNN based speech recognition system on MFCC features is given as baseline. Conventional feature processing [136] and DNN parameters optimized on dev data provide state-of-the-art MFCC baseline. The IFCC features extracted using IF-Smoothed method performed best, with 21.8%, among all the IF estimation methods. No significant change was observed in PER when linearly spaced filter-bank was replaced with mel filter-bank. Also, filter-bank outputs without applying DCT were used as features which did not result in any improvement in accuracy. Even though the performance of IFCC features is lower than that of MFCC features, it is encouraging to note that analytic phase contains significant speech-specific information.

3.3.2 MBR decoding based system combination

In order to combine the complimentary evidence available from the magnitude and phase of the speech signal, the posterior lattices obtained from the MFCC and IFCC features were combined using minimum Bayes risk (MBR) decoding [137]. MBR decoding minimizes the expected PER across multiple systems to decode an optimal phone sequence given by

$$\mathcal{P}^* = \underset{\mathcal{P}}{\operatorname{argmin}} \left\{ \sum_{i=1}^{n} w_i \sum_{\mathcal{P}'} P_i \left(\mathcal{P}' | \mathbf{O} \right) L \left(\mathcal{P}, \mathcal{P}' \right) \right\} \tag{3.13}$$

where w_i is the weight for the i^{th} system, $P_i\left(\mathcal{P}|\mathbf{O}\right)$ is the posterior probability of phone sequence $\mathcal{P}$ given acoustic observation sequence $\mathbf{O}$ for the i^{th} system and $L\left(\mathcal{P}, \mathcal{P}'\right)$ is the Levenshtein distance between two phone sequences, $\mathcal{P}$ and $\mathcal{P}'$. Best PER was achieved with 70% weighting for MFCC and 30% for IF-Smoothed features as optimized on dev data. The performance of the combined system is 16.8% for test, which is 1.6% better than the system based solely on MFCC features. This system is currently listed among top 6 state-of-the art systems on TIMIT standard test set [138]. Hence, a relative improvement of 8.7% was achieved over the system based on MFCC system by combining the evidence from IFCC features. This study clearly demonstrates the complimentary nature of analytic phase for speech recognition.

3.4 Noise robust speech recognition using IF features

The significance of phase of the speech signal becomes higher in noisy conditions as the speech recognition performance becomes worse in lower SNRs [23]. Therefore, in this work phase based instantaneous frequency features are explored to complement magnitude based features for improving recognition performance in noisy conditions. Feature extraction from instantaneous frequency of speech signals is discussed below.

Fig. 1(a) shows a clean speech signal from TIMIT database, Fig. 1(b) shows the Spectrogram for this signal and Fig. 1(c) shows Pyknogram [130] of Smoothed IFCC features extracted using 64 such filters. Fig. 1(d), Fig. 1(e) and Fig. 1(f) show corresponding figures for a noisy version of the same utterance with 10 dB white noise. The Pyknogram clearly exhibits the variations in formants and slight degradation is visible in case of noisy version.

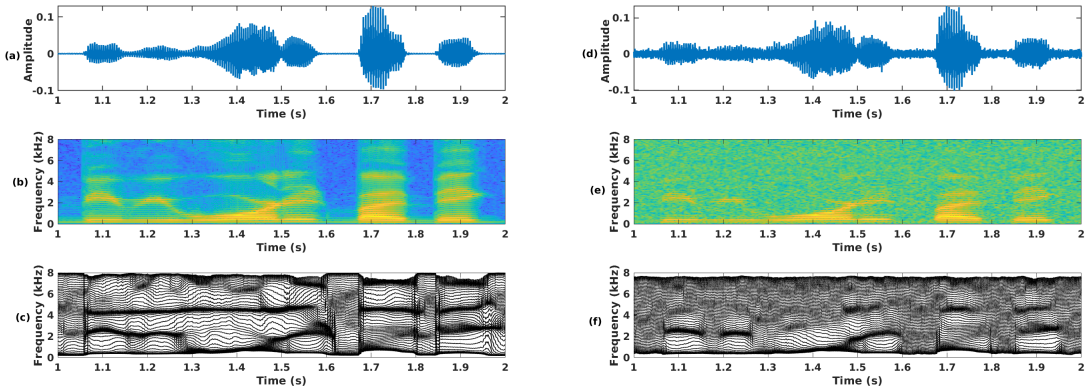


Figure 3.5: a) Speech signal, (b) Spectrogram, and (c) Pyknogram of smoothed IFCC features for clean utterance from TIMIT. (d) Speech signal, (e) Spectrogram, and (f) Pyknogram of smoothed IFCC features for the same utterance with 10 dB white noise.

3.4.1 Database

We created noisy version of TIMIT dev and test sets by adding white, babble and car noise at 10 dB, 15 dB and 20 dB SNR levels. Noise samples are taken from noise files provided with [139]. A bigram language model is used during recognition.

3.4.2 Experimental Results

The phoneme recognition performance is evaluated in terms of PER on clean TIMIT core test set and noisy TIMIT test sets with white, babble and car noises at different SNRs. Table 3.3 shows the comparison of PERs for MFCC based monophone GMM-HMM systems from [32] and current work for the above noises and clean conditions at 10 dB SNR. Also, PERs are evaluated for corresponding MFCC based systems combined with IF-Mean features from [32] and IFCC features from current work. The difference in the MFCC baselines can be attributed to the difference in monophone training strategies. The system combination of proposed features with MFCC provided absolute improvement of 8.39%, 7.95%, 3.8% and 0.1% for clean, white noise, babble noise and car noise conditions respectively over the MFCC+IF-Mean system in [32]. Therefore, the IFCC features provide significantly better combination with MFCCs for recognition on clean speech and with white noise compared to mean IF features.

Table 3.4 shows the results for DNN-HMM systems trained on clean speech and tested in different noise conditions at different SNR levels for MFCC, IFCC and their

Table 3.3: Comparison of phone error rates (%) for different MFCC based monophone GMM-HMM systems (baseline) and their corresponding combined systems with different IF features at 10 dB SNR.

System	TIMIT	TIMIT + White	TIMIT + Babble	TIMIT + Car
MFCC [1] [32]	41.6	82.28	72.29	47.25
MFCC (Proposed)	32.6	70.4	56.9	37.5
MFCC + IF [1] [32]	40.59	73.95	61.4	43.5
MFCC+IFCC (Proposed)	32.2	66.0	57.6	43.4

Table 3.4: Phone error rates (%) in clean and noisy conditions for DNN-HMM systems trained on clean speech.

System	Clean	White			Babble			Car Noise		
SNR(dB)	–	10	15	20	10	15	20	10	15	20
MFCC	18.8	66.9	56.1	44.7	48.3	37.8	29.2	27.7	25.8	29
IFCC	22.1	69.1	53.8	43.1	51.0	41.1	34.1	46.8	42.5	36.6
MFCC+IFCC	17.3	53.9	44.4	35.5	40.6	32.5	26.5	29.7	27.3	24.5

system combinations. The system trained with MFCC features from clean speech recorded a PER of 18.8% on TIMIT core test set which is better than 20.0% PER of the best system reported in [6] based on convolutional DNNs trained with Mel filter-bank features. The system combination of MFCC and IF features provided absolute improvement of 13%, 11.7% and −2% over MFCC alone for white noise, babble noise and car noise respectively at 10 dB SNR. The improvements reduced to 9.2%, 2.7% and 4.5% at 20 dB. This shows the robustness of IFCC features in more noisy conditions. Also, IFCC features provide the highest performance improvement for speech with white noise. As the SNR increases, recognition accuracy improves as it is closer to matched training conditions and hence there is reduction in improvement at higher SNRs. There is slight degradation in performance of the combined system in car noise conditions at lower SNRs as MFCCs are able to model significantly better than IFCCs in this case. But, the combination performs better than MFCCs as the SNR increases to 20 dB.

Table 3.5: Phone error rates (%) for phonetic classes - vowels, fricatives and plosives for white noise.

System	Vowels			Fricatives			Plosives		
SNR(dB)	10	15	20	10	15	20	10	15	20
MFCC	53.5	42.0	34.8	89.3	72.6	53.1	98.6	91.4	73.6
IFCC	60.4	46.5	37.4	98.9	73.8	54.0	94.3	83.2	67.3
MFCC+IFCC	52.9	41.9	34.5	78.3	65.8	46.6	96.8	87.5	72.5

The performance of IFCC features and combined systems for different phonetic classes in white noise conditions is given in Table 3.5. Three broad phonetic classes - vowels, fricatives and plosives are considered for this evaluation. Semi-vowels have also been considered into the vowel category. There is significant improvement of 11%, 6.8% and 6.5% in case of fricatives at 10 dB, 15 dB and 20 dB SNR levels. Slight improvement is observed in case of vowels and plosives as well. This shows that IFCC features along with MFCCs can recognize different phonetic classes both voiced and unvoiced in a better way than only MFCC features in both clean and noisy conditions.

3.5 Summary

This work investigates different instantaneous frequency features as an ASR front-end. Different IF techniques were studied for speech-like synthetic signals and the performance was observed in terms of MSE between true and estimated IFs. Evaluations were conducted for TIMIT phone recognition task using various IF features derived from filter-bank. DNN-HMM systems were trained on magnitude and IF features. The performance of IF features was lower than MFCC features but indicated significant speech-specific information. Magnitude and IF feature based systems were combined using MBR decoding. The system combination outperformed the standalone magnitude or IF features. This demonstrates the complimentary information present in instantaneous frequency features and the significance of combining evidences from magnitude and analytic phase in the context of ASR.

Noise robust speech recognition experiments were conducted for TIMIT phone recognition task under clean and various noisy conditions. The magnitude and IF features based GMM-HMM and DNN-HMM systems were trained on clean speech. MBR decoding was used to combine MFCC and IFCC based systems. System combination of both features delivered absolute improvements of upto 13% over MFCC features alone for DNN-HMM systems under noisy conditions. IFCC features in combination with MFCC features provided significant improvement for all phonetic classes in clean and white noise conditions. The improvement was significantly higher for unvoiced phones. IFCC features are more robust in higher noise levels. This demonstrates the significance of combining evidences from magnitude and phase for noise robust speech recognition. IF based features are effective in conjunction with magnitude based features for different types of noises, different levels of noise and also in different broad phonetic classes.

As low resource recognition accuracies are lower than high resource languages and

is highly dependent on the amount of transcribed resources available, it is imperative to build techniques which could automatically transcribe data without human intervention for building recognizers in unsupervised manner. The next chapter discusses signal to symbol transformation which is an approach towards automatic labeling of speech signals in zero resource settings.

Chapter 4

Signal to symbol transformation : phone-like units

Speech systems for low resource are language specific and do not match the performance of systems developed for high resource languages. Therefore, creating virtual transcriptions from speech signals seems to be a promising direction. It can be generalized to any language and any amount of data could be transcribed using such a technique.

In this chapter, we propose signal to symbol transformation from speech signals obtaining consistent virtual labels which are phone-like units. This ASM approach segments the speech signals into phone-like units using kernel-Gram matrices. These segments are varying-length and are aligned using DTW. Then, the aligned segments are clustered and labeled consistently using a graph clustering and graph growing approach. Acoustic models are trained using speech signals and their virtual transcriptions obtained from segment labeling. The acoustic model can be used for iteratively refining the segment boundaries or can generate posterior probabilities. The evaluation of segmentation and ASM performance is done. Spoken term discovery is performed using these virtual phones on ZeroSpeech 2015 data.

4.1 Kernel-gram segmentation

Let the sequence of states of the vocal-tract system during the production of a speech signal be represented by a sequence of feature vectors $\mathbf{X} = (\mathrm{x}_1, \mathrm{x}_2, \ldots, \mathrm{x}_N)$, where x_i denotes the d-dimensional feature vector extracted from i^{th} frame of the speech signal, and N is the total number of frames. The objective of speech segmentation is to divide the sequence $\mathbf{X}$ into K non-overlapping contiguous segments $\mathbf{S} = (\mathrm{s}_1, \mathrm{s}_2, \ldots \mathrm{s}_K)$, where

s_j denotes j^{th} segment that begins at frame b_j and ends at frame e_j. The segmentation algorithm should ensure that feature vectors in each of the segment s_j are acoustically similar, and represent a phone-like unit. Hence the segmentation algorithm should determine the number of segments and detect their beginning and end points from the acoustic similarity of the feature vectors $\mathbf{X}$.

In the absence of any information about the source distribution, we propose to detect the segment boundaries from the Gram matrix obtained using Gaussian kernel [140]. We assume that two feature vectors from the same segment must have a higher degree of similarity than two feature vectors from different segments. This assumption is justified as the feature vectors from the same segment are drawn from the same source distribution, while the feature vectors from different segments are drawn from different source distributions.

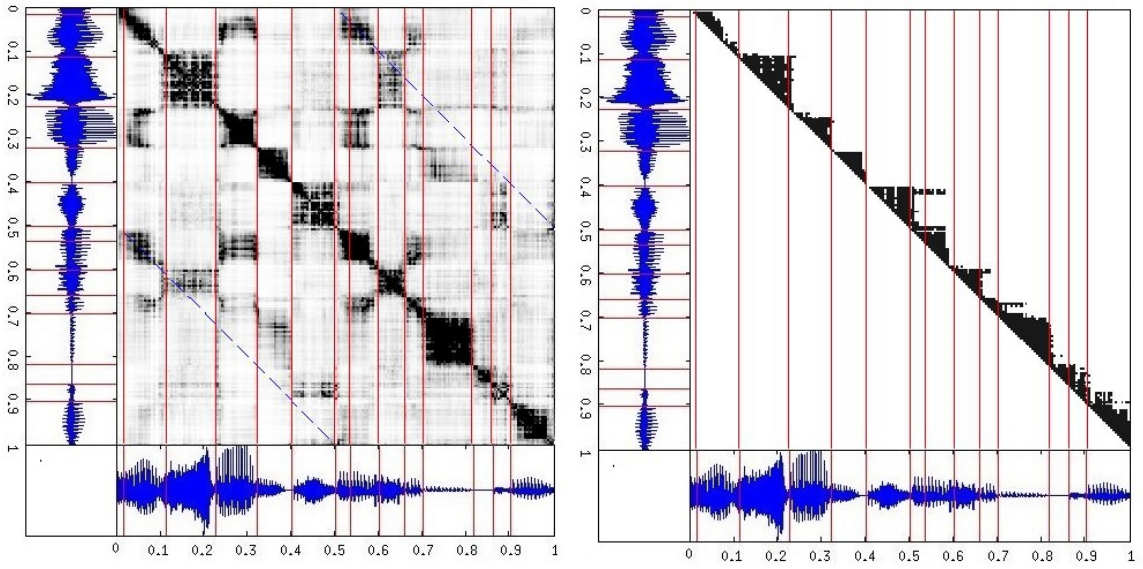
In the proposed approach, the similarity between two feature vectors x_i and x_j is computed using Gaussian kernel as

$$G(i,j) = \exp\left(-\frac{||\mathrm{x}_i - \mathrm{x}_j||}{h}\right) \tag{4.1}$$

where $||.||$ denotes the Euclidean norm of a vector and h is a free parameter which can be used to adjust the width of the Gaussian kernel. Kernel-Gram matrix G can be obtained by computing the similarity between every pair of feature vectors in the sequence $\mathbf{X}$.

Gram matrix computed from 13-dimensional Mel-frequency cepstral coefficient (MFCC) features, extracted from a speech utterance is shown in Fig. 4.1(a). The intensity of a pixel at (i,j) indicates the similarity $G(i,j)$ between the feature vectors x_i and x_j. The region around the principal diagonal corresponds to temporally closer segments. The square patches of higher degree of similarity along the diagonal correspond to acoustically similar segment. Manually marked phone boundaries are also shown in the Fig. 4.1. It is observed that the manually marked boundaries, shown in red colour, coincide exactly with the square patches along the principal diagonal. In this work, the task of speech segmentation is equivalent to identifying the square patches along the main diagonal of the Gram matrix.

As segment boundaries occur in a small neighborhood around the diagonal, the search space can be restricted to a small region parallel to the diagonal. This is analogous to constraining the dynamic time warping path using Itakura parallelogram [141]. The length constraints are shown in Fig. 4.1(a) with dotted lines. As Gram matrix is symmetric, it is enough to compute the upper triangular portion of the Gram matrix. These two constraints lead to significant reduction in computa-



(a) Kernel-Gram matrix. (b) End points detected from pruned matrix.

Figure 4.1: Illustration of Similarity matrices. Red lines indicate manually marked phone boundaries.

tional complexity. The similarity values are higher when the column index j is close to the row index i, i.e., around diagonal, indicating that the frames i and j belong to same segment. On the other hand, as the column index j moves away from the row index i, the similarity values decrease indicating that the frames i and j belong to different segments.

A density-based algorithm is used for identifying the segment boundaries from the kernel gram matrix. We share some concepts such as reachability and ϵ neighbourhood with DBSCAN [142] algorithm which are used in the calculation of segment boundaries.

A frame, x_j, is in ϵ neighbourhood of x_i, if $1-G(i,j) < \epsilon$. Given an appropriate ϵ, all the features that are from the same segment as that of x_i will be in ϵ neighbourhood of x_i. Since segments are continuous only consecutive frames can be in a segment. For frame x_i, we check consecutive frames for the neighbourhood and maintain a run length (number of neighbours) l_i for each frame.

A frame x_{i+l_i} is temporally reachable from x_i, if all the frames in between x_i, x_{i+l_i} are in ϵ neighbourhood of frame x_i. The first temporally unreachable frame from x_i can be considered as the end of the segment containing the x_i frame. However, making a decision with just one single frame could be erroneous due to noise in data. Hence, we use K-step temporal unreachablity, i.e., K successive frames being unreachable, to identify the boundary of the segment containing x_i. The chances of making an

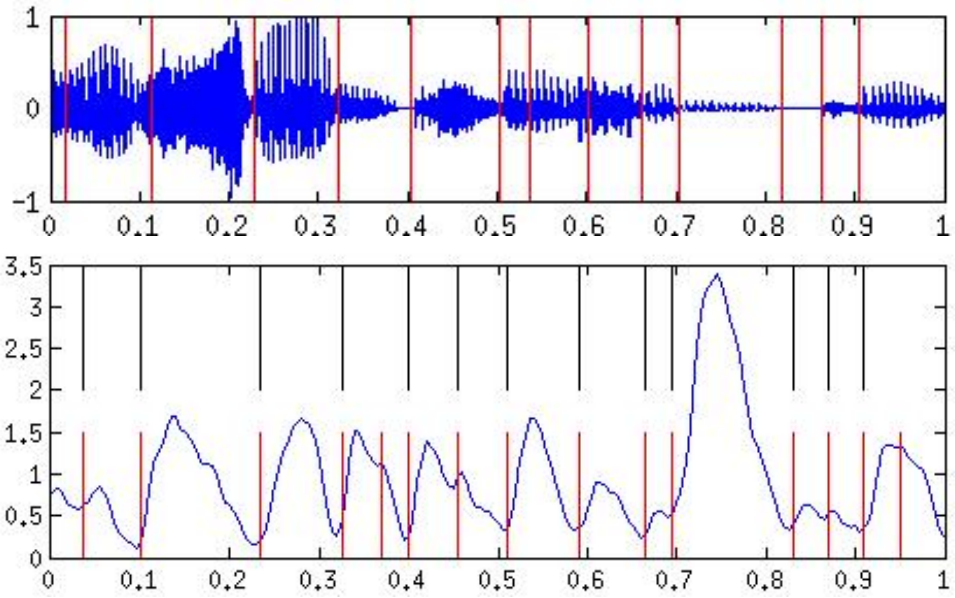


Figure 4.2: Top - Speech signal with manually marked boundaries. Bottom - Segment profile with detected boundaries shown in Red lines. Black lines show manually marked boundaries.

erroneous estimate decrease with increase in K. So for a frame x_i, the end point is estimated at $i + l_i$ if and only if x_{i+l_i} is temporally reachable by x_i and all the frames $x_{i+l_i+1}, \ldots, x_{i+l_i+K}$ are not in ϵ neighbourhood of x_i.

A frame in the beginning of the segment will have the highest number of neighbours and a frame, in the end, will have the lowest number of neighbours. For each frame, we construct the neighbourhood graph, which is basically the sum of all the similarities in the K-point reachablity of that frame. Neighbourhood graph is given as

$$N_G(i) = \Sigma_{j=i}^{l_i} G(i,j) \tag{4.2}$$

where l_i is the run length of x_i frame. For a frame, in the beginning, the value of N_G will be highest since the maximum number of frames contribute to it. As we move towards the end of the segment the value of N_G will keep decreasing as shown in Fig. 4.2. At the end point, the N_G will be minimum, ideally zero. The location of minimas in N_G gives the location of end points of all the segments in the given utterance.

Given the reachability threshold ϵ and parameter K in K-step temporal unreachability, any utterance can be segmented into phone-like units. The segmentation performance critically depends on the choice of ϵ. Acoustic properties of the segments differ across segments. For example, a frame taken from voiced segment will be more similar to another frame taken from voiced segment as compared to a frame taken from an unvoiced segment. This makes finding a global ϵ very hard which is

consistent across different segments. So, we use an ϵ value that adapts itself according to the acoustic properties of the segment.

To automatically determine the value of ϵ and to allow different segments to use separate ϵ threshold, we develop a simple algorithm. The algorithm is based on the observation that similarity between frames of the same segment is higher than the average similarity. New threshold for x_i frame is given as

$$\epsilon_i = \frac{\sum_{j=i}^{\tau} G(i,j)}{\tau} \tag{4.3}$$

where τ is the diagonal window constraint. For each frame, a different ϵ is computed automatically using the acoustic properties of the segment in consideration.

After getting segment boundaries, minimum length criteria is used for avoiding segments that are not possible. Minimum length is fixed to be 20 ms. K is chosen to be equal to the minimum length of the segment because if there is a segment boundary then at least minimum length number of frames will be temporally unreachable after the boundary. For longer segments, the number of unreachable points will exceed minimum length.

4.1.1 Segmentation: experimental evaluation

The speech segmentation algorithm proposed in the paper was evaluated on TIMIT [143] and Zero Resource 2015 databases: Tsonga and English [144]. The following sections explain the performance evaluation on both the benchmarks. The kernel width is simply kept 1 and the algorithm determines the value of ϵ automatically.

Segmentation performance on TIMIT

TIMIT dataset has been used for evaluation segmentation algorithms [47, 145–148]. All the sentences were manually transcribed and segmented at phone level using 61 phone labels. In this work, MFCC features are used to represent the state of the vocal-tract system at a given instant of time. The segment boundaries are extracted from the kernel-Gram matrix, computed from the MFCC features, as discussed in Section 4.1. Let N_C be the number of correctly detected boundaries (within a given tolerance interval), N_T is the total number of detected boundaries and N_G is the total number manual boundaries. The performance of the proposed algorithm is evaluated, by comparing the detected boundaries with the manually marked boundaries, using the following intermediate metrics:

- Hit Rate (HR) : It is the fraction of reference boundaries that are correctly

Table 4.1: Performance comparison of speech segmentation algorithms for 20 ms tolerance window. The $*$ mark represents use of a validation set for parameter fine tuning.

method	F	R
Kernel Width (h = 1)	0.76	0.79
Dusan et. al. [47]	0.71	0.73
Khanagha et. al. [145]	0.74	0.77
Adriana et. al. [146] *	0.76	0.80
Leow et. al. [147] *	0.75	0.78
Rasanen et. al. [148] *	0.76	0.78

detected (N_C/N_G). It is also called recall rate of the segmentation system.

- Over Segmentation (OS) : It represents how many extra (less) boundaries are detected as compared to reference boundaries ($(N_T - N_G)/N_G$).
- False Alarm (FA) : The fraction of incorrectly detected boundaries ($(N_T - N_C)/N_T$).

The overall quantification of segmentation algorithm is done with a global measure, F score, which combines all the intermediate scores.

$$F = \frac{2 * (1 - FA) * HR}{1 - FA + HR} \tag{4.4}$$

There is another global measure, R, which emphasizes more on over segmentation (OS). It argues that recall rate can be increased by inserting random boundaries without changing the algorithm.

$$r_1 = \sqrt{(1 - HR)^2 + (OS)^2};\ r_2 = \frac{-OS + HR - 1}{\sqrt{2}} \tag{4.5}$$

The final metric is defined as

$$R = 1 - \frac{|r_1| + |r_2|}{2} \tag{4.6}$$

We use metrics R and F for evaluating the segmentation algorithm. The performance of the proposed algorithm is given in Table 4.1. For $h = 1$, approximately 73% of the detected boundaries fall within the 20 ms tolerance interval from the manually marked boundaries.

The agglomerative algorithm proposed by Qiao et. al, requires the number of expected segments as input [151]. This method uses manual transcriptions for cal-

Table 4.2: Results (in percentage) for STD task on Zerospeech 2015 databases: English and Xitsonga (in brackets). The best scores for each evaluation metric are highlighted in bold.

System	boundary		
	Precision	Recall	F-score
Baseline [149]	44.1 (22.3)	4.7 (5.6)	8.6 (8.9)
Vseg [58]	**76.1** (26.2)	28.5 (26.3)	41.4 (26.3)
EnvMin [58]	75.7 (16.3)	27.4 (24.4)	40.3 (19.5)
Osc [58]	75.7 **(29.2)**	33.7 (39.4)	46.7 (33.5)
CC-PLP [150]	39.6 (19.4)	7.5 (11.2)	12.7 (14.2)
CC-FDPLS [150]	35.4 (18.8)	38.5 (64)	36.9 (29)
Proposed	41.2 (22.5)	**71.1** (**74.8**)	**52.2** (**34.6**)

culating the exact number of segments for the input utterance. The neural network based segmentation method proposed by Vuuren et. al. [152] used transcriptions for entire train data to learn the probability distribution of segment lengths. Both these approaches achieve very high performance but due to their strong prior requirements, recent works [145, 146] have put them in the category of semi-supervised approaches and performance comparison is done only with zero or minimal fine tuning approaches. We follow the same practice. Adriana et. al. [146] used a small validation set for adjusting the minimum peak height in probability function. Also, the beginning and end silence regions were trimmed to 50 ms which contribute a high number of spectral discontinuities in input signal. Leow et. al. [147] found the best performing system by evaluating the performance and then choosing the parameters of the best system. In the proposed method, the kernel width is simply kept 1. The proposed algorithm selects the optimal number of segments automatically.

Segmentation performance on zero resource 2015 dataset

We also evaluate the performance of the proposed segmentation method on the zero speech challenge 2015 datasets. This dataset consists of 10.5 hours of casual conversations in American English, and 5 hours of read speech in Xitsonga. The aim of the challenge (Track 2) was to discover recurring speech patterns in an unsupervised manner. Evaluation kit for measuring quality of discovered sub-words were provided as part of the challenge. Segments are clustered and combined to discover large segments but the boundaries are not altered during that step. In the present work, we only evaluate the segmentation performance. Recall measures the probability of finding a manual boundary within 30 ms of a discovered boundary. Precision measures the probability that a discovered boundary is within 30 ms of a manual boundary.

The F-score is the harmonic mean of precision and recall. If the algorithm predicts boundaries only where manual boundaries are, then both precision and recall will be 1. The recall can be increased by predicting more boundaries but that would decrease the precision. Similarly, precision can be increased by predicting limited number of boundaries. The precision and recall can be traded off for each other. The F-score combines both of them and is used as a global measure for segmentation evaluation.

4.2 Segment labeling

The segmentation step divides the data into a large number of varying length segments. Segment labeling is the process of assigning unique labels to the acoustically similar segments. In this study, we propose to use graph clustering method for segment labeling. Graph clustering aims to find a subset of nodes that are tightly connected, where the edge weight connecting two nodes can be defined using an appropriate similarity measure.

In this work, an undirected adjacency graph is constructed with each segment as a node and similarity between the segments as edge weights. The graph construction from segmented speech utterance is illustrated in Fig. 4.3, where each node corresponds to a segment in the utterance and thickness of an edge represents the similarity between the nodes. To define the similarity between the nodes, we need to compare two varying-length segments. It is difficult to arrive at a fixed dimensional representation for varying length segments. Though there have been attempts to represent the segment by its mean cepstral vector, it does not capture the temporal dynamics of the segment. We compute the similarity by aligning the sequence of frames in a pair of segments using dynamic time warping and then using the cosine distance on the aligned segments. Given two segments, $\boldsymbol{y}_i := (x_{b_i}, ..., x_{e_i})$ and $\boldsymbol{y}_j := (x_{b_j}, ..., x_{e_j})$, DTW finds an optimal match between them by non-linearly time-warping the sequences. The resulting warping path can be backtracked and the indices encountered in the path can be used to align the two sequences. DTW is used for aligning the two segments and then cosine distance is calculated. Cosine distance lies between 0 and 1, so it can be directly used as graph weights for clustering. Cosine distance between two segments is given as

$$w_{ij} = \frac{\phi(\boldsymbol{y}_i)\phi(\boldsymbol{y}_j)}{\|\phi(\boldsymbol{y}_i)\|.\|\phi(\boldsymbol{y}_j)\|} \tag{4.7}$$

where $\phi(\boldsymbol{y}_i)$ represents the time warped version of the segment $\boldsymbol{y}_i$. Once the graph

is formed with nodes as segments and edges or weights as cosine distance, graph clustering can be done to identify the similar segments. Graph clustering technique is discussed below.

4.2.1 Graph clustering

Graph clustering is the process of identifying subgraphs within a graph which have stronger connections. Graph partition and clustering is an active area of research. There is a myriad of available clustering techniques. After the graph is formed, a suitable algorithm for clustering the graph is required.

In this approach, the problem of graph clustering is mapped onto finding the ground state of an infinite range Potts spin glass [153]. We chose this particular clustering technique because of the following reasons. A single parameter γ controls the weights of missing and existing links in the quality function. Also, it is computationally less demanding, and expectation values of the modularity can be calculated analytically. It helps in determining the clustering tendency of the graph.

Modularity measures the quality of division of a network into clusters. Graphs with high modularity exhibit dense connections between nodes within a cluster and sparse connections between nodes in different clusters.

Various quality functions have been used for evaluating clustering techniques. An ideal quality function should - (i) strengthen the internal edges within a cluster (same spin state), (ii) penalize the missing edges within a clusters, (iii) weaken the edges between clusters and (iv) encourage non-links between two different clusters.

(i) and (ii) encourage the dense connection in a cluster. (iii) and (iv) reduce the overlap between two different clusters.

The energy of a spin glass system is same as the quality function of the clustering with spin states as cluster indices. For a weighted undirected graph, Hamiltonian of the spin glass system is given by

$$\mathcal{J}(\{\sigma\}) = -\sum_{i \neq j} (W_{ij} - \gamma p_{ij}) \delta(\sigma_i, \sigma_j) \tag{4.8}$$

where W represents the adjacency matrix of the graph, $\sigma_i = \{1, 2, ..., q\}$ are possible cluster indices for node i in the graph and δ is the Dirac delta function. The number of spin states controls the maximum number of clusters allowed. The Hamiltonian compares the true distribution of links in the graph with the expected distribution under a model which defines p_{ij}. It has been shown that simulated annealing yields high-quality results for the Potts-model. The use of simulated annealing is general purpose and simple to implement [153]. Simulated annealing algorithm is used to

minimize this Hamiltonian. Minimizing Hamiltonian is equivalent to maximizing the modularity of the graph [153].

It is easy to compute and many other measures also reduce to modularity under appropriate assumptions. It is the comparative measure used in the present work. Hamiltonian and modularity are related by the following equation:

$$Q = -\frac{1}{|V|}\mathcal{J}(\sigma) \tag{4.9}$$

where $|V|$ is the total number of edges in the graph.

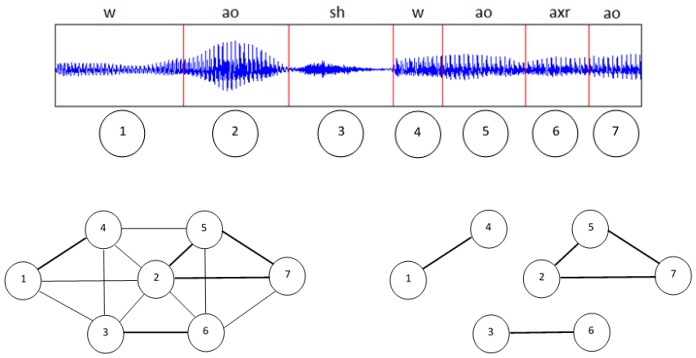


Figure 4.3: Overview of segment labeling process. Top - Segments assigned with labels. Bottom left - Graph formed with similar segments having stronger weights (darker edges). Bottom right - Clustered graph

4.2.2 Graph growing: seeded graph clustering

Similar segments are clustered together using graph clustering technique discussed in the previous section. Then, each cluster is assigned a label which is the initial or the most probable label for all the nodes (speech segments) in the cluster. There are a few issues in applying this algorithm directly. This approach works considerably well for our ASM when the issues mentioned below are addressed:

1. *Size of graph*: The average number of phones in English varies from an average 9.4 to 13.83 per second for speech ranging from poetry to sports commentary [154]. Hence, for an hour of poetic speech, there will be around 33840 phones which would generate a graph with approximately 10^9 edges. The number of edges increases exponentially with increase in data (multiplies by 10^2/hour). Storing and operating such big graphs is very hard and not

straightforward. As a node can potentially belong to any cluster, so the heuristic algorithms can not be applied. Therefore, the entire graph has to be taken into consideration.

2. *Offline clustering*: The system works in an offline manner i.e., it labels the entire data together. To label a new speech utterance, entire graph clustering algorithm has to be re-run which is redundant and is very time-consuming. This technique can not be used to label a segment that was not a part of the system initially. Also, only those graphs which can be accommodated in the physical memory of the system can be clustered in a go. For huge graphs, this algorithm can not be used.

To address the above issues, we propose graph growing as an alternate method to label the segments. In our approach, a small portion of the graph is clustered first. The small graph gives us initial information about the number of clusters present and the distribution of these clusters. It is used as a seed and nodes are added incrementally. We compute the average connectedness of a new acoustic segment to every cluster in the clustered graph. Average connectedness between a cluster and a segment is defined as the mean similarity of the segment to every other segment in the cluster. The new segment is assigned to the cluster with the highest average connectedness. This describes the labeling of segments in an online manner.

4.3 Unsupervised acoustic modeling

We have used DNNs for unsupervised acoustic modeling from a large corpus of speech signals and their corresponding virtual phone transcriptions (cluster indices). In this work, we have used 50 virtual phone labels to transcribe the speech data. Each virtual phone is modeled as a 3-state continuous density hidden Markov model. The model parameters are estimated using Baum-Welch embedded re-estimation from the virtual phone labels. The trained HMM models are used to force align the virtual phones to refine the boundaries obtained from the segmentation step. The state level alignments obtained from the HMM modeling are used as targets to train a DNN classifier. DNN, being a discriminative model, provides better estimates for emission probabilities of the HMM. We used a 6-layer DNN, with 1024 rectified linear units in each layer, to estimate posterior probabilities of the virtual phone states from the acoustic input. The input to the DNN is 39-dimensional (13 MFCCs + deltas + deltas deltas) MFCC features with 7-frame context window.

The trained DNN is capable of generating the virtual phone state posteriors, which can be used as a representative of speech specific information in the speech signal. The state posterior features should, in principle, be speaker-invariant as the speaker-specific information is marginalized during the clustering stage. The DNN, in combination with the HMM, is used to decode the best possible virtual phone sequence for a given speech signal.

4.3.1 Homogeneity of segment labels

Ideally, a labeling algorithm should give the same label to all the instances of a particular class. To check the uniformity of the labels, we performed the following experiments. Two utterances from two different speakers with the same transcription from train section of TIMIT are segmented and labeled using graph labeling. It is evident from Fig. 4.4 that graph labeling is consistent across utterances from different speakers. The sequence of labels obtained from proposed ASM (shown as numbers within black boundaries in Fig. 4.4) is same for both utterances though the time span of these virtual phones is different as both speakers articulate phones of different lengths.

4.3.2 Evaluation metric and comparison with other techniques

Acoustic segment labeling is essentially like solving a clustering problem. If the ASM is perfect, then ASM segments would match the linguistically defined phones. The metrics which are used for evaluating clustering techniques are used here for evaluating ASM performance. In this section, we evaluate the ASM performance by a direct comparison between manual phonetic transcriptions and labels obtained from the ASM.

Let $\mathcal{C} = \{c_1, c_2, ..., c_P\}$ and $\Omega = \{\omega_1, \omega_2, ..., \omega_Q\}$ represent the set of discovered clusters (ASM units) and the set of linguistically defined acoustic units respectively. Here, P and Q denote the total number of ASM units and linguistically defined acoustic units respectively. ω_q represents the number of linguistically defined acoustic units present in q^{th} cluster and c_p represents the number of segments that are assigned label p. Purity is computed by assigning each discovered cluster to the class which is linguistically most frequent in the cluster. ASM units assigned to cluster c_p and having actual label ω_q are represented by $\omega_q \cap c_p$. The most commonly used evaluation metrics for measuring clustering performance are:

- *Purity* measures the fraction of data points assigned to the right cluster. It

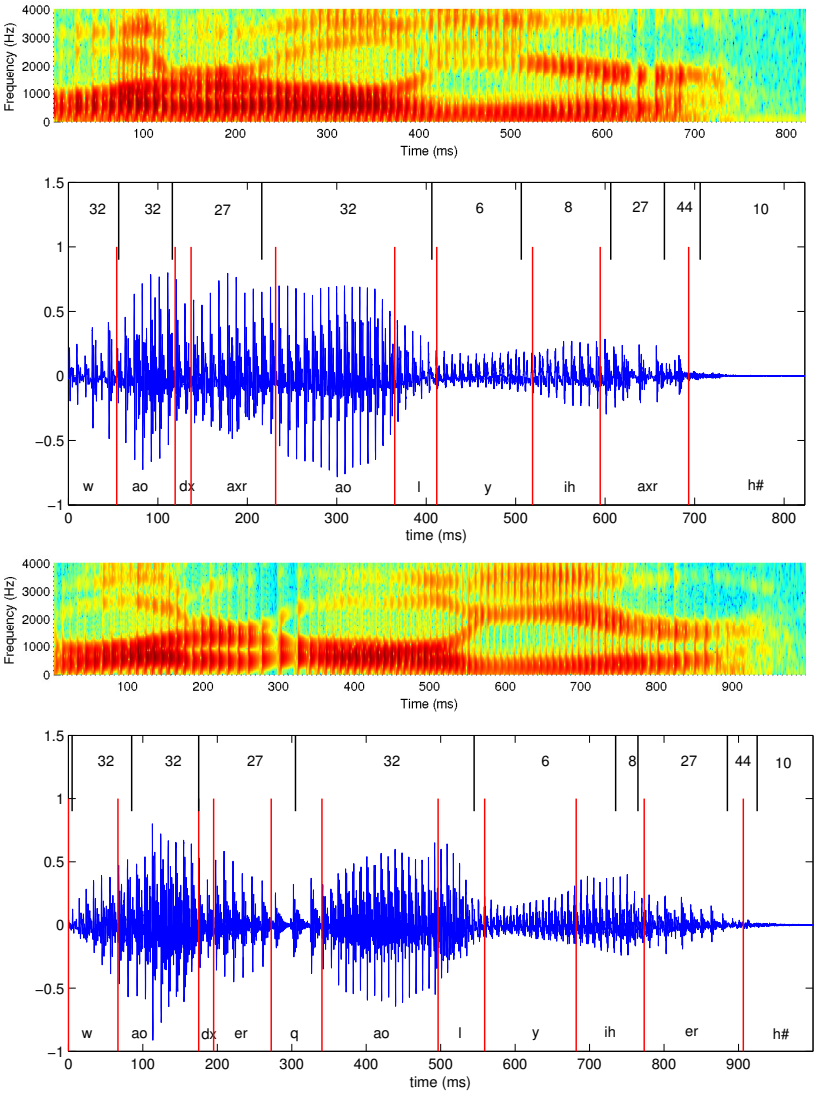


Figure 4.4: Comparison of labels of same TIMIT utterance "Water all year" for different speakers. Top Spectrogram and waveform correspond to the first speaker. Bottom Spectrogram and waveform correspond to the second speaker. Red lines in the waveform show manually marked boundaries with corresponding phonetic transcriptions. Black lines in the waveform show boundaries obtained by the proposed method and corresponding virtual phones.

measures the compactness of clustering algorithm. Purity for a label p can be expressed as follows:

$$purity(p) = \frac{max_p|\omega_q \cap c_p|}{\sum_{q=1}^{Q} |\omega_q \cap c_p|} \tag{4.10}$$

Overall purity can be defined as:

$$purity = \frac{1}{|E|} \sum_{q=1}^{Q} max_p|\omega_q \cap c_p| \tag{4.11}$$

where $|E|$ is the total number of segments.

- *NMI* measures the amount of information that can be obtained about manual labeling from virtual phone labeling.

$$NMI(\Omega, \mathcal{C}) = \frac{I(\Omega; \mathcal{C})}{[\mathcal{H}(\Omega) + \mathcal{H}(\mathcal{C})]/2} \tag{4.12}$$

where I is mutual information and is given by:

$$I(\Omega; \mathcal{C}) = \sum_q \sum_p \frac{|\omega_q \cap c_p|}{|E|} log\left(\frac{|E|.|\omega_q \cap c_p|}{|\omega_q||c_p|}\right) \tag{4.13}$$

$$\mathcal{H}(\Omega) = -\sum_q \frac{|\omega_q|}{|E|} log\left(\frac{|\omega_q|}{|E|}\right) \tag{4.14}$$

A high value of purity is easy to achieve as it increases with increase in the number of clusters. Purity is 1 when the number of clusters is same as the number of nodes or when each node gets its own cluster. This limitation of purity can be overcome by using NMI as an additional measure. The denominator in NMI increases with increase in the number of clusters and reaches its maximum when the number of clusters becomes same as the number of nodes, which makes NMI low.

The performance of the proposed algorithm is compared with four other ASM techniques in Table 4.3. The proposed algorithm outperforms the baseline approaches with significant margin. In vector quantization (VQ) approach [63], the segments are represented using the means of the segments. In VQ, clustering is done by applying k-means directly on mean MFCC features. In GMM labeling [64] technique, a GMM is trained from all the training data. The segment is assigned to the Gaussian component yielding maximum likelihood. Gaussian Clustering (GCC) and Segment clustering

Algorithm	NMI	Purity
VQ [63]	0.161	0.232
GMM labeling [64]	0.198	0.244
GCC [67]	0.208	0.272
SC [67]	0.192	0.269
Graph clustering	0.363	0.440
Iterative modeling	0.405	0.451

Table 4.3: Comparison of available ASM algorithms on TIMIT.

(SC) [67] are spectral clustering based methods which utilize the eigenvalues of the adjacency matrix for clustering the data. The average Gaussian posterior of the acoustic segments is used as an input representation and the inner product between posteriors is used as a similarity measure for constructing the adjacency matrix. GCC and SC require silence regions to be removed before clustering, which is done by using manual transcriptions. The agglomerative segmentation [151] method is used for obtaining the segments. It works well but requires the number of segments per utterance to be known before segmentation. Our proposed algorithm does not have any pre-processing requirements. Iterative modeling is done to further improve the quality of the virtual phones obtained from the graph clustering.

To demonstrate the impact of initial segmentation on labeling performance, the following experiments were conducted. In the first experiment, segmentation is done in an unsupervised manner by using kernel gram segmentation and in the second experiment, manual segmentation is used. The acoustic segments are then clustered using the same graph growing method. NMI and purity are used as evaluation metrics. Better segmentation facilitates more consistent clustering of acoustic units by reducing the differences in similarity scores for different instances of same units. Clustering depends on the adjacency matrix. A better adjacency matrix produces higher quality labels even when the clustering algorithm is kept the same. This is also evident from Table 4.4, where manual segmentation performs better than completely unsupervised segmentation. Manual segmentation is the ground-truth for all the segmentation algorithms, so the later experiment is also the upper bound on maximum achievable NMI and purity for the proposed labeling algorithm.

Labeling Method	NMI	Purity
Unsupervised	0.363	0.440
Supervised	0.460	0.488

Table 4.4: Performance comparison of segment labeling using unsupervised and supervised (manual) segmentation.

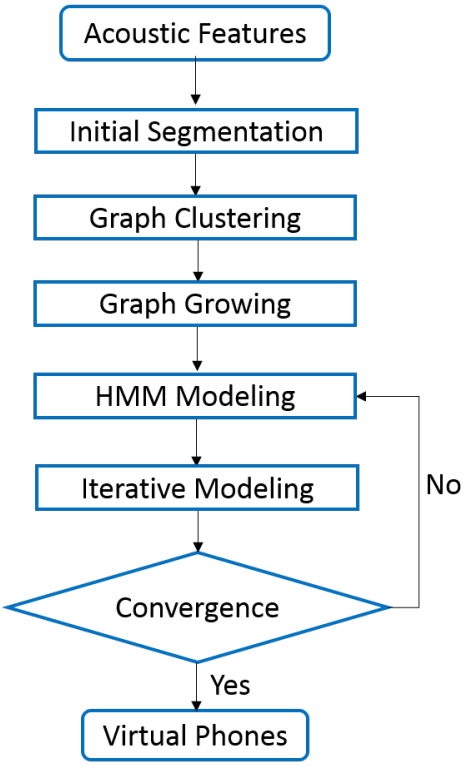


Figure 4.5: Overview of Signal to Symbol Transformation

In our approach, DTW is used to quantify the similarity between two acoustic segments. We have compared the usage of DTW score with the mean of the segment as a representation for clustering in Table 4.5. DTW considers segments in entirety for comparison whereas mean loses the vital temporal information present in the segments. Thus DTW score acts as a better similarity measure and hence improves clustering performance. But this performance improvement comes with a significant increase in the computational cost. All the above experiments are done with the number of clusters fixed as 50.

Algorithm	NMI	Purity
Mean	0.333	0.387
DTW	0.363	0.440

Table 4.5: Comparison of segment mean or DTW score between two segments as a representation for similarity measurement.

4.4 Iterative refinement of segmentation

The labels obtained from graph clustering are used to train the acoustic models for iterative refinement. HMMs provide an efficient framework for modeling time-varying speech signal. Each virtual phone is modeled using a left-to-right HMM, in which each state is modeled using a GMM. Each ASM unit is characterized by an HMM so that all training data can be decoded into sequences of ASM units using Viterbi decoding.

Given initial virtual labels $\mathbf{O}$ and observations $\mathbf{X}$, iterative modeling tries to re-estimate boundaries and build acoustic models that best fit the observations. The problem can be formulated as maximum likelihood problem [66].

$$\lambda^*, O^* = arg \max_{\lambda} \max_{O} p(\mathbf{X}, \mathbf{O}|\lambda) \tag{4.15}$$

A two-step iterative optimization procedure, which successively optimizes labels and acoustic models, is used for refining both the labels and models.

$$O^i = arg\max_{O} p(\mathbf{X}, \mathbf{O}|\lambda^i) \tag{4.16a}$$

$$\lambda^i = arg\max_{\lambda} p(\mathbf{X}, O^{i-1}|\lambda) \tag{4.16b}$$

We find the best label sequences using the existing acoustic models (4.16a). Viterbi decoding [155] is used for this step. We then update the acoustic models λ^i using labels O^{i-1} from previous iteration (4.16b). It is same as training HMMs with new labels. The likelihood of the observations improves after every iteration. The iterative process is stopped when there is no change in the likelihood or the change in successive iterations is less than a threshold. This iterative process converges towards local optimum value and is very sensitive to initial labeling. In general, three to five iterations of re-estimation is enough to reach convergence. Information from all the segmented utterances is used for training the models. Many possible instances of same acoustic units are used to cover all the possible variations and learn better models. Acoustic models trained on the virtual phones can be used to refine segment boundaries. Virtual phone HMMs are used to force align the labeled data to produce the segment boundaries. Global segmentation uses information from all the utterances to segment the input utterance as opposed to Kernel-Gram segmentation which uses local information.

This technique to transform acoustic features from speech signals to virtual phone sequences is referred to as signal to symbol transformation. The process flow of signal

to symbol transformation is shown in Fig. 4.5.

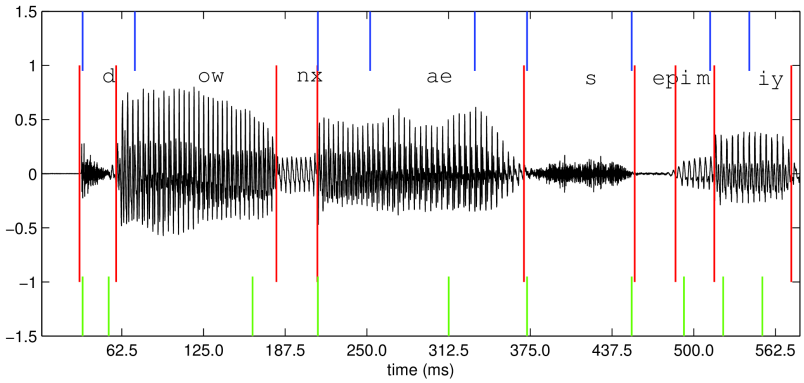


Figure 4.6: Comparison of labels obtained using HMM and Graphical model. Green lines represent boundaries obtained by HMM, red lines represent manual segmentation boundaries and blue lines represent boundaries by Kernel-Gram matrix.

4.5 Spoken term discovery using virtual phones

Motivated from infant learning, the ZeroSpeech 2015 Challenge [156] was aimed at discovering acoustic word units directly from the speech signal without any transcription. This task is known as spoken term discovery (STD). Given a speech signal, the STD system should produce the segments of speech with time-stamps along with labels corresponding to each segment which define the category of each segment.

The STD task involves three steps, namely, matching, clustering and parsing. Matching refers to finding pairs of segments of speech which are similar. This involves speech segmentation followed by template matching. Clustering refers to assigning all the matching pairs unique cluster labels and thus building a corresponding lexicon. Parsing refers to segmenting any given speech signal and assigning the segments labels using the lexicon of cluster labels.

The aim of the STD task is the unsupervised discovery of "words" defined as recurring speech fragments. The systems should take raw speech as input and output a list of speech fragments (time-stamps referring to the original audio file) together with a discrete label for category membership.

The evaluation will use the suite of F-score metrics described in [157], which enables detailed assessment of the different components of a spoken term discovery pipeline (matching, clustering, segmentation, parsing) and so will support a direct comparison with the unsupervised word segmentation models. We propose to identify the repeating word like patterns from the sequence of virtual phones decoded from

the speech signal. The choice of the length of the word plays an important role in searching for words from discrete symbols.

The performance of the STD task is benchmarked against several well established unsupervised term discovery metrics [157]. Normalized edit distance (NED) measures the variability among the phone sequences of a word class, while coverage (Cov) measures the portion of the phone sequences covered in the discovered word units. Other evaluation metrics include token recall, type, and boundary. The token recall is the probability that a gold word (manual word transcription) token is found in obtained word classes. Token precision is the probability that an obtained word token would match a gold word token. A similar definition is used for calculation of type performance. Finally, the segmentation measures the accuracy of boundaries of discovered phone classes with respect to actual word boundaries.

The proposed unsupervised acoustic modeling can be used to obtain either a continuous representation of the speech signal in terms of (virtual phone) state posteriors or alternatively a discrete representation in terms of a sequence of virtual phones.The effectiveness of the continuous and discrete representations of the speech signal, obtained using the proposed method, is illustrated on the STD task.

The performance of the proposed method on these two tasks is evaluated on the zero speech challenge 2015 dataset. This dataset consists of 10.5 hours of casual conversations in American English, and 5 hours of read speech in Xitsonga. Evaluation kit for the STD task was provided as part of the challenge.

The performance of the proposed method on STD task is given in Table 4.6, for both English and Xitsonga. The most prominent finding is a full coverage segmentation algorithm with very high word segmentation accuracy on both languages. In English, found patterns cover the entirety of the speech data with 41.2% of the found boundaries matching a true boundary. It finds 71% of the existing boundaries in the data. Similar performance is observed for Tsonga, found patterns cover 96% of the data and locate 75% of the boundaries. The baseline system and other STD system achieve better precision. The high precision might be due to very selective nature of the systems (less coverage). So, we performed additional experiments to allow only high-quality patterns (proposed 2). We used a minimum similarity threshold while growing the graph. The precision of the new system increased as expected and is now almost twice the previous value. The precision reaches the topline precision. There is a decrease in recall performance. The overall boundary performance (F-score) is still better than the baseline and other STD algorithms. Precision and recall can be traded for each other depending on the application in hand. Higher type and token performance demonstrate the quality of obtained word units using the proposed al-

System	NLP	•	type		•	token			boundary		
•	NED	Cov	P	R	F	P	R	F	P	R	F
Baseline [149]	**21.9** (**12**)	16.3 (16.2)	6.2 (3.2)	1.9 (1.4)	2.9 (2.0)	5.5 (2.6)	0.4 (0.5)	0.8 (0.8)	44.1 (22.3)	4.7 (5.6)	8.6 (8.9)
Vseg [58]	89.6 (78.4)	40.6 (77.7)	13.5 (1.7)	11.3 (4.1)	12.3 (2.4)	21.6 (1.8)	4.8 (1.8)	7.9 (1.8)	76.1 (26.2)	28.5 (26.3)	41.4 (26.3)
EnvMin [58]	88 (61.2)	42.2 (95)	12.7 (1.1)	10.8 (3.3)	11.6 (1.7)	21.6 (0.8)	4.7 (1.3)	7.8 (1.0)	75.7 (16.3)	27.4 (24.4)	40.3 (19.5)
Osc [58]	70.8 (63.1)	42.4 (94.7)	**14.1** (2.2)	12.9 (6.2)	**13.5** (3.3)	**22.6** (2.3)	6.1 (3.4)	9.6 (2.7)	75.7 (29.2)	33.7 (39.4)	46.7 (33.5)
CC-PLP [150]	77.3 (36.1)	25.5 (30.2)	4.7 (3.0)	2.5 (2.7)	3.3 (2.8)	4.2 (2.0)	0.6 (0.9)	1.0 (1.2)	39.6 (19.4)	7.5 (11.2)	12.7 (14.2)
CC-FDPLS [150]	61.2 (43.2)	80.2 (89.4)	3.1 (**4.9**)	9.2 (**18.8**)	4.6 (**7.8**)	2.4 (2.2)	3.5 (**12.6**)	2.8 (3.8)	35.4 (18.8)	38.5 (64)	36.9 (29)
proposed	85.0 (66)	**100** (**95.8**)	5.4 (2.3)	**24.8** (8.0)	8.9 (3.6)	7.9 (**2.7**)	**13.9** (8.5)	**10.1** (**4.1**)	41.2 (22.5)	**71.1** (**74.8**)	**52.2** (**34.6**)
proposed 2	91.3 (80)	5.1 (4.9)							**81.4** (**61.2**)	15.7 (27.8)	26.2 (38.2)
Topline (supervised)	0 (0)	100 (100)	50.3 (15.1)	56.2 (18.1)	53.1 (16.5)	68.2 (34.1)	60.8 (49.7)	64.3 (40.4)	88.4 (66.6)	86.7 (91.9)	87.5 (77.2)

Table 4.6: Results (in percentage) for STD task on Zerospeech 2015 datasets: English and Xitsonga (in brackets). The best scores for each evaluation metric are highlighted in bold. Topline performance is obtained with manual labels.

gorithm. Overall, our algorithm achieves the best performance in the highest number of evaluation metrics on both the languages.

4.6 Summary

An end-to-end framework for unsupervised acoustic segment modeling is proposed in this chapter. Novel algorithms are proposed for improved segmentation and labeling. The segmentation algorithm proposed in the paper automatically selects the optimal number of segments in an utterance without any supervision. The proposed kernel-Gram segmentation provides better performance than the current existing approaches. A graph growing based clustering method is proposed to label large datasets with lesser computational resources. Iterative modeling is done to refine the segment boundaries and labels. Experimental analysis is done to find an empirical relationship between the initial segmentation performance and the quality of virtual phones finally obtained.

The effectiveness of the proposed approach is evaluated on STD task using spon-

taneous American English and Tsonga language datasets, provided as part of zero resource 2015 challenge. It is observed that the proposed system outperforms baselines, supplied along the datasets, in both the tasks without any task specific modifications.

Chapter 5

Signal to symbol transformation : syllable-like units

The phone-like units obtained from the ASM approach in the previous chapter are small units. DTW matching can work better for larger units. Also, larger units like syllables could be potentially more suitable candidates for spoken term discovery as we need to discover word like units.

A sequence of speech sounds with a maximum of sonority between two minima of sonority is known as a syllable. A syllable has a central part known as the nucleus (mostly a vowel) and optional parts namely - onset (if present, at the beginning of the syllable) and coda (if present, at the end of the syllable).

Early research showed that the syllable is the fundamental unit for the perception of speech in infants [53]. Syllables have also been proved to be effective as a basis for the segmentation of speech signals [54]. Syllable-like units have been used as fundamental units for applications such as speech recognition [55], speaker verification [56], language identification [57] etc.

In this work, we use signal processing methods to detect vowel end points (VEPs) which are used as anchor points to identify the syllable-like units. Multiple evidences extracted from the source and spectral characteristics of the speech signal are used for accurate VEP detection. The evidences from the excitation source information include zero frequency filtered signal and Hilbert envelope of linear prediction (LP) residual of speech signal [56], while the evidences from the spectral characteristics include spectral peaks and modulation spectrum energies [61]. The evidences from the source and spectral features are finally combined with the evidence from the Bessel features to arrive at accurate VEP locations. The significance of these evidences for VEP detection is briefly described in the following subsections.

5.1 Multiple evidences for VEP detection

In this work, we use multiple evidences extracted from the source and spectral characteristics of the speech signal for accurate VEP detection. The evidences from the excitation source information include zero frequency filtered signal and Hilbert envelope of linear prediction (LP) residual of speech signal [56], while the evidences from the spectral characteristics include spectral peaks and modulation spectrum energies [61]. The evidences from the source and spectral features are finally combined with the evidence from the Bessel features to arrive at accurate VEP locations. The significance of these evidences for VEP detection is briefly described in the following subsections.

5.1.1 Evidence for VEP from source features

Speech production model is considered to be a time-varying system excited with quasi-periodic sequence of impulses or noise for voiced or unvoiced sounds respectively [158]. The change in the nature of excitation from voiced to unvoiced is an important clue to detect the VEPs. This excitation source information extracted from the Hilbert envelope of LP residual and zero-frequency filtered signal [56] has been exploited to detect the VEPs.

Hilbert envelope of LP residual: It is the magnitude of the complex analytic signal formed from the LP residual [159]. It preserves the excitation source characteristics and is given by the following equation -

$$H_e(n) = \sqrt{e^2(n) + \hat{e}^2(n)} \tag{5.1}$$

where $e(n)$ is the LP residual and $\hat{e}(n)$ is its Hilbert transform. The Hilbert envelope is smoothed by retaining the maximum value for every 5ms with a shift of one sample. The smoothed Hilbert envelope provides evidence for the detection of VEPs.

Zero Frequency Filtered Signal (ZFFS): The ZFFS [129] is obtained by passing the pre-emphasized speech signal through a cascade of two ideal zero frequency resonators, and subtracting the trend from the resulting signal.

$$\hat{y}(n) = -\sum_{k=1}^{4} c_k y(n-k) + s(n) - s(n-1) \tag{5.2}$$

$$\hat{y}(n) = y[n] - \frac{1}{2N+1} \sum_{n=-N}^{N} y(n) \tag{5.3}$$

where the filter coefficients are $c_1 = 4, c_2 = -6, c_3 = 4, c_4 = -1$. The average pitch period is used as the window length $2N+1$ for the trend removal. Since $\hat{y}(n)$, referred to as ZFFS, is obtained by passing speech signal though a narrowband filter centered around 0 Hz, it predominantly contains the excitation strength information.

To detect the VEP locations, the points at which there is significant change in the excitation information are detected by convolving from right to left, the Hilbert Envelope of LP residual or ZFFS with a 100 ms length first order Gaussian differentiator (FOGD) with standard deviation of one sixth of window length. These evidences are summed up and normalized by the maximum value. The resulting envelope provides excitation source based evidence for the VEP locations.

5.1.2 Evidence for VEP from spectral features

The vowels are produced by a relatively open and relatively stationary vocal-tract system compared to the consonants. Hence, the strength of the formants and rate of change of the spectral content provides a strong evidence for the detection of the vowels and their end points.

- **Spectral Peaks**: The shape of the vocal tract which leads to the production of different vowels can be estimated by selecting a few largest spectral peaks. Speech signal is windowed into frames of 20 ms with a frame shift of 10 ms. A 256-point discrete Fourier transform (DFT) is applied to each frame and the sum of ten largest peaks from the first 128 points is computed. This spectral peak sum preserves the evidence for VEP detection [61].

- **Modulation spectrum:** Change in modulation spectrum energy also corresponds to the vowel end points as it represents change in slowly varying temporal and frequency components of speech signal [160]. The VEP evidence is obtained from the modulation spectrum by passing the speech signal through a band of 18 critical trapezoidal shaped band pass filters in the range of 0-4 kHz. Then, the amplitude envelope of the signal is computed by half wave rectification and low pass filtering at 28 Hz. Thereafter, utterance level normalization is done for the amplitude envelope in each band by dividing by its average value. Further, DFT is computed by using a Hamming window with 250 ms width and 12.5 ms shift to analyze the modulations of the processed amplitude envelope in the range of 4-16 kHz. Finally, the energies from all the bands in this frequency range are summed to get the modulation spectrum energy [161]. The change is further enhanced by computing the slope of the modulation spectrum energy.

Significant changes in the spectral peak and modulation energy envelopes provide evidence for VEP detection. Hence, the individual evidences are convolved with the FOGD operator and added to enhance the VEPs.

5.1.3 Evidence for VEP from Bessel features

Schroeder argued that any arbitrary signal can be effectively represented by using basis functions which resemble the signal itself [162]. Speech signal can also be considered to be generated by an under-damped time-varying all pole system with a periodic train of impulses or a random noise excitation which produces series of decaying quasi periodic sinusoids resembling voiced speech or narrowband signals resembling whispered/unvoiced speech respectively [163]. Bessel basis functions are damped sinusoids with decaying amplitude and regular zero crossings which makes them suitable representation for speech signals, and for vowel end point detection as well [164]. The k-th order Bessel function is given by

$$J_k(\lambda) = \sum_{r=0}^{\infty} \frac{(-1)^r}{r!\Gamma(r+k+1)} \left(\frac{\lambda}{2}\right)^{2r+k} \tag{5.4}$$

A speech signal $s(t)$ can be represented in terms of Bessel functions in the time interval $(0, l)$ as

$$s(t) = \sum_{r=1}^{\infty} C_r J_0(\frac{\lambda_r t}{l}), \quad 0 < t < l \tag{5.5}$$

where $J_0(.)$ represents 0^{th}-order Bessel funtion, C_r are the coefficients of the Bessel function and $\lambda_r, r = 1, 2, \ldots$ are the positive roots of $J_0(\lambda) = 0$ in the ascending order. Bessel coefficients are given by

$$C_r = \frac{2\int_0^l t s(t) J_0(\frac{\lambda_r t}{l})}{l^2[J_1(\lambda_r)]^2} \tag{5.6}$$

where $J_1(.)$ represents the Bessel function of first order, $r = 1, 2, \ldots, R$, and the order of Bessel function is R. Bessel coefficients contain both magnitude and phase information and are real [165]. The relation between the index of Bessel coefficient r and the corresponding frequency of the signal f_r at which the maximum peak is achieved can be expressed as

$$f_r = \frac{r f_s}{2N} \tag{5.7}$$

where f_s is the sampling frequency and N is the number of samples in the duration l.

Representation of speech signal in terms of Bessel functions is effective in enhancing vowel-like regions by considering the appropriate range of Bessel coefficients [62]. The signal $s(t)$ can be bandpass filtered in the discrete range of Bessel coefficients (r_1,r_2) corresponding to the vowel region as computed by (5.7) using frequency range of the vowel region.

$$\hat{s}(t) = \sum_{r=r_1}^{r_2} C_r J_0(\frac{\lambda_r t}{k}) \tag{5.8}$$

The discrete version of the bandpass filtered signal $\hat{s}[n]$ is considered an AM-FM signal and its amplitude envelope is extracted using discrete energy separation algorithm. This amplitude envelope is smoothed using a moving average filter of 1 ms duration. A 100 ms size FOGD with 10 ms variance is convolved with the smoothed amplitude envelope from right to left to get the VEP evidence [62].

All the evidences are summed up and normalized by the sum's maximum value. To obtain the VOP locations, the convolution is performed from left to right. Figure 5.1(b) shows evidences from Hilbert envelope of LP residual for detection of vowel onset point (EVI-HE VOP) and endpoint (EVI-HE VEP) for the speech signal shown in Figure 5.1(a) with manually marked boundaries. Also, the corresponding evidences from ZFFS are shown (EVI-ZF VOP, EVI-ZF VEP). Figure 5.1(c) shows final evidences including the ones from Bessel representation and source, spectral peaks, and modulation spectrum energies for vowel onset (Final-EVI VOP) and endpoint detection (Final-EVI VEP). The peaks in Final-EVI VOP and Final-EVI VEP correspond to the detected VOPs and VEPs. The peaks in final evidences are more close to the manually marked boundaries compared to the peaks in other evidences. These multiple evidences provide reliable vowel change point locations from the speech signals.

5.2 Detection of syllable-like units using theta oscillator

Recently, an oscillator based on theta-rate neural oscillations in auditory cortex regions of brain was proposed for unsupervised spoken word discovery [58], which achieved a high word segmentation accuracy on multiple languages. These oscillations coincide well with the syllabic rate according to the speech perception studies [166]. Therefore, Räsänen et al. [58] proposed a damped harmonic oscillator to model the syllabic rate. The input to the oscillator is the amplitude envelope of speech and the minima in the amplitude of the oscillator represent the boundaries of the syllable-like

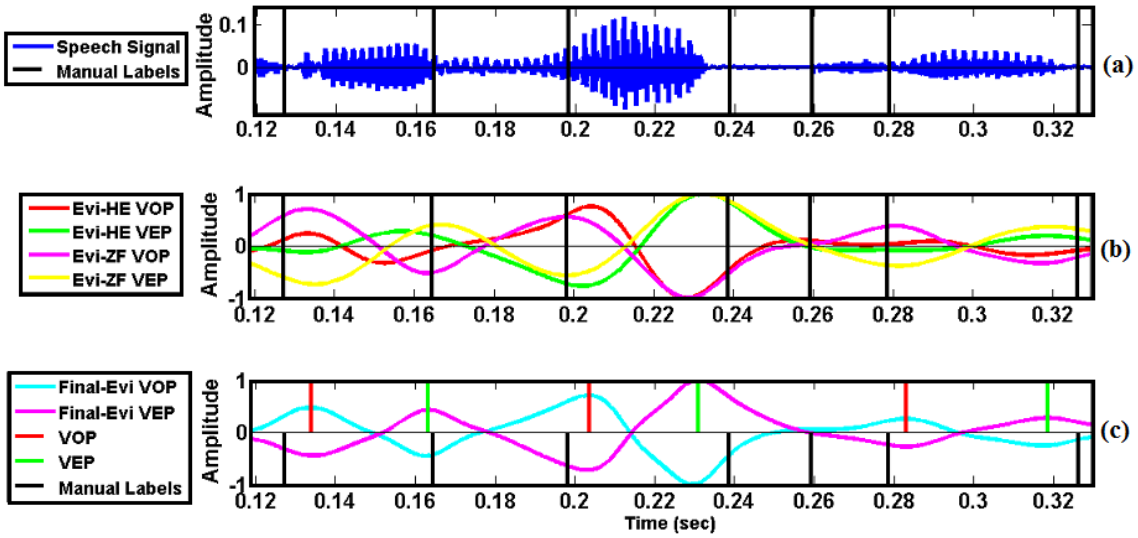


Figure 5.1: Evidences for vowel change points for an English utterance. (a) Speech signal and manual phonetic boundaries (b) Evidences for vowel onset and end points from Hilbert envelope of linear prediction residual and zero frequency filtered signal (c) Final evidences and locations of vowel onset and end points including Bessels features. Manual boundaries are shown in black.

units. The oscillator is modeled as

$$f(t) = e(t) - \frac{1}{f_s}x(t-2)v(t-1) - \frac{2\pi\Delta f}{f_s}v(t-2)f(t-1) \tag{5.9}$$

where, $e(t)$, $x(t)$, $v(t)$, $f(t)$ denote the amplitude envelope of the speech signal, amplitude, velocity and force of the oscillator, respectively. f_s denotes the sampling frequency and Δf is the bandwidth of the oscillator which is fixed to 8 Hz for critical damping.

The syllable-like units provide an alternate segmentation to the kernel-Gram based phonetic segmentation. Graph formation can be done using syllable-like units as nodes in place of virtual phones. Then, graph growing and graph clustering based segment labeling can be applied. The similar units will be assigned a unique label. The virtual labels obtained from syllable-like units can be used to train an acoustic model. Thus, syllable-like units form an alternate representation to virtual phones for zero resource speech processing.

5.3 Segment labeling of syllable-like units

Previous section described the segmentation of speech into syllable-like units using multiple evidences. These units are varying length segments with several instances oc-

curring within the speech data. The aim is to label each acoustically similar syllable-like unit uniquely. Therefore, all similar syllable-like units should be clustered into a single cluster assigned with a unique label. Here, a clustering approach based on k-means is proposed for labeling the syllable-like units.

The syllable-like units are varying in length and should be converted to appropriate fixed-dimensional representation in order to cluster them. A simple downsampling approach is used for obtaining the fixed-dimensional embeddings [1]. Each syllable-like unit is divided into a fixed number of segments and the mean feature vectors are obtained for all the frames in each segment.

The speech data is completely segmented into syllable-like units, which are transformed to fixed-dimensional embedding vectors after feature extraction. These embedding vectors corresponding to the syllable-like units are clustered into K classes. The current work provides an effective initial segmentation into syllable-like units in zero resource conditions. Therefore, the standard k-means can be applied on the embedding vectors directly making the clustering simpler and computationally efficient. The vectors are assigned to the cluster centroids and the means are updated alternately. The objective function is given by

$$\min_{S} \sum_{c=1}^{K} \sum_{x \varepsilon s_c} \left\{ 1 - \frac{x \mu_c^T}{\sqrt{x x^T} \sqrt{\mu_c \mu_c^T}} \right\} \tag{5.10}$$

where μ_c denotes the mean of the cluster c, S denotes the cluster assignments i.e., $s_c \varepsilon S$ represents that corresponding x belongs to the cluster c. A unique label is assigned to each cluster. If a new utterance is to be added, the clustering should be done again. Therefore, a seeded clustering approach is proposed, where a smaller portion of the speech data is clustered first to obtain the number of clusters and their distribution. When a new segment has to be added, the mean similarity of the corresponding fixed-dimensional embedding vector is computed for each cluster. The cluster which renders the highest mean similarity retains the new embedding vector. This provides a procedure for the online labeling of the segments.

The number of clusters is chosen to be 10% of the total syllable-like units detected in a language. The number of clusters is $3,700$ for Mandarin, $41,000$ for French and $60,000$ for English. Greenberg et al. [167] studied syllable structure of English and reported the total number of syllables in Switchboard corpus close to $50,000$. The number of clusters or the number of unique syllables in a corpus is directly related to the size of the corpus as larger amount of speech consists of more variety of words and corresponding syllabic units.

5.4 Spoken term discovery using syllable-like units

The syllable-like units as segment labels are evaluated on the STD task as described in Section 4.5. Firstly, we discuss the choice of the type of syllable-like units for this task.

5.4.1 Choice of syllable type for spoken term discovery

Since the goal of the STD task is to discover the word units from speech in an unsupervised manner, the choice of syllable for segment labeling has significant impact on the STD performance. Let C^* denote a group of consonants and V denote a single vowel. A single consonant or vowel is equivalent to a monophone and is similar to phonetic segmentation. Hence, the next possible larger units C^*V and VC^* are selected for STD. The type of syllables along with detected VOPs and VEPs and labels from manually marked boundaries for an English utterance are shown in Figure 5.2.

As evident from the position of the vowel, C^*V type of units are obtained from the VEPs and those of VC^* type are obtained from the VOPs. The performance of STD for Mandarin is evaluated for these syllable types and is reported in Table 5.1. It is observed from these results that the C^*V type of syllables outperform VC^* by a huge margin for all the measures. Also, lexical studies by Greenberg [167] on spontaneous speech in English for Switchboard corpus shows that C^*V type of syllables form much higher percentage of the corpus and its transcriptions compared to VC^* type of syllables. The type of syllable is fixed to be C^*V further in this work for all the languages without taking into consideration any phonetic knowledge to keep the system unsupervised or zero resource. Figure 5.2 illustrates the different types of syllable-like units formed from the detected vowel endpoints. Figure 5.3 shows the histograms of durations of syllable-like units for English, French and Mandarin. Mandarin shows a distinctly prominent Gaussian distribution over durations of C^*V types of units. The mean and standard deviations for the durations of these units for all the languages are quite similar despite the outliers in case of English and French.

5.5 Syllable-like units for spoken term discovery

The evaluations are carried on the Zero Resource Speech Challenge 2017 [17]. The data consists of Mandarin, French and English with 2.5 hours, 24 hours and 45 hours of total audio duration, respectively. The proposed approach is compared with ES-KMeans, PES-KMeans and baseline results from the challenge.

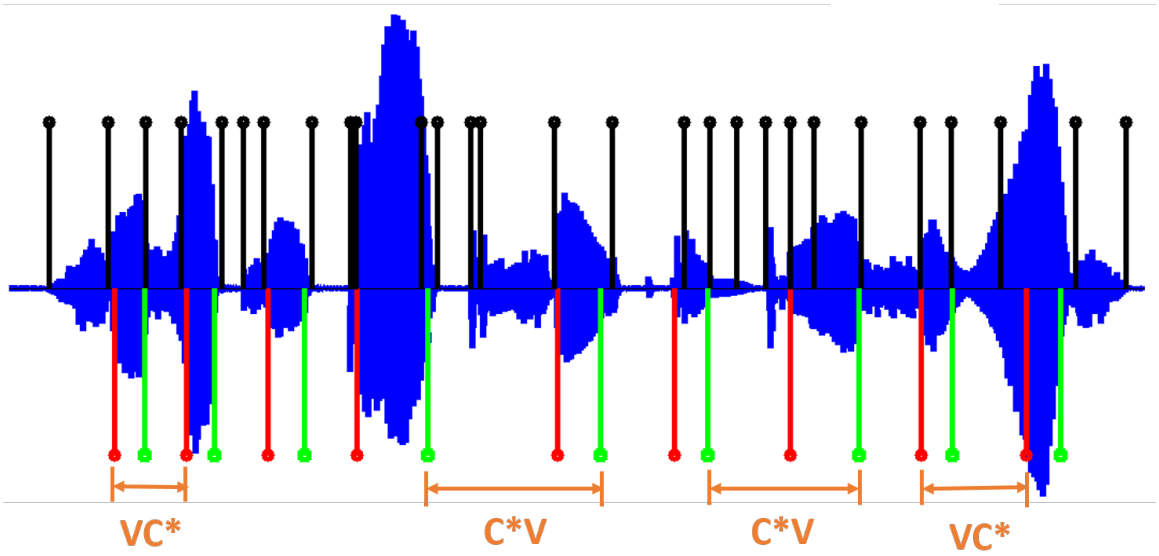


Figure 5.2: An illustration of detected vowel endpoints for an English utterance. The manual phonetic boundaries, vowel onset and end points are shown in black, red and green color respectively. Different types of syllable-like units are shown below.

Table 5.1: Performance comparison of the proposed syllable-like units based approach for different syllable types for Mandarin data of Zero Resource Speech Challenge 2017

Syllable type	NLP		type			token			boundary		
	NED	Cov	P	R	F	P	R	F	P	R	F
C^*V	**80.6**	**116.8**	**10.8**	**14.3**	**12.3**	**10.5**	**19.9**	**13.8**	**46.4**	**78.7**	**58.4**
VC^*	98.6	44.4	3.2	2.3	2.7	2.2	1.8	2.0	31.0	22.8	26.2

The coverage results for all the languages are close to 100%, which indicates that the proposed syllabic-units based clusters are able to cover the complete corpus as shown in Table 5.2. The baseline system provides the best NED scores for Mandarin, French, and English but its coverage is extremely low. The proposed method gives the best performance on type, token and boundary scores for Mandarin and French, whereas ES-KMeans provides the best performance on these metrics for English. PES-KMeans also provides better performance than the proposed approach for English. The reason for this lies in the rhythmic structures of these languages. English is a stress-timed language with a large range of syllabic structures and the speakers do not identify the syllable boundaries clearly [168]. French and Mandarin are syllable-timed rhythmic languages [169] which have more regular syllabic structures compared to stress-timed languages.

Table 5.2: Zero Resource Speech Challenge 2017: Baseline system, ES-KMeans, phoneme based ES-KMeans and the proposed syllable-like units based approach results

Language	System	NLP		type			token			boundary		
		NED	Cov	P	R	F	P	R	F	P	R	F
Mandarin (2.5 hours)	Baseline	**30.7**	2.9	4.5	0.1	0.2	4.0	0.1	0.1	37.5	0.9	1.8
	ES-KMeans	88.1	100	2.5	4.1	3.1	2.5	3.4	2.9	36.5	47.1	41.1
	PES-KMeans	80.0	**117.5**	7.7	10.4	8.8	6.9	11.5	8.7	43.8	66.8	52.9
	Proposed	80.6	116.8	**10.8**	**14.3**	**12.3**	**10.5**	**19.9**	**13.8**	**46.4**	**78.7**	**58.4**
French (24 hours)	Baseline	**25.4**	1.6	**6.9**	0.2	0.3	5.2	0.1	0.1	30.9	0.6	1.1
	ES-KMeans	67.3	97.2	3.1	6.3	4.2	3.5	3.9	3.7	**37.8**	41.6	39.6
	PES-KMeans	68.1	**97.5**	4.2	7.9	5.5	4.8	7.6	5.9	25.4	38.4	30.6
	Proposed	70.4	96.3	6.4	**11.7**	**8.3**	**9.7**	**20.8**	**13.3**	30.2	**59.1**	**40.0**
English (45 hours)	Baseline	**30.7**	2.9	4.5	0.1	0.2	4.0	0.1	0.1	37.5	0.9	1.8
	ES-KMeans	72.6	100	**8.3**	**16.7**	**11.1**	**13.0**	**14.1**	**13.5**	**51.0**	**54.4**	**52.7**
	PES-KMeans	72.2	**100.9**	4.5	9.4	6.1	5.0	8.2	6.2	26.4	41.2	32.2
	Proposed	75.5	99.5	3.1	7.2	4.3	3.9	7.8	5.2	23.4	43.9	30.5

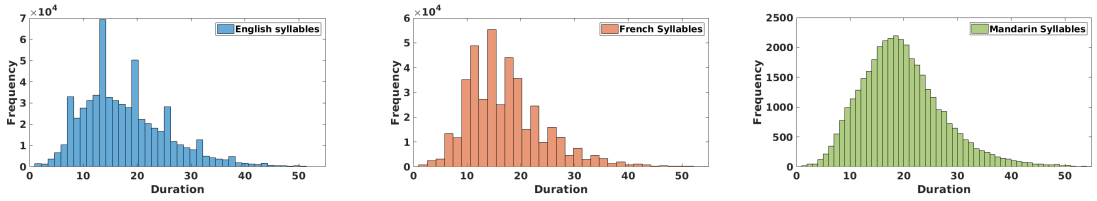


Figure 5.3: Histograms of C^*V type of syllable-like units for English, French and Mandarin. The duration is in terms of frames (normalized by 10 ms). Statistics : English - Mean= 17.78, STD= 8.18 French - Mean= 17.02, STD= 7.59, Mandarin - Mean= 19.23, STD= 7.77

5.6 Summary

In this chapter, zero resource methods for detecting boundaries of syllable-like units are proposed. Multiple evidences from excitation and source information from the speech production model along with the evidences from Bessel feature based representations are used for detecting vowel end points and in turn the boundaries of syllable-like units. A recent zero resource syllabification algorithm based on theta-rate oscillations at the syllabic rate is also described. These syllable-like units can be used as ASM units instead of virtual phones to cluster the data into homogeneous segments and for zero resource speech applications. Syllable-like units are more linguistically closer to actual word units and could be effective in tasks like spoken term discovery.

This chapter also emphasizes the importance of syllable-like units for automatic word discovery from speech. Evidences derived from different sources and representations of speech signals are used to obtain vowel-like regions. These boundaries

are used to form different syllabic patterns which are clustered into few number of classes of syllable-like units. Speech signals are parsed using these clusters and are evaluated for the STD task from Zero Resource 2017 Challenge. The evaluations suggest that the proposed syllable based approach clearly outperforms other techniques for syllable-timed rhythmic languages like French and Mandarin. However, other techniques work well for stress based language like English. These conclusions are based on the experimentation which is zero resource i.e., free from the usage of any transcriptions and also do not make use of any prior phonetic knowledge or syllabic structures in a language.

The future work could be towards combining phonetic and syllabic segmentation methods for better STD performance. Another interesting direction could be to come up with a technique to use all different syllable types together for the STD task which is quite complex given only phone and/or vowel boundaries in a zero resource scenario.

Chapter 6

Applications of zero resource speech processing

Several speech applications do not require manual labels if virtual labels are available. Three zero resource applications are discussed in this chapter, namely - language identification, speaking rate estimation and low bitrate coding. Language identification is based on the co-occurrence statistics of virtual phones. Speaking rate estimation is based on the segmentation of speech into syllable-like units. Low bitrate coding involves signal to symbol transformation to convert speech signals into sequence of virtual labels. These virtual labels are encoded at low bitrates and are used to re-synthesize the speech signals in target speaker's voice. The applications are discussed in the following sections.

6.1 Phonotactic language identification using virtual phones

The process of the unsupervised speech signal to symbol transformation has been described in the previous chapters. This technique provides an effective means to obtain automatic transcriptions for unlabeled speech data in terms of virtual phones. Here, we propose to use these automatic transcriptions for language identification (LID) in zero resource settings. To check the effectiveness of labels, we compare the performance of the above system with LID built-in supervision of manual labels. The rest of this chapter discusses a phonotactic approach to LID using the unsupervised signal to symbol transformation followed by experiments and results.

The phonotactic information provides an effective basis to discriminate between languages and to identify them correctly. Most common phonotactic LID systems are

based on Phone Recognition followed by Language Modeling (PRLM) [78]. PRLM approach uses a phone recognizer trained on a single language for tokenization of speech into a sequence of phonemes [170]. Individual N-gram language models $(\alpha_1, \alpha_2, ..., \alpha_R)$ are trained corresponding to each target language $(L_1, L_2, ..., L_R)$. A probabilistic framework is used for identifying the spoken language from speech. The task is to detect the language given an observation or feature vector corresponding to an utterance.

During the testing phase, a given utterance is tokenized into a sequence of phonemes $\mathbf{O} = (o_1, o_2, ...o_P)$ by trained phone recognizers where P is the length of the decoded phoneme sequence. Log-likelihood score for each language is computed for a test utterance as

$$\mathcal{L}(\mathbf{O}|\alpha_r) = \sum_{i=1}^{P} log(\mathcal{P}_{\alpha_r}(o_i|o_{i-1}, ..., o_{i-(N-1))})),\ 1 \leq r \leq R \tag{6.1}$$

where, α_r is the N-gram language model for language r and R is the total number of languages in the system.

Then, the most likely language is identified as

$$\hat{L} = \underset{1 \leq r \leq R}{\arg\max} \mathcal{P}(\mathbf{O}|\alpha_r) \tag{6.2}$$

A single language phone recognizer may not be able to incorporate all sounds from the prospective languages. Therefore, multiple language-dependent phone recognizers are used in Parallel Phone Recognition followed by Language Modeling (PPRLM) approach [78]. This approach involves tokenization using separate phone recognizers for each language and then modeling thus obtained token sequences. Based on the availability of labeled training data, more sophisticated strategies for LID have been developed such as Parallel Phone Recognition (PPR) which integrates acoustic and phonotactic models. Such a system makes use of language-specific phonotactic constraints during the decoding process in the phone recognizer, thus producing optimal phone sequence [89].

Typically, phone recognizers are trained using manually labeled data which is very hard to obtain. We propose a new unsupervised PPR based LID where each phone recognizer is trained on virtually transcribed data obtained by ASM process as described in Sections **??**, 4.2 and 4.4. Acoustic segment modeling can label speech data in an unsupervised manner. The proposed unsupervised PPR based LID system is shown in Fig. 6.1. The first block is feature extraction module where MFCC features are extracted for any given utterance. The second level of blocks represents

ASM for automatically transcribing the speech for each language using the extracted features. The third level of blocks corresponds to integrated phone recognizers and bi-gram language models built using virtual phones for each of the language. The last block represents a classifier to decide the language to be identified.

During the training phase, utterances from each language in the training set are first tokenized into a sequence of virtual phones by the ASM tokenizer for that particular language. These virtually transcribed labels are used to train the phone recognizers and language models for each language. In the testing phase, a test utterance is decoded through phone recognizers using corresponding language models for every language. Finally, the log likelihood scores for each language are obtained and are used to identify the most likely language.

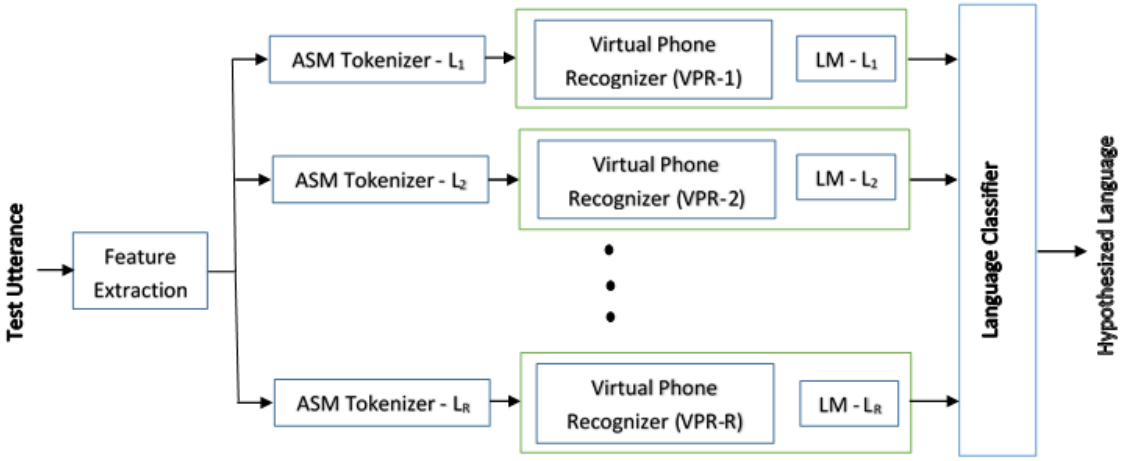


Figure 6.1: *Unsupervised Parallel Phone Recognition based LID*

The posterior probability of language, $\mathcal{P}(l_r|\mathbf{O})$, using Bayes theorem can be converted to a function of language priors, $\mathcal{P}(l_r)$ and the probability of the observation coming from a given language, $\mathcal{P}(\mathbf{O}|l_r)$.

$$\mathcal{P}(l_r|\mathbf{O}) = \frac{\mathcal{P}(\mathbf{O}|l_r)\mathcal{P}(l_r)}{\sum_{i=1}^{R} \mathcal{P}(\mathbf{O}|l_i)\mathcal{P}(l_i)}, \; 1 \leq r \leq R \tag{6.3}$$

where O_r is a token sequence obtained using the r^{th} phone recognizer. Summation over all possible token sequences is impractical. Approximate calculation of the sum can be done by finding the most likely token sequence. For each model, the most likely token sequence is calculated by doing a Viterbi search over the tokenizer.

$$\hat{O}_r = \underset{\mathbf{O} \in Z_r}{argmax} \mathcal{P}(\mathbf{O}|\alpha_r) \tag{6.4}$$

where Z_r is the set of all possible token sequences that can be generated using the r^{th} model. The model consists of both acoustic and language models. The decoder considers both language and acoustic model while searching for the most likely sequence. The language prior, $\mathcal{P}(l_r)$, is kept same for every language. Hence, the modified likelihood is given by

$$\mathcal{P}(l_r|\mathbf{O}) = \frac{\mathcal{P}(\hat{O}_r|l_r)}{\sum_{i=1}^{R} \mathcal{P}(\hat{O}_i|l_i)},\ 1{\leq}r{\leq}R \tag{6.5}$$

The decision is made in favour of the language yielding maximum probability.

$$\hat{L} = \underset{1\leq r\leq R}{\arg\max} \mathcal{P}(l_r|\mathbf{O}) \tag{6.6}$$

6.1.1 Language identification experiments

The experiments were performed on eight Indian languages: Gujarati, Hindi, Kannada, Malayalam, Manipuri, Punjabi, Telugu and Urdu. The dataset used for the experiments is a subset of the Indian languages database described in [171]. The amount of data for each language used in training and testing phase is approximately 2 hours and 1 hour respectively. This choice is made to keep the low resource setting throughout the experiments. The languages selected are quite diverse with respect to phonotactics. Table 6.1 shows the number of speakers for training and testing including male and female speakers.

Language	#Spk_Train	#Spk_Test
Gujarati	46	17
Hindi	28	8
Kannada	10	2
Malayalam	7	6
Manipuri	7	5
Punjabi	4	2
Telugu	23	20
Urdu	46	8

Table 6.1: Database description

The proposed ASM technique is used for tokenizing the utterances and training phone recognizers using these tokens. We also trained phone recognizers with annotated corpora. Standard monophone HMM based phone recognizers and bi-gram language models were trained for each language. The phone models are left-to-right

HMMs with Gaussian mixture observation densities. Each Gaussian mixture model consists of 32 Gaussian components. We used 39-dimensional MFCCs comprising of 13 cepstral coefficients and their first and second order derivatives. Viterbi decoding for each test utterance was done using each trained phone recognizer. Per frame log-likelihood scores were computed for each decoded phone. These scores were processed and normalized according to (6.5). Finally, a decision is made to select the most likely language based on the normalized likelihood scores using (6.6). This system trained using virtual labels from the proposed ASM is referred to as unsupervised parallel phone recognition (UPPR) based LID.

An i-vector based language identification system is built for comparison [80]. The proposed UPPR and i-vector based LID systems are completely unsupervised with zero resource requirement for manual transcriptions. The scores from these two unsupervised systems are fused to provide stronger evidence for identifying the language without any labels.

6.1.2 Language identification results

Equal error rate (EER) [172] is used as performance measure for comparing the proposed UPPR approach with the other methods. The UPPR model uses a bigram language model built with virtual phones. To see the effect of this virtual language model on the performance of the LID, we conducted the experiments without the language model.

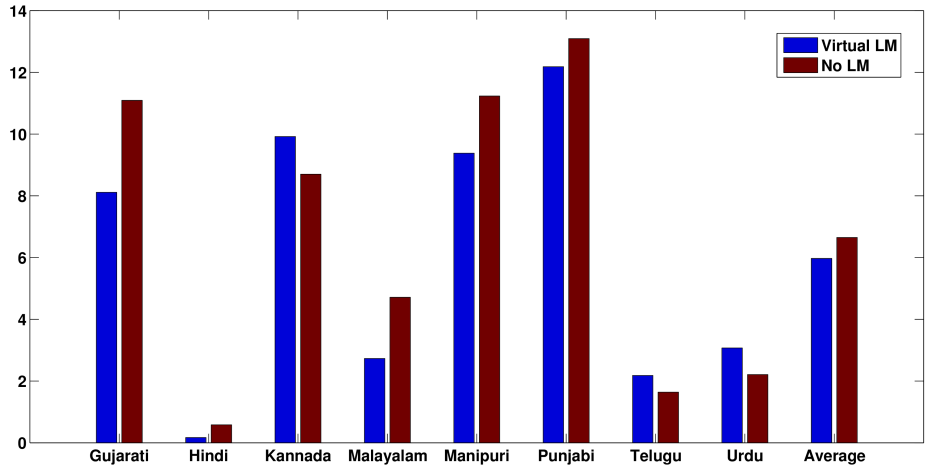


Figure 6.2: Equal error rates for no language model and language model built using virtual phones

Fig. 6.2 shows the EERs of individual languages with no language model and with

virtual language model. The model with no language model gave an average EER of 6.65% for these eight languages where as system with virtual language model had an EER of 5.97%. Thus, virtual language model provided a relative improvement of 10.22% over no language model. This shows the impact of co-occurrence statistics of virtual phones on the performance of the LID, indicating that the sequence of virtual phones indeed captured the language structure. The UPPR system mentioned hereafter uses language models trained on virtual phones. The supervised system used language models trained on actual phones.

Language	EER UPPR	EER i-vector	EER Combined	EER Supervised
Gujarati	8.11	10.29	5.95	7.52
Hindi	0.17	2.40	0.17	0.16
Kannada	9.92	7.12	4.89	11.57
Malayalam	2.73	9.41	2.26	2.07
Manipuri	9.38	0.68	1.55	6.24
Punjabi	12.18	6.14	12.18	11.79
Telugu	2.18	1.44	0.53	1.23
Urdu	3.07	5.03	1.59	1.72
Average	5.97	5.31	3.64	5.29

Table 6.2: EER comparison for UPPR, i-vector, their fusion and supervised systems.

Table 6.2 summarizes the performance comparison of the proposed algorithm with the supervised and i-vector approaches. UPPR achieved average EER of 5.97% which is comparable to the EERs of i-vector approach i.e., 5.31% and supervised approach i.e., 5.29%. This is intuitive that the supervised approach gives slightly better performance than both unsupervised approaches due to the use of actual language models and manual transcriptions. Scores from UPPR and i-vector based unsupervised approaches are fused using Bosaris toolkit [173]. A linear fuser is trained with the same in-set scores to obtain optimized weights of the convex combination of the two score sets. These optimized weights are used to obtain the final fused scores. The EER from this combined system is 3.64% which provides 31.19% relative improvement over supervised approach.

6.2 Zero resource speaking rate estimation

Speaking rate is an important attribute of the speech signal which plays a crucial role in the performance of automatic speech processing systems. The main factors influencing the speaking rate are learned behavioral characteristics of the speaker,

mode of speech, and the emotion of the speaker.

6.2.1 Speaking rate estimation from syllable-like units

In this work, we propose to estimate the speaking rate by segmenting the speech into syllable-like units using end point detection algorithms which do not require any training and fine-tuning. Also, there are no predefined constraints on the expected number of syllabic segments. The syllable-like units are obtained only from speech signal to estimate the speaking rate without any requirement of transcriptions or phonetic knowledge of the speech data. A recent theta-rate oscillator based syllabification algorithm is also employed for speaking rate estimation. The performance is evaluated on TIMIT corpus and spontaneous speech from Switchboard corpus. The correlation results are comparable to recent algorithms which are trained with specific training set and/or make use of the available transcriptions.

The multiple evidence based approach described in Chapter 5 is used to detect the VEPs, and the region between two successive VEPs is considered as a syllable-like unit of C^*V-type, where C^* denotes a non-vowel like region usually consisting of a single or a group of consonants and V denotes a single vowel.

The number of detected syllable-like units per second is used to quantify the speaking rate. A recent approach based on theta-rate oscillations [174] to detect boundaries of syllable-like units was proposed for unsupervised word discovery [58]. This approach is also used to compare with the multiple evidence based approach in the zero-resource settings. The SRE evaluations are done on TIMIT and Switchboard corpus in terms of correlation between the actual and the estimated number of units and the speaking rate.

The earlier works in SRE defined the speaking rate as the number of syllables per second or the number of phones per second in a given segment of speech. A syllable is a subword linguistic unit consisting usually of a vowel as a nucleus with a preceding onset and a succeeding coda, both are optional and are generally consonants. Jiao et al. [112] exploited this almost certain presence of a vowel in a syllable to reformulate the SRE problem as equivalent to estimating the number of vowels per second in a segment of speech. The authors posed the SRE as a convex optimization problem in which an optimum weighting function has to be determined for the features derived from the speech segment to estimate the number of vowels per second in that segment. In this work, we use signal processing methods to detect VEPs which are used as anchor points to identify the syllable-like units. Unlike the referred work, our approach does not require any labeled speech data for training.

The combined evidences as described in 5.1 provide very reliable and stronger

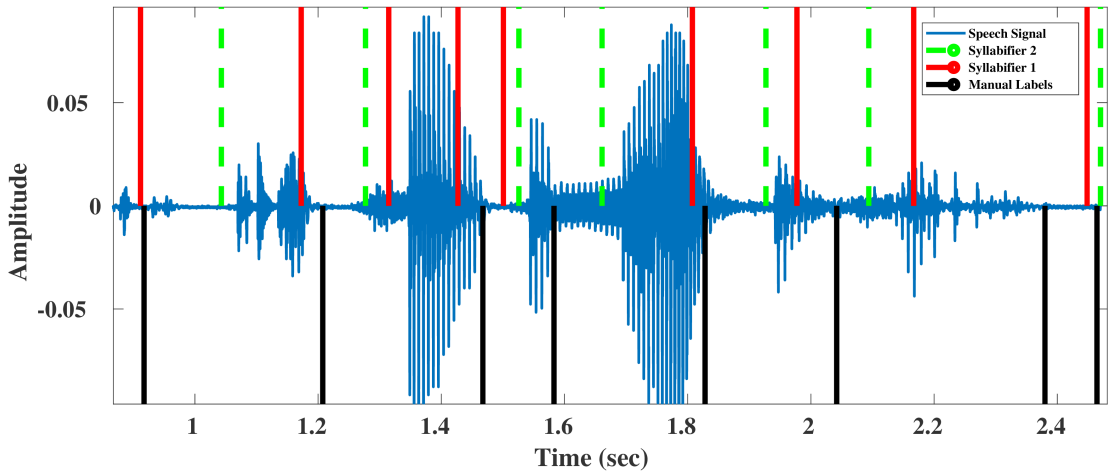


Figure 6.3: A Switchboard utterance with manually marked labels (black) and labels from Syllabifier-1 (red) and Syllabifier-2 (green).

VEPs obtained in conjunction from different sources. The VEPs directly provide the estimate of the number of vowels and in turn the number of syllables in a given segment, the syllable-type being C^*V between consecutive VEPs. This multiple evidence based method for vowel end point detection is further referred to as Syllabifier-1.

The theta-oscillator based method described in 5.2 was originally proposed for unsupervised word discovery from speech. It is computationally efficient, simple and unsupervised. The oscillator is tuned to match the rhythm of syllables. Therefore, in this work we use this method for comparison against the proposed method for SRE. It is referred to as Syllabifier-2 in the rest of the paper. Figure 6.3 shows a Switchboard utterance segmented into syllable-like units by Syllabifier-1, Syllabifier-2 and the corresponding manual boundaries.

6.2.2 Evaluation on TIMIT corpus

The TIMIT test set consists of 1680 sentences on which all the results are reported [143]. The results are compared with intensity based Praat script (Praat) [111], the sub-band and temporal correlation-based method (Sub-band Corr) [110], the GMM based method (GMM) [109], the convex weighting criteria method (Convex OPT) [112]. Sub-band Corr uses TIMIT training set for Monte-Carlo training as in [110]. GMM based model is also trained using the same training set. The Convex OPT method is shown to be dependent on the number of training sentences with speaking rate error reducing almost monotonically with increase in the number of training sentences [110]. Also, the weighting vectors are speaker adapted using a sentence from the test set for

each speaker for achieving further improvements.

The Praat script was directly evaluated on TIMIT test set. The syllabifier-1 (proposed) and syllabifier-2 also do not require any training and are evaluated on the test set directly. Further, there is no parameter fine-tuning or cross-validation done using any labeled data in terms of phones/syllables to keep the methods completely zero resource.

The evaluation metrics are the correlation between the actual and the estimated number of vowels, absolute mean error and the corresponding standard deviation (Stddev error), speaking rate (SR) error rate defined by absolute difference between the actual and the predicted vowels normalized by the actual number of vowels, SR mean and stddev error computed from the absolute difference between actual and predicted SR [112]. Table 6.3 shows the results on TIMIT test set. The correlation for Syllabifier-1 and Syllabifier-2 was found to be comparable to all the methods except Praat which gives relatively higher SR error rate compared to other methods. The mean error, the stddev error and the SR error rate for Syllabifier-1 is better than almost all methods except Convex-OPT which is speaker adapted using test utterances. SR mean error and SR stddev error are also comparable to other methods. This shows that zero resource syllabifiers perform on par with the state-of-the-art on TIMIT without any parameter tuning.

6.2.3 Evaluation on Switchboard corpus

Spontaneous speech consists of inconsistent number of pauses with varied duration and spoken phrases. This makes speaking rate estimation a difficult task for spontaneous discourse. The syllabification algorithms are evaluated on ICSI Switchboard corpus subset with 5564 spontaneous speech utterances which have syllable based manual transcriptions [175]. Acoustically based methods enrate, sub-mrate and mrate are compared which do not require any manual transcriptions for training [107]. The results are also compared with a broad phonetic class recognizer (Broad Class) [115] trained on SCOTUS corpus consisting of large number of tokens for each phonetic class. The methods enrate, sub-mrate, enrate, Sub-band Corr use the pause and noise

Table 6.3: Speaking rate estimation results for TIMIT test set

Method	**Correlation**	**Mean error**	**Stddev error**	**SR error rate%**	**SR mean error**	**SR stddev error**
Pratt	0.890	1.93	1.38	15.4	0.639	0.49
Sub-band Corr	0.830	1.82	1.48	15.0	0.610	0.40
GMM	0.805	1.61	1.41	14.0	0.528	0.41
Convex-OPT	0.869	1.39	1.24	12.2	0.462	0.36
Syllabifier-1 (Proposed)	0.854	1.60	1.39	13.2	0.537	0.44
Syllabifier-2	0.840	1.98	1.58	15.5	0.662	0.51

Table 6.4: Syllable count correlation and statistics for switchboard spontaneous speech

Method	**Correlation**	**Mean error**	**Stddev error**
Convex-OPT	0.971	1.30	1.31
Syllabifier-1 (Proposed)	0.970	1.42	1.59
Syllabifier-2	0.960	1.84	1.82

Table 6.5: Syllable rate correlation and statistics for switchboard spontaneous speech

Method	**Correlation**	**Mean error**	**Stddev error**
enrate	0.415	0.747	1.405
sub-mrate	0.637	0.530	1.219
mrate	0.671	0.464	1.121
Convex-OPT	0.744	0.600	0.490
Sub-band Corr	0.745	0.339	0.796
Broad Class	0.763	-0.161	0.780
Syllabifier-1 (Proposed)	0.655	0.639	0.668
Syllabifier-2	0.517	0.932	0.830

labels in the manual transcriptions to split the utterances into spurts. Convex-OPT, Sub-band Corr and Broad Class use utterances for training/development set in some form or the other. Syllabifier-1 and Syllabifier-2 are completely zero resource methods and do not use any training/development set. Both these methods also do not use spurts obtained using transcriptions.

Table 6.4 shows the correlation between the actual and the estimated syllable counts and the mean and standard deviation between the absolute error between the two. The correlation and other statistics for Syllabifier-1 are comparable to Convex-OPT which is a training based method whereas Syllabifier-1 and Syllabifier-2 are directly evaluated on entire Switchboard corpus.

Table 6.5 shows the correlation between the actual and the estimated syllable based speaking rate and the mean and standard deviation between the corresponding absolute error. The correlation for Syllabifier-1 is comparable to the other zero resource methods and has lesser stddev error compared to the acoustically based methods. The correlations of all the training based methods Convex-OPT, Sub-band Corr, Broad Class are higher than other methods. But the zero resource methods are more generic and can work on any database or language without any fine tuning under new or unknown settings. This can also be inferred by the fact that the zero resource methods are evaluated as it is on both TIMIT and Switchboard corpus without any specific training/fine-tuning for either of the corpus.

6.3 Virtual phone recognition/synthesis for ultra low bitrate coding

This application is based on ZeroSpeech 2019 challenge : TTS without T [117]. The challenge requires building speech synthesis systems without using text or linguistic information only from speech signals. This involves spoken term discovery for discovering acoustic subword units in an unsupervised manner. The speech signals should be re-synthesized using these units.

The objective of this work is to achieve ultra low bitrate coding of speech signals by transforming them into a sequence of virtual phones. Virtual phone units are discovered automatically from the given speech signals in an unsupervised manner. The given speech signal is initially segmented into acoustically homogeneous segments using the kernel-Gram segmentation discussed in Chapter 4. These segments are then clustered using different clustering techniques. These cluster labels are considered as virtual phone units which can be used to transcribe any given speech signal. This approach has been shown to perform well on spoken term discovery task on ZeroSpeech 2015 and 2017 data [176, 177].

The virtual phones for the sentences to be synthesized are encoded as one-hot vector sequences. Deep neural network based duration model and acoustic model are trained for synthesis using these sequences. A vocoder is used to synthesize speech in target speaker's voice from the features estimated by the acoustic model.

Ultra low bitrate coding is achieved as speech signals at higher bitrate are converted into discrete virtual phone sequences which can be coded as one hot vectors in ultra low bit rates. A symbol to signal synthesizer can re-synthesize the speech signal in a target speaker's voice. The ultra low bitrate coding results in relative degradation of intelligibility.

Since, the target classes are virtual phone units, the number of classes is small, close to the number of phonetic units present in a language. The virtual phone sequences are converted into one hot vector encoding which are ultra low bitrate representations of speech signals. These sequences are used to re-synthesize the speech signals using a deep neural network (DNN) based speech synthesis system [178].

The performance evaluation is done on ZeroSpeech 2019 challenge on English and Indonesian language. The bitrate and speaker similarity were found to be better than the ZeroSpeech 2019 baseline with slightly lower intelligibility due to the compact encoding.

6.3.1 Virtual phone recognition from speech signals

The signal to symbol transformation approach described in Chapter 4 is employed for virtual phone recognition from speech signals. The process flow of virtual phone recognition/synthesis for ultra low bitrate coding is shown in Fig. 6.4 and Fig. 6.5. Stage 1 consists of speech segmentation, Stage 2 consists of segment labeling and Stage 3 is synthesis using virtual labels.

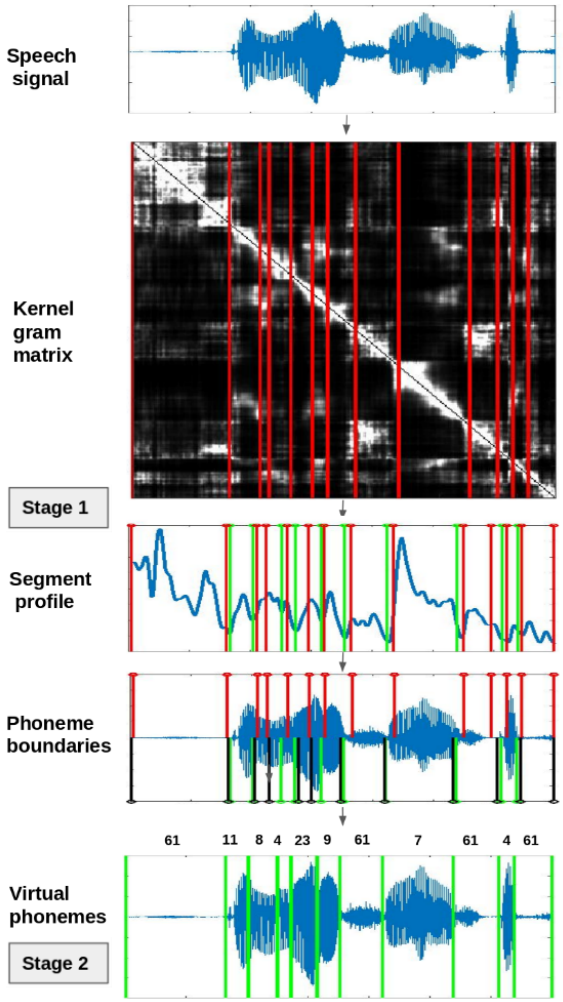


Figure 6.4: Process flow for virtual phone recognition. Stage 1 represents kernel-Gram segmentation and Stage 2 represents segment labeling using different clustering techniques. Manually marked boundaries are represented through red vertical lines and detected boundaries from proposed segmentation using MFCC (AE-BN) features are shown as green (black) vertical lines.

Two different features are used for kernel-Gram segmentation: MFCC and autoencoder bottleneck (AE-BN) features. The AE-BN features are derived with a 5 hidden layer DNN with 32-dimensional bottleneck layer. The input to the autoencoder has

9 frames context (four left, current, four right). In Fig. 6.4, : Phoneme Boundaries, we can observe the segment boundaries from kernel-Gram segmentation using MFCC and AE-BN features respectively are close to the manually marked boundaries in red. AE-BN features provided better segmentation compared to MFCC features. The next task is to cluster these segments.

For clustering the segments and in order to create a lexicon of virtual phone classes, the fixed dimensional representation should be obtained for varying length segments. In this work, feature vector of concatenated sub-segment means is used as the fixed dimensional representation [179]. Further, three different methods are used for clustering - K-means, spectral clustering [67] and embedded segmental K-means(ES-KMeans) [59].

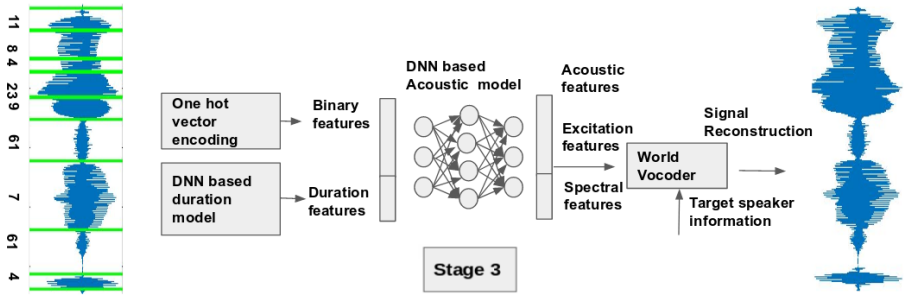


Figure 6.5: Process flow for virtual phone based synthesis

6.3.2 Speech synthesis from virtual labels

To build TTS without text, virtual phones units are obtained by the virtual phone recognition discussed above. These virtual phones are used to train a DNN based synthesis system. Merlin [178] open source toolkit is used in this work to build a neural network based TTS system. Fig. 6.5 shows the block diagram of Stage 3: virtual phone based synthesis. A DNN based duration model is trained to predict duration features from virtual labels. HMM based forced alignment is done to align speech signals with virtual labels and obtain durations to train the DNN based duration model. The virtual labels are converted to one-hot vectors to get binary features. These binary features and duration features are concatenated to form front-end features to train a DNN based acoustic model to predict the acoustic features.

These features are used to synthesize the speech signal using WORLD vocoder [180]. The target speaker information is provided to synthesize the speech signals in the desired voice. The vocoder estimates the spectral envelope, fundamental frequency, aperiodicity and generates speech with these estimated parameters.

6.3.3 Evaluation metrics

Evaluation metrics of ZeroSpeech 2019 measure the quality and the accuracy of re-synthesized speech and the intermediate discrete symbol representation. A brief description of these metrics are given here for the analysis of the results [117].

Embedding metrics

1. **Bitrate:** The entire test set audio is expressed as a sequence of vectors S of length N: S=[s_1,s_2,s_3...,s_N]. The bitrate for S is given by,

$$B(S) = \frac{N.\sum_{i=1}^{N} P(s_i)\log_2 P(s_i)}{D} \tag{6.7}$$

   where P(s_i) is the probability of symbol s_i. $B(S)$ provides the number of bits required to transmit discrete symbols. D is the duration of the audio files. Lower the bitrate, the better.

2. **ABX Score:** Discovered embedding correspond to linguistic units such as syllables, phonemes, etc. ABX discriminability provides the probability that A and X are closer than B and X, if A and B are same. ABX score ranges form 0 to 100, the best being 0 for the gold transcriptions.

Synthesis quality and accuracy metrics

1. **MOS:** It provides the overall quality of the synthesized audio files on a 1 to 5 scale, bigger score is better.

2. **Speaker similarity:** Speaker similarity indicates that whether the re-synthesized utterance is closer to the source or the target speaker. The range of speaker similarity score is from 1 to 5, the bigger score indicates that the re-synthesized file has the generated voice closer to the target voice.

6.3.4 Experimental results

Database

ZeroSpeech 2019 challenge consists of English and Indonesian with nearly 15 hours of data for acoustic unit construction for both languages from approximately 100 speakers each. English train voice dataset contains one male speaker 2 hrs and one female speaker 2 hours 40 mins. Indonesian train voice data contains one female speaker 1 hour 30 mins. This train voice data is used to build synthesis system. Test

dataset contains 28 min with 24 speakers in English and 29 min with 15 speakers in Indonesian dataset.

Impact of clustering methods

Initially, we built our systems with English data for finding the best hyper parameters. Then, we used the same hyper parameters for the Indonesian data set. For Kernel-Gram segmentation, System 1 uses 39 dimensional MFCC features (including Δ and $\Delta\Delta$ features) with 5ms frame shift and System 2 uses 32 dimensional AE-BN features. Kernel width is fixed to 1 for all the experiments.

Table 6.6: Performance evaluation of different clustering methods for virtual phone recognition/synthesis for English from Zero Resource Speech Challenge 2019.

Metric	Clustering Method		
	K-means	Spectral	ESK-means
Bitrate	**37.22**	50.74	41.23
ABX	47.41	**45.12**	46.47

Table 6.6 shows the clustering results for English and Indonesian data set. K-means clustering with AE-BN features provided lowest bitrate whereas ABX results were in close range for all clustering methods. Both the systems submitted to the challenge use K-means clustering.

Table 6.7: Performance comparison of the virtual phone recognition/synthesis approach for two different systems using MFCC (System 1) and AE-BN features (System 2) for English and Indonesian data from Zero Resource Speech Challenge 2019. Up arrows indicate that the higher is better and down arrows indicate that the lower is better.

Model / Metric	English				Indonesian			
	Baseline	System 1	System 2	Topline	Baseline	System 1	System 2	Topline
Bitrate ↓	71.98	52.21	**37.22**	37.73	74.55	46.07	44.07	**35.2**
ABX ↓	35.63	45.54	47.41	**29.85**	27.46	40.17	45.64	**16.09**
MOS ↑	2.5	2.18	1.84	**2.99**	2.07	1.82	1.44	**3.92**
Similarity ↑	2.97	**3.01**	2.59	2.77	3.41	3.3	3.02	**3.95**

ZeroSpeech 2019 challenge evaluation

Table 6.7 shows the results of the two systems submitted to the challenge [1]. The baseline system uses variational inference for unit discovery followed by Merlin based

[1]The complete output of the proposed system and samples of synthesized files can be found here: https://zerospeech.com/2019/results.html

synthesis [94] using these units. Topline system is trained using Kaldi toolkit [181] for recognition followed by Merlin for synthesis using gold transcriptions. Our systems performed better than the baseline and most of the other submissions and in terms of bitrate for both English and Indonesian.

System 1 outperformed the topline in terms of speaker similarity which shows that our synthesis results are closer to the target speakers' voices. MOS and ABX for both the systems were lower than the baseline. Slightly lower synthesis quality can be attributed to very lower bitrates. The bitrate for System 2 is 37.22, which is very close to the topline bitrate 37.73, which shows that the number of virtual labels produced by this system are very close to the number of manually marked labels. Fig. 6.6 shows the speaker similarity versus bitrate for all the systems submitted to the challenge and the baseline and topline systems. It is evident from the figure that the System 1 and the System 2 are in the region of highest similarity and lowest bitrate for both English and Indonesian.

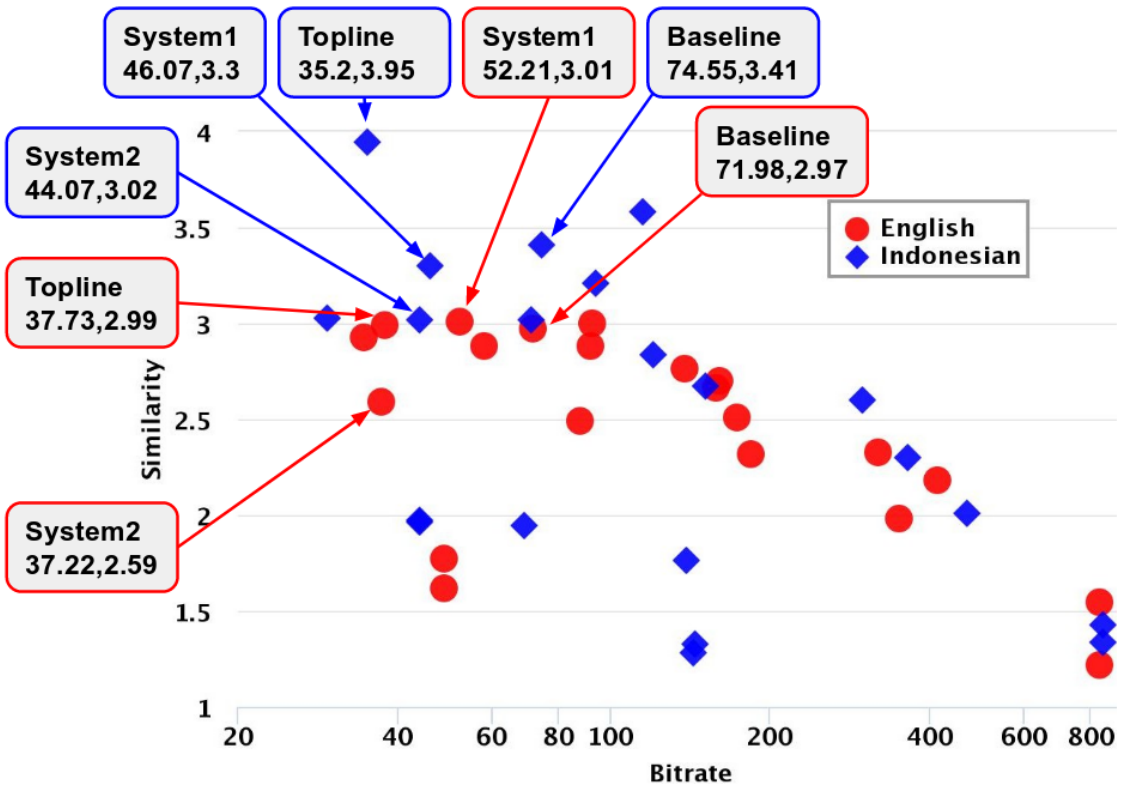


Figure 6.6: Bitrate v/s Speaker similarity. X-axis is displayed in logarithmic scale.

6.4 Summary

The virtual labels obtained from the ASM technique proposed in Chapter 4 are used for language identification. Unsupervised parallel phone recognition is proposed as an approach for LID where virtual labels are obtained from speech signals then phoneme recognizers and language models are built using these virtual labels. The proposed unsupervised LID performed close to the supervised LID using manual transcriptions and also to the i-vector based LID system. Fusion of scores from the two unsupervised approaches significantly outperformed the supervised LID system. This shows the potential of virtual labels from ASM to provide at par performance to the supervised systems for language identification in under-resourced conditions.

Speaking rate estimation is done using syllable-like units obtained from the multiple evidence based approach proposed in Chapter 5. This method provides comparable performance with existing methods for speaking rate estimation on TIMIT and Switchboard corpus. A recent zero resource syllabification algorithm based on theta-rate oscillations at the syllabic rate is also evaluated for speaking rate estimation and is shown to perform closer to other methods. Zero resource based methods for speaking rate estimation can be used for any language or database without any training or fine-tuning of parameters using labeled data. Thus, zero resource methods are better suited for improving speech recognition accuracy through speaking rate dependent decoding in low resource settings.

A virtual phone recognition/synthesis technique is proposed for ultra low bitrate coding. The speech signals are transformed to a sequence of virtual phones which can be coded into one hot vectors in very low bitrates as low as 37.22 bits/sec. The bitrates close to supervised topline system indicate that the number of virtual phone units may be quite close to the actual linguistic units present in the text. These encoded vectors are again re-synthesized back to speech signals using neural network based speech synthesis without any text. No textual or linguistic information of any kind is used throughout this work. The resulting synthesized speech signals have high speaker similarity to the target speakers and slight degradation in synthesis quality. Virtual phones are therefore reliable representations of speech signals leading to ultra low bit rate coding.

Chapter 7

Conclusion and future work

This thesis highlights the problems associated with low resource speech processing and proposes signal to symbol transformation approach towards zero resource speech processing. The proposed approach includes improved segmentation, labeling and modeling. The applications to which the proposed approach is applied successfully include spoken term discovery, language identification, speaking rate estimation and low bitrate coding.

Zero resource is mainly inspired from language acquisition in infants. We draw our motivation from the problem of low resource speech recognition. Low resource languages lack in transcribed data which is required to train acoustic models for building speech systems. Traditional approaches towards low resource speech recognition use different strategies such as data augmentation or building multilingual systems. These methods provide limited success and are nowhere close to the high resource language systems accuracy. Magnitude based features, specifically, MFCC features form an integral part of modern ASR systems including low resource recognition systems. These features are extracted from magnitude spectrum calculated from speech signals thereby ignoring the phase information completely. IF estimation methods are investigated on speech-like synthetic signals and speech signals. IF derived using properties of Fourier Transform from analytic speech signal after smoothing gives reliable estimate of actual IF.

We propose alternate feature representations from this instantaneous frequency derived from the analytic phase of speech signals. The performance of IFCC features is evaluated on low resource and noise robust speech recognition. The PERs obtained from IFCC features is comparable to MFCC features for clean speech, given that IFCC features are phase based and do not use magnitude information at all. This shows that analytic phase contains important information for speech recognition. The lattice level system combination of MFCC and IFCC features provided significant

improvement over MFCC based system. This shows the significance of using both magnitude and phase information for improving ASR performance. Improvements in accuracy were observed using IFCC features in case of speech recognition in different noisy conditions over MFCC features. The system combination provided significant reduction in PERs in almost all noisy conditions. This highlights the importance of phase for speech recognition in noisy conditions.

The low resource problem is data dependant and the performance is inferior compared to high resource despite several strategies being developed specifically for low resource problems. The ideal solution to this would be towards development of completely unsupervised techniques which would be free from use of manual transcriptions. Such techniques can be generalized to any language and would work irrespective of the amount of labeled data available. This is known as zero resource speech processing and is inspired from language acquisition in infants. Towards this goal, a completely zero resource technique is proposed for automatic transcription of speech data into virtual labels. The virtual labels thus obtained are evaluated on spoken term discovery task and are found to be efficient at discovering word units from speech in an unsupervised way.

The virtual labels are used for language identification task. A phonotactic approach based on co-occurrence statistics of virtual labels is proposed. This approach is termed as Unsupervised Parallel Phone Recognition. The phone recognizers and language models are trained using virtual phones for each language in the system. Given a test utterance, the likelihood is calculated for the utterance being from each language. The language which gives maximum likelihood is the hypothesized language. The database consists of eight Indian languages. The experiments suggest that the proposed approach performs close to the supervised system trained with manually marked labels. The score fusion of the proposed approach with i-vector based approach outperforms the supervised system. This shows that LID systems can be built in an unsupervised manner with at par or better performance than supervised systems.

Since, syllables are larger units than phones, they could be better suited for tasks like spoken term discovery. Therefore, we investigate vowel endpoint detection techniques for estimating the boundaries of syllable-like units. Evidences from source features, spectral features and Bessels features are combined for reliable vowel onset/endpoint detection. Different types of syllable-like units are detected using vowel change points and are evaluated for spoken term discovery task. The type of syllable-like units is kept fixed for entire data and across languages to keep the method zero resource. The syllable-like units seem to perform better for French and Mandarin

from ZeroSpeech 2017 database. The performance is relatively lower for English. This indicates that syllable-like units are better suited for syllable-timed rhythmic languages compared to stress based language like English.

The syllable-like units are used for speaking rate estimation in a zero resource manner. The vowel like regions obtained as above are used as anchor points for detecting the syllable-like units. No label or linguistic information is used to estimate the number of syllables per second as the speaking rate. A theta oscillator based approach for detecting syllable-like units from speech in an unsupervised manner is evaluated for speaking rate estimation. This method was originally proposed for spoken term discovery. The evaluations are done on TIMIT and Switchboard corpus. The proposed zero resource method provided performance comparable to the existing methods which are supervised or semi-supervised. This shows that speaking rate estimation can be approached effectively in unsupervised way.

The final application discussed is low bitrate coding. The virtual phones are compact representations of speech signals and can be encoded as one hot vectors at very low bitrates. The encoded vectors are used to re-synthesize the speech signals in a target speaker's voice using neural network based speech synthesis. The proposed system provided ultra low bitrates and high speaker similarity for ZeroSpeech 2019 data. The MOS and ABX scores were relatively worse than the provided baseline systems. This application achieved two tasks in parallel - encoding speech at ultra low bit rates while achieving voice conversion into target speaker's voice.

The major conclusions from this work are highlighted below -

- Phase is important in low resource ASR. IF derived from analytic phase provide reliable features for speech recognition.
- Unsupervised speech systems provide performance comparable to the supervised systems for certain applications.
- Virtual phones and syllable-like units can be used for zero resource speech processing.
- Syllable-like units perform better in case of syllable-timed rhythmic languages.
- Virtual labels can be encoded in ultra low bitrates and can be used for re-synthesizing the speech signals.

7.1 Major Contributions

The major contributions of the thesis are enlisted as follows -

- Importance of analytic phase for ASR in low resource conditions has been demonstrated.
- Instantaneous frequency estimation methods have been investigated for extracting features for speech recognition.
- A novel unsupervised acoustic segment modeling approach has been proposed for zero resource speech processing.
- Segmentation of speech signals into syllable-like units has been studied.
- A phonotactic approach for language identification using virtual phones has been proposed.
- Virtual labels and syllable-like units provided good performance for spoken term discovery.
- Zero resource speaking rate estimation has been proposed using syllable-like units.
- A virtual phone recognition/synthesis approach for ultra low bitrate coding is discussed as a zero resource application.

7.2 Future directions

7.2.1 Transfer learning using virtual phones

A possible approach to effectively utilize the virtual phones for improving the low resource recognition accuracy could be using transfer learning. Knowledge transfer from an acoustic model trained with large amount of unlabeled data which is virtually transcribed using unsupervised speech signal to symbol transformation to an acoustic model trained with low resource data could improve the recognition accuracy in low resource conditions. This approach could be domain dependant as the knowledge transfer from out of domain data could lead to degradation in accuracy.

7.2.2 Keyword spotting and topic identification of spoken documents

Another potential zero resource application is keyword spotting. The query and reference audio can be both transcribed in terms of virtual phones and then a string search can be made to detect query instances in the reference virtual transcriptions.

Alternatively, spoken word retrieval can be performed by matching posteriors for query and test utterances obtained from acoustic models trained using virtual phones.

Identification of topic from spoken documents can be approached in supervised manner. A recent work used infinite phone-loop model for acoustic unit discovery and fed to conventional topic identification based on phoneme trigrams [42]. On the similar lines, the proposed signal to symbol transformation approach can be used for unit discovery and better topic identification methods can be used to improve the accuracy.

7.2.3 Analysis of relationship between virtual phones and linguistic units

The virtual phones obtained from acoustic segment modeling are consistent labels but their relationship with the linguistic counterparts is not known. Measures like purity which is associated with clustering evaluation does not explicitly indicate linguistic relevance. Therefore, in a recent work symmetric KL divergence based metric was proposed to measure the distance between ASM labels and actual phones [182]. Also, the coverage provided by the ASM units of the actual phones was measured. Analysis on similar notes is required to create a one to one mapping between the virtual phones and linguistic phones. This could lead to better understanding of acoustic segment modeling and could be an important milestone towards completely unsupervised speech recognition.

7.2.4 Towards unsupervised speech recognition

Despite the success of virtual phones in applications such as spoken term discovery, language identification etc., their usability in a phoneme recognition or word recognition task is limited due to the fact that there is no actual correspondence between virtual phones and the real phones. A recent work proposed to learn mapping relationships between these two using generative adversarial networks [183]. The phoneme accuracy for completely unrelated text is still far away from acceptable performance.

Another recent work learned the speech and text embedding spaces and aligned the two spaces using adversarial learning [184]. This approach was shown to be effective for spoken word classification and translation tasks. These methods show the possibility and directions for unsupervised speech recognition.

Bibliography

[1] H. Kamper, A. Jansen, and S. Goldwater, "A segmental framework for fully-unsupervised large-vocabulary speech recognition," *Computer Speech & Language*, vol. 46, pp. 154–174, 2017.

[2] P. K. Kuhl, "Early language acquisition: cracking the speech code," *Nature reviews neuroscience*, vol. 5, no. 11, p. 831, 2004.

[3] E. Dupoux, "Cognitive science in the era of artificial intelligence: A roadmap for reverse-engineering the infant language-learner," *Cognition*, vol. 173, pp. 43–59, 2018.

[4] B. Srivastava, S. Sitaram, R. Kumar Mehta, K. Doss Mohan, P. Matani, S. Satpal, K. Bali, R. Srikanth, and N. Nayak, "Interspeech 2018 low resource automatic speech recognition challenge for indian languages," in *Proc. Interspeech*, 2018, pp. 11–14.

[5] O. Adams, T. Cohn, G. Neubig, and A. Michaud, "Phonemic transcription of low-resource tonal languages," in *Proc. Australasian Language Technology Association Workshop*, 2017, pp. 53–60.

[6] G. Hinton, L. Deng, D. Yu, G. E. Dahl, A. R. Mohamed, N. Jaitly, A. Senior, V. Vanhoucke, P. Nguyen, T. N. Sainath *et al.*, "Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups," *IEEE Signal Processing Magazine*, vol. 29, no. 6, pp. 82–97, 2012.

[7] J. Michálek and J. Vaněk, "A survey of recent dnn architectures on the timit phone recognition task," in *Proc. International Workshop on Temporal, Spatial, and Spatio-Temporal Data Mining*. Springer, 2018, pp. 436–444.

[8] A. Graves, A.-r. Mohamed, and G. Hinton, "Speech recognition with deep recurrent neural networks," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 6645–6649.

[9] L. Tóth, "Combining time-and frequency-domain convolution in convolutional neural network-based phone recognition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 190–194.

[10] A. Ragni, K. M. Knill, S. P. Rath, and M. J. Gales, "Data augmentation for low resource languages," in *Proc. INTERSPEECH*, 2014.

[11] N. Kanda, R. Takeda, and Y. Obuchi, "Elastic spectral distortion for low resource speech recognition with deep neural networks," in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2013, pp. 309–314.

[12] A. Ghoshal, P. Swietojanski, and S. Renals, "Multilingual training of deep neural networks," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 7319–7323.

[13] J.-T. Huang, J. Li, D. Yu, L. Deng, and Y. Gong, "Cross-language knowledge transfer using multilingual deep neural network with shared hidden layers," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 7304–7308.

[14] S. Zhou, Y. Zhao, S. Xu, B. Xu *et al.*, "Multilingual recurrent neural networks with residual learning for low-resource speech recognition," in *Proc. INTERSPEECH*, 2017, pp. 704–708.

[15] U. Uebler, "Multilingual speech recognition in seven languages," *Speech Communication*, vol. 35, no. 1-2, pp. 53–69, 2001.

[16] Z. Tüske, P. Golik, R. Schlüter, and H. Ney, "Acoustic modeling with deep neural networks using raw time signal for lvcsr," in *Proc. INTERSPEECH*, 2014.

[17] E. Dunbar, X. N. Cao, J. Benjumea, J. Karadayi, M. Bernard, L. Besacier, X. Anguera, and E. Dupoux, "The zero resource speech challenge 2017," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2017, pp. 323–330.

[18] Y. Shekofteh, F. AlmasGanj, A. Rezaei, and M. M. Goodarzi, "Two novel fdlp based feature extraction methods for improvement of speech recognition," in *Proc. International Symposium on Telecommunications (IST)*, 2010, pp. 600–603.

[19] R. S. Turner, "The ohm-seebeck dispute, hermann von helmholtz, and the origins of physiological acoustics," *The British Journal for the History of Science*, vol. 10, no. 01, pp. 1–24, 1977.

[20] J. Wolfe, E. C. Schafer, B. Heldner, H. Mülder, E. Ward, and B. Vincent, "Evaluation of speech recognition in noise with cochlear implants and dynamic fm," *Journal of the American Academy of Audiology*, vol. 20, no. 7, pp. 409–421, 2009.

[21] T. J. Gardner and M. O. Magnasco, "Sparse time-frequency representations," *Proceedings of the National Academy of Sciences*, vol. 103, no. 16, pp. 6094–6099, 2006.

[22] F.-G. Zeng, K. Nie, G. S. Stickney, Y.-Y. Kong, M. Vongphoe, A. Bhargave, C. Wei, and K. Cao, "Speech recognition with amplitude and frequency modulations," *Proc. National Academy of Sciences*, vol. 102, no. 7, pp. 2293–2298, 2005.

[23] G. Shi, M. M. Shanechi, and P. Aarabi, "On the importance of phase in human speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1867–1874, 2006.

[24] K. K. Paliwal and L. D. Alsteris, "Usefulness of phase spectrum in human speech perception." in *INTERSPEECH*, 2003.

[25] P. Mowlaee, R. Saeidi, and Y. Stylanou, "Interspeech 2014 special session: Phase importance in speech processing applications," in *Proc. Interspeech*, 2014, pp. 1623–1627.

[26] P. Mowlaee, R. Saeidi, and Y. Stylianou, "Advances in phase-aware signal processing in speech communication," *Speech Communication*, vol. 81, pp. 1–29, 2016.

[27] K. K. Paliwal and B. S. Atal, "Frequency-related represention of speech," *Power (dB)*, vol. 70, no. 80, p. 90, 2003.

[28] L. D. Alsteris and K. K. Paliwal, "Short-time phase spectrum in speech processing: A review and some experimental results," *Digital Signal Processing*, vol. 17, no. 3, pp. 578–616, 2007.

[29] R. M. Hegde, H. A. Murthy, and V. R. R. Gadde, "Significance of the modified group delay feature in speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 1, pp. 190–202, 2007.

[30] E. Loweimi, S. M. Ahadi, and T. Drugman, "A new phase-based feature representation for robust speech recognition," in *Proc. IEEE International conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 7155–7159.

[31] B. Boashash, "Estimating and interpreting the instantaneous frequency of a signal. ii. algorithms and applications," *Proc. IEEE*, vol. 80, no. 4, pp. 540–568, 1992.

[32] D. Dimitriadis, P. Maragos, and A. Potamianos, "Robust am-fm features for speech recognition," *IEEE Signal Processing Letters*, vol. 12, no. 9, pp. 621–624, 2005.

[33] P. Tsiakoulis, A. Potamianos, and D. Dimitriadis, "Short-time instantaneous frequency and bandwidth features for speech recognition," in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2009, pp. 103–106.

[34] H. Yin, V. Hohmann, and C. Nadeu, "Acoustic features for speech recognition based on gammatone filterbank and instantaneous frequency," *Speech communication*, vol. 53, no. 5, pp. 707–715, 2011.

[35] Y. Wang, J. Hansen, G. K. Allu, and R. Kumaresan, "Average instantaneous frequency (AIF) and average log-envelopes (ALE) for asr with the aurora 2 database." in *INTERSPEECH*, 2003.

[36] Y. Kubo, S. Okawa, A. Kurematsu, and K. Shirai, "Temporal am–fm combination for robust speech recognition," *Speech Communication*, vol. 53, no. 5, pp. 716–725, 2011.

[37] A. Biswas, P. K. Sahu, A. Bhowmick, and M. Chandra, "Feature extraction technique using erb like wavelet sub-band periodic and aperiodic decomposition for timit phoneme recognition," *International Journal of Speech Technology*, vol. 17, no. 4, pp. 389–399, 2014.

[38] H. Li, B. Ma, and C.-H. Lee, "A vector space modeling approach to spoken language identification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 1, pp. 271–284, 2007.

[39] R. Thiolliere, E. Dunbar, G. Synnaeve, M. Versteegh, and E. Dupoux, "A hybrid dynamic time warping-deep neural network architecture for unsupervised acoustic modeling." in *INTERSPEECH*, 2015, pp. 3179–3183.

[40] J. Mrozinski, E. W. Whittaker, P. Chatain, and S. Furui, "Automatic sentence segmentation of speech for automatic summarization," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1. IEEE, 2006, pp. I–I.

[41] M.-H. Siu, H. Gish, A. Chan, and W. Belfield, "Improved topic classification and keyword discovery using an hmm-based speech recognizer trained without supervision," in *Proc. Annual Conference of the International Speech Communication Association*, 2010.

[42] S. Kesiraju, R. Pappagari, L. Ondel, L. Burget, N. Dehak, S. Khudanpur, J. Černockỳ, and S. V. Gangashetty, "Topic identification of spoken documents using unsupervised acoustic unit discovery," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5745–5749.

[43] D. R. Reddy, "Segmentation of speech sounds," *The Journal of the Acoustical Society of America*, vol. 40, no. 2, pp. 307–312, 1966. [Online]. Available: https://doi.org/10.1121/1.1910071

[44] H. Kasuya and H. Wakita, "Speech segmentation and feature normalization based on area functions," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1, Apr 1976, pp. 29–32.

[45] Y. Qiao, N. Shimomura, and N. Minematsu, "Unsupervised optimal phoneme segmentation: Objectives, algorithm and comparisons," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2008, pp. 3989–3992.

[46] M. Bacchiani and M. Ostendorf, "Joint lexicon, acoustic unit inventory and model design," *Speech Communication*, vol. 29, no. 2, pp. 99–114, 1999.

[47] S. Dusan and L. Rabiner, "On the relation between maximum spectral transition positions and phone boundaries," in *International Conference on Spoken Language Processing*, 2006.

[48] Y. P. Estevan, V. Wan, and O. Scharenborg, "Finding maximum margin segments in speech," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 4. IEEE, 2007, pp. IV–937.

[49] M. M. Goodwin and J. Laroche, "Audio segmentation by feature-space clustering using linear discriminant analysis and dynamic programming," in *Proc.*

IEEE Workshop on Applications of Signal Processing to Audio and Acoustics. IEEE, 2003, pp. 131–134.

[50] A. Sarkar and T. V. Sreenivas, "Automatic speech segmentation using average level crossing rate information," 2005.

[51] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, D. S. Pallett, N. L. Dahlgren, and V. Zue, "Timit acoustic-phonetic continuous speech corpus," *Linguistic data consortium*, vol. 10, no. 5, p. 0, 1993.

[52] O. Scharenborg, V. Wan, and M. Ernestus, "Unsupervised speech segmentation: An analysis of the hypothesized phone boundaries," *The Journal of the Acoustical Society of America*, vol. 127, no. 2, pp. 1084–1095, 2010.

[53] J. Mehler, "The role of syllables in speech processing: Infant and adult data," *Phil. Trans. R. Soc. Lond. B*, vol. 295, no. 1077, pp. 333–352, 1981.

[54] J. Mehler, J. Y. Dommergues, U. H. Frauenfelder, and J. Segui, "The syllable's role in speech segmentation," *Journal of verbal learning and verbal behavior*, vol. 20, no. 3, pp. 298–305, 1981.

[55] S.-L. Wu, M. L. Shire, S. Greenberg, and N. Morgan, "Integrating syllable boundary information into speech recognition," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 2. IEEE, 1997, pp. 987–990.

[56] G. Pradhan and S. M. Prasanna, "Speaker verification by vowel and nonvowel like segmentation," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 21, no. 4, pp. 854–867, 2013.

[57] T. Nagarajan and H. A. Murthy, "Language identification using parallel syllable-like unit recognition," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing.* IEEE, 2004, pp. I–401.

[58] O. Räsänen, G. Doyle, and M. C. Frank, "Unsupervised word discovery from speech using automatic segmentation into syllable-like units," in *Proc. Interspeech*, 2015.

[59] H. Kamper, K. Livescu, and S. Goldwater, "An embedded segmental k-means model for unsupervised segmentation and clustering of speech," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2017, pp. 719–726.

[60] S. Bhati, S. Nayak, and K. Sri Rama Murty, "Unsupervised segmentation of speech signals using kernel-gram matrices," in *Proc. National Conference on Computer Vision, Pattern Recognition, Image Processing, and Graphics (NCVPRIPG), Revised Selected Papers 6*. Springer, 2017, pp. 139–149.

[61] S. M. Prasanna, B. S. Reddy, and P. Krishnamoorthy, "Vowel onset point detection using source, spectral peaks, and modulation spectrum energies," *IEEE Transactions on audio, speech, and language processing*, vol. 17, no. 4, pp. 556–565, 2009.

[62] B. D. Sarma, S. S. Prajwal, and S. M. Prasanna, "Improved vowel onset and offset points detection using bessel features," in *Proc. International Conference on Signal Processing and Communications (SPCOM)*, 2014, pp. 1–6.

[63] C.-H. Lee, F. K. Soong, and B.-H. Juang, "A segment model based approach to speech recognition," in *Acoustics, Speech, and Signal Processing, 1988. ICASSP-88., 1988 International Conference on*. IEEE, 1988, pp. 501–541.

[64] H. Wang, C.-C. Leung, T. Lee, B. Ma, and H. Li, "An acoustic segment modeling approach to query-by-example spoken term detection," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 5157–5160.

[65] H. Gish and K. Ng, "A segmental speech model with applications to word spotting," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 2. IEEE, 1993, pp. 447–450.

[66] M.-h. Siu, H. Gish, A. Chan, W. Belfield, and S. Lowe, "Unsupervised training of an hmm-based self-organizing unit recognizer with applications to topic classification and keyword discovery," *Computer Speech & Language*, vol. 28, no. 1, pp. 210–223, 2014.

[67] H. Wang, T. Lee, C.-C. Leung, B. Ma, and H. Li, "Acoustic segment modeling with spectral clustering methods," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 23, no. 2, pp. 264–277, 2015.

[68] A. S. Park and J. R. Glass, "Unsupervised pattern discovery in speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 1, pp. 186–197, 2008.

[69] A. Park and J. R. Glass, "Towards unsupervised pattern discovery in speech," in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding*. IEEE, 2005, pp. 53–58.

[70] J. Reed and C.-H. Lee, "A study on music genre classification based on universal acoustic models." in *ISMIR*, 2006, pp. 89–94.

[71] M. Bacchiani, M. Ostendorf, Y. Sagisaka, and K. Paliwal, "Unsupervised learning of non-uniform segmental units for acoustic modeling in speech recognition,"," in *Proc. IEEE ASR Workshop*, 1995, pp. 141–142.

[72] C.-y. Lee and J. Glass, "A nonparametric bayesian approach to acoustic model discovery," in *Proc. Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*. Association for Computational Linguistics, 2012, pp. 40–49.

[73] F. Brugnara, D. Falavigna, and M. Omologo, "Automatic segmentation and labeling of speech based on hidden markov models," *Speech Communication*, vol. 12, no. 4, pp. 357–370, 1993.

[74] B. Ma, C. Guan, H. Li, and C.-H. Lee, "Multilingual speech recognition with language identification." in *Proc. INTERSPEECH*, 2002.

[75] P. Dai, U. Iurgel, and G. Rigoll, "A novel feature combination approach for spoken document classification with support vector machines," in *Proc. Multimedia information retrieval workshop*. Citeseer, 2003, pp. 1–5.

[76] Y. K. Muthusamy, E. Barnard, and R. A. Cole, "Reviewing automatic language identification," *IEEE Signal Processing Magazine*, vol. 11, no. 4, pp. 33–41, 1994.

[77] H. Li, B. Ma, and K. A. Lee, "Spoken language recognition: from fundamentals to practice," *Proc. IEEE*, vol. 101, no. 5, pp. 1136–1159, 2013.

[78] E. Ambikairajah, H. Li, L. Wang, B. Yin, and V. Sethu, "Language identification: A tutorial," *IEEE Circuits and Systems Magazine*, vol. 11, no. 2, pp. 82–108, 2011.

[79] R. Tong, B. Ma, D. Zhu, H. Li, and E. S. Chng, "Integrating acoustic, prosodic and phonotactic features for spoken language identification," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 1. IEEE, 2006, pp. I–I.

[80] N. Dehak, P. A. Torres-Carrasquillo, D. Reynolds, and R. Dehak, "Language recognition via i-vectors and dimensionality reduction," in *Proc. Annual Conference of the International Speech Communication Association*, 2011.

[81] Y. Yan and E. Barnard, "An approach to automatic language identification based on language-dependent phone recognition," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 5. IEEE, 1995, pp. 3511–3514.

[82] J.-L. Gauvain, A. Messaoudi, and H. Schwenk, "Language recognition using phone latices." in *Proc. INTERSPEECH*, 2004.

[83] C. Corredor-Ardoy, J.-L. Gauvain, M. Adda-Decker, and L. Lamel, "Language identification with language-independent acoustic models." in *Eurospeech*. Citeseer, 1997.

[84] W. Campbell, T. Gleason, J. Navratil, D. Reynolds, W. Shen, E. Singer, and P. Torres-Carrasquillo, "Advanced language recognition using cepstra and phonotactics: Mitll system performance on the nist 2005 language recognition evaluation," in *Proc. IEEE Odyssey: Speaker and Language Recognition Workshop*. IEEE, 2006, pp. 1–8.

[85] H. Li and B. Ma, "A phonotactic language model for spoken language identification," in *Proc. Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, 2005, pp. 515–522.

[86] R. Tong, B. Ma, H. Li, and E. S. Chng, "A target-oriented phonotactic frontend for spoken language recognition," *IEEE transactions on audio, speech, and language processing*, vol. 17, no. 7, pp. 1335–1347, 2009.

[87] L. F. D'Haro, R. Cordoba, C. Salamea, and J. D. Echeverry, "Extended phone log-likelihood ratio features and acoustic-based i-vectors for language recognition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 5342–5346.

[88] M. Diez, A. Varona, M. Penagarikano, L. J. Rodriguez-Fuentes, and G. Bordel, "On the projection of pllrs for unbounded feature distributions in spoken language recognition," *IEEE Signal Processing Letters*, vol. 21, no. 9, pp. 1073–1077, 2014.

[89] M. A. Zissman *et al.*, "Comparison of four approaches to automatic language identification of telephone speech," *IEEE Transactions on Speech and Audio Processing*, vol. 4, no. 1, p. 31, 1996.

[90] K. Vijayan, H. Li, H. Sun, and K. A. Lee, "On the importance of analytic phase of speech signals in spoken language recognition," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*. IEEE, 2018, pp. 5194–5198.

[91] W. Zhang, R. A. Clark, Y. Wang, and W. Li, "Unsupervised language identification based on latent dirichlet allocation," *Computer Speech & Language*, vol. 39, pp. 47–66, 2016.

[92] A. S. Jayram, V. Ramasubramanian, and T. V. Sreenivas, "Language identification using parallel sub-word recognition," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 1. IEEE, 2003, pp. I–32.

[93] C.-y. Lee and J. Glass, "A nonparametric bayesian approach to acoustic model discovery," in *Proc. Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, 2012, pp. 40–49.

[94] L. Ondel, L. Burget, and J. Černockỳ, "Variational inference for acoustic unit discovery," *Procedia Computer Science*, vol. 81, pp. 80–86, 2016.

[95] A. F. Myrman and G. Salvi, "Partitioning of posteriorgrams using siamese models for unsupervised acoustic modelling," in *Proc. International Workshop on Grounding Language Understanding*, 2017, pp. 27–31.

[96] M. Heck, S. Sakti, and S. Nakamura, "Unsupervised linear discriminant analysis for supporting dpgmm clustering in the zero resource scenario," *Procedia Computer Science*, vol. 81, pp. 73–79, 2016.

[97] A. Jansen, S. Thomas, and H. Hermansky, "Weak top-down constraints for unsupervised acoustic model training," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2013, pp. 8091–8095.

[98] G. Synnaeve, T. Schatz, and E. Dupoux, "Phonetics embedding learning with side information," in *Proc. IEEE Spoken Language Technology Workshop (SLT)*, 2014, pp. 106–111.

[99] D. Renshaw, H. Kamper, A. Jansen, and S. Goldwater, "A comparison of neural network methods for unsupervised representation learning on the zero resource speech challenge," in *Proc. INTERSPEECH*, 2015, pp. 3199–3203.

[100] L. Badino, C. Canevari, L. Fadiga, and G. Metta, "An auto-encoder based approach to unsupervised learning of subword units," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 7634–7638.

[101] T. Tsuchiya, N. Tawara, T. Ogawa, and T. Kobayashi, "Speaker invariant feature extraction for zero-resource languages with adversarial learning," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 2381–2385.

[102] E. Fosler-Lussier and N. Morgan, "Effects of speaking rate and word frequency on pronunciations in conversational speech," *Speech Communication*, vol. 29, no. 2, pp. 137–158, 1999.

[103] H. Nanjo and T. Kawahara, "Speaking-rate dependent decoding and adaptation for spontaneous lecture speech recognition," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 2002, pp. I–725 – I–728.

[104] M. Benzeghiba, R. De Mori, O. Deroo, S. Dupont, T. Erbes, D. Jouvet, L. Fissore, P. Laface, A. Mertins, C. Ris *et al.*, "Automatic speech recognition and speech variability: A review," *Speech communication*, vol. 49, no. 10, pp. 763–786, 2007.

[105] H. Nanjo and T. Kawahara, "Language model and speaking rate adaptation for spontaneous presentation speech recognition," *IEEE Transactions on Speech and Audio Processing*, vol. 12, no. 4, pp. 391–400, 2004.

[106] A. Verma and A. Kumar, "Modeling speaking rate for voice fonts," in *Proc. European Conference on Speech Communication and Technology*, 2003, pp. 2917–2920.

[107] N. Morgan and E. Fosler-Lussier, "Combining multiple estimators of speaking rate," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, vol. 2, 1998, pp. 729–732.

[108] T. Pfau and G. Ruske, "Estimating the speaking rate by vowel detection," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 1998*, 1998, pp. 945–948.

[109] R. Faltlhauser, T. Pfau, and G. Ruske, "On-line speaking rate estimation using gaussian mixture models," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2000, pp. 1355–1358.

[110] D. Wang and S. S. Narayanan, "Robust speech rate estimation for spontaneous speech," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 15, no. 8, pp. 2190–2201, 2007.

[111] N. H. De Jong and T. Wempe, "Praat script to detect syllable nuclei and measure speech rate automatically," *Behavior research methods*, vol. 41, no. 2, pp. 385–390, 2009.

[112] Y. Jiao, V. Berisha, M. Tu, and J. Liss, "Convex weighting criteria for speaking rate estimation," *IEEE/ACM Transactions on audio, speech, and language processing*, vol. 23, no. 9, pp. 1421–1430, 2015.

[113] Y. Jiao, M. Tu, V. Berisha, and J. Liss, "Online speaking rate estimation using recurrent neural networks," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 5245–5249.

[114] S. Nagesh, C. Yarra, O. D. Deshmukh, and P. K. Ghosh, "A robust speech rate estimation based on the activation profile from the selected acoustic unit dictionary," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2016, pp. 5400–5404.

[115] J. Yuan and M. Liberman, "Robust speaking rate estimation using broad phonetic class recognition," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2010, pp. 4222–4225.

[116] N. Mirghafori, E. Fosler, and N. Morgan, "Towards robustness to fast speech in ASR," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1996, pp. 335–338.

[117] E. Dunbar, R. Algayres, J. Karadayi, M. Bernard, J. Benjumea, X.-N. Cao, L. Miskic, C. Dugrain, L. Ondel, A. W. Black, L. Besacier, S. Sakti, and E. Dupoux, "The zero resource speech challenge 2019: TTS without T," 2019.

[118] T. Dutoit, *An introduction to text-to-speech synthesis*. Springer Science & Business Media, 1997, vol. 3.

[119] P. K. Muthukumar and A. W. Black, "Automatic discovery of a phonetic inventory for unwritten languages for statistical speech synthesis," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2014, pp. 2594–2598.

[120] A. Tjandra, S. Sakti, and S. Nakamura, "Listening while speaking: Speech chain by deep learning," in *Proc. IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2017, pp. 301–308.

[121] Y. Gao, R. Singh, and B. Raj, "Voice impersonation using generative adversarial networks," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 2506–2510.

[122] C.-C. Hsu, H.-T. Hwang, Y.-C. Wu, Y. Tsao, and H.-M. Wang, "Voice conversion from non-parallel corpora using variational auto-encoder," in *Proc. Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA)*, 2016, pp. 1–6.

[123] J.-c. Chou, C.-c. Yeh, H.-y. Lee, and L.-s. Lee, "Multi-target voice conversion without parallel data by adversarially learning disentangled audio representations," *arXiv preprint arXiv:1804.02812*, 2018.

[124] T. Kaneko and H. Kameoka, "Parallel-data-free voice conversion using cycle-consistent adversarial networks," *arXiv preprint arXiv:1711.11293*, 2017.

[125] L. Cohen, "Time-frequency analysis: theory and applications," *USA: Prentice Hall*, 1995.

[126] S. C. Sekhar and T. V. Sreenivas, "Adaptive window zero-crossing-based instantaneous frequency estimation," *EURASIP Journal on Advances in Signal Processing*, vol. 2004, no. 12, p. 249858, 2004.

[127] J. J. Shynk *et al.*, "Frequency-domain and multirate adaptive filtering," *IEEE Signal Processing Magazine*, vol. 9, no. 1, pp. 14–37, 1992.

[128] D. Rudoy, T. F. Quatieri, and P. J. Wolfe, "Time-varying autoregressions in speech: Detection theory and applications," *IEEE Transactions on audio, Speech, and Language processing*, vol. 19, no. 4, pp. 977–989, 2011.

[129] K. S. R. Murty and B. Yegnanarayana, "Epoch extraction from speech signals," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 16, no. 8, pp. 1602–1613, 2008.

[130] A. Potamianos and P. Maragos, "Speech formant frequency and bandwidth tracking using multiband energy demodulation," *The Journal of the Acoustical Society of America*, vol. 99, no. 6, pp. 3795–3806, 1996.

[131] K. Vijayan, P. R. Reddy, and K. S. R. Murty, "Significance of analytic phase of speech signals in speaker verification," *Speech Communication*, vol. 81, pp. 54–71, 2016.

[132] L. F. Lamel, R. H. Kassel, and S. Seneff, "Speech database development: Design and analysis of the acoustic-phonetic corpus," in *Speech Input/Output Assessment and Speech Databases*, 1989.

[133] A.-r. Mohamed, G. Dahl, and G. Hinton, "Deep belief networks for phone recognition," in *Proc. Nips workshop on deep learning for speech recognition and related applications*, vol. 1, no. 9, 2009, p. 39.

[134] G. E. Dahl, D. Yu, L. Deng, and A. Acero, "Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 20, no. 1, pp. 30–42, 2012.

[135] L. R. Rabiner, "A tutorial on hidden markov models and selected applications in speech recognition," *Proc. IEEE*, vol. 77, no. 2, pp. 257–286, Feb 1989.

[136] K. Veselỳ, A. Ghoshal, L. Burget, and D. Povey, "Sequence-discriminative training of deep neural networks." in *Interspeech*, 2013, pp. 2345–2349.

[137] H. Xu, D. Povey, L. Mangu, and J. Zhu, "Minimum bayes risk decoding and system combination based on a recursion for edit distance," *Computer Speech & Language*, vol. 25, no. 4, pp. 802–828, 2011.

[138] "WER are we," https://github.com/syhw/wer_are_we#timit, 2019, [Online; accessed 29-May-2019].

[139] P. C. Loizou, *Speech enhancement: theory and practice*. CRC press, 2013.

[140] J.-P. Vert, K. Tsuda, and B. Schölkopf, "A primer on kernel methods," *Kernel Methods in Computational Biology*, pp. 35–70, 2004.

[141] L. R. Rabiner, *Multirate digital signal processing*. Prentice Hall PTR, 1996.

[142] M. Ester, H.-P. Kriegel, J. Sander, X. Xu *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise." in *Kdd*, vol. 96, no. 34, 1996, pp. 226–231.

[143] J. S. Garofolo, L. F. Lamel, W. M. Fisher, J. G. Fiscus, and D. S. Pallett, "Darpa timit acoustic-phonetic continous speech corpus cd-rom. nist speech disc 1-1.1," *NASA STI/Recon technical report n*, vol. 93, 1993.

[144] M. Versteegh, R. Thiolliere, T. Schatz, X.-N. Cao, X. Anguera, A. Jansen, and E. Dupoux, "The zero resource speech challenge 2015." in *Interspeech*, 2015, pp. 3169–3173.

[145] V. Khanagha, K. Daoudi, O. Pont, and H. Yahia, "Phonetic segmentation of speech signal using local singularity analysis," *Digital Signal Processing*, vol. 35, pp. 86–94, 2014.

[146] A. Stan, C. Valentini-Botinhao, B. Orza, and M. Giurgiu, "Blind speech segmentation using spectrogram image-based features and mel cepstral coefficients," in *Proc. IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2016, pp. 597–602.

[147] S. J. Leow, E. S. Chng, and C.-H. Lee, "Language-resource independent speech segmentation using cues from a spectrogram image," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2015, pp. 5813–5817.

[148] O. Rasanen, U. Laine, and T. Altosaar, "Blind segmentation of speech using non-linear filtering methods," in *Speech Technologies*. InTech, 2011.

[149] A. Jansen and B. Van Durme, "Efficient spoken term discovery using randomized algorithms," in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding (ASRU)*, 2011, pp. 401–406.

[150] V. Lyzinski, G. Sell, and A. Jansen, "An evaluation of graph clustering methods for unsupervised term discovery," in *Proc. Interspeech*, 2015.

[151] Y. Qiao, N. Shimomura, and N. Minematsu, "Unsupervised optimal phoneme segmentation: Objectives, algorithm and comparisons," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2008, pp. 3989–3992.

[152] V. Vuuren, L. Bosch, and T. Niesler, "Unconstrained speech segmentation using deep neural networks," in *Proc. International conference on pattern recognition applications and methods (ICPRAM)*, vol. 1, 2015, pp. 248–254.

[153] J. Reichardt and S. Bornholdt, "Statistical mechanics of community detection," *Physical Review E*, vol. 74, no. 1, p. 016110, 2006.

[154] I. Fonagy and K. Magdics, "Speed of utterance in phrases of different lengths," *Language and Speech*, vol. 3, no. 4, pp. 179–192, 1960.

[155] G. D. Forney, "The viterbi algorithm," *Proceedings of the IEEE*, vol. 61, no. 3, pp. 268–278, 1973.

[156] M. Versteegh, X. Anguera, A. Jansen, and E. Dupoux, "The zero resource speech challenge 2015: Proposed approaches and results," *Procedia Computer Science*, vol. 81, pp. 67–72, 2016.

[157] B. Ludusan, M. Versteegh, A. Jansen, G. Gravier, X.-N. Cao, M. Johnson, and E. Dupoux, "Bridging the gap between speech technology and natural language processing: an evaluation toolbox for term discovery systems," in *Language Resources and Evaluation Conference*, 2014.

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

[159] S. M. Prasanna and B. Yegnanarayana, "Detection of vowel onset point events using excitation information," in *Proc. Ninth European Conference on Speech Communication and Technology*, 2005, pp. 1133–1136.

[160] S. Greenberg and B. Kingsbury, "The modulation spectrogram: In pursuit of an invariant representation of speech," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1997, pp. 1647–1650.

[161] B. E. Kingsbury, N. Morgan, and S. Greenberg, "Robust speech recognition using the modulation spectrogram," *Speech communication*, vol. 25, no. 1-3, pp. 117–132, 1998.

[162] J. Schroeder, "Signal processing via fourier-bessel series expansion." DENVER UNIV CO COLL OF ENGINEERING, Tech. Rep., 1994.

[163] S. Nayak, S. Bhati, and K. S. R. Murty, "An investigation into instantaneous frequency estimation methods for improved speech recognition features," in *Proc. IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2017, pp. 363–367.

[164] C. Prakash, N. Dhananjaya, and S. V. Gangashetty, "Bessel features for detection of voice onset time using AM-FM signal," in *Proc. IEEE International Conference on Systems, Signals and Image Processing (IWSSIP)*, 2011, pp. 1–4.

[165] C. Prakash, D. N. Gowda, and S. V. Gangashetty, "Analysis of acoustic events in speech signals using bessel series expansion," *Circuits, Systems, and Signal Processing*, vol. 32, no. 6, pp. 2915–2938, 2013.

[166] A.-L. Giraud and D. Poeppel, "Cortical oscillations and speech processing: emerging computational principles and operations," *Nature neuroscience*, vol. 15, no. 4, p. 511, 2012.

[167] S. Greenberg, "Speaking in shorthand–a syllable-centric perspective for understanding pronunciation variation," *Speech Communication*, vol. 29, no. 2-4, pp. 159–176, 1999.

[168] A. Cutler and D. Norris, "The role of strong syllables in segmentation for lexical access." *Journal of Experimental Psychology: Human perception and performance*, vol. 14, no. 1, p. 113, 1988.

[169] P. Mok, "On the syllable-timing of cantonese and beijing mandarin," *Chinese Journal of Phonetics*, vol. 2, pp. 148–154, 2009.

[170] M. A. Zissman and E. Singer, "Automatic language identification of telephone speech messages using phoneme recognition and n-gram modeling," in *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, vol. 1. IEEE, 1994, pp. I–305.

[171] K. Mounika, L. H. Sivanand Achanta, V. G. Suryakanth, and A. K. Vuppala, "An investigation of deep neural network architectures for language recognition in indian languages," *Interspeech 2016*, pp. 2930–2933, 2016.

[172] A. Martin, G. Doddington, T. Kamm, M. Ordowski, and M. Przybocki, "The det curve in assessment of detection task performance," DTIC Document, Tech. Rep., 1997.

[173] N. Brümmer and E. De Villiers, "The bosaris toolkit: Theory, algorithms and code for surviving the new dcf," *arXiv preprint arXiv:1304.2865*, 2013.

[174] O. Räsänen, G. Doyle, and M. C. Frank, "Pre-linguistic segmentation of speech into syllable-like units," *Cognition*, vol. 171, pp. 130–150, 2018.

[175] J. J. Godfrey, E. C. Holliman, and J. McDaniel, "Switchboard: Telephone speech corpus for research and development," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 1992, pp. 517–520.

[176] S. Bhati, S. Nayak, and K. S. R. Murty, "Unsupervised speech signal to symbol transformation for zero resource speech applications." in *Proc. INTERSPEECH*, 2017, pp. 2133–2137.

[177] S. Bhati, H. Kamper, and K. S. R. Murty, "Phoneme based embedded segmental k-means for unsupervised term discovery," in *Proc. IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2018, pp. 5169–5173.

[178] Z. Wu, O. Watts, and S. King, "Merlin: An open source neural network speech synthesis system." in *SSW*, 2016, pp. 202–207.

[179] K. Levin, K. Henry, A. Jansen, and K. Livescu, "Fixed-dimensional acoustic embeddings of variable-length segments in low-resource settings," in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding*, 2013, pp. 410–415.

[180] M. Morise, F. Yokomori, and K. Ozawa, "World: a vocoder-based high-quality speech synthesis system for real-time applications," *IEICE TRANSACTIONS on Information and Systems*, vol. 99, no. 7, pp. 1877–1884, 2016.

[181] D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz *et al.*, "The kaldi speech recognition toolkit," in *Proc. IEEE workshop on automatic speech recognition and understanding*, no. EPFL-CONF-192584. IEEE Signal Processing Society, 2011.

[182] S. Feng and T. Lee, "On the linguistic relevance of speech units learned by unsupervised acoustic modeling." in *Proc. INTERSPEECH*, 2017, pp. 2068–2072.

[183] D.-R. Liu, K.-Y. Chen, H.-Y. Lee, and L.-s. Lee, "Completely unsupervised phoneme recognition by adversarially learning mapping relationships from audio embeddings," *arXiv preprint arXiv:1804.00316*, 2018.

[184] Y.-A. Chung, W.-H. Weng, S. Tong, and J. Glass, "Unsupervised cross-modal alignment of speech and text embedding spaces," in *Advances in Neural Information Processing Systems*, 2018, pp. 7354–7364.

List of Publications

Journals:

- Nayak S., Bhati S. and Murty, K.S.R., "Syllable-like Units based Zero-Resource Spoken Term Discovery," to be submitted to *IEEE Signal Processing letters.*

- Bhati S.*, Nayak S.* and Murty, K.S.R., "Unsupervised Speech Signal to Symbol Transformation for Language Identification," submitted to *Circuits, Systems and Signal Processing*, Springer (* equal contribution).

Conferences:

- Nayak, S., Bhati, S. and Murty, K.S.R., "Zero resource speaking rate estimation from change point detection of syllable-like units", in *Proc. IEEE Int. Conf. on Acoust., Speech and Signal Process. (ICASSP)*, May 2019.

- Nayak S., Dhar S., Bhati S., Bramhendra K. and Murty, K.S.R., "Instantaneous Frequency Features for Noise Robust Speech Recognition", in *Proc. Twenty Fifth National Conference on Communications (NCC)*, Feb. 2019.

- Nayak, S., Bhati, S. and Murty, K.S.R., "An investigation into instantaneous frequency estimation methods for improved speech recognition features", in *Proc. IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, Nov. 2017, pp. 363-367.

- Bhati, S., Nayak, S. and Murty, K.S.R., "Unsupervised Speech Signal to Symbol Transformation for Zero Resource Speech Applications," in *Proc. INTERSPEECH*, Aug. 2017, pp. 2133-2137.

- Bhati, S., Nayak, S. and Murty, K.S.R., "Unsupervised segmentation of speech signals using kernel-gram matrices," in *Proc. 6th National Conference on Computer Vision, Pattern Recognition, Image Processing, and Graphics (NCVPRIPG)*, Mandi, India, Dec. 16-19, 2017, pp. 139-149, Springer Singapore.

- Nayak, S., Shiva Kumar C., Ramesh G., Bhati, S. and Murty, K.S.R., "Virtual Phone Recognition/Synthesis for Ultra Low Bitrate Coding," submitted to *INTERSPEECH 2019.*